```
1
  3
  4
  5
  6
  8
  9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
100
101
102
103
104
105
```

```
; **************************
; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.2
; Last Update: 01/09/2020
; Beginning: 04/01/2016
; Assembler: NASM version 2.14 (trdos386.s)
; -----
; Turkish Rational DOS
; Operating System Project v2.0 by ERDOGAN TAN (Beginning: 04/01/2016)
; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
; unix386.s (03/01/2016)
; Derived from TRDOS Operating System v1.0 (8086) source code by Erdogan Tan
; TRDOS2.ASM (09/11/2011)
; Derived from 'IBM PC-XT-286' BIOS source code (1986)
; nasm trdos386.s -1 trdos386.txt -o TRDOS386.SYS
KLOAD equ 10000h ; Kernel loading address
     ; NOTE: Retro UNIX 8086 v1 /boot code loads kernel at 1000h:0000h
KCODE equ 08h ; Code segment descriptor (ring 0) KDATA equ 10h ; Data segment descriptor (ring 0)
; 19/03/2015
UCODE equ 1Bh ; 18h + 3h (ring 3)
UDATA equ 23h ; 20h + 3h (ring 3)
; 24/03/2015
                   ; Task state segment descriptor (ring 0)
TSS equ 28h
; 19/03/2015
CORE equ 400000h ; Start of USER's virtual/linear address space
                  ; (at the end of the 1st 4MB)
ECORE equ OFFC00000h; End of USER's virtual address space (4GB - 4MB)
                  ; ULIMIT = (ECORE/4096) - 1 = OFFBFFh (in GDT)
;; 27/12/2013
; KEND equ KLOAD + 65536 ; (28/12/2013) (end of kernel space)
; 04/07/2016
KEND equ KERNELFSIZE + KLOAD
; IBM PC/AT BIOS ---- 10/06/85 (postequ.inc)
;----- CMOS TABLE LOCATION ADDRESS'S ------
                              ; SECONDS (BCD)
CMOS_SECONDS EQU 00H
CMOS_SEC_ALARM EQU 01H ; SECONDS ALARM (BCD)
CMOS_MINUTES EQU 02H ; MINUTES (BCD)
CMOS_MIN_ALARM EQU 03H ; MINUTES ALARM (BCD)
CMOS_HOURS EQU 04H ; HOURS (BCD
CMOS_HR ALARM EOU 005H . HOURS ALARM (BCD)
CMOS_HR_ALARM EQU 005H ; HOURS ALARM (BCD)
CMOS_DAY_WEEK EQU 06H ; DAY OF THE WEEK (BCD)
CMOS_DAY_MONTH EQU 07H ; DAY OF THE MONTH (BCD)
CMOS DAY MONTH EQU 07H ; DAY OF THE MONTH (BCD)

CMOS MONTH EQU 08H ; MONTH (BCD)

CMOS YEAR EQU 09H ; YEAR (TWO DIGITS) (BCD)

CMOS CENTURY EQU 32H ; DATE CENTURY BYTE (BCD)

CMOS REG A EQU 0AH ; STATUS REGISTER A

CMOS REG B EQU 00CH ; STATUS REGISTER C FLAGS

CMOS REG C EQU 00CH ; STATUS REGISTER C FLAGS

CMOS REG D EQU 0DH ; STATUS REGISTER D BATTERY

CMOS SHUT DOWN EQU 0FH ; SHUTDOWN STATUS COMMAND BYTE
;-----
; CMOS EQUATES FOR THIS SYSTEM ;
            EQU 070H ; I/O ADDRESS OF CMOS ADDRESS PORT EQU 071H ; I/O ADDRESS OF CMOS DATA PORT
CMOS_PORT
CMOS DATA
             EQU 10000000B ; DISABLE NMI INTERRUPTS MASK -
                                   ; HIGH BIT OF CMOS LOCATION ADDRESS
; Memory Allocation Table Address
; 05/11/2014
; 31/10/2014
                     equ 100000h
MEM_ALLOC_TBL
                                                 ; Memory Allocation Table at the end of
                                  ; the 1st 1 MB memory space.
                                   ; (This address must be aligned
                                   ; on 128 KB boundary, if it will be
                                   ; changed later.)
                                   ; ((lower 17 bits of 32 bit M.A.T.
                                       address must be ZERO)).
                                    ; ((((Reason: 32 bit allocation
                                          instructions, dword steps)))
                                    ; (((byte >> 12 --> page >> 5)))
;04/11/2014
PDE A PRESENT
                                          ; Present flag for PDE
                     equ
PDE A WRITE equ
                     2
                                   ; Writable (write permission) flag
                                   ; User (non-system/kernel) page flag
PDE_A_USER equ
PTE_A_PRESENT
                                           ; Present flag for PTE (bit 0)
                     equ
PTE A WRITE equ
                                   ; Writable (write permission) flag (bit 1)
                     2
                    4
PTE_A_USER equ
                                   ; User (non-system/kernel) page flag (bit 2)
                                   ; Accessed flag (bit 5) ; 09/03/2015
PTE_A_ACCESS
                 equ32
; 17/02/2015 (unix386.s)
; 10/12/2014 - 30/12/2014 (OB000h -> 9000h) (dsectrm2.s)
DPT_SEGM equ 09000h ; FDPT segment (EDD v1.1, EDD v3)
HDO DPT
                         ; Disk parameter table address for hd0
               equ 0
               equ 32
                               ; Disk parameter table address for hd1
HD1 DPT
               equ 64
HD2 DPT
                                ; Disk parameter table address for hd2
HD3_DPT
               equ 96
                                ; Disk parameter table address for hd3
; FDPT (Phoenix, Enhanced Disk Drive Specification v1.1, v3.0)
        (HDPT: Programmer's Guide to the AMIBIOS, 1993)
```

```
106
107
                                    FDPT CYLS
                                                equ 0 ; 1 word, number of cylinders
                                    FDPT HDS
108
                                                equ 2 ; 1 byte, number of heads
109
                                    FDPT TT
                                                      equ 3 ; 1 byte, A0h = translated FDPT with logical values
110
                                                      ; otherwise it is standard FDPT with physical values
111
                                    FDPT_PCMP
                                                equ 5 ; 1 word, starting write precompensation cylinder
                                                     ; (obsolete for IDE/ATA drives)
112
                                                      equ 8 ; 1 byte, drive control byte
                                    FDPT CB
113
114
                                                      ; Bits 7-6 : Enable or disable retries (00h = enable)
115
                                                      ; Bit 5 : 1 = Defect map is located at last cyl. + 1
                                                      ; Bit 4 : Reserved. Always 0
116
                                                      ; Bit 3 : Set to 1 if more than 8 heads
117
                                                      ; Bit 2-0 : Reserved. Alsways 0
118
                                    FDPT LZ
                                                      equ 12 ; 1 word, landing zone (obsolete for IDE/ATA drives)
119
120
                                    FDPT SPT
                                                equ 14 ; 1 byte, sectors per track
121
                                    ; Floppy Drive Parameters Table (Programmer's Guide to the AMIBIOS, 1993)
122
                                    ; (11 bytes long) will be used by diskette handler/bios
123
124
                                    ; which is derived from IBM PC-AT BIOS (DISKETTE.ASM, 21/04/1986).
125
126
                                    ; 01/02/2016
                                    Logical DOSDisks equ 90000h + 100h; 26*256 = 6656 bytes
127
                                    Directory_Buffer equ 80000h; max = 64K Bytes
128
129
                                    FAT_Buffer equ 91C00h; 1536 bytes (3 sectors)
130
                                    ; 15/02/2016
131
                                    Cluster_Buffer
                                                       equ 70000h; max = 64K Bytes; buffer for file read & write
132
                                    ; 11/04/2016
                                    Env Page: equ 93000h; 512 bytes (4096 bytes)
133
134
                                    Env_Page_Size
                                                       equ 512 ; (4096 bytes)
135
                                    ; 30/07/2016
136
                                    Video_Pg_Backup
                                                       equ 98000h; Mode 3h, video page backup (32K, 8 pages)
137
138
                                    [BITS 16]
                                                 ; We need 16-bit intructions for Real mode
139
140
                                    [ORG 0]
                                         ; 12/11/2014
142
                                         ; Save boot drive number (that is default root drive)
143 00000000 8816[525D]
                                              [boot_drv], dl ; physical drv number
144
145
                                         ; Determine installed memory
                                         ; 31/10/2014
146
147
148 00000004 B801E8
                                                ax, 0E801h ; Get memory size
                                         mov
149 00000007 CD15
                                               15h ; for large configurations
                                         int
150 00000009 7308
                                          jnc
                                                short chk_ms
151 0000000B B488
                                         mov
                                                ah, 88h ; Get extended memory size
152 0000000D CD15
                                         int
                                               15h
153
154
                                               al, 17h
                                                           ; Extended memory (1K blocks) low byte
                                         ; mov
155
                                          ;out 70h, al ; select CMOS register
156
                                          ;in
                                               al, 71h; read data (1 byte)
157
                                          ; mov
                                               cl, al
158
                                          ;mov al, 18h; Extended memory (1K blocks) high byte
159
                                               70h, al ; select CMOS register
                                          ;out
160
                                          ;in
                                               al, 71h; read data (1 byte)
161
                                         ;mov
                                               ch, al
162
                                          ;
163 0000000F 89C1
                                         mov
                                                cx, ax
164 00000011 31D2
                                         xor
                                                dx, dx
165
                                    chk_ms:
166 00000013 890E[4E5D]
                                         mov
                                                [mem 1m 1k], cx
                                               [mem_16m_64k], dx
167 00000017 8916[505D]
                                         mov
                                         ; 05/11/2014
168
169
                                         ; and dx, dx
                                         jz short L2
170
171 0000001B 81F90004
                                          cmp cx, 1024
172 0000001F 7351
                                               short L0
                                         jnb
173
                                                ; insufficient memory_error
174
                                                 ; Minimum 2 MB memory is needed...
                                          ; 05/11/2014
175
176
                                          ; (real mode error printing)
177 00000021 FB
                                          sti
178 00000022 BE[3600]
                                               si, msg_out_of_memory
179 00000025 BB0700
                                               bx, 7
                                         mov
180 00000028 B40E
                                               ah, OEh
                                                           ; write tty
                                         mov
                                    oom 1:
182 0000002A AC
                                         lodsb
183 0000002B 08C0
                                               al, al
                                          or
184 0000002D 7404
                                                short oom_2
                                          jΖ
185 0000002F CD10
                                          int
                                               10h
186 00000031 EBF7
                                          jmp
                                                short oom 1
                                    oom_2:
188 00000033 F4
                                           hlt
189 00000034 EBFD
                                          jmp short oom 2
190
                                    ; 20/02/2017
192
                                    ; 05/11/2014
193
                                    msg_out_of_memory:
                                         db 07h, 0Dh, 0Ah
db 'Insufficient memory!'
194 00000036 070D0A
195 00000039 496E73756666696369-
195 00000042 656E74206D656D6F72-
195 0000004B 792021
196 0000004E 0D0A
                                          db
                                               ODh, OAh
197
                                    int13h 48h buffer: ; 07/07/2016
198 00000050 284D696E696D756D20-
                                          db
                                               '(Minimum 2MB memory is needed.)'
198 00000059 324D42206D656D6F72-
198 00000062 79206973206E656564-
198 0000006B 65642E29
199 0000006F 0D0A00
                                                0Dh, 0Ah, 0
                                          db
200
201
202
                                    L0:
                                    %include 'diskinit.s'; 07/03/2015
203
                                1
```

```
4
                              <1> ; Last Update: 29/08/2020
                              <1> ; ------
 5
 6
                              <1>; Beginning: 24/01/2016
 7
 8
                              <1>; Assembler: NASM version 2.14 (trdos386.s)
 9
                              <1>; ------
 10
                              <1> ; Turkish Rational DOS
                              <1>; Operating System Project v2.0 by ERDOGAN TAN (Beginning: 04/01/2016)
 11
 12
                              <1>;
                              <1>; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
 13
14
                              <1>; diskinit.inc (10/07/2015)
                              <1> ;
15
                              <1>; Derived from 'IBM PC-XT-286' BIOS source code (1986)
 16
                              17
 18
                              <1>; Retro UNIX 386 v1 Kernel - DISKINIT.INC
 19
 20
                              <1> ; Last Modification: 10/07/2015
 21
                              <1>
                              <1>; DISK I/O SYSTEM INITIALIZATION - Erdogan Tan (Retro UNIX 386 v1 project)
 22
 23
                              <1>; ////// DISK I/O SYSTEM STRUCTURE INITIALIZATION //////////
 24
 25
                                      ; 29/08/2020
 26
                              <1>
 27
                              <1>
                                      ; 17/07/2020
 28
                              <1>
                                      ; 14/07/2020 (TRDOS 386 v2.0.2)
                                      ; 10/12/2014 - 02/02/2015 - dsectrm2.s
 29
                              <1>
 30
                              <1> ;L0:
 31
                                       ; 12/11/2014 (Retro UNIX 386 v1 - beginning)
                              <1>
 32
                              <1>
                                       ; Detecting disk drives... (by help of ROM-BIOS)
 33 00000072 BA7F00
                              <1>
                                      mov
                                            dx, 7Fh
                             <1> L1:
 34
 35 00000075 FEC2
                             <1>
                                      inc
                                            dl
                                            ah, 41h; Check extensions present
 36 00000077 B441
                             <1>
                                      mov
 37
                             <1>
                                                  ; Phoenix EDD v1.1 - EDD v3
 38 00000079 BBAA55
                             <1>
                                      mov
                                            bx, 55AAh
                                      int
 39 0000007C CD13
                                            13h
                             <1>
 40 0000007E 721A
                             <1>
                                            short L2
                                      jс
                             <1>
 41
 42 00000080 81FB55AA
                                      cmp bx, 0AA55h
                             <1>
 43 00000084 7514
                             <1>
                                      jne
                                            short L2
                             <1>
                                      inc byte [hdc] ; count of hard disks (EDD present)
 44 00000086 FE06[555D]
 45 0000008A 8816[545D]
                             <1>
                                             [last_drv], dl ; last hard disk number
 46 0000008E BB[D85C]
                             <1>
                                      mov bx, hd0_type - 80h
 47 00000091 01D3
                             <1>
                                      add
                                            bx, dx
 48 00000093 880F
                             <1>
                                      mov [bx], cl; Interface support bit map in CX
                                                   ; Bit 0 - 1, Fixed disk access subset ready
 49
                             <1>
 50
                              <1>
                                                    ; Bit 1 - 1, Drv locking and ejecting ready
 51
                              <1>
                                                   ; Bit 2 - 1, Enhanced Disk Drive Support
 52
                              <1>
                                                                   (EDD) ready (DPTE ready)
 53
                              <1>
                                                    ; Bit 3 - 1, 64bit extensions are present
                                                               (EDD-3)
 54
                              <1>
                                                       ;
                                                   ; Bit 4 to 15 - 0, Reserved
 55
                             <1>
                                                      ; drive number < 83h
 56 00000095 80FA83
                             <1>
                                            dl, 83h
                                       cmp
 57 00000098 72DB
                             <1>
                                       jb
                                            short L1
 58
                             <1> L2:
                                      ; 23/11/2014
 59
                             <1>
 60
                             <1>
                                       ; 19/11/2014
 61 0000009A 30D2
                             <1>
                                      xor dl, dl ; 0
                             <1>
                                      ; 04/02/2016 (esi -> si)
 63 0000009C BE[565D]
                             <1>
                                      mov si, fd0_type
                             <1> L3:
 64
 65
                             <1>
                                      ; 14/01/2015
 66 0000009F 8816[535D]
                             <1>
                                      mov [drv], dl
 67
                             <1>
                                      mov
 68 000000A3 B408
                             <1>
                                            ah, 08h ; Return drive parameters
 69 000000A5 CD13
                             <1>
                                      int 13h
 70 000000A7 7210
                             <1>
                                            short L4
 71
                              <1>
                                            ; BL = drive type (for floppy drives)
                                            ; DL = number of floppy drives
 72
                              <1>
 73
                              <1>
                                            ; ES:DI = Address of DPT from BIOS
 74
                              <1>
 75
                              <1>
 76 000000A9 881C
                              <1>
                                           [si], bl ; Drive type
                                     mov
 77
                             <1>
                                                  ; 4 = 1.44 MB, 80 track, 3 1/2"
                                      ; 14/01/2015
 78
                             <1>
 79 000000AB E8DE01
                                      call set_disk_parms
                              <1>
                                      ; 10/12/2014
                              <1>
 81 000000AE 81FE[565D]
                                       cmp si, fd0_type
                              <1>
 82 000000B2 7705
                              <1>
                                       ja short L4
 83 000000B4 46
                                          si ; fd1_type
                              <1>
                                       inc
 84 000000B5 B201
                              <1>
                                       mov
                                             dl, 1
 85 000000B7 EBE6
                              <1>
                                            short L3
                                       jmp
                              <1> L4:
 86
 87 000000B9 B27F
                                            dl, 7Fh
                              <1>
                                       mov
                              <1>
                                       ; 24/12/2014
 89 000000BB 803E[555D]00
                              <1>
                                       cmp byte [hdc], 0 ; EDD present or not ?
 90 000000C0 0F878700
                              <1>
                                                        ; yes, all fixed disk operations
                                       jа
                              <1>
                                                    ; will be performed according to
 92
                              <1>
                                                    ; present EDD specification
 93
                              <1>
                                       ; 17/07/2020
 94
                              <1>
 95
                              <1>
                                       ; Note: Virtual CPU will not come here while
 96
                              <1>
                                       ; running in QEMU, Bochs, VirtualBox emulators !!!
 97
                              <1>
 98
                              <1>
 99
                              <1>
                                       ; Older BIOS (INT 13h, AH = 48h is not available)
100
                              <1> L6:
101 000000C4 FEC2
                              <1>
                                       inc
102 000000C6 8816[535D]
                             <1>
                                       mov
                                                [drv], dl
103 000000CA 8816[545D]
                              <1>
                                        mov
                                                [last_drv], dl ; 14/01/2015
                                       mov ah, 08h ; Return drive parameters
104 000000CE B408
                             <1>
105 000000D0 CD13
                              <1>
                                       int 13h ; (conventional function)
106 000000D2 0F82A801
                              <1>
                                       jc L13; fixed disk drive not ready
```

<1>; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.2 - diskinit.s

2

```
107 000000D6 8816[555D]
                           <1>
                                                 [hdc], dl ; number of drives
                                        mov
                              <1>
                                       ;; 14/01/2013
109
                              <1>
                                       ;;push cx
                                      call set_disk_parms
110 000000DA E8AF01
                              <1>
                              <1>
111
                                       ;;pop cx
112
                              <1>
                                       ;;and cl, 3Fh
                                                           ; sectors per track (bits 0-6)
113
                              <1>
114 000000DD 8A16[535D]
                                        mov dl, [drv]
                              <1>
115 000000E1 BB0401
                              <1>
                                        mov bx, 65*4; hd0 parameters table (INT 41h)
                                        cmp dl, 80h
116 000000E4 80FA80
                             <1>
117 000000E7 7603
                             <1>
                                        jna short L7
118 000000E9 83C314
                              <1>
                                        add bx, 5*4
                                                         ; hdl parameters table (INT 46h)
                              <1> L7:
119
120 000000EC 31C0
                              <1>
                                        xor ax, ax
                             <1>
<1>
121 000000EE 8ED8
                                       mov ds, ax
                                      mov si, [bx]
mov ax, [bx+2]
122 000000F0 8B37
                          <1>
<1>
123 000000F2 8B4702
124 000000F5 8ED8
                                       mov ds, ax
                                      cmp cl, [si+FDPT_SPT] ; sectors per track
jne L12 ; invalid FDPT
125 000000F7 3A4C0E
                              <1>
                             <1>
126 000000FA 0F857C01
127 000000FE BF0000
                             <1>
                                     mov di, HD0_DPT
                                       cmp dl, 80h
jna short L8
128 00000101 80FA80
                              <1>
                              <1>
129 00000104 7603
130 00000106 BF2000
                            <1>
                                       mov di, HD1_DPT
                              <1> L8:
131
                                        ; 30/12/2014
132
                              <1>
133 00000109 B80090
                              <1>
                                       mov ax, DPT_SEGM
                              <1>
134 0000010C 8EC0
                                       mov es, ax
                              <1>
                                        ; 24/12/2014
135
136 0000010E B90800
                             <1>
                                       mov cx, 8
137 00000111 F3A5
                              <1>
                                        rep movsw ; copy 16 bytes to the kernel's DPT location
138 00000113 8CC8
                              <1>
                                       mov
                                             ax, cs
                                       mov ds, ax
139 00000115 8ED8
                              <1>
140
                              <1>
                                      ; 02/02/2015
141
                              <1>
                                       ;mov cl, [drv]
142
                              <1>
                                       ;mov bl, cl
143
                               <1>
144
                               <1>
                                       ;mov ax, 1F0h
145
                               <1>
                                        ;and bl, 1
                                       jz short L9
                              <1>
146
147
                              <1>
                                       ;shl bl, 4
148
                               <1>
                                       ;sub ax, 1F0h-170h
                              <1>
149
                                       ; 17/07/2020
150
                              <1>
                                       ; (Only 1F0h port address must be valid for old ROM BIOSes)
151
                              <1>
152 00000117 B8F001
                              <1>
                                        mov ax, 1F0h
153 0000011A B3A0
                              <1>
                                        mov bl, 0A0h
154 0000011C 80FA80
                              <1>
                                       cmp dl, 80h
jna short L9
155 0000011F 7603
                              <1>
                                        ; dl = 81h
                              <1>
156
157 00000121 80C310
                              <1>
                                        add bl, 10h ; slave disk
158
                              <1>
                                        ;sub ax, 1F0h-170h
                              <1> L9:
159
160 00000124 AB
                              <1>
                                        stosw ; I/O PORT Base Address (1F0h, 170h)
                             <1>
<1>
161 00000125 050602
                                        add ax, 206h
162 00000128 AB
                                        stosw ; CONTROL PORT Address (3F6h, 376h)
163 00000129 88D8
                            <1>
                                       mov al, bl ; bit 4, master/slave disk bit
                              <1>
                                       ;add al, 0A0h ; 17/07/2020
164
165 0000012B AA
                              <1>
                                        stosb ; Device/Head Register upper nibble
                       166
167 0000012C FE06[535D]
                                     inc byte [drv]
168 00000130 BB[D85C]
                                       mov
                                             bx, hd0_type - 80h
                                       add bx, cx
                             <1>
169 00000133 01CB
170 00000135 800F80
                            <1>
                                       or byte [bx], 80h ; present sign (when lower nibble is 0)
                                       mov al, [hdc]
171 00000138 A0[555D]
                              <1>
172 0000013B FEC8
                              <1>
                                        dec
                                             al
                                        jz
173 0000013D 0F843D01
                             <1>
                              <1>
174 00000141 80FA80
                                        cmp dl, 80h
                                       jna L6 ; Max. 2 hard disks ; 17/07/2020
175 00000144 0F867CFF
                              <1>
176 00000148 E93301
                              <1>
                                         jmp L13
177
                              <1> L10:
178 0000014B FEC2
                              <1>
                                       inc
                                             dl
                                        ; 25/12/2014
179
                              <1>
180 0000014D 8816[535D]
                             <1>
                                      mov
181 00000151 B408
                              <1>
                                             ah, 08h ; Return drive parameters
                                       int 13h; (conventional function) jc L13
182 00000153 CD13
                              <1>
183 00000155 0F822501
                             <1>
                                       ; 14/01/2015
184
                              <1>
185 00000159 8A16[535D]
                              <1>
                                       mov dl, [drv]
                                       push dx
186 0000015D 52
                              <1>
187 0000015E 51
                              <1>
                                        push cx
                                             set_disk_parms
188 0000015F E82A01
                               <1>
                                        call
189 00000162 59
                               <1>
                                             CX
                                        pop
                               <1>
190 00000163 5A
                                        pop
                               <1>
                                        ; 06/07/2016 (BugFix for >64K kernel files)
191
192
                               <1>
                                        ; 04/02/2016 (esi -> si)
                                        ;mov si, _end ; 30 byte temporary buffer address
193
                               <1>
194
                               <1>
                                                   ; at the '_end' of kernel.
195
                               <1>
                                        ;mov word [si], 30
                               <1>
                                        ; 06/07/2016
196
197 00000164 BE[5000]
                               <1>
                                        mov si, _int13h_48h_buffer
198
                               <1>
                                        ; 09/07/2016
199 00000167 B81E00
                              <1>
                                        mov ax, 001Eh
200 0000016A 8824
                              <1>
                                        mov
                                             [si], ah ; 0
201 0000016C 46
                               <1>
                                        inc
                                            si
word [si], ax
202 0000016D 8904
                              <1>
                                        mov
                               <1>
                                        ; word [si] = 30
                              <1>
204
                                        ;
205 0000016F B448
                               <1>
                                        mov
                                             ah, 48h
                                                          ; Get drive parameters (EDD function)
206 00000171 CD13
                                        int 13h
                              <1>
207 00000173 0F820701
                              <1>
                                        jс
                                                 L13
                               <1>
209
                               <1>
                                       ; 29/08/2020
210
                               <1>
                                       ; 04/02/2016 (ebx -> bx)
211
                               <1>
                                        ; 14/01/2015
```

```
bh, bh
212 00000177 28FF
                               <1>
                                        sub
213 00000179 88D3
                               <1>
                                        mov
                                              bl, dl
                                        ;sub bl, 80h
                               <1>
214
215
                               <1>
                                        ; 29/08/2020
216 0000017B 81C3[D85C]
                               <1>
                                         add bx, (hd0_type - 80h)
217
                               <1>
                                         ;mov
                                              al, [bx]
                                        mov al, [bx]
218 0000017F 8A07
                               <1>
219 00000181 0C80
                                        or
                               <1>
                                              al, 80h
220 00000183 8807
                               <1>
                                               [bx], al
                                        mov
                                        sub bx, hd0_type - 2; 15/01/2015
221 00000185 81EB[565D]
                               <1>
222
                               <1>
                                       ;add bx, drv.status
223
                               <1>
                                        ;mov
                                               [bx], al
224
                               <1>
                                        ; 29/08/2020
225 00000189 8887[A25D]
                               <1>
                                        mov [bx+drv.status], al
                               <1>
                                        ; 04/02/2016 (eax -> ax)
226
                                         ;mov ax, [si+16]
227
                               <1>
                                         ; 14/07/2020
228
                               <1>
229
                               <1>
                                         ;mov di, [si+18]
230
                               <1>
                                         ;;test ax, [si+18]
                                         ;test ax, di ; 14/07/2020
231
                               <1>
232
                               <1>
                                         ;jz short L10_A0h ; (!) ; 17/07/2020
                                                     ; 'CHS only' disks on EDD system
233
                                <1>
234
                               <1>
                                                     ; are reported with ZERO disk size
235
                               <1>
                                                     ; (if so, we must not overwrite
236
                               <1>
                                                      ; calculated disk size in 'set_disk_parms')
                                         ; 29/08/2020
237
                               <1>
238 0000018D 8B4410
                               <1>
                                         mov ax, [si+16]
239 00000190 8B7C12
                               <1>
                                         mov
                                              di, [si+18]
240 00000193 09C0
                               <1>
                                         or
                                               ax, ax
241 00000195 7504
                                              short L10 LBA
                               <1>
                                         jnz
242 00000197 09FF
                               <1>
                                         or
                                               di, di
243 00000199 740B
                               <1>
                                               short L10_A0h
                                         jΖ
                               <1> L10_LBA:
244
245
                               <1>
                                    ;sub bx, drv.status
246 0000019B C1E302
                               <1>
                                         shl
                                               bx, 2
247
                               <1>
                                         ;add bx, drv.size; disk size (in sectors)
248
                               <1>
                                        ;mov [bx], ax
                               <1>
                                        ; 29/08/2020
249
250 0000019E 8987[865D]
                               <1>
                                              [bx+drv.size], ax
                                        mov
                                        ;mov ax, [si+18]
251
                               <1>
252
                               <1>
                                        ;;mov [bx], ax
253
                               <1>
                                        ; mov [bx+2], ax; BugFix; 15/07/2020
                                         ; 14/07/2020
254
                               <1>
                                        ;mov [bx+2], di ; 15/07/2020
255
                               <1>
                                        ; 29/08/2020
256
                               <1>
257 000001A2 89BF[885D]
                               <1>
                                         mov
                                              [bx+drv.size+2], di
258
                               <1> L10 A0h:
                                      ; 17/07/2020
259
                               <1>
260
                               <1>
                                         ; Note: Virtual CPU will jump here from above (!) test
261
                               <1>
                                               while running in QEMU
262
                               <1>
263
                               <1>
                                        ; Jump here to fix a ZERO (LBA) disk size problem
264
                               <1>
                                         ; for CHS disks (28/02/2015)
265
                               <1>
                               <1>
                                        ; 30/12/2014
266
267 000001A6 BF0000
                               <1>
                                         mov
                                              di, HD0_DPT
268 000001A9 88D0
                               <1>
                                               al, dl
                                         mov
269 000001AB 83E003
                               <1>
                                         and
                                               ax, 3
270 000001AE C0E005
                               <1>
                                               al, 5 ; * 32
                                         shl
271 000001B1 01C7
                               <1>
                                         add
                                              di, ax
272 000001B3 B80090
                               <1>
                                         mov
                                               ax, DPT_SEGM
273 000001B6 8EC0
                               <1>
                                               es, ax
                                        mov
274
                               <1>
275 000001B8 88E8
                               <1>
                                               al, ch; max. cylinder number (bits 0-7)
                                         mov
276 000001BA 88CC
                               <1>
                                               ah, cl
                                         mov
277 000001BC C0EC06
                               <1>
                                         shr
                                               ah, 6 ; max. cylinder number (bits 8-9)
278 000001BF 40
                               <1>
                                         inc
                                                   ; logical cylinders (limit 1024)
                                               ax
279 000001C0 AB
                               <1>
                                         stosw
280 000001C1 88F0
                               <1>
                                              al, dh; max. head number
                                         mov
281
                               <1>
                                               dh, dh ; 29/08/2020 (dh = 0 is needed here)
282 000001C3 30F6
                               <1>
                                         xor
283
                               <1>
284 000001C5 FEC0
                               <1>
                                         inc
285 000001C7 AA
                               <1>
                                         stosb
                                                     ; logical heads (limits 256)
286 000001C8 B0A0
                               <1>
                                              al, OAOh ; Indicates translated table
                                         mov
287 000001CA AA
                               <1>
                                         stosb
288 000001CB 8A440C
                               <1>
                                         mov
                                               al, [si+12]
289 000001CE AA
                               <1>
                                         stosb
                                                     ; physical sectors per track
290 000001CF 31C0
                               <1>
                                         xor
291
                               <1>
                                         ;dec ax ; 02/01/2015
292 000001D1 AB
                                                      ; precompensation (obsolete)
                               <1>
                                         stosw
                                <1>
                                         ;xor
                                               al, al ; 02/01/2015
294 000001D2 AA
                               <1>
                                         stosb
                                                    ; reserved
295 000001D3 B008
                               <1>
                                              al, 8 ; drive control byte
                                                      ; (do not disable retries,
296
                               <1>
297
                               <1>
                                                       ; more than 8 heads)
298 000001D5 AA
                              <1>
                                         stosb
                              <1>
299 000001D6 8B4404
                                         mov ax, [si+4]
                                                   ; physical number of cylinders
300 000001D9 AB
                               <1>
                                         stosw
                                                      ; 02/01/2015
                              <1>
                                         ;push ax
302 000001DA 8A4408
                              <1>
                                         mov al, [si+8]
                               <1>
303 000001DD AA
                                         stosb ; physical num. of heads (limit 16)
304 000001DE 29C0
                                         sub ax, ax
                              <1>
305
                              <1>
                                         ;pop ax ; 02/01/2015
306 000001E0 AB
                                                      ; landing zone (obsolete)
                               <1>
                                         stosw
307 000001E1 88C8
                              <1>
                                         mov al, cl ; logical sectors per track (limit 63)
308 000001E3 243F
                              <1>
                                         and al, 3Fh
309 000001E5 AA
                                         stosb
                               <1>
310
                               <1>
                                         ; sub al, al ; checksum
311
                               <1>
                                         :stosb
312
                               <1>
313 000001E6 83C61A
                               <1>
                                         add si, 26 ; (BIOS) DPTE address pointer
314 000001E9 AD
                               <1>
                                         lodsw
315 000001EA 50
                               <1>
                                         push ax ; * ; (BIOS) DPTE offset
316 000001EB AD
                               <1>
                                         lodsw
```

```
push ax ; ** ; (BIOS) DPTE segment
                               <1>
319
                                        ; checksum calculation
                               <1>
320 000001ED 89FE
                               <1>
                                        mov si, di
321 000001EF 06
                               <1>
                                        push es
                                        pop
322 000001F0 1F
                               <1>
                                              ds
                              <1>
                                        ;mov cx, 16
324 000001F1 B90F00
                              <1>
                                        mov cx, 15
325 000001F4 29CE
                              <1>
                                         sub
                                              si, cx
                                         xor ah, ah
326 000001F6 30E4
                              <1>
327
                              <1>
                                        ;del cl
                               <1> L11:
328
329 000001F8 AC
                              <1>
                                        lodsb
330 000001F9 00C4
                              <1>
                                         add ah, al
                                        loop L11
331 000001FB E2FB
                              <1>
332
                              <1>
333 000001FD 88E0
                              <1>
                                        mov al, ah
334 000001FF F6D8
                                        neg al ; -x+x = 0
                              <1>
                                         stosb
335 00000201 AA
                               <1>
                                                   ; put checksum in byte 15 of the tbl
                              <1>
                                        pop ds; ** ; (BIOS) DPTE segment
337 00000202 1F
                              <1>
                                        pop si; *; (BIOS) DPTE offset
338 00000203 5E
                               <1>
                              <1>
339
                                        ; 14/07/2020
340
                              <1>
                                        ; OFFFFh:OFFFFh = invalid DPTE address
341
                               <1>
342 00000204 8B0C
                              <1>
                                        mov cx, [si]
343 00000206 8B4402
                              <1>
                                       mov ax, [si+2]
                              <1>
344 00000209 21C1
                                       and cx, ax
                                        inc cx
jz short L11c; 0FFFFh:0FFFFh
345 0000020B 41
                              <1>
346 0000020C 7404
                              <1>
347 0000020E 0B04
                              <1>
                                        or ax, [si]
348 00000210 752A
                               <1>
                                        jnz short L11e; <> 0
                               <1> L11c:
349
                                    ; 17/07/2020
350
                               <1>
                                       ; TRDOS 386 v2 DRVINIT assumptions:
351
                               <1>
                                        ; (also by regarding QEMU, Bochs and VirtualBox settings)
352
                               <1>
                                        ; Hard disk 0 port address: 1F0h
353
                               <1>
                                              Hard disk 1 port address: 1F0h
354
                               <1>
355
                               <1>
                                              Hard disk 2 port address: 170h
                                        ; Hard disk 3 port address: 170h
356
                               <1>
357
                               <1>
                                        ; in QEMU, hda=hd0 (1F0h) and hdb=hd1 (1F0h) -IRQ14-
358
                               <1>
                                              and hdc=hd2 (170h) and hdd=hd3 (170h) -IRQ15-
359
                               <1>
360
                               <1>
361 00000212 B8F001
                               <1>
                                        mov ax, 1F0h
362
                               <1>
                                       ; 15/07/2020
363
                               <1>
                                        ; 14/07/2020
364
                               <1>
                                        ; Invalid DPTE address...
365
                               <1>
                               <1>
                                        ; Default DPTE parms must be set for DISK_IO_CONT
366
367
                               <1>
                                        ; (diskio.s)
368
                               <1>
                                        ; 17/07/2020
369
                               <1>
370
                               <1>
                                        ;mov bl, dl
                                        ;and bl, 1
371
                               <1>
372
                               <1>
                                              short L11d
                                        ;jz
373
                               <1>
374 00000215 B3A0
                                        mov bl, 0A0h
                               <1>
375
                               <1>
376 00000217 F6C201
                                         test dl, 1
                               <1>
377 0000021A 7403
                               <1>
                                         jz short L11g ; Master (as default, for 80h & 82h))
                                         ;shl bl, 4; bl = 16 (bit 4 = 1 \rightarrow slave)
                               <1>
379 0000021C 80C310
                              <1>
                                         add bl, 10h; Slave (as default, for 81h & 83h)
380
                               <1> L11g:
                                        ; 17/07/2020
381
                               <1>
                                        cmp dl, 82h
jb short L11
382 0000021F 80FA82
                               <1>
                                                           ; Hard disk 3 or 4 ?
383 00000222 7203
                                              short L11d; Primary ATA channel (hd0, hd1)
                               <1>
                               <1>
                                                       ; (port address = 1F0h)
384
385
                               <1>
386
                               <1>
                                        ; Secondary ATA channel (hd2, hd3)
387
                               <1>
                                        ; (port address = 170h)
388
                               <1>
389 00000224 2D8000
                                         sub ax, 1F0h-170h
                               <1>
390
                               <1> L11d:
                                        ; 14/07/2020
391
                               <1>
392 00000227 AB
                               <1>
                                         stosw ; I/O PORT Base Address (1F0h, 170h)
393 00000228 050602
                                        add ax, 206h
                              <1>
                                         stosw ; CONTROL PORT Address (3F6h, 376h)
394 0000022B AB
                               <1>
395 0000022C 88D8
                               <1>
                                        mov al, bl ; Master/Slave bit (0 = Master)
396
                               <1>
                                        ; 17/07/2020
                                        ;or al, 0A0h; CHS (LBA enable bit = 0)
397
                               <1>
398
                               <1>
                                                      ; (Bits 5\&7, reserved bits = 1)
                                              ah, ah
399 0000022E 30E4
                               <1>
                               <1>
                                         ;stosb; Device/Head Register upper nibble
401 00000230 AB
                               <1>
                                         stosw
402 00000231 30C0
                               <1>
                                         xor
                                              al, al
403 00000233 B90500
                              <1>
                                              cx, 5
                                        mov
404 00000236 F3AB
                              <1>
                                              stosw; clear remain part of the (fake) DPTE
                                        rep
405 00000238 OE
                               <1>
                                         push cs
406 00000239 1F
                               <1>
                                              ds
                                         pop
407 0000023A EB2E
                               <1>
                                              short L11f
                                         jmp
408
                               <1> L11e:
409
                                        ; 23/02/2015
                               <1>
410 0000023C 57
                               <1>
                                        ; ES:DI points to DPTE (FDPTE) location
411
                               <1>
412
                               <1>
                                         ;;mov cx, 8
                                        ;mov cl, 8
                               <1>
                                         mov cx, 8; 14/07/2020
414 0000023D B90800
                               <1>
415 00000240 F3A5
                               <1>
                                         rep
                                              movsw
                               <1>
416
                                         ; 23/02/2015
417
                               <1>
418
                               <1>
                                         ; (P) ATA drive and LBA validation
419
                               <1>
                                         ; (invalidating SATA drives and setting
                                         ; CHS type I/O for old type fixed disks)
420
                               <1>
421 00000242 5B
                               <1>
                                         pop bx
```

317 000001EC 50

<1>

```
422 00000243 8CC8
                               <1>
                                        mov
                                               ax, cs
423 00000245 8ED8
                               <1>
                                         mov
                                               ds, ax
424 00000247 268B07
                               <1>
                                         mov
                                               ax, [es:bx]
425 0000024A 3DF001
                              <1>
                                              ax, 1F0h
                                         cmp
426 0000024D 7413
                               <1>
                                         jе
                                               short L11a
427 0000024F 3D7001
                               <1>
                                         cmp
                                               ax, 170h
                                             short L11a
428 00000252 740E
                               <1>
                                       jе
                                       ; invalidation
                               <1>
429
430
                               <1>
                                        ; (because base port address is not 1F0h or 170h)
431
                               <1>
                                        ;xor bh, bh
432
                               <1>
                                        ;mov bl, dl
                                        ; 29/08/2020
433
                               <1>
434
                               <1>
                                        ;xor dh, dh ; 0
435 00000254 89D3
                               <1>
                                        mov bx, dx
                               <1>
436
                                        ;sub bl, 80h
                                        ;mov byte [bx+hd0_type], 0 ; not a valid disk drive !
437
                               <1>
                               <1>
                                         ;or byte [bx+drv.status+2], OFOh ; (failure sign)
                                         ; 29/08/2020
439
                               <1>
440 00000256 C687[D85C]00
                               <1>
                                         mov byte [bx+hd0_type-80h], 0
441 0000025B 808F[245D]F0
                                               byte [bx+drv.status-7Eh], 0F0h
                               <1>
                                         or
442 00000260 EB0F
                                         jmp short L11b
                               <1>
                               <1> L11a:
444
                               <1>
                                        ; LBA validation
445 00000262 268A4704
                               <1>
                                         mov al, [es:bx+4]; Head register upper nibble
                               <1>
446 00000266 A840
                                         test al, 40h; LBA bit (bit 6)
447 00000268 7507
                               <1>
                                               short L11b; LBA type I/O is OK! (E0h or F0h)
                                         jnz
448
                               <1> L11f:
                                        ; force CHS type I/O for this drive (A0h or B0h)
449
                               <1>
450
                               <1>
                                         ; sub bh, bh
                                        ;mov bl, dl
451
                               <1>
452
                               <1>
                                        ; 29/08/2020
                                        ;xor dh, dh; 0 mov bx, dx
453
                               <1>
454 0000026A 89D3
                               <1>
455
                               <1>
                                        ;sub bl, 80h; 26/02/2015
456
                               <1>
                                         ;and byte [bx+drv.status+2], OFEh ; clear bit 0
457
                               <1>
                                                            ; bit 0 = LBA ready bit
                                         ; 29/08/2020
                               <1>
459 0000026C 80A7[245D]FE
                               <1>
                                         and byte [bx+drv.status-7Eh], 0FEh
460
                               <1>
                                         ; 'diskio' procedure will check this bit !
                               <1> L11b:
461
462 00000271 3A16[545D]
                               <1>
                                         cmp dl, [last_drv]; 25/12/2014
463 00000275 7307
                               <1>
                                          jnb
                                                  short L13
464 00000277 E9D1FE
                               <1>
                                           jmp
                                                   L10
465
                               <1>
                               <1> L12:
466
467
                               <1>
                                         ; Restore data registers
468 0000027A 8CC8
                               <1>
                                         mov ax, cs
469 0000027C 8ED8
                               <1>
                                         mov ds, ax
470
                               <1> L13:
471
                               <1>
                                         ; 13/12/2014
472 0000027E 0E
                               <1>
                                         push cs
473 0000027F 07
                               <1>
                                         pop es
                               <1> L14:
474
475
                               <1>
                                         ; clear keyboard buffer
476 00000280 B411
                               <1>
                                         mov ah, 11h
477 00000282 CD16
                               <1>
                                         int
                                               16h
478 00000284 744D
                              <1>
                                               short L16 ; no keys in keyboard buffer
                                         jΖ
479 00000286 B010
                               <1>
                                         mov al, 10h
480 00000288 CD16
                               <1>
                                         int
                                               16h
481 0000028A EBF4
                               <1>
                                         jmp
                                              short L14
482
                               <1>
                               <1> set disk parms:
483
                                      ; 29/08/2020
                               <1>
484
485
                               <1>
                                        ; 04/02/2016 (ebx -> bx)
486
                               <1>
                                        ; 10/07/2015
487
                               <1>
                                         ; 14/01/2015
                               <1>
                                        ;push bx
489 0000028C 28FF
                                              bh, bh
                               <1>
                                         sub
                                               bl, [drv]
490 0000028E 8A1E[535D]
                               <1>
                                               bl, 80h
491 00000292 80FB80
                               <1>
                                         cmp
492 00000295 7203
                               <1>
                                         jb
                                               short sdp0
493 00000297 80EB7E
                               <1>
                                         sub
                                               bl, 7Eh
                               <1> sdp0:
494
                                     ;add
                                               bx, drv.status
495
                               <1>
                                         ;mov
                                               byte [bx], 80h; 'Present' flag
496
                               <1>
497
                               <1>
                                         ; 29/08/2020
498 0000029A C687[A25D]80
                               <1>
                                         mov byte [bx+drv.status], 80h
499
                               <1>
500 0000029F 88E8
                               <1>
                                               al, ch; last cylinder (bits 0-7)
501 000002A1 88CC
                                               ah, cl;
                               <1>
                                         mov
502 000002A3 C0EC06
                               <1>
                                         shr
                                               ah, 6 ; last cylinder (bits 8-9)
                                               bx, drv.status
                                <1>
                                         ;sub
504 000002A6 D0E3
                               <1>
                                               bl, 1
                               <1>
                                         ;add bx, drv.cylinders
506 000002A8 40
                                              ax ; convert max. cyl number to cyl count
                               <1>
                                         inc
507
                               <1>
                                         ;mov
                                              [bx], ax
                                         ; 29/08/2020
                               <1>
509 000002A9 8987[5C5D]
                               <1>
                                              [bx+drv.cylinders], ax
                                        mov
510 000002AD 50
                               <1>
                                              ax ; ** cylinders
                                        push
511
                               <1>
                                         ; sub bx, drv.cylinders
512
                               <1>
                                         ;add bx, drv.heads
513 000002AE 30E4
                               <1>
                                         xor
                                               ah, ah
514 000002B0 88F0
                                               al, dh ; heads
                               <1>
                                         mov
515 000002B2 40
                               <1>
                                         inc ax
516
                               <1>
                                         ;mov [bx], ax
517
                               <1>
                                         ; 29/08/2020
518 000002B3 8987[6A5D]
                               <1>
                                         mov [bx+drv.heads], ax
                                         ;sub
                                          ; sub bx, drv.heads
; add bx, drv.spt
519
                               <1>
520
                               <1>
                                         xor ch, ch
521 000002B7 30ED
                              <1>
522 000002B9 80E13F
                              <1>
                                         and cl, 3Fh
                                                            ; sectors (bits 0-6)
                               <1>
                                        ;mov [bx], cx
                                         ; 29/08/2020
524
                               <1>
525 000002BC 898F[785D]
                               <1>
                                         mov [bx+drv.spt], cx
                               <1>
526
                                         ; sub bx, drv.spt
```

```
<1>
                                          ;add bx, drv.size; disk size (in sectors)
                                          ; LBA size = cylinders * heads * secpertrack
529
                                <1>
530 000002C2 F7E1
                                <1>
                                          mul cx
531 000002C4 89C2
                                <1>
                                          mov
                                               dx, ax; heads*spt
                                                ax ; ** cylinders
532 000002C6 58
                                <1>
                                          pop
533 000002C7 48
                               <1>
                                          dec ax ; 1 cylinder reserved (!?)
534 000002C8 F7E2
                                               dx ; cylinders * (heads*spt)
                                <1>
                                         mul
535
                                <1>
                                          ;mov
                                               [bx], ax
                                               [bx+2], dx
536
                                <1>
                                         ;mov
                                         ; 29/08/2020
537
                                <1>
538 000002CA 8987[865D]
                                <1>
                                         mov
                                               [bx+drv.size], ax
539 000002CE 8997[885D]
                                <1>
                                                [bx+drv.size+2], dx
                                          mov
540
                                <1>
                                <1>
541
                                          ;pop bx
542 000002D2 C3
                                <1>
                                          retn
543
                                <1>
                                <1> L16: ; 28/05/2016
544
204
205
                                          ; 10/11/2014
206 000002D3 FA
                                          cli ; Disable interrupts (clear interrupt flag)
                                                ; Reset Interrupt MASK Registers (Master&Slave)
207
208
                                          ;mov al, OFFh ; mask off all interrupts
209
                                          ;out 21h, al
                                                                ; on master PIC (8259)
                                          ;jmp $+2 ; (delay)
;out 0A1h, al ; on slave PIC (8259)
210
211
212
                                          ; Disable NMI
213
214 000002D4 B080
                                               al, 80h
                                          mov
                                               70h, al
215 000002D6 E670
                                          out
                                                                   ; set bit 7 to 1 for disabling NMI
216
                                          ;23/02/2015
217
                                          ;nop
218
                                                al, 71h
                                                                   ; read in 71h just after writing out to 70h
                                          ;in
219
                                                             ; for preventing unknown state (!?)
220
                                          ; 20/08/2014
221
                                          ; Moving the kernel 64 KB back (to physical address 0)
222
                                          ; DS = \overline{CS} = 1000h
223
224
                                          ; 05/11/2014
225 000002D8 31C0
                                          xor ax, ax
226 000002DA 8EC0
                                          mov es, ax; ES = 0
227
                                          ; 04/07/2016 - TRDOS 386 (64K - 128K kernel)
228
229 000002DC 31F6
                                                xor si, si
230 000002DE 31FF
                                                di, di
                                          xor
231 000002E0 B90040
                                          mov
                                                cx, 16384
232 000002E3 F366A5
                                                movsd
                                          rep
233
234 000002E6 06
                                          push es; 0
235 000002E7 68[EB02]
                                          push L17
236 000002EA CB
                                    L17:
238 000002EB B90010
                                          mov
                                                cx, 1000h
239 000002EE 8EC1
                                                es, cx ; 1000h
240 000002F0 01C9
                                          add
                                                CX, CX
241 000002F2 8ED9
                                          mov
                                                ds, cx ; 2000h
242 000002F4 29F6
                                          sub
                                                si, si
243 000002F6 29FF
                                                di, di
                                          sub
244 000002F8 B90040
                                          mov
                                                cx, 16384
245 000002FB F366A5
                                          rep
                                                movsd
246
                                          ; Turn off the floppy drive motor
248 000002FE BAF203
                                            mov dx, 3F2h
249 00000301 EE
                                                   dx, al ; 0 ; 31/12/2013
250
251
                                          ; Enable access to memory above one megabyte
253 00000302 E464
                                                al, 64h
                                          in
254 00000304 A802
                                          test al, 2
255 00000306 75FA
                                          jnz short L18
256 00000308 B0D1
                                          mov al, OD1h ; Write output port
257 0000030A E664
                                                64h, al
                                          out
258
                                    L19:
259 0000030C E464
                                          in
                                                al, 64h
260 0000030E A802
                                          test al, 2
261 00000310 75FA
                                          jnz short L19
262 00000312 BODF
                                          mov al, ODFh ; Enable A20 line
263 00000314 E660
                                          out
                                               60h, al
264
                                    ;L20:
265
266
                                          ; Load global descriptor table register
267
268
                                            ; mov
                                                     ax, cs
                                            ;mov
                                                     ds, ax
270
271 00000316 2E0F0116[C05C]
                                            lgdt
                                                    [cs:gdtd]
273 0000031C 0F20C0
                                            mov
                                                    eax, cr0
274
                                          ; or eax, 1
275 0000031F 40
                                          inc
                                                 ax
276 00000320 0F22C0
                                                 cr0, eax
                                          mov
278
                                          ; Jump to 32 bit code
279
280 00000323 66
                                          db 66h
                                                                    ; Prefix for 32-bit
281 00000324 EA
                                          db 0EAh
                                                            ; Opcode for far jump
282 00000325 [2B030000]
                                          dd StartPM
                                                             ; Offset to start, 32-bit
                                                            ; (1000h:StartPM = StartPM + 10000h)
283
                                          dw KCODE
284 00000329 0800
                                                             ; This is the selector for CODE32_DESCRIPTOR,
285
                                                             ; assuming that StartPM resides in code32
286
287
                                    ; 20/02/2017
288
289
290
                                    [BITS 32]
```

527 000002C0 D1E3

<1>

shl bx, 1

```
291
292
                                     StartPM:
293
                                          ; Kernel Base Address = 0 ; 30/12/2013
294 0000032B 66B81000
                                          mov ax, KDATA ; Save data segment identifier
                                                                  ; Move a valid data segment into DS register
295 0000032F 8ED8
                                           mov ds, ax
                                                                     ; Move data segment into ES register
296 00000331 8EC0
                                               mov es, ax
                                                                        ; Move data segment into FS register
297 00000333 8EE0
                                                mov fs, ax
                                                                       ; Move data segment into GS register
                                                mov gs, ax
298 00000335 8EE8
299 00000337 8ED0
                                                                    ; Move data segment into SS register
                                             mov ss, ax
                                            mov esp, 90000h
300 00000339 BC00000900
                                                                    ; Move the stack pointer to 090000h
301
302
                                     clear bss: ; Clear uninitialized data area
303
                                          ; 11/03/2015
304 0000033E 31C0
                                           xor eax, eax; 0
305 00000340 B9F26E0000
                                          mov ecx, (bss_end - bss_start)/4
306
                                          ;shr ecx, 2; bss section is already aligned for double words
307 00000345 BF[D2550100]
                                          mov edi, bss start
308 0000034A F3AB
                                          rep stosd
309
310
                                     memory_init:
311
                                         ; Initialize memory allocation table and page tables
312
                                          ; 16/11/2014
313
                                          ; 15/11/2014
314
                                          ; 07/11/2014
315
                                          ; 06/11/2014
316
                                           ; 05/11/2014
317
                                           ; 04/11/2014
                                           ; 31/10/2014 (Retro UNIX 386 v1 - Beginning)
318
319
320
                                           xor
                                                 eax, eax
321
                                           xor
                                                 ecx, ecx
322 0000034C B108
                                           mov
                                                 cl, 8
323 0000034E BF00001000
                                                 edi, MEM_ALLOC_TBL
                                           mov
324 00000353 F3AB
                                                           ; clear Memory Allocation Table
                                           rep
325
                                                                 ; for the first 1 MB memory
326
327 00000355 668B0D[4E5D0000]
                                                 cx, [mem 1m 1k]
                                                                      ; Number of contiguous KB between
                                          mov
                                                                ; 1 and 16 MB, max. 3C00h = 15 MB.
328
329 0000035C 66C1E902
                                           shr
                                                                 ; convert 1 KB count to 4 KB count
330 00000360 890D[C8580100]
                                                 [free_pages], ecx
                                          mov
331 00000366 668B15[505D0000]
                                                 dx, [mem_16m_64k] ; Number of contiguous 64 KB blocks
                                          mov
                                                                 ; between 16 MB and 4 GB.
333 0000036D 6609D2
                                                 dx, dx
                                           or
334 00000370 7413
                                                 short mi_0
                                          jz
335
                                          ;
336 00000372 6689D0
                                          mov
                                                 ax, dx
337 00000375 C1E004
                                                 eax, 4
                                          shl
                                                                 ; 64 KB -> 4 KB (page count)
338 00000378 0105[C8580100]
                                                 [free_pages], eax
                                           add
                                                 eax, 4096
                                                               ; 16 \text{ MB} = 4096 \text{ pages}
339 0000037E 0500100000
                                           add
340 00000383 EB07
                                                 {	t short mi\_1}
                                           jmp
341
                                     mi_0:
342 00000385 6689C8
                                          mov
                                                 ax, cx
                                                                        ; add 256 pages for the first 1 MB
343 00000388 66050001
                                           add
                                                 ax, 256
                                     mi_1:
345 0000038C A3[C4580100]
                                                 [memory_size], eax ; Total available memory in pages
                                          mov
346
                                                                 ; 1 alloc. tbl. bit = 1 memory page
                                                                 ; 32 allocation bits = 32 mem. pages
348
349 00000391 05FF7F0000
                                           add
                                                  eax, 32767
                                                                 ; 32768 memory pages per 1 M.A.T. page
                                                                    ; ((32768 * x) + y) pages (y < 32768)
350 00000396 C1E80F
                                           shr
                                                 eax, 15
351
                                                                   --> x + 1 M.A.T. pages, if y > 0
                                                                 ; --> x M.A.T. pages, if y = 0
                                                                 ; Memory Alloc. Table Size in pages
; 1 M.A.T. page = 4096 bytes
353 00000399 66A3[D8580100]
                                           mov
                                                 [mat_size], ax
                                                 eax, 12
354 0000039F C1E00C
                                           shl
                                                                 ; Max. 32 M.A.T. pages (4 GB memory)
355
                                           ;
356 000003A2 89C3
                                           mov
                                                                 ; M.A.T. size in bytes
                                          ; Set/Calculate Kernel's Page Directory Address
358 000003A4 81C300001000
                                           add ebx, MEM ALLOC TBL
359 000003AA 891D[C0580100]
                                                 [k_page_dir], ebx ; Kernel's Page Directory address
                                          mov
360
                                                                 ; just after the last M.A.T. page
361
362 000003B0 83E804
                                                                 ; convert M.A.T. size to offset value
                                           sub
                                                 eax, 4
                                                 [last_page], eax ; last page ofset in the M.A.T.
363 000003B3 A3[D0580100]
                                           mov
                                                                ; (allocation status search must be
364
                                           ;
365
                                                                 ; stopped after here)
366 000003B8 31C0
                                           xor
                                                 eax, eax
367 000003BA 48
                                           dec
                                                                 ; FFFFFFFFh (set all bits to 1)
                                                eax
368 000003BB 6651
                                           push cx
369 000003BD C1E905
                                                                 ; convert 1 - 16 MB page count to
                                                 ecx, 5
370
                                                                 ; count of 32 allocation bits
371 000003C0 F3AB
                                           rep
                                                 stosd
372 000003C2 6659
                                           pop
                                                 CX
373 000003C4 40
                                           inc
                                                 eax
                                                                 ; 0
374 000003C5 80E11F
                                                 cl, 31
                                                                 ; remain bits
375 000003C8 7412
                                                 short mi 4
                                           jΖ
                                                                 ; reset
376 000003CA 8907
                                           mov
                                                 [edi], eax
                                     mi 2:
378 000003CC 0FAB07
                                                 [edi], eax
                                                                 ; 06/11/2014
                                           bts
379 000003CF FEC9
                                           dec
                                                 cl
380 000003D1 7404
                                                 short mi_3
                                           jΖ
381 000003D3 FEC0
                                           inc
                                                 al
382 000003D5 EBF5
                                           qmj
                                                 short mi 2
                                     mi_3:
383
                                                 al, al
                                                                 ; 0
384 000003D7 28C0
                                           sub
385 000003D9 83C704
                                           add
                                                 edi, 4
                                                                 ; 15/11/2014
386
                                     mi_4:
387 000003DC 6609D2
                                                                ; check 16M to 4G memory space
                                                  dx, dx
                                           or
388 000003DF 7421
                                                 short mi_6
                                                                ; max. 16 MB memory, no more...
                                           jΖ
389
390 000003E1 B900021000
                                                 ecx, MEM_ALLOC_TBL + 512 ; End of first 16 MB memory
                                           mov
391
392 000003E6 29F9
                                           sub
                                                 ecx, edi
                                                                ; displacement (to end of 16 MB)
393 000003E8 7406
                                           jz
                                                  short mi_5
                                                                ; jump if EDI points to
                                                                ; end of first 16 MB
394
395 000003EA D1E9
                                                 ecx. 1
                                                                ; convert to dword count
                                           shr
```

```
396 000003EC D1E9
                                                               ; (shift 2 bits right)
                                          shr
                                                 ecx, 1
397 000003EE F3AB
                                          rep
                                                stosd
                                                               ; reset all bits for reserved pages
398
                                                               ; (memory hole under 16 MB)
399
400 000003F0 6689D1
                                                 cx. dx
                                                               ; count of 64 KB memory blocks
                                          mov
401 000003F3 D1E9
                                          shr
                                                 ecx, 1
                                                                ; 1 alloc. dword per 128 KB memory
402 000003F5 9C
                                                                ; 16/11/2014
                                          pushf
403 000003F6 48
                                                                ; FFFFFFFFh (set all bits to 1)
                                          dec
                                                 eax
404 000003F7 F3AB
                                                 stosd
                                          rep
405 000003F9 40
                                          inc
                                                 eax
406 000003FA 9D
                                                                ; 16/11/2014
                                          popf
407 000003FB 7305
                                          jnc
                                                short mi 6
408 000003FD 6648
                                                                ; eax = 0000FFFFh
                                          dec
                                                ax
409 000003FF AB
                                          stosd
410 00000400 6640
                                                               ; 0
                                          inc
                                                ax
411
                                    mi_6:
                                                               ; check if EDI points to
412 00000402 39DF
                                                 edi, ebx
                                          cmp
413 00000404 730A
                                                 short mi 7
                                                               ; end of memory allocation table
                                          jnb
414
                                                               ; (>= MEM_ALLOC_TBL + 4906)
415 00000406 89D9
                                                               ; end of memory allocation table
                                                 ecx, ebx
                                          mov
416 00000408 29F9
                                          sub
                                                 ecx, edi
                                                               ; convert displacement/offset
417 0000040A D1E9
                                          shr
                                                 ecx, 1
                                                               ; to dword count
418 0000040C D1E9
                                                ecx, 1
                                                               ; (shift 2 bits right)
                                          shr
419 0000040E F3AB
                                          rep
                                                               ; reset all remain M.A.T. bits
420
                                    mi 7:
                                          ; Reset M.A.T. bits in M.A.T. (allocate M.A.T. pages)
421
                                          mov edx, MEM_ALLOC_TBL
422 00000410 BA00001000
                                                             ; Mem. Alloc. Tbl. size in bytes
                                          ;sub ebx, edx
423
                                                ebx, 12
                                                                     ; Mem. Alloc. Tbl. size in pages
                                          ;shr
                                                                      ; Mem. Alloc. Tbl. size in pages
425 00000415 668B0D[D8580100]
                                                cx, [mat_size]
                                          mov
426 0000041C 89D7
                                          mov
                                                edi, edx
427 0000041E C1EF0F
                                          shr
                                                edi, 15
                                                                     ; convert M.A.T. address to
                                                                ; byte offset in M.A.T.
428
429
                                                                ; (1 M.A.T. byte points to
                                                                ; 32768 bytes)
430
                                                                ; Note: MEM_ALLOC_TBL address
431
                                                                ; must be aligned on 128 KB
433
                                                                ; boundary!
434 00000421 01D7
                                          add edi, edx
                                                               ; points to M.A.T.'s itself
                                          ; eax = 0
435
436 00000423 290D[C8580100]
                                          sub
                                                [free_pages], ecx; 07/11/2014
437
                                    mi 8:
438 00000429 OFB307
                                                 [edi], eax
                                                             ; clear bit 0 to bit x (1 to 31)
                                          btr
                                          ;dec bl
439
440 0000042C FEC9
                                          dec
                                                cl
441 0000042E 7404
                                          jz
                                                short mi_9
                                                al
442 00000430 FEC0
                                          inc
443 00000432 EBF5
                                                short mi_8
                                          jmp
444
                                    mi_9:
445
446
                                          ; Reset Kernel's Page Dir. and Page Table bits in M.A.T.
447
                                                     (allocate pages for system page tables)
448
449
                                          ; edx = MEM ALLOC TBL
450 00000434 8B0D[C4580100]
                                          mov ecx, [memory_size] ; memory size in pages (PTEs)
                                                            ; round up (1024 PTEs per table)
451 0000043A 81C1FF030000
                                          add
                                                ecx, 1023
452 00000440 C1E90A
                                          shr
                                                ecx, 10
                                                                 ; convert memory page count to
                                                               ; page table count (PDE count)
453
454
455 00000443 51
                                          push
                                                              ; (**) PDE count (<= 1024)
                                                ecx
456
457 00000444 41
                                          inc
                                                              ; +1 for kernel page directory
458
459 00000445 290D[C8580100]
                                          sub
                                                 [free_pages], ecx; 07/11/2014
460
                                          ;
461 0000044B 8B35[C0580100]
                                          mov
                                                 esi, [k_page_dir] ; Kernel's Page Directory address
462 00000451 C1EE0C
                                          shr
                                                 esi, 12
                                                                    ; convert to page number
                                    mi_10:
463
                                                 eax, esi
464 00000454 89F0
                                                              ; allocation bit offset
                                          mov
465 00000456 89C3
                                                 ebx, eax
                                          mov
                                                 ebx, 3
                                                               ; convert to alloc. byte offset
466 00000458 C1EB03
                                          shr
467 0000045B 80E3FC
                                                 bl, OFCh
                                                               ; clear bit 0 and bit 1
                                          and
                                                               ; to align on dword boundary
468
469 0000045E 83E01F
                                                 eax, 31
                                                                     ; set allocation bit position
                                                               ; (bit 0 to bit 31)
470
471
472 00000461 01D3
                                          add
                                                 ebx, edx
                                                              ; offset in M.A.T. + M.A.T. address
473
474 00000463 OFB303
                                                 [ebx], eax
                                                               ; reset relevant bit (0 to 31)
                                          btr
475
476 00000466 46
                                          inc
                                                 esi
                                                               ; next page table
                                                               ; allocate next kernel page table
477 00000467 E2EB
                                          loop
                                                mi 10
478
                                                               ; (ecx = page table count + 1)
479
480 00000469 59
                                                               ; (**) PDE count (= pg. tbl. count)
                                          pop
                                                ecx
481
                                          ; Initialize Kernel Page Directory and Kernel Page Tables
483
484
                                          ; Initialize Kernel's Page Directory
485 0000046A 8B3D[C0580100]
                                          mov edi, [k_page_dir]
486 00000470 89F8
                                          mov
                                                eax, edi
487 00000472 0C03
                                          or
                                                al, PDE A PRESENT + PDE A WRITE
                                                             ; supervisor + read&write + present
488
489 00000474 89CA
                                                 edx, ecx
                                                             ; (**) PDE count (= pg. tbl. count)
490
                                    mi_11:
491 00000476 0500100000
                                                            ; Add page size (PGSZ)
                                          add
                                                 eax, 4096
                                                              ; EAX points to next page table
493 0000047B AB
                                          stosd
494 0000047C E2F8
                                          loop mi 11
495 0000047E 29C0
                                                eax, eax
                                                             ; Empty PDE
                                          sub
496 00000480 66B90004
                                          mov
                                                cx, 1024
                                                             ; Entry count (PGSZ/4)
497 00000484 29D1
                                          sub
                                                ecx, edx
498 00000486 7402
                                          jz
                                                 short mi_12
499 00000488 F3AB
                                                            ; clear remain (empty) PDEs
                                          rep
500
```

```
501
                                          ; Initialization of Kernel's Page Directory is OK, here.
502
                                    mi 12:
                                         ; Initialize Kernel's Page Tables
503
504
505
                                         ; (EDI points to address of page table 0)
                                          ; eax = 0
506
507 0000048A 8B0D[C4580100]
                                         mov ecx, [memory_size] ; memory size in pages
508 00000490 89CA
                                         mov edx, ecx ; (***)
509 00000492 B003
                                               al, PTE_A_PRESENT + PTE_A_WRITE
                                         mov
                                                        ; supervisor + read&write + present
510
511
                                    mi_13:
512 00000494 AB
                                         stosd
513 00000495 0500100000
                                          add eax, 4096
514 0000049A E2F8
                                          loop mi_13
                                                             ; (***)
515 0000049C 6681E2FF03
                                          and dx, 1023
516 000004A1 740B
                                                short mi 14
                                          jz
517 000004A3 66B90004
                                          mov
                                               cx, 1024
518 000004A7 6629D1
                                          sub
                                               cx, dx
                                                            ; from dx (<= 1023) to 1024
519 000004AA 31C0
                                          xor
                                                eax, eax
520 000004AC F3AB
                                               stosd
                                                            ; clear remain (empty) PTEs
                                         rep
521
                                                            ; of the last page table
522
                                    mi 14:
                                         ; Initialization of Kernel's Page Tables is OK, here.
523
524
525 000004AE 89F8
                                                            ; end of the last page table page
                                                eax, edi
                                          mov
526
                                                             ; (beginging of user space pages)
                                                eax, 15
527 000004B0 C1E80F
                                          shr
                                                                  ; convert to M.A.T. byte offset
                                                al, OFCh
                                                            ; clear bit 0 and bit 1 for
528 000004B3 24FC
                                          and
529
                                                             ; aligning on dword boundary
530
531 000004B5 A3[D4580100]
                                          mov
                                                [first_page], eax
532 000004BA A3[CC580100]
                                         mov
                                                [next_page], eax ; The first free page pointer
                                                             ; for user programs
533
534
                                                              ; (Offset in Mem. Alloc. Tbl.)
535
536
                                          ; Linear/FLAT (1 to 1) memory paging for the kernel is OK, here.
537
                                          ;
538
                                          ; Enable paging
539
540
                                          ;
541 000004BF A1[C0580100]
                                          mov
                                                   eax, [k_page_dir]
542 000004C4 0F22D8
                                          mov cr3, eax
543 000004C7 0F20C0
                                               eax, cr0
                                          mov
544 000004CA 0D00000080
                                          or eax, 80000000h ; set paging bit (bit 31)
545 000004CF 0F22C0
                                          mov cr0, eax
                                           ;jmp KCODE:StartPMP
546
547
548 000004D2 EA
                                          db 0EAh
                                                            ; Opcode for far jump
                                                           ; 32 bit offset
549 000004D3 [D9040000]
                                           dd StartPMP
550 000004D7 0800
                                          dw KCODE
                                                            ; kernel code segment descriptor
551
552
                                    StartPMP:
553
                                         ; 06/11//2014
554
555
                                          ; Clear video page 0
556
                                          ; Temporary Code
558
559 000004D9 B9E8030000
                                               ecx, 80*25/2
                                              edi, 0B8000h
560 000004DE BF00800B00
                                          mov
561
                                          ; 30/01/2016
562
                                                            ; black background, black fore color
                                          ;xor eax, eax
563 000004E3 B800070007
                                               eax, 07000700h ; black background, light gray fore color
                                          mov
564 000004E8 F3AB
                                          rep stosd
565
566
                                          ; 19/08/2014
567
                                         ; Kernel Base Address = 0
                                          ; It is mapped to (physically) 0 in the page table.
568
569
                                          ; So, here is exactly 'StartPMP' address.
570
571
                                          ; 29/01/2016 (TRDOS 386 = TRDOS v2.0)
572 000004EA BE[1D1A0100]
                                          mov esi, starting msg
                                          ;; 14/08/2015 (kernel version message will appear
573
                                                       when protected mode and paging is enabled)
574
                                          ;;
575 000004EF BF00800B00
                                               edi, 0B8000h ; 27/08/2014
                                          mov
576 000004F4 B40A
                                          mov
                                               ah, OAh ; Black background, light green forecolor
                                          ; 20/08/2014
578 000004F6 E88F010000
                                          call printk
579
580
                                          ; 'UNIX v7/x86' source code by Robert Nordier (1999)
581
                                          ; // Set IRQ offsets
582
583
                                          ; Linux (v0.12) source code by Linus Torvalds (1991)
585
                                                                   ;; ICW1
586 000004FB B011
                                          mov al, 11h
                                                                         ; Initialization sequence
587 000004FD E620
                                          out 20h, al
                                                                          ; 8259A-1
                                          ; jmp $+2
588
589 000004FF E6A0
                                               0A0h, al
                                                                        8259A-2
                                                                   ;; ICW2
590
591 00000501 B020
                                                                         ; Start of hardware ints (20h)
                                          mov
                                               al, 20h
                                                21h, al
592 00000503 E621
                                          out
                                                                         ; for 8259A-1
                                          ; jmp $+2
593
594 00000505 B028
                                          mov al, 28h
                                                                         ; Start of hardware ints (28h)
                                                                         for 8259A-2
595 00000507 E6A1
                                          out
                                               0A1h, al
                                                                   ;
596
597 00000509 B004
                                               al, 04h
                                                                        ;; ICW3
                                          mov
598 0000050B E621
                                          out 21h, al
                                                                               IRQ2 of 8259A-1 (master)
                                                                         ;
                                          ; jmp $+2
599
600 0000050D B002
                                          mov al, 02h
                                                                               is 8259A-2 (slave)
601 0000050F E6A1
                                          out
                                                0A1h, al
                                                                   ;; ICW4
603 00000511 B001
                                          mov
                                               al, 01h
                                          out 21h, al
604 00000513 E621
                                                                                8086 mode, normal EOI
605
                                          ; jmp $+2
```

```
606 00000515 E6A1
                                         out OA1h, al
                                                                       for both chips.
607
608
                                         ;mov al, OFFh ; mask off all interrupts for now
                                         out 21h, al
609
610
                                         ;; jmp $+2
611
                                         ;out OA1h, al
612
                                         ; 02/04/2015
613
614
                                         ; 26/03/2015 System call (INT 30h) modification
                                         ; DPL = 3 (Interrupt service routine can be called from user mode)
615
616
                                         ;; Linux (v0.12) source code by Linus Torvalds (1991)
617
618
                                         ; setup_idt:
619
620
                                          ;; 16/02/2015
                                         ;;mov dword [DISKETTE_INT], fdc_int ; IRQ 6 handler
621
                                         ; 21/08/2014 (timer int)
                                         mov esi, ilist lea edi, [idt]
623 00000517 BE[E0160100]
624 0000051C 8D3D[D8550100]
                                         ; 26/03/2015
626 00000522 B930000000
                                         mov ecx, 48
                                                                ; 48 hardware interrupts (INT 0 to INT 2Fh)
                                         ; 02/04/2015
                                         mov ebx, 80000h
628 00000527 BB00000800
                                    rp_sidt1:
629
630 0000052C AD
                                         lodsd
631 0000052D 89C2
                                         mov edx, eax
632 0000052F 66BA008E
                                         mov
                                              dx, 8E00h
633 00000533 6689C3
                                         mov bx, ax
                                                           ; /* selector = 0x0008 = cs */
634 00000536 89D8
                                         mov eax, ebx
635
                                                                ; /* interrupt gate - dpl=0, present */
636 00000538 AB
                                        stosd ; selector & offset bits 0-15
637 00000539 89D0
                                         mov eax, edx
                                         stosd ; attributes & offset bits 16-23
638 0000053B AB
639 0000053C E2EE
                                         loop rp_sidt1
                                         ; 15/04/2016
640
                                         ; TRDOS 386 (TRDOS v2.0) /// 32 sofware interrupts ///
641
                                         ;mov cl, 16 ; 16 software interrupts (INT 30h to INT 3Fh)
642
643 0000053E B120
                                         mov cl, 32
                                                           ; 32 software interrupts (INT 30h to INT 4Fh)
644
                                   rp_sidt2:
645 00000540 AD
                                        lodsd
646 00000541 21C0
                                         and eax, eax
647 00000543 7413
                                         jz
                                                short rp sidt3
648 00000545 89C2
                                               edx, eax
                                         mov
                                               dx, 0EE00h ; P=1b/DPL=11b/01110b
649 00000547 66BA00EE
                                         mov
650 0000054B 6689C3
                                         mov
                                               bx, ax
651 0000054E 89D8
                                         mov
                                               eax, ebx
                                                          ; selector & offset bits 0-15
652 00000550 AB
                                         stosd
653 00000551 89D0
                                         mov eax, edx
654 00000553 AB
                                         stosd
                                         loop rp sidt2
655 00000554 E2EA
656 00000556 EB16
                                         jmp
                                               short sidt_OK
                                   rp_sidt3:
658 00000558 B8[B30A0000]
                                        mov
                                                eax, ignore_int
659 0000055D 89C2
                                         mov
                                               edx, eax
660 0000055F 66BA00EE
                                                dx, 0EE00h ; P=1b/DPL=11b/01110b
                                        mov
661 00000563 6689C3
                                         mov
                                               bx, ax
662 00000566 89D8
                                               eax, ebx
                                                          ; selector & offset bits 0-15
                                         mov
663
                                   rp_sidt4:
664 00000568 AB
                                         stosd
665 00000569 92
                                         xchg eax, edx
666 0000056A AB
                                         stosd
667 0000056B 92
                                         xchg edx, eax
668 0000056C E2FA
                                         loop rp_sidt4
                                    sidt_OK:
669
670 0000056E 0F011D[C65C0000]
                                         lidt [idtd]
671
                                         ; TSS descriptor setup ; 24/03/2015
673 00000575 B8[58580100]
                                        mov eax, task_state_segment
674 0000057A 66A3[BA5C0000]
                                                [gdt_tss0], ax
                                         mov
                                        rol
675 00000580 C1C010
                                               eax, 16
                                               [gdt tss1], al
676 00000583 A2[BC5C0000]
                                        mov
677 00000588 8825[BF5C0000]
                                               [gdt_tss2], ah
                                         mov
678 0000058E 66C705[BE580100]68-
                                               word [tss.IOPB], tss_end - task_state_segment
                                         mov
678 00000596 00
679
                                                ; IO Map Base address (When this address points
680
681
                                                ; to end of the TSS, CPU does not use IO port
                                                ; permission bit map for RING 3 IO permissions,
682
683
                                                ; access to any IO ports in ring 3 will be forbidden.)
684
685
                                          ;mov [tss.esp0], esp ; TSS offset 4
                                          ; mov
                                                word [tss.ss0], KDATA; TSS offset 8 (SS)
                                                ax, TSS ; It is needed when an interrupt
687 00000597 66B82800
                                         mov
                                                       ; occurs (or a system call -software INT- is requested)
                                                       ; while cpu running in ring 3 (in user mode).
689
690
                                                       ; (Kernel stack pointer and segment will be loaded
                                                       ; from offset 4 and 8 of the TSS, by the CPU.)
692 0000059B 0F00D8
                                               ax ; Load task register
                                         ltr
693
694
                                    esp0_set0:
                                         ; 30/07/2015
695
                                         mov
696 0000059E 8B0D[C4580100]
                                               ecx, [memory size] ; memory size in pages
697 000005A4 C1E10C
                                                ecx, 12 ; convert page count to byte count
                                         shl
698 000005A7 81F900004000
                                                ecx, CORE; beginning of user's memory space (400000h)
                                                       ; (kernel mode virtual address)
699
700 000005AD 7605
                                         jna
                                                short esp0_set1
702
                                         ; If available memory > CORE (end of the 1st 4 MB)
                                         ; set stack pointer to CORE
703
                                         ; (Because, PDE 0 is reserved for kernel space in user's page directory)
704
                                         ;(PDE 0 points to page table of the 1st 4 MB virtual address space)
705
706 000005AF B900004000
                                         mov ecx, CORE
707
                                    esp0_set1:
708 000005B4 89CC
                                               esp, ecx; top of kernel stack (**tss.esp0**)
                                        mov
709
                                    esp0_set_ok:
```

```
; 30/07/2015 (**tss.esp0**)
710
711 000005B6 8925[5C580100]
                                         mov [tss.esp0], esp
712 000005BC 66C705[60580100]10-
                                          mov word [tss.ss0], KDATA
712 000005C4 00
713
                                          ; 14/08/2015
714
                                          ; 10/11/2014 (Retro UNIX 386 v1 - Erdogan Tan)
715
716
                                          ;cli ; Disable interrupts (for CPU)
717
                                              (CPU will not handle hardware interrupts, except NMI!)
718
719 000005C5 30C0
                                          xor
                                               al, al
                                                            ; Enable all hardware interrupts!
720 000005C7 E621
                                                21h, al
                                                                  ; (IBM PC-AT compatibility)
                                          out
721 000005C9 EB00
                                                             ; (All conventional PC-AT hardware
                                          jmp
                                               $+2
722 000005CB E6A1
                                               0A1h, al
                                                            ; interrupts will be in use.)
723
                                                            ; (Even if related hardware component
724
                                                             ; does not exist!)
                                         ; Enable NMI
725
726 000005CD B07F
                                          mov al, 7Fh
                                                                   ; Clear bit 7 to enable NMI (again)
                                                70h, al
727 000005CF E670
                                          out
                                          ; 23/02/2015
729 000005D1 90
                                          nop
730 000005D2 E471
                                          in
                                                al, 71h
                                                                   ; read in 71h just after writing out to 70h
731
                                                             ; for preventing unknown state (!?)
732
733
                                          ; Only a NMI can occur here... (Before a 'STI' instruction)
734
                                          ; 02/09/2014
736 000005D4 6631DB
                                          xor bx, bx
                                          mov dx, 0200h call _set_cpos
737 000005D7 66BA0002
                                                            ; Row 2, column 0 ; 07/03/2015
738 000005DB E871170000
                                                           ; 24/01/2016
739
                                          ; 06/11/2014
741 000005E0 E8782C0000
                                          call memory_info
742
                                          ; 14/08/2015
                                          ; call getch ; 28/02/2015
743
744
                                    drv_init:
745 000005E5 FB
                                                ; Enable Interrupts
                                          ; 06/02/2015
746
747 000005E6 8B15[585D0000]
                                          mov edx, [hd0_type]; hd0, hd1, hd2, hd3
748 000005EC 668B1D[565D0000]
                                                bx, [fd0_type]; fd0, fd1
                                         mov
                                          ; 22/02/2015
749
750 000005F3 6621DB
                                          and
                                               bx, bx
751 000005F6 751C
                                               short dil
                                          jnz
752
753 000005F8 09D2
                                          or
                                                edx, edx
754 000005FA 752A
                                          jnz
                                               short di2
                                         ;
756
                                    setup_error:
757 000005FC BE[E6190100]
                                          mov esi, setup_error_msg
                                    psem:
759 00000601 AC
                                          lodsb
760 00000602 08C0
                                          or
                                               al, al
                                               short haltx ; 22/02/2015
761
                                          ;jz
                                          jz short di3
762 00000604 7427
763 00000606 56
                                          push esi
764
                                          ; 13/05/2016
765 00000607 BB07000000
                                          mov ebx, 7; Black background,
                                                ; light gray forecolor
766
767
                                                      ; Video page 0 (BH=0)
                                          call _write_tty
768 0000060C E8AA160000
769 00000611 5E
                                          pop
                                               esi
770 00000612 EBED
                                               short psem
                                          j mp
771
772
                                    dil:
773
                                          ; supress 'jmp short T6'
774
                                          ; (activate fdc motor control code)
775 00000614 66C705[F4060000]90-
                                         mov word [T5], 9090h; nop
775 0000061C 90
776
777
                                          ;mov ax, int OEh ; IRQ 6 handler
                                          ; mov di, 0Eh^{\star}4 ; IRQ 6 vector
778
779
                                          ;stosw
780
                                          ;mov ax, cs
781
                                          ;stosw
782
                                          ;; 16/02/2015
                                                    dword [DISKETTE INT], fdc int ; IRQ 6 handler
783
                                           ;;mov
785 0000061D E8CD3B0000
                                          CALL DSKETTE_SETUP; Initialize Floppy Disks
786
787 00000622 09D2
                                                edx, edx
                                          or
788 00000624 7407
                                           jz short di3
789
                                    di2:
790 00000626 E80B3C0000
                                                     DISK_SETUP ; Initialize Fixed Disks
                                          call
791 0000062B 72CF
                                                    short setup_error
                                           jс
792
                                    di3:
                                          call setup_rtc_int; 22/05/2015 (dsectrpm.s)
793 0000062D E8FF2B0000
795 00000632 E845120100
                                          call display_disks ; 07/03/2015 (Temporary)
796
                                    ;haltx:
                                         ; 14/08/2015
797
                                          ;call getch ; 22/02/2015
798
799
                                         ;sti ; Enable interrupts (for CPU)
                                          ; 29/01/2016
800
801
                                          sub ah, ah; read time count
802
                                          call int1Ah
803
                                         mov edx, ecx; 18.2 * seconds
                                    ;md info msg wait1:
804
                                         ; 29/01/2016
805
806
                                               ah, 1
807
                                         call int16h
808
                                          jz short md_info_msg_wait2
809
                                               ah, ah ; 0
                                         xor
810
                                          call int16h
                                          jmp short md info msg ok
811
812
                                    ;md_info_msg_wait2:
```

```
813
                                          sub
                                                ah, ah ; read time count
814
                                          call int1Ah
                                     ;
                                                edx, ecx ; ; 18.2 * seconds
815
                                     ;
                                          cmp
816
                                          jna short md_info_msg_wait3
817
                                          xchg edx, ecx
818
                                     ;md_info_msg_wait3:
819
                                          sub ecx, edx
                                     ;
                                           cmp ecx, 127; 7 seconds (18.2 * 7)
820
821
                                           jb
                                                short md_info_msg_wait1
822
                                     ;md info msg ok:
                                          ; 08/09/2016
823
824 00000637 0F20C0
                                          mov eax, cr0
                                          test al, 10h ; Bit 4, ET (Extension Type)
825 0000063A A810
826 0000063C 7408
                                          jz short sysinit
827
                                          ; 27/02/2017
828 0000063E FE05[7C650100]
                                          inc byte [fpready]
                                           ; 80387 (FPU) is ready
830 00000644 DBE3
                                           fninit ; Initialize Floating-Point Unit
831
                                     sysinit:
                                          ; 30/06/2015
832
833 00000646 E8635C0000
                                           call sys_init
834
835
                                          ;jmp cpu_reset ; 22/02/2015
836
                                          ; 23/02/2015
837
838
                                           ;sti
                                                              ; Enable interrupts
839 0000064B F4
                                          hlt
840
                                          ;
841
                                           ;nop
                                          ;; 03/12/2014
842
843
                                           ;; 28/08/2014
844
                                          ;mov ah, 11h
845
                                          ;call getc
846
                                          ;jz
847
                                          ; 23/02/2015
848
                                          ; 06/02/2015
849
850
                                          ; 07/09/2014
851 0000064C 31DB
                                                 ebx, ebx
                                          xor
852 0000064E 8A1D[EE580100]
                                                 bl, [ptty]
                                          mov
                                                             ; active_page
853 00000654 89DE
                                          mov
                                                 esi, ebx
854 00000656 66D1E6
                                          shl
                                                 si, 1
855 00000659 81C6[F0580100]
                                                 esi, ttychr
                                          add
856 0000065F 668B06
                                          mov
                                                 ax, [esi]
857 00000662 6621C0
                                          and
                                                 ax, ax
858
                                           ;jz
                                                 short _c8
859 00000665 74E4
                                                 short hang
                                          jz
860 00000667 66C7060000
                                                 word [esi], 0
                                          mov
861 0000066C 80FB03
                                          cmp
                                                 bl, 3
                                                              ; Video page 3
                                                short c8
862
                                          ;jb
863 0000066F 72DA
                                           jb
                                                 short hang
864
                                          ; 13/05/2016
865
                                          ; 07/09/2014
866
867
                                     nxtl:
868 00000671 6653
                                          push bx
869 00000673 66BB0E00
                                                 bx, OEh
                                                              ; Yellow character
                                          mov
                                                              ; on black background
870
871
                                                              ; bh = 0 (video page 0)
                                                              ; Retro UNIX 386 v1 - Video Mode 0
872
873
                                                              ; (PC/AT Video Mode 3 - 80x25 Alpha.)
874 00000677 6650
                                          push ax
875 00000679 E83D160000
                                          call
                                                 _write_tty
876 0000067E 6658
                                                 ax
                                          pop
877 00000680 665B
                                                 bx
                                          pop
878 00000682 3C0D
                                           cmp
                                                 al, ODh
                                                                     ; carriage return (enter)
                                                short c8
                                          ;jne
880 00000684 75C5
                                                 short hang
                                          jne
881 00000686 B00A
                                          mov
                                                 al, OAh
                                                                     ; next line
882 00000688 EBE7
                                                short nxtl
                                          jmp
883
                                    ;_c8:
884
                                          ; 25/08/2014
885
                                     ;
886
                                           cli
                                                                     ; Disable interrupts
                                                 al, [scounter + 1]
887
                                     ;
                                          mov
888
                                           and
                                                al, al
889
                                                hang
                                          jnz
                                     ;
890
                                          call rtc p
891
                                                  hang
                                          jmp
892
893
                                          ; 27/08/2014
894
                                          ; 20/08/2014
895
896
897
                                             ; mov
                                                    edi, [scr_row]
898
                                     pkl:
899 0000068A AC
                                           lodsb
900 0000068B 08C0
                                           or
                                                 al, al
901 0000068D 7404
                                           jΖ
                                                 short pkr
902 0000068F 66AB
                                          stosw
903 00000691 EBF7
                                           jmp
                                                 short pkl
904
                                     pkr:
905 00000693 C3
                                           retn
906
907
                                     ; 28/02/2017
908
                                     ; 22/01/2017
909
                                     ; 15/01/2017
910
                                     ; 14/01/2017
911
                                     ; 02/01/2017
912
                                     ; 25/12/2016
                                     ; 19/12/2016
913
914
                                    ; 10/12/2016 (callback)
915
                                     ; 06/06/2016
                                    ; 23/05/2016
916
917
                                     ; 22/05/2016 - TRDOS 386 (TRDOS v2.0) Timer Event Modifications
```

```
918
                                     ; 25/07/2015
 919
                                    ; 14/05/2015 (multi tasking -time sharing- 'clock', x timer)
 920
                                     : 17/02/2015
 921
                                     ; 06/02/2015 (unix386.s)
 922
                                     ; 11/12/2014 - 22/12/2014 (dsectrm2.s)
923
                                     ; IBM PC-XT Model 286 Source Code - BIOS2.ASM (06/10/85)
 925
926
                                     ;-- HARDWARE INT 08 H - ( IRQ LEVEL 0 ) ------
                                          THIS ROUTINE HANDLES THE TIMER INTERRUPT FROM FROM CHANNEL 0 OF
 927
 928
                                          THE 8254 TIMER. INPUT FREQUENCY IS 1.19318 MHZ AND THE DIVISOR
 929
                                          IS 65536, RESULTING IN APPROXIMATELY 18.2 INTERRUPTS EVERY SECOND.
 930
931
                                          THE INTERRUPT HANDLER MAINTAINS A COUNT (40:6C) OF INTERRUPTS SINCE
 932
                                          POWER ON TIME, WHICH MAY BE USED TO ESTABLISH TIME OF DAY.
933
                                          THE INTERRUPT HANDLER ALSO DECREMENTS THE MOTOR CONTROL COUNT (40:40) :
                                          OF THE DISKETTE, AND WHEN IT EXPIRES, WILL TURN OFF THE
                                          DISKETTE MOTOR(s), AND RESET THE MOTOR RUNNING FLAGS.
 935
 936
                                          THE INTERRUPT HANDLER WILL ALSO INVOKE A USER ROUTINE THROUGH
                                          INTERRUPT 1CH AT EVERY TIME TICK. THE USER MUST CODE A
 937
 938
                                          ROUTINE AND PLACE THE CORRECT ADDRESS IN THE VECTOR TABLE.
 939
 940
 941
 942
                                     timer_int: ; IRQ 0
 943
                                     ;int_08h:
                                                 ; Timer
 944
                                          ; 14/10/2015
 945
                                          ; Here, we are simulating system call entry (for task switch)
 946
                                          ; (If multitasking is enabled,
                                          ; 'clock' procedure may jump to 'sysrelease')
 947
 948
 949 00000694 1E
                                          push ds
 950 00000695 06
                                          push es
 951 00000696 OFA0
                                          push fs
 952 00000698 0FA8
                                          push gs
953
 954 0000069A 60
                                          pushad; eax, ecx, edx, ebx, esp -before pushad-, ebp, esi, edi
 955 0000069B 66B91000
                                          mov cx, KDATA
 956 0000069F 8ED9
                                           mov
                                                    ds, cx
 957 000006A1 8EC1
                                            mov
                                                    es, cx
 958 000006A3 8EE1
                                            mov
                                                   fs, cx
 959 000006A5 8EE9
                                           mov
                                                   gs, cx
960
 961 000006A7 0F20D9
                                          mov ecx, cr3
 962 000006AA 890D[5C040300]
                                          mov [cr3reg], ecx; save current cr3 register value/content
963
                                          ; 14/01/2017
 965 000006B0 3B0D[C0580100]
                                          cmp ecx, [k_page_dir]
 966 000006B6 7409
                                                 short T3
 968 000006B8 8B0D[C0580100]
                                                 ecx, [k_page_dir]
                                          mov
 969 000006BE 0F22D9
                                          mov
                                                 cr3, ecx
970
                                    т3:
 971
                                          ;sti
                                                                   ; INTERRUPTS BACK ON
 972 000006C1 66FF05[40590100]
                                          INC
                                                 word [TIMER_LOW] ; INCREMENT TIME
 973 000006C8 7507
                                          JNZ
                                                 short T4
                                                                    ; GO TO TEST_DAY
                                                 word [TIMER_HIGH] ; INCREMENT HIGH WORD OF TIME
 974 000006CA 66FF05[42590100]
 975
                                     T4:
                                                                    ; TEST_DAY
 976 000006D1 66833D[42590100]18
                                          CMP
                                                 word [TIMER_HIGH],018H ; TEST FOR COUNT EQUALING 24 HOURS
                                                 short T5
 977 000006D9 7519
                                          JNZ
                                                                  ; GO TO DISKETTE CTL
 978 000006DB 66813D[40590100]B0-
                                          CMP
                                                 word [TIMER_LOW], 0B0H
 978 000006E3 00
 979 000006E4 750E
                                          JNZ
                                                                  ; GO TO DISKETTE_CTL
                                                 short T5
 980
 981
                                                 TIMER HAS GONE 24 HOURS
 982
                                          ;;SUB AX,AX
 983
                                          ;MOV [TIMER HIGH],AX
                                          ; MOV
                                                [TIMER_LOW],AX
 984
 985 000006E6 29C0
                                          sub
                                                 eax, eax
 986 000006E8 A3[40590100]
                                                [TIMER_LH], eax
                                          mov
 988 000006ED C605[44590100]01
                                          MOV
                                                byte [TIMER_OFL],1
 989
                                     ;----
 990
                                                 TEST FOR DISKETTE TIME OUT
 991
992
                                     T5:
                                          ; 23/12/2014
                                          jmp short T6
 994 000006F4 EB1D
                                                                    ; will be replaced with nop, nop
                                                                    ; (9090h) if a floppy disk
 995
 996
                                                                    ; is detected.
                                           ;mov al,[CS:MOTOR COUNT]
 997
 998 000006F6 A0[47590100]
                                          mov
                                                 al, [MOTOR_COUNT]
 999 000006FB FEC8
                                          dec
                                                [CS:MOTOR_COUNT], al
                                                                          ; DECREMENT DISKETTE MOTOR CONTROL
1001 000006FD A2[47590100]
                                                 [MOTOR COUNT], al
                                          mov
                                               [ORG MOTOR COUNT], al
1002
                                          ;mov
1003 00000702 750F
                                                              ; RETURN IF COUNT NOT OUT
                                          JNZ short T6
1004 00000704 B0F0
                                          mov
                                                al,0F0h
                                          ;AND [CS:MOTOR STATUS],al
1005
                                                                         ; TURN OFF MOTOR RUNNING BITS
1006 00000706 2005[46590100]
                                                 [MOTOR STATUS], al
                                          and
1007
                                          ;and [ORG_MOTOR_STATUS], al
1008 0000070C B00C
                                                                  ; bit 3 = enable IRQ & DMA,
                                          MOV
                                                AL, OCH
                                                                    ; bit 2 = enable controller
1009
1010
                                                                         1 = normal operation
1011
                                                                          0 = reset
                                                                    ; bit 0, 1 = drive select
1012
                                                                   ; bit 4-7 = motor running bits
                                                                   ; FDC CTL PORT
1014 0000070E 66BAF203
                                          MOV
                                                DX,03F2H
1015 00000712 EE
                                          OUT
                                                 DX,AL
                                                                    ; TURN OFF THE MOTOR
                                     T6:
1016
1017
                                           ;inc word [CS:wait_count]
                                                                         ; 22/12/2014 (byte -> word)
1018
                                                                   ; TIMER TICK INTERRUPT
                                          ;;inc word [wait_count] ;;27/02/2015
1019
                                                                   ; TRANSFER CONTROL TO A USER ROUTINE
1020
                                          ;INT 1CH
1021
                                          ;cli
```

```
1022 00000713 E857040000
                                                                              ; TRANSFER CONTROL TO A USER ROUTINE
                                            call u timer
                                            ; 23/05/2016
1024 00000718 E89EF20000
                                            call clock
                                                                      ; Multi Tasking control procedure
1025
1026
                                            ; 14/10/2015
                                                                       ; GET END OF INTERRUPT MASK
1027 0000071D B020
                                            MOV AL, EOI
1028 0000071F FA
                                            CLI
                                                                      ; DISABLE INTERRUPTS TILL STACK CLEARED
1029 00000720 E620
                                                  INTA00,AL
                                            OUT
                                                                     ; END OF INTERRUPT TO 8259 - 1
1030
                                      rtc_int 2:
1031
1032
                                            ; 26/12/2016
1033
                                            ;mov ecx, [cr3reg]
                                            ; 13/01/2017
1034
                                            cmp byte [u.t_lock], 0 ; T_LOCK
ja short timer_int_return ; Timer Lock : 'sysrele' is needed !
1035 00000722 803D[D4030300]00
1036 00000729 7730
                                            ; 28/02/2017
1037
1038
                                            ; We need to exit if the user's IRQ callback service is in progress!
                                            ; (To prevent a conflict!)
1039
                                            cmp byte [u.r_lock], 0 ; R_LOCK, IRQ callback service lock !
ja short timer_int_return ; Timer Lock : 'sysrele' is needed !
1040 0000072B 803D[D8030300]00
1041 00000732 7727
                                            ; 15/01/2017
1042
                                            cmp byte [priority], 2
jnb short T8 ; current process has a timer event (15/01/2017)
1043 00000734 803D[50650100]02
1044 0000073B 733A
                                            ; 22/05/2016
1045
                                            cmp byte [p_change], 0 ; in 'set_run_sequence', in 'rtc_p'
jna short timer_int_return ; 23/05/2016
1046 0000073D 803D[51650100]00
1047 00000744 7615
1048
                                            ; 15/01/2017
1049
1050
1051
                                            ; present process must be changed with high priority process
1052
                                            ;xor al, al
1053 00000746 31C0
                                            xor
                                                  eax, eax ; 26/12/2016
1054 00000748 A2[51650100]
                                                   [p_change], al ; 0
                                            mov
1055
                                            ;mov byte [priority], 2 ; 15/01/2017 (there is a timer event)
1056
1057 0000074D 803D[5B030300]FF
                                            cmp
                                                    byte [sysflg], OFFh; user or system space?
1058 00000754 7416
                                                   short rtc_int_3 ; user space ([sysflg] = 0FFh)
                                            jе
1059
1060
                                            ; system space, wait for 'sysret'
                                            ; to change running process
1061
1062
                                            ; with high priority (event) process
1063
                                            mov [u.quant], al; 0
1064 00000756 A2[A8030300]
1065
1066
                                      timer_int_return: ; 23/05/2016 - jump from 'rtc_int' ('rtc_int_2')
1067 0000075B 8B0D[5C040300]
                                            mov ecx, [cr3reg] ; previous value/content of cr3 register
1068 00000761 0F22D9
                                            mov cr3, ecx ; restore cr3 register content
1069
1070 00000764 61
                                            popad; edi, esi, ebp, temp (icrement esp by 4), ebx, edx, ecx, eax
1071
1072 00000765 0FA9
                                            pop
                                                  gs
1073 00000767 0FA1
                                                  fs
                                            pop
1074 00000769 07
                                            pop
                                                  es
1075 0000076A 1F
                                            pop
                                                   ds
1076
1077 0000076B CF
                                            iretd ; return from interrupt
1078
1079
                                      rtc_int_3:
1080 0000076C FE05[5B030300]
                                            inc byte [sysflg]
                                                                     ; now, we are in system space
1081
1082 00000772 E90BC00000
                                              jmp
                                                       sysrelease; change running process immediatelly
1083
                                      T8:
1084
1085
                                            ; 13/01/2017 (eax -> ebx)
1086
                                            ; callback checking... (19/12/2016)
1087 00000777 31DB
                                            xor ebx, ebx
1088 00000779 871D[D0030300]
                                            xchg ebx, [u.tcb] ; callback address (0 = normal return)
1089 0000077F 09DB
                                            or
                                                  ebx, ebx
1090 00000781 74D8
                                                  short timer_int_return
                                            jΖ
1091
1092
                                            ; Set user's callback routine as return address from this interrupt
                                            ; and set normal return address as return address from callback
1093
1094
                                            ; routine!!! (19/12/2016)
1095
                                            ; 14/01/2017
1096
                                            ; 13/01/2017 - Timer Lock (T_LOCK)
1097
1098 00000783 FE05[D4030300]
                                            inc byte [u.t lock]
1099 00000789 8A0D[5B030300]
                                            mov cl, [sysflg]
1100 0000078F 880D[D5030300]
                                                  [u.t mode], cl
                                            mov
1101
1102 00000795 8B2D[5C580100]
                                            mov
                                                   ebp, [tss.esp0]; kernel stack address (for ring 0)
1103 0000079B 83ED14
                                            sub
                                                   ebp, 20
                                                            ; eip, cs, eflags, esp, ss
1104 0000079E 892D[5C030300]
                                            mov
                                                   [u.sp], ebp
1105 000007A4 8925[60030300]
                                            mov
                                                   [u.usp], esp
1106
                                                   word [ebp+8], 200h; 22/01/2017, force enabling interrupts
1107
                                            ;or
1108
1109 000007AA 8B44241C
                                                   eax, [esp+28]; pushed eax
                                            mov
1110 000007AE A3[64030300]
                                                   [u.r0], eax
                                            mov
1111
1112 000007B3 E8F9DE0000
                                            call wswap; save user's registers & status
1113
                                            ; software int is in ring 0 but timer int must return to ring 3
1114
1115
                                            ; so, ring 3 return address and stack registers
1116
                                            ; (eip, cs, eflags, esp, ss)
1117
                                            ; must be copied to timer int return
1118
                                            ; eip will be replaced by callback service routine address
1119
1120 000007B8 C605[5B030300]FF
                                                   byte [sysflg], OFFh; user mode
1121
                                            ; system mode (system call)
1122
1123
                                            ;mov ebp, [u.sp]; EIP (u), CS (UCODE), EFLAGS (u),
1124
                                                              ; ESP (u), SS (UDATA)
1125
1126 000007BF 8B4510
                                                   eax, [ebp+16]; SS (UDATA
                                            mov
```

```
1127 000007C2 89E6
                                            mov
                                                  esi, esp
1128 000007C4 50
                                           push eax
1129 000007C5 50
                                            push eax
1130 000007C6 89E7
                                            mov
                                                   edi, esp
1131 000007C8 893D[60030300]
                                                  [u.usp], edi
                                            mov
1132 000007CE B908000000
                                            mov
                                                   ecx, ((ESPACE/4) - 4); except DS, ES, FS, GS
1133 000007D3 F3A5
                                           rep
1134 000007D5 B104
                                                  cl, 4
                                           mov
1135 000007D7 F3AB
                                                  stosd
                                            rep
1136 000007D9 893D[5C030300]
                                           mov
                                                  [u.sp], edi
1137 000007DF 89EE
                                            mov
                                                  esi, ebp
1138 000007E1 B105
                                                  cl, 5; EIP (u), CS (UCODE), EFLAGS (u), ESP (u), SS (UDATA)
                                            mov
1139 000007E3 F3A5
                                            rep
                                                  movsd
1140
1141 000007E5 8B0D[B8030300]
                                           mov
                                                   ecx, [u.pgdir]
1142 000007EB 890D[5C040300]
                                            mov
                                                  [cr3reg], ecx
1144
                                            ; 13/01/207 (eax -> ebx)
1145
                                            ; EBX = callback routine address (virtual, not physical address!)
1146
1147
                                            ; 09/01/2017
1148
                                            ; !!! CALLBACK ROUTINE MUST BE ENDED/RETURNED WITH 'sysrele'
1149
                                                  system call !!!
1150
                                            ; 25/12/2016
                                            ; Callback Note: (19/12/2016)
1151
                                            ; !!! CALLBACK ROUTINE MUST BE ENDED/RETURNED WITH 'RETN' !!!
1152
1153
                                                  pushf ; save flags
1154
                                                   <callback service code>
1155
                                                  popf ; restore flags
1156
                                                  retn ; return to normal running address
1157
1158
                                            ; 15/01/2017
1159
1160
                                            ; 14/01/2017
                                            ; 13/01/2017 (eax -> ebx)
1161
                                            ; 10/01/2017
1162
                                      set callback addr:
1163
1164
                                           ; 09/01/2017 (**)
                                            ; 02/01/2017 (*)
1165
1166
                                            ; 25/12/2016 (*)
                                            ; 19/12/2016 (TRDOS 386 feature only!)
1167
1168
                                            ; This routine sets return address
1169
1170
                                            ; to start of user's interrupt
1171
                                            ; service (callback) address
1172
                                            ;; and sets callback 'retn' address to normal
                                            ;; return address of user's running code!
1173
1174
1175
                                            ; INPUT:
1176
                                                   EBX = callback routine/service address
                                                       (virtual, not physical address!)
1177
1178
                                                   [u.sp] = kernel stack, points to
1179
                                                         user's EIP,CS,EFLAGS,ESP,SS
1180
                                                          registers.
1181
                                            ; OUTPUT:
1182
                                                   EIP (user) = callback (service) address
                                                   CS (user) = UCODE
1183
1184
                                                   EFLAGS (user) = flags before callback
1185
                                                    ESP (user) = ESP-4 (user, before callback)
                                                   [ESP] (user) = EIP (user) before callback
1186
1187
1188
                                            ; Note: If CPU was in user mode while entering
1189
                                                   the timer interrupt service routine,
1190
                                                   'IRET' will get return to callback routine
                                                   immediately. If CPU was in system/kernel\ mode
1191
1192
                                                   'iret' will get return to system call and
1193
                                                   then, callback routine will be return address
1194
                                                   from system call. (User's callback/service code
1195
                                                   will be able to return to normal return address
1196
                                                   via an 'retn' at the end.)
1197
1198
                                            ; Note(**): User's callback service code must be ended
                                                   with a 'sysrele' sytstem call ! (09/01/2017)
1199
1200
1201
                                                   For example:
1202
1203
                                                   timer_callback:
1204
1205
                                                       inc
                                                                dword [time_counter]
1206
                                                       mov eax, 39 ; 'sysrele'
1207
                                                       int 40h ; TRDOS 386 system call (interrupt)
1208
1209
1210
                                            ;; Note(*): User's callback service code must preserve cpu
1211
1212
                                                   flags if it has any instructions which changes
1213
                                                   flags in the service code. (25/12/2016)
                                            ;;
1214
                                            ;;
                                                   For example:
1215
                                            ;;
1216
                                            ;;
1217
                                                   timer_callback:
                                            ;;
1218
                                                      pushf ; save flags
                                            ;;
1219
                                            ;;
                                                       ; this instruction changes zero flag
1220
                                                              dword [time_counter]
                                            ;;
1221
                                                      popf ; restore flags
                                            ;;
                                                       retn ; return to normal user code
1222
                                            ;;
1223
                                                          (which is interrupted by the
                                            ;;
1224
                                                            timer interput)
                                            ;;
1225
                                            ;;
1226
                                            ; 15/01/2017
1227
1228 000007F1 8B2D[5C030300]
                                                  ebp, [u.sp]; kernel's stack, points to EIP (user)
                                            mov
1229 000007F7 895D00
                                            mov
                                                   [ebp], ebx
                                                  timer int return
1230 000007FA E95CFFFFFF
                                            jmp
```

```
1232
                                           ; 15/01/2017
1233
                                          ; 13/01/2017
                                          ; 19/12/2016
1234
1235
                                          ; 06/06/2016
1236
                                          ; 23/05/2016
1237
                                           ; 22/05/2016
                                          ; 19/05/2016 - TRDOS 386 (TRDOS v2.0)
1238
1239
                                          ; 26/02/2015
1240
                                          ; 07/09/2014
1241
                                          ; 25/08/2014
1242
                                     rtc_int:
                                                 ; Real Time Clock Interrupt (IRQ 8)
                                         ; 22/05/2016
1243
                                           push ds; **; 23/05/2016
1244 000007FF 1E
                                          push eax ; *
1245 00000800 50
1246 00000801 66B81000
                                          mov
                                                ax, KDATA
1247 00000805 8ED8
                                          mov
                                                ds, ax
                                          ;
1249 00000807 8A25[3E590100]
                                                ah, [RTC_2Hz]; 2 Hz interrupt to 1 Hz function
                                          mov
1250 0000080D 80F401
                                          xor
1251 00000810 8825[3E590100]
                                                [RTC 2Hz], ah; 1 = 0.5 second, 0 = 1 second
                                          mov
                                          jnz short rtc_int_return ; half second
1252 00000816 753B
1253
                                          ; 1 second
1254
                                     rtc_int_0:
1255
                                          ; 22/05/2016
1256 00000818 58
                                          pop eax; *
1257
1258
                                          ; 14/10/2015 ('timer int')
1259
                                          ; Here, we are simulating system call entry (for task switch)
1260
                                           ; (If multitasking is enabled,
1261
                                           ; 'clock' procedure may jump to 'sysrelease')
1262
                                           ;push ds ; ** ; 23/05/2016
                                          push es
push fs
1263 00000819 06
1264 0000081A 0FA0
1265 0000081C 0FA8
                                          push gs
1266 0000081E 60
                                          pushad ; eax, ecx, edx, ebx, esp -before pushad-, ebp, esi, edi
1267 0000081F 66B91000
                                          mov cx, KDATA
                                           ;mov ds, cx; 06/06/2016
1269 00000823 8EC1
                                            mov
                                                    es, cx
1270 00000825 8EE1
                                            mov
                                                    fs, cx
1271 00000827 8EE9
                                                    gs, cx
                                            mov
1272
1273 00000829 0F20D9
                                          mov
                                                 ecx, cr3
1274 0000082C 890D[5C040300]
                                                 [cr3reg], ecx; save current cr3 register value/content
                                          mov
1275
1276 00000832 803D[D4030300]00
                                                byte [u.t_lock], 0 ; timer lock (callback) status ?
                                          cmp
1277 00000839 7711
                                           jа
                                                 short rtc_int_1
                                                                       ; yes
1278
                                          ; 15/01/2017
1279
                                                ecx, [k page dir]
1280 0000083B 3B0D[C0580100]
                                           cmp
                                                 short rtc_int 1
1281 00000841 7409
                                           jе
1282
1283 00000843 8B0D[C0580100]
                                                 ecx, [k_page_dir]
                                          mov
1284 00000849 0F22D9
                                          mov
                                                 cr3, ecx
1285
                                     rtc_int_1:
1286
                                          ; Timer event (kernel) functions must be performed with
1287
                                           ; 1 second intervals - TRDOS 386 (TRDOS v2.0) feature ! -
                                           ; 25/08/2014
1289
1290 0000084C E81A030000
                                           call rtc_p ; 19/05/2016 - major modification
1291
1292
                                           ; 23/05/2016
1293 00000851 28E4
                                           sub ah, ah; 0
                                           ; 22/05/2016 - TRDOS 386 timer event modifications
1294
1295
                                     rtc_int_return: ; 19/05/2016
                                          ; 22/02/2015 - dsectpm.s
1296
1297
                                           ; [ source: http://wiki.osdev.org/RTC ]
                                           ; read status register C to complete procedure
1298
1299
                                           ; (it is needed to get a next IRQ 8)
1300 00000853 B00C
                                                al, 0Ch ;
                                           mov
                                               70h, al ; select register C
1301 00000855 E670
                                           out
1302 00000857 90
                                           nop
1303 00000858 E471
                                                 al, 71h; just throw away contents
                                           in
                                           ; 22/02/2015
1304
                                                             ; END OF INTERRUPT
1305 0000085A B020
                                           MOV AL, EOI
                                                            ; DISABLE INTERRUPTS TILL STACK CLEARED ; FOR CONTROLLER #2
1306
                                           ;CLI
1307 0000085C E6A0
                                                INTB00,AL
                                           OUT
1308
1309
                                           ; 23/05/2016
                                                             ; GET END OF INTERRUPT MASK
1310 0000085E B020
                                           VOM
                                                AL,EOI
                                                             ; DISABLE INTERRUPTS TILL STACK CLEARED
1311 00000860 FA
                                           CLI
1312 00000861 E620
                                                INTA00,AL
                                                            ; END OF INTERRUPT TO 8259 - 1
                                           OUT
                                           ; 23/05/2016
1315 00000863 20E4
                                           and ah, ah
1316 00000865 0F84B7FEFFFF
                                           jz rtc_int_2
1317
                                           ; ah = 1 (half second)
1319 0000086B 58
                                           pop eax; *
1320 0000086C 1F
                                                ds ; **
                                           pop
1321 0000086D CF
                                           iretd
1322
                                     1323
1324
1325
                                          ; 28/08/2014
1326
                                     irq0:
1327 0000086E 6A00
                                            push
                                                      dword 0
1328 00000870 EB48
                                           jmp short which_irq
1329
                                     irq1:
1330 00000872 6A01
                                            push
                                                       dword 1
1331 00000874 EB44
                                           jmp short which_irq
                                     irq2:
1332
                                                   dword 2
1333 00000876 6A02
                                            push
1334 00000878 EB40
                                           jmp short which_irq
1335
                                     irq3:
1336
                                           ; 20/11/2015
```

```
; 24/10/2015
1337
                                         call dword [cs:com2_irq3]
push dword 3
1338 0000087A 2EFF15[79FC0000]
1339 00000881 6A03
1340 00000883 EB35
                                         jmp short which_irq
1341
                                    irq4:
                                         ; 20/11/2015
1342
                                         ; 24/10/2015
1343
1344 00000885 2EFF15[75FC0000]
                                         call dword [cs:com1_irq4]
                                          push dword 4
1345 0000088C 6A04
1346 0000088E EB2A
                                          jmp short which_irq
1347
1348 00000890 6A05
                                           push
                                                    dword 5
1349 00000892 EB26
                                          jmp short which_irq
1350
                                           push dword 6
1351 00000894 6A06
                                          jmp short which_irq
1352 00000896 EB22
                                    irq7:
                                           push dword 7
1354 00000898 6A07
                                         jmp short which_irq
1355 0000089A EB1E
1356
                                    irq8:
                                          push dword 8
1357 0000089C 6A08
1358 0000089E EB1A
                                         jmp short which_irq
1359
                                    irq9:
                                           push dword 9
1360 000008A0 6A09
                                         jmp short which_irq
1361 000008A2 EB16
1362
                                    irq10:
                                          push dword 10
1363 000008A4 6A0A
1364 000008A6 EB12
                                          jmp short which_irq
1365
                                    irq11:
                                          push dword 11
1366 000008A8 6A0B
1367 000008AA EB0E
                                          jmp short which_irq
                                    irq12:
1368
                                           push
                                                     dword 12
1369 000008AC 6A0C
1370 000008AE EB0A
                                          jmp short which_irq
1371
                                    irq13:
                                           push dword 13
1372 000008B0 6A0D
1373 000008B2 EB06
                                         jmp short which_irq
1374
                                    irq14:
                                           push dword 14
1375 000008B4 6A0E
                                          jmp short which_irq
1376 000008B6 EB02
                                    irq15:
1377
                                          push dword 15
1378 000008B8 6A0F
1379
                                         ;jmp short which_irq
1380
                                         ; 22/01/2017
1381
                                         ; 19/10/2015
1382
1383
                                         ; 29/08/2014
                                         ; 21/08/2014
1384
1385
                                    which_irq:
                                         xchg eax, [esp] ; 28/08/2014
1386 000008BA 870424
1387 000008BD 53
                                         push ebx
1388 000008BE 56
                                         push esi
1389 000008BF 57
                                         push edi
                                         push
1390 000008C0 1E
                                               ds
1391 000008C1 06
                                         push
                                               es
1392
1393 000008C2 88C3
                                         mov
                                               bl, al
1394
1395 000008C4 B810000000
                                         mov
                                                eax, KDATA
1396 000008C9 8ED8
                                         mov
                                               ds, ax
1397 000008CB 8EC0
                                          mov es, ax
1398
                                         ; 19/10/2015
1399 000008CD FC
                                          cld
                                          ; 27/08/2014
1400
                                           add dword [scr_row], 0A0h
1401 000008CE 8105[D6160100]A000-
1401 000008D6 0000
1402
                                         ;
1403 000008D8 B417
                                                          ; blue (1) background,
                                                ah, 17h
                                         mov
1404
                                                    ; light gray (7) forecolor
                                                  edi, [scr_row]
1405 000008DA 8B3D[D6160100]
                                           mov
1406 000008E0 B049
                                          mov al, 'I'
1407 000008E2 66AB
                                          stosw
                                         mov al, 'R'
1408 000008E4 B052
1409 000008E6 66AB
                                          stosw
1410 000008E8 B051
                                         mov al, 'Q'
1411 000008EA 66AB
                                         stosw
1412 000008EC B020
                                         mov al, ''
1413 000008EE 66AB
                                          stosw
1414 000008F0 88D8
                                               al, bl
                                          mov
1415 000008F2 3C0A
                                               al, 10
                                          cmp
1416 000008F4 7208
                                          jb
                                               short iil
1417 000008F6 B031
                                         mov
                                                al, '1'
1418 000008F8 66AB
                                          stosw
1419 000008FA 88D8
                                                al, bl
1420 000008FC 2C0A
                                         sub
                                               al, 10
                                    ii1:
1421
1422 000008FE 0430
                                         add
                                               al, '0'
1423 00000900 66AB
                                          stosw
                                               al, ' '
1424 00000902 B020
                                          mov
1425 00000904 66AB
                                         stosw
                                               al, '!'
1426 00000906 B021
                                         mov
1427 00000908 66AB
                                          stosw
                                         mov al, ''
1428 0000090A B020
1429 0000090C 66AB
                                          stosw
                                         ; 23/02/2015
1430
1431 0000090E 80FB07
                                          cmp
                                               bl, 7; check for IRQ 8 to IRQ 15
1432 00000911 7604
                                          jna
                                          ; 22/01/2017
1433
                                               al, 20h ; END OF INTERRUPT COMMAND TO
1434 00000913 B020
                                                OAOh, al ; the 2nd 8259
1435 00000915 E6A0
                                         out
                                    ii2:
1436
1437 00000917 B020
                                                al, 20h ; END OF INTERRUPT COMMAND TO
                                         mov
1438 00000919 E620
                                          out
                                                20h, al ; the 2nd 8259
1439 0000091B E9CD010000
                                                 iiret
                                          jmp
                                          ;
```

```
; 22/08/2014
1441
1442
                                            ;mov al, 20h; END OF INTERRUPT COMMAND TO 8259
                                                 20h, al ; 8259 PORT
1443
                                            ;out
1444
1445
                                            ;pop
                                                  es
1446
                                            ;pop
                                                  ds
1447
                                            ;pop
                                                  edi
1448
                                                  esi
                                            ;pop
1449
                                                  ebx
                                            ;pop
1450
                                            ;pop
                                                  eax
1451
                                            ;iret
1452
                                            ; 02/04/2015
1453
                                            ; 25/08/2014
1454
1455
                                      exc0:
1456 00000920 6A00
                                              push
                                                        dword 0
1457 00000922 E990000000
                                              jmp
                                                      cpu_except
1458
                                      exc1:
1459 00000927 6A01
                                              push
                                                         dword 1
1460 00000929 E989000000
                                                      cpu_except
                                              jmp
1461
                                      exc2:
1462 0000092E 6A02
                                              push
                                                        dword 2
1463 00000930 E982000000
                                              jmp
                                                      cpu_except
1464
                                      exc3:
1465 00000935 6A03
                                              push
                                                        dword 3
1466 00000937 EB7E
                                              jmp
                                                      cpu_except
1467
                                      exc4:
1468 00000939 6A04
                                             push
                                                         dword 4
1469 0000093B EB7A
                                              jmp
                                                      cpu_except
1470
                                      exc5:
1471 0000093D 6A05
                                              push
                                                        dword 5
1472 0000093F EB76
                                                      cpu_except
                                              jmp
1473
                                      exc6:
1474 00000941 6A06
                                              push
                                                         dword 6
1475 00000943 EB72
                                                      cpu_except
                                              jmp
1476
                                      exc7:
1477 00000945 6A07
                                              push
                                                        dword 7
1478 00000947 EB6E
                                              jmp
                                                      cpu_except
1479
                                      exc8:
1480
                                           ; [esp]
                                                   = Error code
1481 00000949 6A08
                                              push
                                                        dword 8
1482 0000094B EB5C
                                              jmp
                                                      cpu_except_en
1483
                                      exc9:
                                             push
1484 0000094D 6A09
                                                        dword 9
1485 0000094F EB66
                                                      cpu_except
                                              jmp
1486
                                      exc10:
1487
                                           ; [esp] = Error code
1488 00000951 6A0A
                                              push
                                                        dword 10
                                                      cpu except en
1489 00000953 EB54
                                              jmp
1490
                                      exc11:
1491
                                            ; [esp] = Error code
1492 00000955 6A0B
                                             push
                                                       dword 11
1493 00000957 EB50
                                              jmp
                                                      cpu_except_en
1494
                                      exc12:
1495
                                           ; [esp] = Error code
1496 00000959 6A0C
                                              push
                                                        dword 12
                                                      cpu_except en
1497 0000095B EB4C
                                              jmp
                                      exc13:
1498
1499
                                           ; [esp] = Error code
1500 0000095D 6A0D
                                             push
                                                      dword 13
1501 0000095F EB48
                                              jmp
                                                      cpu_except_en
1502
                                      exc14:
1503
                                           ; [esp] = Error code
1504 00000961 6A0E
                                             push
                                                       dword 14
1505 00000963 EB44
                                                short cpu_except_en
                                            jmp
1506
                                      exc15:
1507 00000965 6A0F
                                              push
                                                        dword 15
1508 00000967 EB4E
                                              jmp
                                                      cpu_except
1509
                                      exc16:
1510 00000969 6A10
                                              push
                                                        dword 16
1511 0000096B EB4A
                                              jmp
                                                      cpu_except
1512
                                      exc17:
1513
                                           ; [esp] = Error code
1514 0000096D 6A11
                                                        dword 17
                                             push
1515 0000096F EB38
                                                short cpu_except_en
                                            jmp
1516
                                      exc18:
1517 00000971 6A12
                                             push
                                                        dword 18
1518 00000973 EB42
                                            jmp
                                                  short cpu_except
1519
                                      exc19:
1520 00000975 6A13
                                             push
                                                        dword 19
1521 00000977 EB3E
                                            jmp
                                                 short cpu_except
                                      exc20:
1522
1523 00000979 6A14
                                             push dword 20
1524 0000097B EB3A
                                            jmp short cpu_except
1525
                                      exc21:
1526 0000097D 6A15
                                             push
                                                        dword 21
1527 0000097F EB36
                                            jmp short cpu except
                                      exc22:
1528
                                                      dword 22
1529 00000981 6A16
                                             push
                                            jmp short cpu_except
1530 00000983 EB32
                                      exc23:
1531
1532 00000985 6A17
                                             push
                                                       dword 23
1533 00000987 EB2E
                                            jmp short cpu_except
                                      exc24:
1534
                                             push
1535 00000989 6A18
                                                       dword 24
1536 0000098B EB2A
                                           jmp short cpu_except
                                      exc25:
1538 0000098D 6A19
                                                       dword 25
                                             push
1539 0000098F EB26
                                           jmp short cpu_except
1540
                                      exc26:
                                             push
1541 00000991 6A1A
                                                       dword 26
1542 00000993 EB22
                                           jmp short cpu except
                                      exc27:
1543
                                             push
1544 00000995 6A1B
                                                    dword 27
1545 00000997 EB1E
                                            jmp short cpu_except
```

```
exc28:
1546
1547 00000999 6A1C
                                           push
                                                     dword 28
1548 0000099B EB1A
                                          jmp short cpu_except
                                     exc29:
1549
1550 0000099D 6A1D
                                           push
                                                      dword 29
1551 0000099F EB16
                                          jmp short cpu_except
1553 000009A1 6A1E
                                           push
                                                      dword 30
1554 000009A3 EB04
                                          jmp short cpu_except_en
1555
                                     exc31:
1556 000009A5 6A1F
                                            push
                                                       dword 31
1557 000009A7 EB0E
                                            jmp
                                                    short cpu_except
1558
1559
                                          ; 19/10/2015
                                          ; 19/09/2015
1560
                                          ; 01/09/2015
1561
                                          ; 28/08/2015
1562
1563
                                          ; 28/08/2014
1564
                                     cpu_except_en:
1565 000009A9 87442404
                                         xchg eax, [esp+4]; Error code
1566 000009AD 36A3[78050300]
                                          mov [ss:error_code], eax
1567 000009B3 58
                                                eax ; Exception number
                                          pop
                                          xchg eax, [esp]
1568 000009B4 870424
1569
                                                 ; eax = eax before exception
                                                 ; [esp] -> exception number
1570
1571
                                                 ; [esp+4] \rightarrow EIP to return
1572
                                          ; 22/01/2017
1573
                                          ; 19/10/2015
1574
                                          ; 19/09/2015
1575
                                          ; 01/09/2015
1576
                                          ; 28/08/2015
                                          ; 29/08/2014
1577
1578
                                          ; 28/08/2014
                                          ; 25/08/2014
1579
                                          ; 21/08/2014
1580
                                     cpu_except: ; CPU Exceptions
1581
1582 000009B7 FC
                                          cld
1583 000009B8 870424
                                          xchg eax, [esp]
1584
                                                ; eax = Exception number
1585
                                                 ; [esp] = eax (before exception)
1586 000009BB 53
                                          push ebx
1587 000009BC 56
                                          push esi
1588 000009BD 57
                                          push edi
1589 000009BE 1E
                                          push ds
                                          push es
1590 000009BF 06
                                          ; 28/08/2015
1591
                                          mov bx, KDATA
1592 000009C0 66BB1000
1593 000009C4 8EDB
                                          mov
                                                ds, bx
1594 000009C6 8EC3
                                          mov
                                                es, bx
1595 000009C8 0F20DB
                                                ebx, cr3
                                          mov
1596 000009CB 53
                                          push ebx; (*) page directory
1597
                                          ; 19/10/2015
1598 000009CC FC
                                          cld
                                          ; 25/03/2015
1599
1600 000009CD 8B1D[C0580100]
                                          mov ebx, [k_page_dir]
1601 000009D3 0F22DB
                                          mov
                                                cr3, ebx
                                          ; 28/08/2015
1603 000009D6 83F80E
                                          cmp eax, OEh ; 14, PAGE FAULT
1604 000009D9 750F
                                          jne
                                                short cpu_except_nfp
1605 000009DB E8CF440000
                                          call page_fault_handler
1606 000009E0 21C0
                                          and eax, eax
1607 000009E2 0F8401010000
                                          jz iiretp ; 01/09/2015
1608 000009E8 B00E
                                          mov al, 0Eh; 14
1609
                                     cpu_except_nfp:
1610
                                          ; 23/08/2016
1611 000009EA 803D[225F0000]03
                                          cmp byte [CRT_MODE], 3
1612 000009F1 7409
                                          jе
                                                short cpu_except_mode_3
1613 000009F3 50
                                          push eax
                                                al, 3
1614 000009F4 B003
                                          mov
                                          call _set_mode
1615 000009F6 E8730B0000
1616 000009FB 58
                                          pop eax
                                     cpu except mode 3:
1617
1618
                                          ; 02/04/2015
1619 000009FC BB[4B060000]
                                          mov ebx, hang
1620 00000A01 875C241C
                                          xchg ebx, [esp+28]
1621
                                                 ; EIP (points to instruction which faults)
                                                 ; New EIP (hang)
1623 00000A05 891D[7C050300]
                                                [FaultOffset], ebx
                                          mov
1624 00000A0B C744242008000000
                                          mov
                                                 dword [esp+32], KCODE ; kernel's code segment
1625 00000A13 814C242400020000
                                                 dword [esp+36], 200h; enable interrupts (set IF)
                                          or
1626
1627 00000A1B 88C4
                                          mov
                                                 ah, al
1628 00000A1D 240F
                                          and
                                                al, OFh
1629 00000A1F 3C09
                                                al, 9
1630 00000A21 7602
                                          jna
                                                 short hlok
                                                 al, 'A'-':'
1631 00000A23 0407
                                          add
                                     hlok:
1633 00000A25 C0EC04
                                                 ah, 4
                                          shr
1634 00000A28 80FC09
                                                 ah, 9
                                          cmp
1635 00000A2B 7603
                                                 short h2ok
                                          jna
                                                 ah, 'A'-':'
1636 00000A2D 80C407
                                          add
                                    h2ok:
1638 00000A30 86E0
                                                 ah, al
                                          xchg
1639 00000A32 66053030
                                                 ax, '00'
                                          add
1640 00000A36 66A3[30190100]
                                          mov
                                                 [excnstr], ax
1641
                                          ; 29/08/2014
1642
1643 00000A3C A1[7C050300]
                                                eax, [FaultOffset]
                                          mov
1644 00000A41 51
                                          push
                                                ecx
1645 00000A42 52
                                          push edx
                                                ebx, esp
1646 00000A43 89E3
                                          mov
1647
                                          ; 28/08/2015
1648 00000A45 B910000000
                                          mov ecx, 16
                                                             ; divisor value to convert binary number
1649
                                                       ; to hexadecimal string
1650
                                          ; mov ecx, 10 ; divisor to convert
```

```
1651
                                                             ; binary number to decimal string
1652
                                     b2d1:
1653 00000A4A 31D2
                                           xor
                                                 edx, edx
1654 00000A4C F7F1
                                           div
                                                  ecx
1655 00000A4E 6652
                                           push dx
1656 00000A50 39C8
                                           cmp
                                                  eax, ecx
1657 00000A52 73F6
                                           jnb
                                                  short b2d1
1658 00000A54 BF[3B190100]
                                                  edi, EIPstr ; EIP value
                                           mov
1659
                                                            ; points to instruction which faults
1660
                                           ; 28/08/2015
1661 00000A59 89C2
                                           mov
                                                  edx, eax
                                     b2d2:
1662
                                                 al, '0'
1663
                                           ;add
1664 00000A5B 8A82[24330000]
                                                  al, [edx+hexchrs]
                                           mov
1665 00000A61 AA
                                                            ; write hexadecimal digit to its place
                                           stosb
1666 00000A62 39E3
                                           cmp
                                                  ebx, esp
1667 00000A64 7606
                                           jna
                                                  short b2d3
1668 00000A66 6658
                                           pop
                                                  ax
1669 00000A68 88C2
                                           mov
                                                  dl, al
1670 00000A6A EBEF
                                                  short b2d2
                                           jmp
                                     b2d3:
1671
1672 00000A6C B068
                                                  al, 'h'; 28/08/2015
                                           mov
1673 00000A6E AA
                                           stosb
1674 00000A6F B020
                                           mov
                                                  al, 20h
                                                                 ; space
1675 00000A71 AA
                                           stosb
1676 00000A72 30C0
                                           xor
                                                 al, al
                                                            ; to do it an ASCIIZ string
1677 00000A74 AA
                                           stosb
1678
                                           ;
1679 00000A75 5A
                                           pop
                                                  edx
1680 00000A76 59
                                           pop
                                                  ecx
1681
1682 00000A77 B44F
                                                  ah, 4Fh
                                                             ; red (4) background,
                                           mov
                                                      ; white (F) forecolor
1683
1684 00000A79 BE[20190100]
                                           mov
                                                  esi, exc_msg ; message offset
1685
                                           ; 20/01/2017 (!cpu exception!)
1686
1687
1688 00000A7E 8105[D6160100]A000-
                                             add
                                                    dword [scr_row], 0A0h
1688 00000A86 0000
1689 00000A88 8B3D[D6160100]
                                                    edi, [scr_row]
                                             mov
1690
1691 00000A8E C605[5B030300]00
                                           mov
                                                 byte [sysflg], 0 ; system mode
1692 00000A95 FB
                                             sti
1693
                                                 printk
1694 00000A96 E8EFFBFFFF
                                           call
1695
1696 00000A9B B410
                                           mov
                                                  ah, 10h
1697 00000A9D E87D010000
                                                 int16h ; getc
                                           call
1698
1699 00000AA2 B003
                                                  al, 3
                                           mov
1700 00000AA4 E8C50A0000
                                           call
                                                  _set_mode
                                           ;
1702 00000AA9 B801000000
                                           mov
                                                  eax, 1
1703 00000AAE E936BE0000
                                                 sysexit; terminate process !!!
                                           jmp
1704
1705
                                           ; 22/01/2017
1706
                                           ; 18/04/2016
                                           ; 28/08/2015
1707
                                           ; 23/02/2015
1708
1709
                                           ; 20/08/2014
1710
                                      ignore_int:
1711 00000AB3 50
                                           push eax
1712 00000AB4 53
                                           push ebx; 23/02/2015
1713 00000AB5 56
                                           push esi
1714 00000AB6 57
                                           push
                                                  edi
1715 00000AB7 1E
                                           push
                                                  ds
                                           push
1716 00000AB8 06
                                                 es
1717
                                           ; 18/04/2016
1718 00000AB9 66B81000
                                           mov
                                                 ax, KDATA
1719 00000ABD 8ED8
                                                 ds, ax
                                           mov
1720 00000ABF 8EC0
                                           mov
                                                  es, ax
1721
                                           ; 28/08/2015
1722 00000AC1 0F20D8
                                           mov
                                                  eax, cr3
1723 00000AC4 50
                                           push eax; (*) page directory
1724
                                           ;
1725 00000AC5 B467
                                           mov
                                                  ah, 67h
                                                              ; brown (6) background,
1726
                                                  ; light gray (7) forecolor
1727 00000AC7 BE[E8170100]
                                           mov
                                                  esi, int msg ; message offset
1728
                                      piemsg:
1729
                                             ; 27/08/2014
1730 00000ACC 8105[D6160100]A000-
                                             add dword [scr_row], 0A0h
1730 00000AD4 0000
1731 00000AD6 8B3D[D6160100]
                                                     edi, [scr_row]
                                             mov
1732
                                           call printk
1733 00000ADC E8A9FBFFFF
1734
                                           ; 23/02/2015
1735
                                           mov al, 20h; END OF INTERRUPT COMMAND TO out 0A0h, al; the 2nd 8259
1736 00000AE1 B020
1737 00000AE3 E6A0
1738
                                           ; 22/08/2014
                                           mov al, 20h; END OF INTERRUPT COMMAND TO 8259
1739 00000AE5 B020
1740 00000AE7 E620
                                                 20h, al ; 8259 PORT
                                           out
1741
                                      iiretp:
                                          ; 22/01/2017
1742
1743
                                           ; 01/09/2015
1744
                                           ; 28/08/2015
1745 00000AE9 58
                                           pop eax; (*) page directory
1746 00000AEA 0F22D8
                                           mov cr3, eax
                                      iiret:
1747
1748 00000AED 07
                                           pop
                                                  es
1749 00000AEE 1F
                                           pop
                                                 ds
1750 00000AEF 5F
                                                  edi
                                           pop
1751 00000AF0 5E
                                           pop
                                                 esi
                                                 ebx ; 29/08/2014
1752 00000AF1 5B
                                           pop
1753 00000AF2 58
                                                 eax
                                           pop
```

```
1754 00000AF3 CF
                                             iretd
1755
1756
                                             ; 23/05/2016
1757
                                             ; 22/08/2014
                                             ; IBM PC/AT BIOS source code ---- 10/06/85 (bios.asm)
1758
1759
                                              ; (INT 1Ah)
1760
                                             ;; Linux (v0.12) source code (main.c) by Linus Torvalds (1991)
1761
                                       time_of_day:
                                       call UPD_IPR
1762 00000AF4 E842510000
                                                                                ; WAIT TILL UPDATE NOT IN PROGRESS
                                             jc short time_of_day_retn ; 23/05/2016
1763 00000AF9 726F
1764 00000AFB B000
                                           mov al, CMOS_SECONDS
                                        mov [time_seconds], a
mov al, CMOS_MINUTES
call CMOS_READ
mov [time_minutes],
mov al, CMOS_HOURS
call CMOS_READ
                                            call CMOS_READ
mov [time_seconds], al
1765 00000AFD E854510000
1766 00000B02 A2[30590100]
1767 00000B07 B002
1768 00000B09 E848510000
1769 00000B0E A2[31590100]
                                                    [time_minutes], al
                                       mov al, CMOS_HOURS
call CMOS_READ
mov [time_hours], a
mov al, CMOS_DAY_WEEK
call CMOS_READ
mov [date_wday], al
mov al, CMOS_DAY_MONTH
call CMOS_READ
mov [date_day], al
mov al, CMOS_MONTH
call CMOS_READ
mov [date_month], al
mov al, CMOS_YEAR
1770 00000B13 B004
1771 00000B15 E83C510000
1772 00000B1A A2[32590100]
                                             mov [time_hours], al
1773 00000B1F B006
1774 00000B21 E830510000
1775 00000B26 A2[33590100]
1776 00000B2B B007
1777 00000B2D E824510000
1778 00000B32 A2[34590100]
1779 00000B37 B008
1780 00000B39 E818510000
1781 00000B3E A2[35590100]
1782 00000B43 B009
                                                    al, CMOS_YEAR
                                             mov
                                            call CMOS READ
1783 00000B45 E80C510000
                                          mov
1784 00000B4A A2[36590100]
                                                    [date_year], al
1785 00000B4F B032
                                                    al, CMOS CENTURY
                                             mov
1786 00000B51 E800510000
                                             call CMOS READ
1787 00000B56 A2[37590100]
                                           mov
                                                    [date_century], al
                                            ;
1788
1789 00000B5B B000
                                                    al, CMOS_SECONDS
                                            mov
                                           call CMOS_READ
1790 00000B5D E8F4500000
1791 00000B62 3A05[30590100]
                                             cmp al, [time_seconds]
1792 00000B68 758A
                                                    short time_of_day
                                             jne
1793
1794
                                       time_of_day_retn:
1795 00000B6A C3
                                             retn
1796
1797
                                             ; 15/01/2017
1798
                                             ; 10/06/2016
1799
                                             ; 07/06/2016
                                             ; 06/06/2016
1800
1801
                                             ; 23/05/2016
1802
                                       rtc_p:
1803 00000B6B B101
                                            mov cl, 1; 15/01/2017
1804 00000B6D EB02
                                             jmp short rtc_p0
1805
                                       u timer:
                                            ; Timer Events with 18.2 Hz Timer Ticks
1806
1807
                                              ; (and also timer events with RTC seconds)
1808 00000B6F 28C9
                                             sub cl, cl; mov cl, 0; 15/01/2017
1809
                                        rtc_p0:
                                             ; 19/05/2016 - TRDOS 386 (TRDOS v2.0)
1810
                                              ; Major Modification:
1811
1812
                                              ; Check and Perform Timer Events (for RTC)
1813
                                             ; 25/08/2014 - 07/09/2014
1814
                                             ; Retro UNIX 386 v1:
1815
                                             ; Print Real Time Clock content
1816
                                             ; 15/01/2017
1818 00000B71 880D[50650100]
                                             mov byte [priority], cl; 0 or 1 (not 2)
1819 00000B77 8A2D[53650100]
                                             mov
                                                    ch, [timer_events]
1820 00000B7D 20ED
                                             and ch, ch
1821 00000B7F 7420
                                             jz
                                                    short rtc_p3
1822
1823 00000B81 BE[60040300]
                                             mov
                                                    esi, timer_set ; beginning address of
1824
                                                                 ; timer events space
1825
                                       rtc_p1:
1826 00000B86 8B06
                                                    eax, [esi]
                                             mov
                                                    al, al; 0 = free, >0 = process no.
1827 00000B88 20C0
                                              and
1828 00000B8A 7416
                                                    short rtc_p4
                                             jг
1829
1830 00000B8C C1C810
                                             ror eax, 16
                                              ; ah = response value, al = interrupt type
1831
1832
                                              ; 15/01/2017
1833
                                              ; cl = interrupt source
1834
                                                    1 = RTC, 0 = PIT
1835 00000B8F 38C8
                                                     al, cl
                                              cmp
1836 00000B91 750A
                                                     short rtc_p2; not as requested or undefined!
1837 00000B93 3C01
                                                    al, 1; 1; RTC interrupt?
1838 00000B95 7410
                                                    short rtc p5 ; yes, check for response
                                              jе
                                              ; 06/06/2016 - 18.2 Hz Timer Ticks
1839
1840 00000B97 836E080A
                                              sub dword [esi+8], 10; 1 tick = 10
1841 00000B9B 7613
                                                    short rtc_p6 ; continue for responding
                                              jna
                                        rtc_p2:
1842
1843
                                             ; 15/01/2017 (cl -> ch)
1844
                                              ; 07/06/2016
1845 00000B9D FECD
                                              dec
                                                    ch ; remain count of timer events
1846 00000B9F 7501
                                              jnz
                                                    short rtc_p4
                                        rtc_p3:
1848 00000BA1 C3
                                             retn
1849
                                        rtc_p4:
1850
                                             ;cmp esi, timer set + 240 ; 15*16 (last event)
1851
                                              ;jnb short rtc_p3 ; end of timer event space
                                                     esi, 16; next timer event
1852 00000BA2 83C610
1853 00000BA5 EBDF
                                              jmp
                                                    short rtc_p1
1854
                                        rtc_p5:
1855
                                              ; current timer count ; 06/06/2016 (182)
                                                     dword [esi+8], 182; 1 second (10*18.2)
1856 00000BA7 816E08B6000000
                                              sub
1857 00000BAE 77ED
                                                     short rtc_p2 ; check for the next
                                              jа
                                        rtc_p6:
```

```
1859
                                        ; it is the time of response!
                                        mov ebx, [esi+4]; set (count limit) value mov [esi+8], ebx; reset count down value
1860 00000BB0 8B5E04
1861 00000BB3 895E08
                                                         ; to count limit
1862
1863
                                        ; 19/12/2016
                                        ; 10/12/2016 - timer callback modification
1864
1865 00000BB6 8B7E0C
                                        mov edi, [esi+12]; response (or callback) address
                                             byte [esi+1], 0 ; >0 = callback
short rtc_p8
1866 00000BB9 807E0100
                                        cmp
1867 00000BBD 762A
                                        jna
1868
1869
                                        ; timer callback !
1870 00000BBF 0FB61E
                                        movzx ebx, byte [esi] ; process number (>0)
1871 00000BC2 89D8
                                        mov eax, ebx
1872 00000BC4 C0E302
                                        shl bl, 2; *4
                                        mov [ebx+p.tcb-4], edi ; user's callback service addr
1873 00000BC7 89BB[0C010300]
1874 00000BCD 3A05[B3030300]
                                        cmp
                                              al, [u.uno]
1875 00000BD3 7521
                                        jne short rtc p9
1876 00000BD5 893D[D0030300]
                                        mov
                                              [u.tcb], edi
1877
                                   rtc_p7:
                                        ; 15/01/2017
1878
1879 00000BDB B002
                                        mov al, 2
1880 00000BDD A2[50650100]
                                              [priority], al ; 2
                                        mov
                                        ; 10/01/2017
1881
                                        ;mov byte [u.pri], 2
1883 00000BE2 A2[A9030300]
                                        mov [u.pri], al; 2
1884 00000BE7 EBB4
                                        jmp
                                             short rtc_p2
1885
                                   rtc p8:
1886
                                        ; response address is physical address of
1887
                                        ; the program's response (signal return) byte
                                        ; 06/06/2016
1888
1889
                                        ;mov edi, [esi+12] ; response address
1890 00000BE9 8827
                                        mov
                                              [edi], ah ; response value
1891
1892 00000BEB C1C010
                                        rol
                                              eax, 16
1893
                                        ; 15/01/2017
1894 00000BEE 3A05[B3030300]
                                         cmp al, [u.uno] ; running process ?
1895 00000BF4 74E5
                                        jе
                                              short rtc p7
1896
                                   rtc_p9:
1897
                                        ; al = process number ; 10/06/2016
                                        mov dl, 2; priority, 2 = event (high)
1898 00000BF6 B202
1899 00000BF8 E872ED0000
                                        call set_run_sequence ; 19/05/2016
1900 00000BFD EB9E
                                        jmp short rtc_p2 ; 10/06/2016
1901
1902
                                   ; Default IRQ 7 handler against spurious IRQs (from master PIC)
1903
                                   ; 25/02/2015 (source: http://wiki.osdev.org/8259_PIC)
1904
1905
                                   default irq7:
1906 00000BFF 6650
                                        push ax
1907 00000C01 B00B
                                        mov
                                              al, OBh ; In-Service register
                                        out 20h, al
1908 00000C03 E620
1909 00000C05 EB00
                                         jmp short $+2
1910 00000C07 EB00
                                         jmp short $+2
1911 00000C09 E420
                                        in al, 20h
1912 00000C0B 2480
                                         and al, 80h; bit 7 (is it real IRQ 7 or fake?)
                                         jz short irq7_iret ; Fake (spurious) IRQ, do not send EOI
mov al, 20h ; EOI
1913 00000C0D 7404
1914 00000C0F B020
                                        out 20h, al
1915 00000C11 E620
1916
                                   irq7_iret:
1917 00000C13 6658
                                        pop
1918 00000C15 CF
                                        iretd
1919
1920
                                   bcd to ascii:
                                        ; 25/08/2014
1921
1922
                                        ; INPUT ->
1923
                                        ; al = Packed BCD number
                                         ; OUTPUT ->
1924
1925
                                             ax = ASCII word/number
1926
1927
                                         ; Erdogan Tan - 1998 (proc_hex) - TRDOS.ASM (2004-2011)
1928
                                                                     ; Undocumented inst. AAM
1929 00000C16 D410
                                         db 0D4h,10h
                                                                 ; AH = AL / 10h
1930
                                                                 ; AL = AL MOD 10h
1931
                                                                      ; Make it ASCII based
1932 00000C18 660D3030
                                        or ax,'00'
1933
1934 00000C1C 86E0
                                          xchg ah, al
1935
1936 00000C1E C3
                                        retn
1937
1938
                                   %include 'keyboard.s'; 07/03/2015
1939
                                <1> ; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.0 - keyboard.s
  4
                               <1> ; Last Update: 15/01/2017
  5
                               <1> ; -----
  6
                               <1> ; Beginning: 17/01/2016
  7
                               <1>; ------
  8
                               <1>; Assembler: NASM version 2.11 (trdos386.s)
  9
 10
                               <1>; Turkish Rational DOS
 11
                               <1>; Operating System Project v2.0 by ERDOGAN TAN (Beginning: 04/01/2016)
 12
                               <1> :
 13
                               <1>; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
 14
                               <1>; keyboard.inc (17/10/2015)
                               <1>;
 1.5
                               <1>; Derived from 'IBM PC-XT-286' BIOS source code (1986)
 16
                               17
 18
                               <1>
                               <1>; Retro UNIX 386 v1 Kernel - KEYBOARD.INC
 19
                               <1> ; Last Modification: 17/10/2015
 2.0
 21
                               <1>;
                                                  (Keyboard Data is in 'KYBDATA.INC')
 22
                               <1>;
 23
                               <1> ; ////// KEYBOARD FUNCTIONS (PROCEDURES) //////////
 24
```

```
25
                                <1> ; 17/01/2016 (TRDOS 386 = TRDOS v2.0)
 26
 27
                                 <1> ; 03/12/2014
 28
                                 <1> ; 26/08/2014
 29
                                 <1> : KEYBOARD I/O
 30
                                 <1>; (INT 16h - Retro UNIX 8086 v1 - U9.ASM, 30/06/2014)
                                <1> ; NOTE: 'k0' to 'k7' are name of OPMASK registers.
 32
 33
                                 <1>; (The reason of using '_k' labels!!!) (27/08/2014)
                                 <1> ;NOTE: 'NOT' keyword is '~' unary operator in NASM.
 34
                                 <1> ; ('NOT LC_HC' --> '~LC_HC') (bit reversing operator)
 35
 36
                                 <1>
                                <1> int16h: ; 30/06/2015
 37
 38
                                <1> ;getc:
                                     pushfd; 28/08/2014
                                     push cs
call KEYBOARD_IO_1; getc_int
retn
 39 00000C1F 9C
                                <1>
 40 00000C20 0E
                                <1>
 41 00000C21 E801000000
                                <1>
 42 00000C26 C3
                                <1>
 43
                                <1>
 44
                                <1> getc_int:
                                <1> ; 28/02/2015
 45
                                         ; 03/12/2014 (derivation from pc-xt-286 bios source code -1986-,
 46
                                <1>
                                                  instead of pc-at bios - 1985-)
 47
                                <1>
                                     ; 28/08/2014 (_k1d)
 48
                                <1>
                                       ; 30/06/2014
 49
                                <1>
                                        ; 03/03/2014
; 28/02/2014
 50
                                <1>
                                <1>
                                        ; Derived from "KEYBOARD_IO_1" procedure of IBM "pc-xt-286"
 52
                                <1>
                                        ; rombios source code (21/04/1986)
; 'keybd.asm', INT 16H, KEYBOARD_IO
 53
                                <1>
 54
                                <1>
 55
                                <1>
 56
                                 <1>
                                         ; KYBD --- 03/06/86 KEYBOARD BIOS
 57
                                <1>
 58
                                <1>
                                         ;--- INT 16 H ------
                                        ; KEYBOARD I/O
 59
                                <1>
 60
                                <1>
                                         ; THESE ROUTINES PROVIDE READ KEYBOARD SUPPORT
                                          ; INPUT
 61
                                        ; (AH) = 00H READ THE NEXT ASCII CHARACTER ENTERED FROM THE KEYBOARD,
                                <1>
 62
                                                         RETURN THE RESULT IN (AL), SCAN CODE IN (AH).
 63
                                 <1>
                                                         THIS IS THE COMPATIBLE READ INTERFACE, EQUIVALENT TO THE
 64
                                <1>
 65
                                <1>
                                                           STANDARD PC OR PCAT KEYBOARD
 66
                                 <1>
                                              (AH)= 01H SET THE ZERO FLAG TO INDICATE IF AN ASCII CHARACTER IS
 67
                                <1>
                                                       AVAILABLE TO BE READ FROM THE KEYBOARD BUFFER. :
 68
                                <1>
                                                          (ZF) = 0 -- CODE IS AVAILABLE (AX) = CHARACTER
IF (ZF) = 0. The MEYE CULTAGE.
 69
                                <1>
                                                          (ZF) = 1 -- NO CODE AVAILABLE
 70
                                <1>
 71
                                <1>
                                                         IF (ZF) = 0, THE NEXT CHARACTER IN THE BUFFER TO BE READ IS:
                                                    IN (AX), AND THE ENTRY REMAINS IN THE BUFFER.
THIS WILL RETURN ONLY PC/PCAT KEYBOARD COMPATIBLE CODES
 72
                                <1>
 73
                                <1>
 74
                                <1>
                                          ; (AH)= 02H RETURN THE CURRENT SHIFT STATUS IN AL REGISTER :
 75
                                <1>
                                                  THE BIT SETTINGS FOR THIS CODE ARE INDICATED IN THE
 76
                                 <1>
 77
                                <1>
                                                          EQUATES FOR @KB_FLAG
                                          ;-----:
 78
                                <1>
                                              (AH) = 03H SET TYPAMATIC RATE AND DELAY
 79
                                <1>
                                                  (AL) = 05H
(BL) = TYPAMATIC RATE (BITS 5 - 7 MUST BE RESET TO 0)
 80
                                <1>
 81
                                 <1>
                                <1>
 82
 83
                                <1>
                                                               REGISTER RATE REGISTER
                                                                                                  RATE
                                                               VALUE SELECTED VALUE SELECTED
 84
                                <1>
                                                            00H 30.0 10H 7.5
01H 26.7 11H 6.7
02H 24.0 12H 6.0
03H 21.8 13H 5.5
04H 20.0 14H 5.0
05H 18.5 15H 4.6
06H 17.1 16H 4.3
07H 16.0 17H 4.0
08H 15.0 18H 3.7
09H 13.3 19H 3.3
0AH 12.0 1AH 3.0
0BH 10.9 1BH 2.7
0CH 10.0 1CH 2.5
0DH 9.2 1DH 2.3
0EH 8.6 1EH 2.1
0FH 8.0 1FH 2.0
 85
                                <1>
                                                               _____
 86
                                <1>
 87
                                <1>
 88
                                <1>
 89
                                <1>
 90
                                <1>
                                <1>
 92
                                <1>
 93
                                <1>
 94
                                <1>
 95
                                <1>
 96
                                <1>
 97
                                <1>
 98
                                <1>
                                                                                  1DH 2.3
1EH 2.1
1FH 2.0
 99
                                <1>
100
                                <1>
                                                                       8.0
101
                                 <1>
                                                             OFH
102
                                <1>
                                                       (BH) = TYPAMATIC DELAY (BITS 2 - 7 MUST BE RESET TO 0)
103
                                 <1>
104
                                <1>
                                                                            DELAY
                                                               REGISTER
105
                                <1>
106
                                 <1>
                                                                 VALUE
                                                                            VALUE
107
                                <1>
108
                                <1>
                                                             00H 250 ms
109
                                 <1>
                                                             01H
                                                                        500 ms
                                                             01H 500 ms
02H 750 ms
03H 1000 ms
110
                                 <1>
111
                                 <1>
                                <1>
                                          ;-----:
112
113
                                <1>
                                               (AH)= 05H PLACE ASCII CHARACTER/SCAN CODE COMBINATION IN KEYBOARD
                                                        BUFFER AS IF STRUCK FROM KEYBOARD
                                 <1>
114
115
                                <1>
                                                          ENTRY: (CL) = ASCII CHARACTER
116
                                 <1>
                                                                  (CH) = SCAN CODE
                                                                 (AH) = 00H = SUCCESSFUL OPERATION
117
                                <1>
118
                                <1>
                                                                  (AL) = 01H = UNSUCCESSFUL - BUFFER FULL
119
                                <1>
                                                         FLAGS: CARRY IF ERROR
120
                                <1>
                                          ; (AH) = 10H EXTENDED READ INTERFACE FOR THE ENHANCED KEYBOARD, :
121
                                 <1>
                                <1>
                                                  OTHERWISE SAME AS FUNCTION AH=0
122
123
                                <1>
                                              (AH) = 11H EXTENDED ASCII STATUS FOR THE ENHANCED KEYBOARD,
124
                                <1>
                                                       OTHERWISE SAME AS FUNCTION AH=1
125
                                <1>
126
                                 <1>
                                             (AH) = 12H RETURN THE EXTENDED SHIFT STATUS IN AX REGISTER

AL = BITS FROM KB_FLAG, AH = BITS FOR LEFT AND RIGHT:
127
                                <1>
128
                                <1>
129
                                <1>
                                                          CTL AND ALT KEYS FROM KB_FLAG_1 AND KB_FLAG_3
```

```
130
                                                           ; OUTPUT
; AS NOTED ABOVE, ONLY (AX) AND FLAGS CHANGED
; ALL REGISTERS RETAINED
                                                   <1>
 131
 132
                                                   <1>
 133
                                                   <1>
 134
                                                   <1>
                                                   <1> ; 15/01/2017
 135
 136
                                                   <1> ; 14/01/2017
 137
                                                   <1>; 02/01/2017
 138
                                                   <1> ; 29/05/2016
                                                   <1>; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
 139
 140
                                                   <1> int32h: ; Keyboard BIOS
 141
                                                   <1> KEYBOARD IO_1:
 142
 143
                                                  <1> ;sti
                                                                                                      ; INTERRUPTS BACK ON
                                                                 ; 29/05/2016
                                                  <1>
 144
 145 00000C27 80642408BE
                                                                 and byte [esp+8], 101111110b; clear zero flag and cary flag
                                                 <1>
                                                 <1>
                                                          push ds
push ebx
;push ecx
mov bx, KDATA
                                                                                                       ; SAVE CURRENT DS
 147 00000C2C 1E
                                                 <1>
 148 00000C2D 53
                                                  <1>
                                                                                                        ; SAVE BX TEMPORARILY
 149
                                                 <1>
                                                                                                        ; SAVE CX TEMPORARILY
                                           <1>
 150 00000C2E 66BB1000
                                                                                                        ; PUT SEGMENT VALUE OF DATA AREA INTO DS
 151 00000C32 8EDB
                                                  <1>
                                                                mov ds, bx
 152
                                                 <1>
 153
                                                 <1>
                                                                 ; 14/01/2017
                                                             mov ebx, [esp]
 154 00000C34 8B1C24
                                                  <1>
 155
                                                  <1>
                                                                 ;; 15/01/2017
 156
                                                  <1>
                                                                 ; 02/01/2017
or ah, ah ; CHECK FOR (AH) = 00H ; Short K1 ; ASCII_READ ; CHECK FOR (AH) = 0 CHECK FOR (
                                                  <1>
 157
                                                                ;;mov byte [intflg], 32h ; keyboard interrupt
                                                                                                       ; CHECK FOR (AH) = 01H
; ASCII_STATUS
                                                                                                        ; SET TYPAMATIC RATE/DELAY
                                                                                                          ; KEYBOARD WRITE
                                                ; AH = 10H
; EXTENDED ASCII READ
                                            jz short _K1E
                                                                                                     ; CHECK FOR (AH) = 11H
 174 00000C5E FECC
                                                                                                ; EXTENDED_ASCII_STATUS
; CHECK FOR (AH) = 12H
; EXTENDED_SHIFT_STATUS
 175 00000C60 7422
                                                                           short _K2E
 176 00000C62 FECC
 177 00000C64 7458
                                                                          short _K3E
                                                 <1> _KIO_EXIT:
<1> ; 02/01/2017
 178
 179
 180 00000C66 FA
                                                              cli
                                                  <1>
                                                                 ;;mov byte [intflg], 0 ;; 15/01/2017
 181
                                                  <1>
 182
                                                  <1>
 183
                                                 <1>
                                                                                                       ; RECOVER REGISTER
                                                                 ;pop
                                                                                                       ; RECOVER REGISTER
 184 00000C67 5B
                                                  <1>
                                                                 pop ebx
 185 00000C68 1F
                                                  <1>
                                                                          ds
                                                                                                        ; RECOVER SEGMENT
                                                                 pop
 186 00000C69 CF
                                                  <1>
                                                                 iretd
                                                                                                        ; INVALID COMMAND, EXIT
 187
                                                  <1>
 188
                                                  <1>
                                                                 ;---- ASCII CHARACTER
 189
                                                 <1> _K1E:
                                                                 call _K1S ; GET A CHARACTER FROM THE BUFFER (EXTENDED)
call _KIO_E_XLAT ; ROUTINE TO XLATE FOR EXTENDED CALLS
jmp short _KIO_EXIT ; GIVE IT TO THE CALLER
                                                 <1>
 190 00000C6A E8D3000000
 191 00000C6F E848010000
                                                 <1>
                                                 <1>
 192 00000C74 EBF0
                                                 <1> _K1:
                                                                 call _K1S
call _KIO_S_XLAT
                                                  <1>
                                                                                                        ; GET A CHARACTER FROM THE BUFFER
 194 00000C76 E8C7000000
 195 00000C7B E847010000
                                                  <1>
                                                                                                        ; ROUTINE TO XLATE FOR STANDARD CALLS
 196 00000C80 72F4
                                                 <1>
                                                                 jc short _K1
                                                                                                        ; CARRY SET MEANS TROW CODE AWAY
                                                  <1> _K1A:
 197
 198 00000C82 EBE2
                                                  <1>
                                                                 jmp short _KIO_EXIT
                                                                                                     ; RETURN TO CALLER
 199
                                                  <1>
                                                                 ;---- ASCII STATUS
 200
                                                  <1>
 201
                                                  <1> _K2E:
 202 00000C84 E804010000
                                                  <1>
                                                           call _K2S
                                                                                                        ; TEST FOR CHARACTER IN BUFFER (EXTENDED)
 203 00000C89 7420
                                                                                                      ; RETURN IF BUFFER EMPTY
                                                 <1>
                                                                 jz short _K2B
                                                                                                     ; SAVE ZF FROM TEST
                                                                 pushf
 204 00000C8B 9C
                                                  <1>
                                                                 call _KIO_E_XLAT
jmp short _K2A
                                                                                                ; ROUTINE TO XLATE FOR EXTENDED CALLS
 205 00000C8C E82B010000
                                                 <1>
                                                 <1>
 206 00000C91 EB17
                                                                                                        ; GIVE IT TO THE CALLER
                                                  <1> _K2:
 207
                                                           call _K2S
 208 00000C93 E8F5000000
                                                  <1>
                                                                                                        ; TEST FOR CHARACTER IN BUFFER
 209 00000C98 7411
                                                                 jz short K2B
                                                                                                        ; RETURN IF BUFFER EMPTY
                                                  <1>
                                                          pushf
 210 00000C9A 9C
                                                  <1>
                                                                                                      ; SAVE ZF FROM TEST
                                                                                                     ; ROUTINE TO XLATE FOR STANDARD CALLS ; CARRY CLEAR MEANS PASS VALID CODE
 211 00000C9B E827010000
                                                   <1>
                                                                 call _KIO_S_XLAT
 212 00000CA0 7308
                                                  <1>
                                                                            short _K2A
                                                                  jnc
 213 00000CA2 9D
                                                   <1>
                                                                                                        ; INVALID CODE FOR THIS TYPE OF CALL
                                                                  popf
                                                                                                      ; THROW THE CHARACTER AWAY
 214 00000CA3 E89A000000
                                                  <1>
                                                                  call
                                                                            K1S
                                                                            short K2
 215 00000CA8 EBE9
                                                  <1>
                                                                                                        ; GO LOOK FOR NEXT CHAR, IF ANY
                                                                  jmp
                                                  <1> K2A:
 217 00000CAA 9D
                                                  <1>
                                                                                                        ; RESTORE ZF FROM TEST
                                                                 popf
 218
                                                  <1> K2B:
 219
                                                  <1>
                                                                 ; 02/01/2017
 220 00000CAB FA
                                                  <1>
                                                                  cli
                                                                 ;; mov byte [intflq], 0 ;; 15/01/2017
 221
                                                   <1>
                                                   <1>
 222
 223
                                                  <1>
                                                                                                        ; RECOVER REGISTER
                                                                 ;pop ecx
 224 00000CAC 5B
                                                  <1>
                                                                 pop ebx
                                                                                                        ; RECOVER REGISTER
 225 00000CAD 1F
                                                  <1>
                                                                 pop
                                                                          ds
                                                                                                        ; RECOVER SEGMENT
                                                                 ; (*) 29/05/2016
 226
                                                  <1>
                                                                 ; (*) retf 4
                                                                                                        ; THROW AWAY (e) FLAGS
 227
                                                  <1>
                                                                          short _k2d
short k2c
 228 00000CAE 7208
                                                   <1>
                                                                  jс
 229 00000CB0 7505
                                                  <1>
                                                                  jnz
 230 00000CB2 804C240840
                                                                          byte [esp+8], 01000000b ; set zero flag bit of eflags register
                                                  <1>
                                                                  or
                                                   <1> _k2c:
                                                  <1>
 232 00000CB7 CF
                                                                 iretd
 233
                                                   <1> k2d:
 234
                                                   <1>
                                                            ; 29/05/2016 -set carry flag on stack-
```

```
; [esp] = EIP
235
                                <1>
236
                                <1>
                                         ; [esp+4] = CS
                                         ; [esp+8] = E-FLAGS
                                <1>
237
                                       or byte [esp+8], 1 ; set carry bit of eflags register
238 00000CB8 804C240801
                                <1>
                                <1>
                                         ; [esp+12] = ESP (user)
239
240
                                <1>
                                         ; [esp+16] = SS (User)
241 00000CBD CF
                                <1>
                                         iretd
                                <1>
242
243
                                <1>
                                         ; (*) 29/05/2016 - 'ref 4' intruction causes to stack fault
244
                                <1>
                                         ; (OUTER-PRIVILEGE-LEVEL)
245
                                <1>
                                         ; INTEL 80386 PROGRAMMER'S REFERENCE MANUAL 1986
246
                                <1>
247
                                <1>
                                         ; // RETF instruction:
248
                                <1>
                                       ; IF OperandMode=32 THEN
249
                                <1>
                                         ; Load CS:EIP from stack;
250
                                <1>
                                              Set CS RPL to CPL;
                                       ; Increment eSP by 8 plus the immediate offset if it exists;
                                <1>
252
253
                                <1>
                                              Load SS:eSP from stack;
254
                                <1>
                                         ; ELSE (* OperandMode=16 *)
                                       ; Load CS: IP from stack;
255
                                <1>
256
                                <1>
                                              Set CS RPL to CPL;
                                             Increment eSP by 4 plus the immediate offset if it exists;
257
                                <1>
258
                                <1>
                                         ; Load SS:eSP from stack;
                                       ; FI;
259
                                <1>
260
                                <1>
261
                                <1>
                                         ; //
262
                                <1>
                                <1>
                                         ;---- SHIFT STATUS
263
                               <1> _K3E:
                                                                        ; GET THE EXTENDED SHIFT STATUS FLAGS
                                                                  ; GET SYSTEM SHIFT KEY STATUS
265 00000CBE 8A25[EE5E0000]
266 00000CC4 80E404
                                                                  ; SHIFT THEW SYSTEMKEY BIT OVER TO
267
<1> _K3:
<1> mc
                                         mov al, [KB_FLAG] ; GET THE SHIFT STATUS FL.; jmp short KIO_EXIT ; RETURN TO CALLER
                                                                  ; GET THE SHIFT STATUS FLAGS
277 00000CDC A0[ED5E0000]
278
                               <1>
                                         jmp _KIO_EXIT
279 00000CE1 EB83
                               <1>
280
                               <1>
                                         ;---- SET TYPAMATIC RATE AND DELAY
                               <1>
                               <1> _K300:
282
                                         cmp
283 00000CE3 3C05
                               <1>
                               <1>
284
                              <1> mov al, bl
<1> or al, bh
<1> call SND_DATA
<1> jmp _KIO_EXIT
296 00000D09 08F8
                                                                  ; AND DELAY
                                                                ; SEND TO KEYBOARD
297 00000D0B E8CE060000
                                         jmp _KIO_EXIT
                                                                        ; RETURN TO CALLER
298 00000D10 E951FFFFFF
299
                               <1>
300
                               <1>
                                         ;---- WRITE TO KEYBOARD BUFFER
                               <1> _K500:
                                     push esi
302 00000D15 56
                               <1>
                                                                  ; SAVE SI (esi)
                            <1> cli
<1> mov
<1> mov
<1> call
<1> cmp
<1> mov
<1> call
<1> cmp
<1> je
<1> mov
<1> je
<1> je
<1> mov
<1> je
<1 mov
<1 mov
<1 mov
<1 mov
<1 jmp
<1 jmp
<1 jmp
<1 jmp</pre>
303 00000D16 FA
                                               ebx, [BUFFER_TAIL] ; GET THE 'IN TO' POINTER TO THE BUFFER
304 00000D17 8B1D[FE5E0000]
306 00000D1F E8D3000000
                                               esi, ebx ; SAVE A COPY IN CASE BUFFER NOT FULL
                                               _K4
                                                                  ; BUMP THE POINTER TO SEE IF BUFFER IS FULL
307 00000D24 3B1D[FA5E0000]
                                                ebx, [BUFFER HEAD] ; WILL THE BUFFER OVERRUN IF WE STORE THIS?
                                               short _K502 ; YES - INFORM CALLER OF ERROR [esi], cx ; NO - PUT ASCII/SCAN CODE INTO BUFFER
308 00000D2A 740D
309 00000D2C 66890E
                                               [BUFFER_TAIL], ebx ; ADJUST 'IN TO' POINTER TO REFLECT CHANGE
310 00000D2F 891D[FE5E0000]
                                               al, al ; TELL CALLER THAT OPERATION WAS SUCCESSFUL short _K504 ; SUB INSTRUCTION ALSO RESETS CARRY FLAG
311 00000D35 28C0
312 00000D37 EB02
                                <1> _K502: mc
313
314 00000D39 B001
                                                                        ; BUFFER FULL INDICATION
                                               al, 01h
                                       mov
                                <1> _K504:
315
 316 00000D3B FB
                                <1>
                                         sti
                                                               ; RECOVER SI (esi)
 317 00000D3C 5E
                                         pop esi
                               <1>
318 00000D3D E924FFFFFF
                               <1>
                                          jmp _KIO_EXIT
                                                                        ; RETURN TO CALLER WITH STATUS IN AL
319
                                <1>
                                         ;---- READ THE KEY TO FIGURE OUT WHAT TO DO -----
320
                                <1>
                               <1> _K1S:
                                         cli ; 03/12/2014
322 00000D42 FA
                               <1>
                                         mov ebx, [BUFFER_HEAD] ; GET POINTER TO HEAD OF BUFFER cmp ebx, [BUFFER_TAIL] ; TEST END OF BUFFER ; jne short _KlU ; IF ANYTHING IN BUFFER SKIP INTERRUPT
323 00000D43 8B1D[FA5E0000]
                               <1>
324 00000D49 3B1D[FE5E0000]
                               <1>
325
                                <1>
                                         jne short _{k1x}; 03/12/2014
326 00000D4F 750F
                                <1>
                                <1>
327
328
                                <1>
                                         ; 03/12/2014
                                         ; 28/08/2014
329
                                <1>
                                         ; PERFORM OTHER FUNCTION ?? here !
330
                                <1>
331
                                <1>
                                         ;; MOV AX, 9002h ; MOVE IN WAIT CODE & TYPE
                                                                  ; PERFORM OTHER FUNCTION
                                <1>
                                         ;; INT 15H
332
                                <1> K1T:
333
                                                                          ; ASCII READ
                                                                  ; INTERRUPTS BACK ON DURING LOOP
                               <1>
334 00000D51 FB
                                         sti
335 00000D52 90
                               <1>
                                                                  ; ALLOW AN INTERRUPT TO OCCUR
336
                               <1> _K1U:
337 00000D53 FA
                               <1>
                                         cli
                                                                  ; INTERRUPTS BACK OFF
                                                   ebx, [BUFFER_HEAD] ; GET POINTER TO HEAD OF BUFFER
 338 00000D54 8B1D[FA5E0000]
                               <1>
339 00000D5A 3B1D[FE5E0000]
                               <1>
                                                  ebx, [BUFFER TAIL] ; TEST END OF BUFFER
                                           cmp
```

```
340
                                 <1> _k1x:
; LOOP UNTIL SOMETHING IN BUFFER
                                                      _K4 ; MOVE POINTER TO NEXT POSITION ; STORE VALUE IN VARIABLE
 358 00000D8C C3
                                                     ; RETURN
                                 <1>
                                            retn
359
                                 <1>
                                           ;---- READ THE KEY TO SEE IF ONE IS PRESENT ----
360
                                  <1>
 361
                                  <1> _K2S:
<1> _K2T:
                                 <1> pop
<1> popf
<1> sti
 375 00000DB7 6658
                                                                      ; RESTORE CODE
 376 00000DB9 9D
                                           popf
                                                                     ; RESTORE FLAGS
                                                                      ; INTERRUPTS BACK ON
 377 00000DBA FB
                                           sti
 378 00000DBB C3
                                 <1>
                                                                      ; RETURN
379
                                 <1>
                                        ;---- ROUTINE TO TRANSLATE SCAN CODE PAIRS FOR EXTENDED CALLS ----
 380
                                 <1>
                           381
 382 00000DBC 3CF0
                                                                      ; IS IT ONE OF THE FILL-INS?
 383 00000DBE 7506
 384 00000DC0 08E4
 385 00000DC2 7402
                                                                       ; PASS THIS ON UNCHANGED
 386 00000DC4 30C0
                                 <1> _KIO_E_RET:
 387
 388 00000DC6 C3
                                 <1>
                                                                      ; GO BACK

<1> ;---- ROUTINE TO TRANSLATE SCAN CODE PAIRS FOR STANDARD C;
<1> KIO_S_XLAT:
<1> cmp ah, 0E0h ; IS IT KEYPAD ENTER OR / ?
<1> jne short _KIO_S2 ; NO, CONTINUE
<1> cmp al, 0Dh ; KEYPAD ENTER CODE?
<1> je short _KIO_S1 ; YES, MASSAGE A BIT
<1> cmp al, 0Ah ; CTRL KEYPAD ENTER CODE?
<1> je short _KIO_S1 ; YES, MASSAGE THE SAME
<1> mov ah, 35h ; NO, MUST BE KEYPAD /
<1> kio_ret: ; 03/12/2014

389
                                 <1>
                                         ;---- ROUTINE TO TRANSLATE SCAN CODE PAIRS FOR STANDARD CALLS ----
 390
 391
 392 00000DC7 80FCE0
 393 00000DCA 750F
 394 00000DCC 3C0D
 395 00000DCE 7408
 396 00000DD0 3C0A
 397 00000DD2 7404
 398 00000DD4 B435
                                 <1> _kio_ret: ; 03/12/2014
 400 00000DD6 F8
                                 <1> clc
 401 00000DD7 C3
                                 <1>
                                           retn
                                 <1>
                                           ;jmp short _KIO_USE
                                                                           ; GIVE TO CALLER
 402
 403
                                 <1> _KIO_S1:
                                                  ah, 1Ch
                                       mov
                                                                            ; CONVERT TO COMPATIBLE OUTPUT
 404 00000DD8 B41C
                                 <1>
                                                  short _KIO_USE
 405
                                 <1>
                                           ;jmp
                                                                             ; GIVE TO CALLER
                                 <1>
 406 00000DDA C3
                                           retn
                                 <1> _KIO_S2:
 407

<1> cmp ah, 84h ; IS IT ONE OF EXTENDED ONES?
<1> ja short _KIO_DIS ; YES, THROW AWAY AND GET ANOTHER CHAR
<1> cmp al, 0F0h ; IS IT ONE OF THE FILL-INS?
<1> jne short _KIO_S3 ; NO, TRY LAST TEST
<1> or ah, ah ; AH = 0 IS SPECIAL CASE
<1> jz short _KIO_USE ; PASS THIS ON UNCHANGED
<1> jmp short _KIO_DIS ; THROW AWAY THE REST
<1> KIO S3:

                                 408 00000DDB 80FC84
 409 00000DDE 7715
 410 00000DE0 3CF0
 411 00000DE2 7506
 412 00000DE4 08E4
 413 00000DE6 740C
 414 00000DE8 EB0B
415
                                 <1> _KIO_S3:
                                       cmp al, 0E0h ; IS IT AN EXTENSION OF A PREVIOUS ONE?
; jne short KIO_USE ; NO, MUST BE A STANDARD CODE
                                 <1> cmp
 416 00000DEA 3CE0
                                 <1>
 417
                                418 00000DEC 75E8
                                                                     ; AH = 0 IS SPECIAL CASE
 419 00000DEE 08E4
 420 00000DF0 7402
                                                                     ; JUMP IF AH = 0
                                                                      ; CONVERT TO COMPATIBLE OUTPUT
 421 00000DF2 30C0
                                  <1>
                                                  al, al
                                           xor
                                                                           ; PASS IT ON TO CALLER
                                  <1>
                                           ;jmp short _KIO_USE
 423
                                  <1> KIO USE:
                                                                     ; CLEAR CARRY TO INDICATE GOOD CODE
                                  <1>
 424
                                           ;clc
 425 00000DF4 C3
                                  <1>
                                            retn
                                                                      ; RETURN
                                  <1> _KIO_DIS:
                                                                      ; SET CARRY TO INDICATE DISCARD CODE
 427 00000DF5 F9
                                 <1>
                                           stc
 428 00000DF6 C3
                                 <1>
                                                                      ; RETURN
429
                                  <1>
                                            ;---- INCREMENT BUFFER POINTER ROUTINE ----
430
                                  <1>
431
                                  <1> _K4:
432 00000DF7 43
                                 <1>
                                            inc
                                                    ebx
                                                                     ; MOVE TO NEXT WORD IN LIST
 433 00000DF8 43
                                 <1>
                                           cmp ebx, [BUFFER_END] ; AT END OF BUFFER?
434 00000DF9 3B1D[F65E0000]
                                 <1>
                                                    short _K5
                                  <1>
                                             ;jne
                                                                             ; NO, CONTINUE
 436 00000DFF 7206
                                  <1>
                                            jb short K5
                                            mov ebx, [BUFFER_START] ; YES, RESET TO BUFFER BEGINNING
 437 00000E01 8B1D[F25E0000]
                                  <1>
 438
                                  <1> _K5:
 439 00000E07 C3
                                  <1>
                                            retn
440
                                  <1>
 441
                                  <1> ; 20/02/2015
442
                                  <1>; 05/12/2014
                                  <1>; 26/08/2014
 443
 444
                                  <1>; KEYBOARD (HARDWARE) INTERRUPT - IRQ LEVEL 1
```

```
445
                         <1>; (INT_09h - Retro UNIX 8086 v1 - U9.ASM, 07/03/2014)
446
                         <1> ; Derived from "KB_INT_1" procedure of IBM "pc-at"
447
448
                         <1>; rombios source code (06/10/1985)
449
                         <1> ; 'keybd.asm', HARDWARE INT 09h - (IRQ Level 1)
450
                         <1>
                         <1>; EQUATES (IBM PC-XT-286 BIOS, 1986, 'POSQEQU.INC')
451
452
                         <1>
453
                         454
                         \1> DIS_KBD
<1> SHUT_CMD
<1>
455
456
                         <1> ;---- 8042 KEYBOARD INTERFACE AND DIAGNOSTIC CONTROL REGISTERS ------
457
                         <1> STATUS PORT equ 064h ; 8042 STATUS PORT
458
                         459
460
461
462
                         463
464
                         <1> ;----- KEYBOARD/LED COMMANDS ------
465
                         <1> KB_ENABLE equ 0F4h ; KEYBOARD ENABLE
<1> LED_CMD equ 0EDh ; LED WRITE COMMAND
466
467
                         <1> KB_TYPA_RD equ 0F3h ; TYPAMATIC RATE/DELAY COMMAND
468
                         <1> ;----- KEYBOARD SCAN CODES -----
469
                         470
471
                         472
                         473
474
475
476
                                           42 ; SCAN CODE FOR LEFT SHIFT
54 ; SCAN CODE FOR RIGHT SHIFT
equ 84 ; SCAN CODE FOR SYSTEM KEY
477
                         <1> RIGHT_KEY equ
                                          54
478
479
                         <1> SYS KEY
                         <1> ;----- ENHANCED KEYBOARD SCAN CODES ------
480
                         481
                                  equ 041h ; 2ND ID CHARACTER FOR KBX
equ 054h ; ALTERNATE 2ND ID CHARACTER FOR KBX
equ 87 ; F11 KEY MAKE
equ 88 ; F12 KEY MAKE
equ 224 ; GENERAL MARKER CODE
equ 225 ; PAUSE KEY MARKER CODE
482
483
484
485
                         <1> F12_M
                         <1> MC \overline{E}0
486
                         <1> MC_E1
487
488
                         <1> ;----- FLAG EQUATES WITHIN @KB FLAG-----
                                          00000001b ; RIGHT SHIFT KEY DEPRESSED 00000010b ; LEFT SHIFT KEY DEPRESSED
489
                         <1> RIGHT_SHIFT equ
                         <1> LEFT_SHIFT equ 00000010b ; LEFT SHIFT KEY DEPRESSED
<1> CTL_SHIFT equ 00000100b ; CONTROL SHIFT KEY DEPRESSED
490
491
                         492
493
494
                         <1> CAPS_STATE equ 01000000b ; CAPS_LOCK_STATE IS ACTIVE
495
496
                         <1> INS STATE equ
                                           10000000b
                                                     ; INSERT STATE IS ACTIVE
                         497
                         <1> L_CTL_SHIFT equ 00000001b ; LEFT CTL KEY DOWN
498
                                          00000010b ; LEFT ALT KEY DOWN
00000100b ; SYSTEM KEY DEPRESSED AND HELD
00001000b ; SUSPEND KEY HAS BEEN TOGGLED
499
                         <1> L ALT SHIFT equ
                         <1> SYS_SHIFT equ
500
                         <1> HOLD STATE equ
501
                         <1> SCROLL_SHIFT equ
                                           00010000b ; SCROLL LOCK KEY IS DEPRESSED
502
                         503
504
505
                         <1> ;----- FLAGS EQUATES WITHIN @KB FLAG 2 -----
506
                         <1> KB_LEDS
                                     equ 00000111b ; KEYBOARD LED STATE BITS
507
                         <1>;
508
                                      equ 00000001b ; SCROLL LOCK INDICATOR
                                      equ
                                           0000010b ; NUM LOCK INDICATOR
509
                         <1>;
                         ; CAPS LOCK INDICATOR ; RESERVED (MUST BE ZERO)
510
                                      equ
                                           00000100b
                                      equ 00001000b
511
                                                    ; ACKNOWLEDGMENT RECEIVED
                         <1> KB_FA equ 00010000b
<1> KB_FE equ 00100000b
512
513
                                                    ; RESEND RECEIVED FLAG
; MODE INDICATOR UPDATE
                                      equ
                         <1> KB PR_LED equ
514
                                          01000000b
                                           equ 10000000b ; KEYBOARD TRANSMIT ERROR FLAG
515
                         <1> KB_ERR
                         516
                         517
518
                         ; RIGHT CTL KEY DOWN ; RIGHT ALT KEY DOWN
519
520
                         <1> GRAPH ON equ
                                           00001000b ; ALT GRAPHICS KEY DOWN (WT ONLY)
521
                                                    ; ENHANCED KEYBOARD INSTALLED
                         <1> KBX
                                           00010000b
522
                                      equ
                                                    ; FORCE NUM LOCK IF READ ID AND KBX ; LAST CHARACTER WAS FIRST ID CHARACTER
                          <1> SET NUM LK equ
                                           00100000b
523
                         <1> LC_AB equ
<1> RD ID equ
524
                                           01000000b
525
                         <1> RD_ID
                                      equ
                                          10000000b
                                                    ; DOING A READ ID (MUST BE BIT0)
                          <1>;
                         <1> ;----- INTERRUPT EQUATES ------
                         <1> EOI equ 020h
                                                 ; END OF INTERRUPT COMMAND TO 8259
528
                                          equ 020h
529
                         <1> INTA00
                                                        ; 8259 PORT
530
                         <1>
531
                         <1>
                         <1> kb_int:
532
533
                         <1>
                         <1>; 17/10/2015 ('ctrlbrk')
534
                         <1> ; 05/12/2014
535
536
                         <1>; 04/12/2014 (derived from pc-xt-286 bios source code -1986-)
537
                         <1> ; 26/08/2014
538
                         <1>;
539
                         <1> ; 03/06/86 KEYBOARD BIOS
540
                         <1>;
                         <1> ;--- HARDWARE INT 09H -- (IRQ LEVEL 1) -------
541
542
                         <1>;
543
                         <1>;
                                 KEYBOARD INTERRUPT ROUTINE
                                                                               ;
544
                         <1>;
                                                                               ;
                         <1> ;-----
545
546
                         <1>
                         <1> KB_INT_1:
547
548 00000E08 FB
                                                     ; ENABLE INTERRUPTS
                         <1>
                                 ;push ebp
549
                         <1>
```

```
550 00000E09 50
                            <1>
                                    push eax
551 00000E0A 53
                            <1>
                                    push
                                          ebx
552 00000E0B 51
                            <1>
                                    push
                                          ecx
553 00000E0C 52
                           <1>
                                    push
554 00000E0D 56
                            <1>
                                    push
                                          esi
555 00000E0E 57
                            <1>
                                    push
                                          edi
                                    push ds
556 00000E0F 1E
                            <1>
557 00000E10 06
                                   push es
                            <1>
558 00000E11 FC
                            <1>
                                                           ; FORWARD DIRECTION
                                    cld
                                  mov
559 00000E12 66B81000
                           <1>
                                          ax, KDATA
560 00000E16 8ED8
                           <1>
                                  mov
                                          ds, ax
561 00000E18 8EC0
                            <1>
                                    mov
                                          es, ax
562
                           <1>
562
563
564 00000E1A B0AD
565 00000E1C E8A9050000
                           <1>
                                    ;---- WAIT FOR KEYBOARD DISABLE COMMAND TO BE ACCEPTED
                                         al, DIS_KBD ; DISABLE THE KEYBOARD COMMAND
                           <1>
                                    mov
                                          SHIP_IT
                                                         ; EXECUTE DISABLE ; DISABLE INTERRUPTS
                           <1>
                                    call
566 00000E21 FA
                           <1>
                                          ecx, 10000h ; SET MAXIMUM TIMEOUT
567 00000E22 B900000100
                           <1>
                                    mov
568
                            <1> KB_INT_01:
                           <1> in al, STATUS_PORT
569 00000E27 E464
                                                                 ; READ ADAPTER STATUS
                                     test al, INPT_BUF_FULL ; CHECK INPUT BUFFER FULL STATUS BIT
570 00000E29 A802
                           <1>
                                                      ; WAIT FOR COMMAND TO BE ACCEPTED
571 00000E2B E0FA
                            <1>
                                     loopnz KB_INT_01
572
                            <1>
573
                            <1>
                                    ;---- READ CHARACTER FROM KEYBOARD INTERFACE
574 00000E2D E460
                            <1>
                                        al, PORT_A ; READ IN THE CHARACTER
                                    in
575
                            <1>
                            <1>
                                    ;---- SYSTEM HOOK INT 15H - FUNCTION 4FH (ON HARDWARE INT LEVEL 9H)
                                    ;MOV AH, 04FH
                                                     ; SYSTEM INTERCEPT - KEY CODE FUNCTION
577
                            <1>
578
                            <1>
                                    ;STC
                                                           ; SET CY=1 (IN CASE OF IRET)
                                    ;INT 15H
579
                            <1>
                                                           ; CASETTE CALL (AL) = KEY SCAN CODE
580
                            <1>
                                                          ; RETURNS CY=1 FOR INVALID FUNCTION
                                                        ; CONTINUE IF CARRY FLAG SET ((AL)=CODE)
                                         KB INT 02
581
                            <1>
                                    ;JC
                                     ; JMP K2\overline{6}
                                                           ; EXIT IF SYSTEM HANDLES SCAN CODE
                            <1>
582
                            <1>
                                                           ; EXIT HANDLES HARDWARE EOI AND ENABLE
583
584
                            <1>
585
                            <1>
                                     ;---- CHECK FOR A RESEND COMMAND TO KEYBOARD
                            sti ; ENABLE INTERRUPTS AGAIN
cmp al, KB_RESEND ; IS THE INPUT A RESEND
je short KB_INT_4 ; GO IF RESEND
;
587 00000E2F FB
                            <1>
588 00000E30 3CFE
                            <1>
589 00000E32 7411
                           <1>
590
                           <1>
                                  ;---- CHECK FOR RESPONSE TO A COMMAND TO KEYBOARD
591
                            <1>
                                     cmp al, KB_ACK ; IS THE INPUT AN ACKNOWLEDGE
592 00000E34 3CFA
                           <1>
                                    jne short KB_INT_2 ; GO IF NOT
593 00000E36 751A
                           <1>
594
                            <1>
                                    ;
                                     ;---- A COMMAND TO THE KEYBOARD WAS ISSUED
595
                            <1>
                                    cli ; DISABLE INTERROFFO
or byte [KB_FLAG_2], KB_FA ; INDICATE ACK RECEIVED
: RETURN IF NOT (ACK
596 00000E38 FA
                            <1>
597 00000E39 800D[EF5E0000]10
                           <1>
598 00000E40 E97A020000
                                                       ; RETURN IF NOT (ACK RETURNED FOR DATA)
                            <1>
                                    jmp K26
599
                            <1>
600
                            <1>
                                    ;---- RESEND THE LAST BYTE
601
                            <1> KB INT 4:
                            602 00000E45 FA
                                                           ; DISABLE INTERRUPTS
                                     or byte [KB_FLAG_2], KB_FE; INDICATE RESEND RECEIVED
603 00000E46 800D[EF5E0000]20
                           <1>
                            <1> jmp K26 <1> ;
604 00000E4D E96D020000
                                                                 ; RETURN IF NOT ACK RETURNED FOR DATA)
605
                            <1> ;----
606
                                          UPDATE MODE INDICATORS IF CHANGE IN STATE
607
                            <1> KB_INT_2:
; SAVE DATA IN
                                    call MAKE_LED
                                                           ; GO GET MODE INDICATOR DATA BYTE
614 00000E66 E8C0050000
                                    call SND_LED
                                                                ; GO TURN ON MODE INDICATORS
                            <1>
                            <1> UP0:
615
616 00000E6B 6658
                            <1>
                                                           ; RESTORE DATA IN
                                    pop ax
                            <1> ;-----
617
                            <1>; START OF KEY PROCESSING
618
                            <1> ;---
619
620 00000E6D 88C4
                           <1>
                                                         ; SAVE SCAN CODE IN AH ALSO
621
                            <1>
                                     ;---- TEST FOR OVERRUN SCAN CODE FROM KEYBOARD
622
                            <1>
623 00000E6F 3CFF
                                    <1>
                                    je K62
624 00000E71 0F843F050000
                            <1>
625
                            <1>
                            <1> K16:
627 00000E77 8A3D[F05E0000]
                                    mov bh, [KB FLAG 3]
                            <1>
                                                                 ; LOAD FLAGS FOR TESTING
628
                            <1>
629
                                     ;---- TEST TO SEE IF A READ ID IS IN PROGRESS
                            <1>
630 00000E7D F6C7C0
                                     test bh, RD_ID+LC_AB ; ARE WE DOING A READ ID?
                            <1>
                                                      ; CONTINUE IF NOT
631 00000E80 7449
                            <1>
                                          short NOT ID
                                                           ; IS THE RD_ID FLAG ON?
632 00000E82 7917
                                          short TST ID 2
                            <1>
                                     jns
633 00000E84 3CAB
                            <1>
                                          al, ID 1
                                                           ; IS THIS THE 1ST ID CHARACTER?
634 00000E86 7507
                            <1>
                                          short RST RD ID
                                     jne
635 00000E88 800D[F05E0000]40
                                          byte [KB_FLAG_3], LC_AB ; INDICATE 1ST ID WAS OK
                            <1>
                                     or
                            <1> RST RD ID:
637 00000E8F 8025[F05E0000]7F
                                     and byte [KB_FLAG_3], ~RD_ID; RESET THE READ ID FLAG
                           <1>
                                                          ; AND EXIT
638
                            <1>
                                     ;jmp short ID_EX
                                     jmp K26
639 00000E96 E924020000
                            <1>
640
                            <1>
                            <1> TST ID 2:
641
                           <1>
                                    and byte [KB_FLAG_3], ~LC_AB ; RESET FLAG
642 00000E9B 8025[F05E0000]BF
                                                      ; IS THIS THE 2ND ID CHARACTER?
; JUMP IF SO
                                     cmp al, ID_2A
643 00000EA2 3C54
                           <1>
                                 je short KX_BIT
644 00000EA4 7419
                            <1>
                                   645 00000EA6 3C41
                           <1>
                                ;jne shor
jne K26
                           <1>
647 00000EA8 0F8511020000
                           <1>
648
                            <1>
                                  ;
;---- A READ ID SAID THAT IT WAS ENHANCED KEYBOARD
649
                            <1>
                                 test bh, SET_NUM_LK ; SHOULD WE SET NUM LOCK?
jz short KX_BIT ; EXIT IF NOT
650 00000EAE F6C720
                           <1>
651 00000EB1 740C
                            <1>
or byte [KB_FLAG], NUM_STATE; FORCE NUM LOCK ON
                                                      ; GO SET THE NUM LOCK INDICATOR
                            <1> KX BIT:
654
```

```
<1> or
                                                 byte [KB_FLAG_3], KBX
655 00000EBF 800D[F05E0000]10
                                                                        ; INDICATE ENHANCED KEYBOARD WAS FOUND
                                                 \frac{1}{\text{jmp}} \frac{1}{\text{K26}}
656 00000EC6 E9F4010000
                                <1> ID EX:
                                                                           ; EXIT
657
                                <1>;
                                                al, MC_EO ; IS THIS THE GENERAL MARKER CODE? short TEST_E1
658
                                <1> NOT ID:
659 00000ECB 3CE0
                                <1> cmp
                                                al, MC E0
660 00000ECD 750C
                                <1>
                                          jne
661 00000ECF 800D[F05E0000]12 <1> or
                                                byte [KB_FLAG_3], LC_E0+KBX; SET FLAG BIT, SET KBX, AND
                               <1> ;jmp
<1> jmp
                                               short EXIT ; THROW AWAY THIS CODE
663 00000ED6 E9EB010000
                                                 K26A
                                <1> TEST E1:
                                                 aı, MC_E1 ; IS THIS THE PAUSE KEY? short NOT_HC
664
665 00000EDB 3CE1
                                <1> cmp
666 00000EDD 750C
                                <1>
                                          jne
667 00000EDF 800D[F05E0000]11 <1>
                                                 byte [KB_FLAG_3], LC_E1+KBX; SET FLAG BIT, SET KBX, AND
                                          or
668 00000EE6 E9DB010000
                                                          ; THROW AWAY THIS CODE
                                <1> EXIT: jmp
                                <1> ;
669
                                <1> NOT_HC:
670
671 00000EEB 247F
                                                                   ; TURN OFF THE BREAK BIT
                               <1> and al, 07Fh
                                          test bh, LC_E0
                                                bh, LC_EO ; LAST CODE THE EO MARKER CODE short NOT_LC_EO ; JUMP IF NOT
672 00000EED F6C702
                               <1>
673 00000EF0 7414
                                <1>
                                          jz
                                <1>
                                          ;
675 00000EF2 BF[DA5D0000]
                                                                 ; IS THIS A SHIFT KEY?
                               <1>
                                        mov edi, _K6+6
                                     scasb
676 00000EF7 AE
                                <1>
677 00000EFF AE
                               <1>
                                                   K26 ; K16B
                                                                           ; YES, THROW AWAY & RESET FLAG
                                          jе
                               <1>
<1>
<1>
678 00000EFE AE
                                          scasb
                                                                    ; NO, CONTINUE KEY PROCESSING
                                          jne short K16A
679 00000EFF 757C
680
                                <1>
                                          ;jmp
                                                short K16B
                                                                     ; YES, THROW AWAY & RESET FLAG
681 00000F01 E9B9010000
                               <1>
                                                K26
                                          jmp
682
                                <1>
                                <1> NOT_LC_E0:
683
                                          test bh, LC_E1 ; LAST CODE THE E1 MARKER CODE?
jz short T_SYS_KEY ; JUMP IF NOT
684 00000F06 F6C701
                               <1>
685 00000F09 7435
                               <1>
686 00000F0B B90400000
687 00000F10 BF[D85D0000]
686 00000F0B B904000000
                                          mov ecx, 4 ; LENGHT OF SEARCH
mov edi, K6+4 ; IS THIS AN ALT, C
repne scasb ; CHECK IT
; je short EXIT ; THROW AWAY IF SO
                                <1>
                                                                    ; IS THIS AN ALT, CTL, OR SHIFT?
                                <1>
                                     repne scasb
688 00000F15 F2AE
                                <1>
689
                                <1>
690 00000F17 0F84A9010000
                                <1>
                                                 K26A
                                          jе
                                <1>
                                                al, NUM_KEY ; IS IT THE PAUSE KEY?
692 00000F1D 3C45
                                <1>
                                          cmp
                                     ., IRROW AWAY & RESET FL.

cest ah, 80h ; YES, IS IT THE BREA
; jnz short K16B ; YES, THROW THIS AWAY, TOO
jnz K26
; 20/02/2015
test byte [KB FLAC 17
                                                                    ; NO, THROW AWAY & RESET FLAG
                                <1>
693
694 00000F1F 0F859A010000
                                <1>
                                                                           ; YES, IS IT THE BREAK OF THE KEY?
695 00000F25 F6C480
                                <1>
                                <1>
697 00000F28 0F8591010000
                                <1>
                                <1>
698
                                          test byte [KB_FLAG_1], HOLD_STATE; NO, ARE WE PAUSED ALREADY?; jnz short K16B; YES, THROW AWAY
699 00000F2E F605[EE5E0000]08
                                <1>
700
                                <1>
701 00000F35 0F8584010000
                                <1>
                                          jnz
702 00000F3B E9E1020000
                                                                         ; NO, THIS IS THE REAL PAUSE STATE
                                <1>
                                                 K39P
                                          jmp
703
                                <1>
704
                                          ;---- TEST FOR SYSTEM KEY
                                <1>
705
                                <1> T_SYS_KEY:
                                                al, SYS_KEY ; IS IT THE SYSTEM KEY? short K16A ; CONTINUE IF NOT
706 00000F40 3C54
                                <1>
                                          cmp
707 00000F42 7539
                                <1>
                                          jnz
708
                                <1>
709 00000F44 F6C480
                                <1>
                                                                           ; CHECK IF THIS A BREAK CODE
                                          test ah, 80h
                                                short K16C ; DO NOT TOUCH SYSTEM INDICATOR IF TRUE
710 00000F47 7524
                                <1>
                                          jnz
                                <1>
                                          test byte [KB_FLAG_1], SYS_SHIFT; SEE IF IN SYSTEM KEY HELD DOWN
712 00000F49 F605[EE5E0000]04
                                <1>
                                          ; jnz short K16B ; IF YES, DO NOT PROCESS SYSTEM INDICATOR
                                <1>
713
714 00000F50 0F8569010000
                                <1>
                                          jnz K26
715
                                <1>
716 00000F56 800D[EE5E0000]04
                                <1>
                                          or
                                                byte [KB_FLAG_1], SYS_SHIFT; INDICATE SYSTEM KEY DEPRESSED
                                                al, EOI ; END OF INTERRUPT COMMAND
20h, al ;out INTA00, al ; SEND COMMAND TO INTERRUPT CONTROL PORT
717 00000F5D B020
                                <1>
                                          mov
718 00000F5F E620
                                <1>
                                                               ; INTERRUPT-RETURN-NO-EOI
                                <1>
719
                                                                  ; INSURE KEYBOARD IS ENABLED
720 00000F61 B0AE
                                <1>
                                                 al, ENA_KBD
721 00000F63 E862040000
                                <1>
                                          call SHIP IT
                                                                       ; EXECUTE ENABLE
                                          ; !!! SYSREQ !!! function/system call (INTERRUPT) must be here !!!
                                <1>
722
                                                            ; FUNCTION VALUE FOR MAKE OF SYSTEM KEY
723
                                <1>
                                          ;MOV AL, 8500H
724
                                <1>
                                                                    ; MAKE SURE INTERRUPTS ENABLED
                                          ;STI
                                                                   ; USER INTERRUPT
725
                                <1>
                                          ;INT 15H
                                                                          ; END PROCESSING
726 00000F68 E965010000
                                <1>
                                          jmp K27A
                                <1>
727
                                <1> ;K16B:
                                                 jmp K26
                                                                          ; IGNORE SYSTEM KEY
728
729
                                <1>
730
                                <1> K16C:
731 00000F6D 8025[EE5E0000]FB
                                <1> and
                                                byte [KB_FLAG_1], ~SYS_SHIFT; TURN OFF SHIFT KEY HELD DOWN
                                                                          ; END OF INTERRUPT COMMAND
732 00000F74 B020
                                <1>
                                          mov
                                                al, EOI
733 00000F76 E620
                                <1>
                                                20h, al ;out INTA00, al ; SEND COMMAND TO INTERRUPT CONTROL PORT
                                                           ; INTERRUPT-RETURN-NO-EOI
734
                                <1>
                                                               ; INSURE KEYBOARD IS ENABLED
735
                                 <1>
                                          ; MOV AL, ENA_KBD
                                        ;CALL SHIP_IT
                                                                           ; EXECUTE ENABLE
736
                                 <1>
                                 <1>
738
                                 <1>
                                          ;MOV AX, 8501H
                                                                  ; FUNCTION VALUE FOR BREAK OF SYSTEM KEY
                                                                  ; MAKE SURE INTERRUPTS ENABLED
739
                                 <1>
                                          ;STI
                                                15H
740
                                 <1>
                                          ;INT
                                                                    ; USER INTERRUPT
                                                                   ; INGONRE SYSTEM KEY
741
                                <1>
                                          ;JMP
                                               K27A
                                <1>
742
                                          ;
743 00000F78 E94E010000
                                <1>
                                                 K27
                                                                    ; IGNORE SYSTEM KEY
                                          jmp
744
                                <1>
                                          ;---- TEST FOR SHIFT KEYS
745
                                <1>
746
                                <1> K16A:
                                                                 ; PUT STATE FLAGS IN BL
; SHIFT KEY TABLE offset
                                                bl, [KB_FLAG]
747 00000F7D 8A1D[ED5E0000]
                                <1>
                                          mov
748 00000F83 BF[D45D0000]
                                          mov edi, K_6
                               <1>
                                                                   ; LENGTH
749 00000F88 B908000000
                                          mov ecx, _K6L
                                <1>
                                          repne scasb
750 00000F8D F2AE
                                <1>
                                                                    ; LOOK THROUGH THE TABLE FOR A MATCH
                                                                   ; RECOVER SCAN CODE
; IF NO MATCH, THEN SHIFT NOT FOUND
751 00000F8F 88E0
                                <1>
                                          mov al, ah
752 00000F91 0F8510010000
                                <1>
                                          jne K25
753
                                <1>
                                          ;---- SHIFT KEY FOUND
                                <1>
754
755
                                <1> K17:
                                          sub edi, _K6+1 ; ADJUST PTR TO SCAN CODE MATCH
mov ah, [edi+_K7] ; GET MASK INTO AH
mov cl, 2 ; SETUP COUNT FOR FLAG SHIFTS
test al, 80h ; TEST FOR BREAK KEY
756 00000F97 81EF[D55D0000]
                                <1>
757 00000F9D 8AA7[DC5D0000]
                               <1>
758 00000FA3 B102
                                <1>
759 00000FA5 A880
                                <1>
```

```
760 00000FA7 0F8596000000
                                      <1>
                                                            jnz K23
                                                                                                          ; JUMP OF BREAK
                                              <1>
                                             <1>
762
                                                            ;---- SHIFT MAKE FOUND, DETERMINE SET OR TOGGLE
763
                                             <1> K17C:
764 00000FAD 80FC10
                                             <1>
                                                            cmp ah, SCROLL_SHIFT
jae short K18
765 00000FB0 732B
                                             <1>
                                                                                                ; IF SCROLL SHIFT OR ABOVE, TOGGLE KEY
<1>
771 00000FBA 0F84FF000000
772
773 00000FC0 F6C702
774 00000FC3 740B
                                             <1> K17D:
                                                           test bh, LC_EO ; IS THIS ONE OF NEW KEYS?
jz short K17E ; NO, JUMP
or [KB_FLAG_3], ah ; SET BITS FOR RIGHT CTRL, ALT
                                             <1>
; INTERRUPT RETURN
                                                            jmp
                                             <1> K17E:
                                                                    ; MOVE FLAG BITS TWO POSITIONS [KB_FLAG_1], ah : SFT DIMO TO THE ROOT OF THE R
777
778 00000FD0 D2EC
                                              <1>
                                                           shr
                                                                                                  ; SET BITS FOR LEFT CTRL, ALT
779 00000FD2 0825[EE5E0000]
                                             <1>
                                                            or
                                              <1>
                                                            jmp K26
780 00000FD8 E9E2000000
                                              <1>
782
                                                            ;---- TOGGLED SHIFT KEY, TEST FOR 1ST MAKE OR NOT
                                              <1>
                                                            test bl, CTL_SHIFT;
783
                                              <1> K18:
                                                            test bl, CTL_SHIFT ; CHECK CTL SHIFT STATE
; jz short K18A ; JUMP IF NOT CTL STATE
jnz K25 ; JUMP IF CTL STATE
784 00000FDD F6C304
                                              <1>
785
                                              <1>
786 00000FE0 0F85C1000000
                                             <1>
                                             <1> K18A:
787
                                             cmp al, INS_KEY ; CHECK FOR INSERT KEY
ine short K22 ; JUMP IF NOT INSERT KEY
test bl, ALT_SHIFT ; CHECK FOR ALTERNATE SHIFT

ipz short K18B ; JUMP IF NOT ALTERNATE SHIFT

ipz K25 ; JUMP IF ALTERNATE SHIFT
788 00000FE6 3C52
789 00000FE8 7524
                                             <1>
790 00000FEA F6C308
                                             <1>
                                             , Jz shor
jnz K25
<1> K18B:
792 00000FED 0F85B4000000
793
                                                            test bh, LC_E0;20/02/2015; IS THIS NEW INSERT KEY?
794 00000FF3 F6C702
                                             <1>
795 00000FF6 7516
                                             <1>
                                                                    short K22 ; YES, THIS ONE'S NEVER A '0'
                                             <1> K19:
                                                            test bl, NUM_STATE ; CHECK FOR BASE S jnz short K21 ; JUMP IF NUM LOCK IS ON
797 00000FF8 F6C320
                                                                                                          ; CHECK FOR BASE STATE
                                             <1>
798 00000FFB 750C
                                             <1>
                                            <1>
<1>
                                                            test bl, LEFT SHIFT+RIGHT SHIFT; TEST FOR SHIFT STATE
799 00000FFD F6C303
800 00001000 740C
                                                            jz short K22 ; JUMP IF BASE STATE
                                             <1> K20:
                                                                                                ; NUMERIC ZERO, NOT INSERT KEY
                                                                                               ; PUT SCAN CODE BACK IN AH
                                                            mov ah, al
                                             <1>
802 00001002 88C4
                                                            jmp K25
803 00001004 E99E000000
                                             <1>
                                                                                                ; NUMERAL '0', STNDRD. PROCESSING
                                                                                              ; MIGHT BE NUMERIC
                                             <1> K21:
804
805 00001009 F6C303
                                                            test bl, LEFT_SHIFT+RIGHT_SHIFT
                                             <1>
806 0000100C 74F4
                                             <1>
                                                            jz short K20 ; IS NUMERIC, STD. PROC.
                                             <1>
807
                                                            ; SHIFT TOGGLE KEY HIT; PROCESS IT
test ah, [KB_FLAG_1]; IS KEY ALREADY DEPRESSED
; JUMP IF KEY ALREADY DEPRESSED
808
                                              <1> K22:
809 0000100E 8425[EE5E0000]
                                             <1>
810 00001014 0F85A5000000
                                             <1>
811
                                              <1> K22A:
                                                            or [KB_FLAG_1], ah ; INDICATE THAT THE KEY IS DEPRESSED
812 0000101A 0825[EE5E0000]
                                             <1>
                                                                                              ; TOGGLE THE SHIFT STATE
813 00001020 3025[ED5E0000]
                                             <1>
                                                            xor [KB_FLAG], ah
                                              <1>
814
815
                                              <1>
                                                            ;---- TOGGLE LED IF CAPS, NUM OR SCROLL KEY DEPRESSED
                                                            test ah, CAPS SHIFT+NUM SHIFT+SCROLL SHIFT; SHIFT TOGGLE?
816 00001026 F6C470
                                             <1>
817 00001029 7409
                                             <1>
                                                                    short K22B ; GO IF NOT
                                                            jz
                                              <1>
818
                                                                                            ; SAVE SCAN CODE AND SHIFT MASK
                                                            push ax
819 0000102B 6650
                                             <1>
820 0000102D E8F9030000
                                             <1>
                                                            call SND_LED
                                                                                                 ; GO TURN MODE INDICATORS ON
821 00001032 6658
                                              <1>
                                                                                                ; RESTORE SCAN CODE
                                                            pop ax
                                             <1> K22B:
822
                                                           cmp al, INS_KEY
jne K26
mov ah, al
jmp K28
                                             <1>
                                                                                                ; TEST FOR 1ST MAKE OF INSERT KEY
823 00001034 3C52
                                                                                                ; JUMP IF NOT INSERT KEY
824 00001036 0F8583000000
                                             <1>
825 0000103C 88C4
                                             <1>
                                                                                                  ; SCAN CODE IN BOTH HALVES OF AX
826 0000103E E999000000
                                                                                                       ; FLAGS UPDATED, PROC. FOR BUFFER
                                             <1>
827
                                             <1>
                                                            ;---- BREAK SHIFT FOUND
828
                                              <1>
                                              <1> K23:
                                                                                                 ; BREAK-SHIFT-FOUND
829
                                                                    ah, SCROLL SHIFT ; IS THIS A TOGGLE KEY
830 00001043 80FC10
                                             <1>
                                                                    ah ; INVERT MASK short K24 ; YES, HANDLE
831 00001046 F6D4
                                              <1>
                                                            not
                                                                    short K24 ; YES, HANDLE LINE [KB_FLAG], ah ; TURN OFF SHIFT BIT ; IS THIS ALT OR CTL?
832 00001048 7355
                                             <1>
                                                            jae
833 0000104A 2025[ED5E0000]
                                             <1>
                                                            and
834 00001050 80FCFB
                                              <1>
                                                            cmp
                                                                                              ; NO, ALL DONE
835 00001053 7730
                                              <1>
                                                                     short K23D
                                                            jа
                                              <1>
                                                                                      ; 2ND ALT OR CTL?
837 00001055 F6C702
                                                           test bh, LC_E0
                                              <1>
                                                                                              ; NO, HANSLE NORMALLY
                                                                     short K23A
838 00001058 7408
                                              <1>
                                                            jz
                                                                    [KB FLAG 3], ah
839 0000105A 2025[F05E0000]
                                                                                                 ; RESET BIT FOR RIGHT ALT OR CTL
                                             <1>
                                                            and
                                                                                               ; CONTINUE
840 00001060 EB08
                                             <1>
                                                            jmp short K23B
                                              <1> K23A:
                                                                     ah, cl
842 00001062 D2FC
                                                                                                 ; MOVE THE MASK BIT TWO POSITIONS
                                             <1>
                                                            sar
843 00001064 2025[EE5E0000]
                                                                    [KB FLAG 1], ah
                                              <1>
                                                                                                 ; RESET BIT FOR LEFT ALT AND CTL
844
                                              <1> K23B:
845 0000106A 88C4
                                            al, [KB_FLAG_3]; GET RIGHT ALT & CTRL F.

<1> shr al, cl ; MOVE TO BITS 1 & 0

<1> or al, [KB_FLAG_1]; PUT IN LEFT ALST & CTL

<1> shl al, cl ; MOVE BACK TO BITS 3 & 2

<1> and al, ALT_SHIFT+CTL_SHIFT; FILTER OUT OTHER GARBAGE

<1> or [KB_FLAG], al ; PUT RESULT IN THE PEAT BY COLOR

<1> mov al ab
                                             <1>
                                                           mov
                                                                     ah, al
                                                                                                ; SAVE SCAN CODE
846 0000106C A0[F05E0000]
                                                                                                  ; GET RIGHT ALT & CTRL FLAGS
847 00001071 D2E8
                                                                                               ; PUT IN LEFT ALŞT & CTL FLAGS ; MOVE BACK TO BITS 3 & 2
848 00001073 0A05[EE5E0000]
849 00001079 D2E0
850 0000107B 240C
851 0000107D 0805[ED5E0000]
852 00001083 88E0
                                                           mov
                                                                    al, ah
                                             <1> K23D:
854 00001085 3CB8
                                             <1>
                                                                    al, ALT KEY+80h
                                                                                                          ; IS THIS ALTERNATE SHIFT RELEASE
                                                        cmp
855 00001087 7536
                                                                    short K26
                                             <1>
                                                            jne
                                                                                                ; INTERRUPT RETURN
                                             <1>
                                             <1>
                                                           ;---- ALTERNATE SHIFT KEY RELEASED, GET THE VALUE INTO BUFFER
857
857
858 00001089 A0[F15E0000]
                                                           mov al, [ALT_INPUT]
                                             <1>
                                                                                                 ; SCAN CODE OF 0
859 0000108E B400
                                             <1>
                                                                    ah, 0
                                                            mov
860 00001090 8825[F15E0000]
                                            <1>
                                                            mov [ALT INPUT], ah
                                                                                              ; ZERO OUT THE FIELD
                                                           cmp al, 0 ; WAS THE INPUT = 0?
je short K26 ; INTERRUPT_RETURN
861 00001096 3C00
                                              <1>
862 00001098 7425
                                             <1>
                                                            ; 29/01/2016
863
                                              <1>
                                                            ;jmp K61
864
                                              <1>
                                                                                                       ; IT WASN'T, SO PUT IN BUFFER
```

```
865 0000109A E9D0020000
                                              <1>
                                                                     _K60
                                                             jmp
                                               <1>
867
                                              <1> K24:
                                                                                                 ; BREAK-TOGGLE
868 0000109F 2025[EE5E0000]
                                              <1>
                                                             and
                                                                     [KB_FLAG_1], ah ; INDICATE NO LONGER DEPRESSED
869 000010A5 EB18
                                               <1>
                                                                     short K26 ; INTERRUPT_RETURN
                                                             jmp
870
                                               <1>
                                                             ;---- TEST FOR HOLD STATE
871
                                               <1>
                                                                                               ; AL, AH = SCAN CODE
872
                                               <1>
873
                                               <1> K25:
                                                                                                 ; NO-SHIFT-FOUND
                                                                                                    ; TEST FOR BREAK KEY
874 000010A7 3C80
                                               <1>
                                                                     al, 80h
                                                            cmp
                                                             jae short K26
                                                                                                ; NOTHING FOR BREAK CHARS FROM HERE ON
875 000010A9 7314
                                               <1>
876 000010AB F605[EE5E0000]08
                                                            test byte [KB_FLAG_1], HOLD_STATE ; ARE WE IN HOLD STATE
                                               <1>
                                                                                             ; BRANCH AROUND TEST IF NOT
877 000010B2 7428
                                                                     short K28
                                              <1>
                                                            jz
878 000010B4 3C45
                                               <1>
                                                                     al, NUM KEY
879 000010B6 7407
                                               <1>
                                                                                                 ; CAN'T END HOLD ON NUM LOCK
                                                                     short K26
                                                            jе
                                                                     byte [KB_FLAG_1], ~HOLD_STATE ; TURN OFF THE HOLD STATE BIT
880 000010B8 8025[EE5E0000]F7
                                              <1>
                                                             and
                                               <1>
                                               <1> K26:
882
883 000010BF 8025[F05E0000]FC
                                                                      byte [KB FLAG 3], ~(LC E0+LC E1); RESET LAST CHAR H.C. FLAG
                                               <1>
                                                            and
                                                                                                ; INTERRUPT-RETURN
                                              <1> K26A:
884
885 000010C6 FA
                                              <1>
                                                             cli
                                                                                                 ; TURN OFF INTERRUPTS
886 000010C7 B020
                                                                      al, EOI
                                               <1>
                                                             mov
                                                                                                         ; END OF INTERRUPT COMMAND
                                                                                        ;out INTA00, al ; SEND COMMAND TO INTERRUPT CONTROL PORT
887 000010C9 E620
                                              <1>
                                                             out
                                                                      20h, al
                                                                                               ; INTERRUPT-RETURN-NO-EOI
                                               <1> K27:
                                                                                                ; INSURE KEYBOARD IS ENABLED
                                                                      al, ENA KBD
889 000010CB BOAE
                                              <1>
                                                            mov
890 000010CD E8F8020000
                                              <1>
                                                             call
                                                                    SHIP_IT
                                                                                                          ; EXECUTE ENABLE
                                              <1> K27A:
892 000010D2 FA
                                                            cli
                                              <1>
                                                                                                 ; DISABLE INTERRUPTS
893
                                              <1>
                                                             ;;mov byte [intflg], 0 ; 07/01/2017 ;; 15/01/2017
                                                                                                 ; RESTORE REGISTERS
894 000010D3 07
                                              <1>
                                                            pop es
895 000010D4 1F
                                              <1>
                                                                     ds
                                                            pop
896 000010D5 5F
                                              <1>
                                                                     edi
                                                            pop
897 000010D6 5E
                                              <1>
                                                             pop
                                                                     esi
898 000010D7 5A
                                              <1>
                                                            pop
                                                                      edx
899 000010D8 59
                                              <1>
                                                                     ecx
                                                            pop
900 000010D9 5B
                                              <1>
                                                                      ebx
                                                            pop
901 000010DA 58
                                               <1>
                                                             pop
                                                                     eax
902
                                               <1>
                                                             ;pop ebp
903 000010DB CF
                                               <1>
                                                                                                 ; RETURN
                                                             iretd
904
                                               <1>
                                                             ;---- NOT IN HOLD STATE
905
                                              <1>
                                               <1> K28:
 906
                                                                                                 ; NO-HOLD-STATE
                                                                    al, 88
                                                                                                 ; TEST FOR OUT-OF-RANGE SCAN CODES
 907 000010DC 3C58
                                              <1>
                                                             cmp
908 000010DE 77DF
                                              <1>
                                                                     short K26
                                                                                                ; IGNORE IF OUT-OF-RANGE
                                                             jа
909
                                               <1>
910 000010E0 F6C308
                                               <1>
                                                             test bl, ALT_SHIFT
                                                                                                           ; ARE WE IN ALTERNATE SHIFT
                                                            ; jz short \overline{K28A}
                                               <1>
                                                                                                 ; IF NOT ALTERNATE
912 000010E3 0F84F1000000
                                               <1>
                                                                      K38
                                                             jz
913
                                               <1>
                                                            test bh, KBX
914 000010E9 F6C710
                                                                                                           ; IS THIS THE ENCHANCED KEYBOARD?
                                               <1>
                                                            jz short K29
                                                                                                ; NO, ALT STATE IS REAL
915 000010EC 740D
                                               <1>
916
                                               <1>
                                                             ;28/02/2015
917 000010EE F605[EE5E0000]04
                                                             test byte [KB_FLAG_1], SYS_SHIFT; YES, IS SYSREQ KEY DOWN?
                                               <1>
                                                                                      ; NO, ALT STATE IS REAL
                                               <1>
918
                                                            ; j z
                                                                     short K29
919 000010F5 0F85DF000000
                                               <1>
                                                                                                 ; YES, THIS IS PHONY ALT STATE
                                                                     K38
                                                             jnz
920
                                               <1>
                                                                                                 ; DUE TO PRESSING SYSREQ
                                               <1> ; K28A:
921
                                                                     jmp short K38
922
                                               <1>
923
                                               <1>
                                                             ;---- TEST FOR RESET KEY SEQUENCE (CTL ALT DEL)
                                                                                             ; TEST-RESET
924
                                              <1> K29:
925 000010FB F6C304
                                                             test bl, CTL_SHIFT
                                              <1>
                                                                                                          ; ARE WE IN CONTROL SHIFT ALSO?
 926 000010FE 740B
                                              <1>
                                                                     short K31
                                                                                                ; NO RESET
                                                             jz
927 00001100 3C53
                                                                     al, DEL KEY
                                              <1>
                                                                                                 ; CTL-ALT STATE, TEST FOR DELETE KEY
                                                             cmp
928 00001102 7507
                                              <1>
                                                                    short K31
                                                                                                 ; NO_RESET, IGNORE
929
                                              <1>
930
                                              <1>
                                                            ;---- CTL-ALT-DEL HAS BEEN FOUND
931
                                               <1>
                                                            ; 26/08/2014
932
                                               <1> cpu_reset:
933
                                               <1>
                                                            ; IBM PC/AT ROM BIOS source code - 10/06/85 (TEST4.ASM - PROC_SHUTDOWN)
934
                                              <1>
                                                            ; Send FEh (system reset command) to the keyboard controller.
935 00001104 B0FE
                                              <1>
                                                             mov al, SHUT CMD
                                                                                              ; SHUTDOWN COMMAND
 936 00001106 E664
                                                                     STATUS PORT, al
                                                                                                       ; SEND TO KEYBOARD CONTROL PORT
                                              <1>
                                                            out
                                              <1> khere:
937
938 00001108 F4
                                              <1>
                                                            hlt
                                                                                                 ; WAIT FOR 80286 RESET
                                                            jmp
939 00001109 EBFD
                                               <1>
                                                                     short khere
                                                                                                 ; INSURE HALT
940
                                               <1>
941
                                               <1>
                                                             ;---- IN ALTERNATE SHIFT, RESET NOT FOUND
942
                                               <1>
                                                                                               ; NO-RESET
943
                                               <1> K31:
944 0000110B 3C39
                                                                     al, 57
                                                                                                 ; TEST FOR SPACE KEY
                                               <1>
                                                             cmp
                                                                                              ; NOT THERE
                                                                    short K311
945 0000110D 7507
                                               <1>
                                                             jne
                                                                                                           ; SET SPACE CHAR
 946 0000110F B020
                                               <1>
                                                                      al, ' '
                                                             mov
 947 00001111 E948020000
                                                                                                            ; BUFFER FILL
                                             <1>
                                                               jmp
                                                                          K57
                                               <1> K311:
                                              <1> cmp al, 15 ; TEST FOR TAB KEY
<1> jne short K312 ; NOT THERE
<1> mov ax, 0A500h ; SET SPECIAL CODE FOR ALT-TAB
 949 00001116 3C0F
950 00001118 7509
                                             <1>
 951 0000111A 66B800A5
                                             <1>
                                                                                                         ; BUFFER FILL
952 0000111E E93B020000
                                           <1>
                                                            je K37B
954 00001123 3C4A
                                             <1>
                                                      cmp al, 74
955 00001125 0F84A2000000
                                             <1>
                                                            cmp al, 78
 956 0000112B 3C4E
                                              <1>
                                                                                                ; TEST FOR KEY PAD +
957 0000112D 0F849A000000
                                              <1>
                                                            je K37B
                                                                                                 ; GO PROCESS
                                              <1>
                                                            ;---- LOOK FOR KEY PAD ENTRY
959
                                              <1>
| Second | S
                                                                                                ; ALT-KEY-PAD
960
                                              <1> K32:
```

```
970 00001157 F6E4
                                                              mul ah
                                                <1>
                                                                       ax, di ; ADD IN THE LATEST ENTRY [ALT_INPUT], al ; STORE IT AWAY
  971 00001159 6601F8
                                                <1>
                                                               add
  972 0000115C A2[F15E0000]
                                                <1>
                                                               mov
  973
                                                 <1> ;K32A:
                                                               jmp
                                                                                                              ; THROW AWAY THAT KEYSTROKE
  974 00001161 E959FFFFFF
                                                 <1>
                                                                          K26
  975
                                                 <1>
                                                               ; NO-ALT-KEYPAD

mov byte [ALT_INPUT], 0 ; ZERO ANY PREVIOUS ENTRY INTO INPUT

mov ecx, 26 ; (DI), (ES) ALREADY POINTING

repne scasb ; LOOK FOR MARKON ----
                                                               ;---- LOOK FOR SUPERSHIFT ENTRY
  976
                                                 <1>
                                                 <1> K33:
  977
  978 00001166 C605[F15E0000]00
                                                <1>
  979 0000116D B91A000000
                                                 <1>
  980 00001172 F2AE
                                                 <1>
                                                                je short K37A ; MATCH FOUND, GO FILLL THE BUFFER
  981 00001174 7450
                                                 <1>
  982
                                                 <1>
  983
                                                 <1>
                                                               ;---- LOOK FOR TOP ROW OF ALTERNATE SHIFT
                                                                       al, 2
                                                 <1> K34:
  984
  985 00001176 3C02
                                                                                                    ; KEY WITH '1' ON IT
                                                 <1>
                                                                cmp
                                                               jb short K37B; KEY WITH '1' ON IT
jb short K37B; MUST BE ESCAPE
cmp al, 13; IS IT IN THE REGION
ja short K35; NO, ALT SOMETHING ELSE
add ah, 118; CONVERT PSEUDO S
jmp short K37A; GO FILL THE BUFFER
  986 00001178 7253
                                                <1>
  987 0000117A 3C0D
                                                <1>
  988 0000117C 7705
                                                <1>
  989 0000117E 80C476
                                                <1>
                                                                                                         ; CONVERT PSEUDO SCAN CODE TO RANGE
  990 00001181 EB43
                                                <1>
  991
                                                 <1>
  992
                                                               ;---- TRANSLATE ALTERNATE SHIFT PSEUDO SCAN CODES
                                                 <1>
  993
                                                 <1> K35:
                                                                                                    ; ALT-FUNCTION
                                                <1>
  994 00001183 3C57
                                                                        al, F11 M
                                                                                                     ; IS IT F11?
                                                               cmp
                                                                         short K35A; 20/02/2015; NO, BRANCH
  995 00001185 7209
                                                <1>
                                                                jb
  996 00001187 3C58
                                                <1>
                                                                        al, F12 M ; IS IT F12?
                                                               cmp
                                                                        short \overline{K35A}; 20/02/2015; NO, BRANCH
  997 00001189 7705
                                                <1>
                                                               ja
                                                                        ah, 52 ; CONVERT TO PSEUDO SCAN CODE short K37A ; GO FILL THE BUFFER
  998 0000118B 80C434
                                                <1>
                                                               add
 999 0000118E EB36
                                                <1>
                                                              test bh, LC_EO ; DO WE HAVE ONE OF THE NEW KEYS?

jz short K37 ; NO, JUMP

cmp al, 28 ; TEST FOR KEYPAD ENTER

jne short K35B ; NOT THERE

mov ax, 0A600h ; SPECIAL CODE

jmp K57 ; BUFFER ETTT
                                                               jmp
                                                <1> K35A:
1000
1001 00001190 F6C702
                                                <1> test bh, LC_E0
1002 00001193 7422
                                                <1>
1003 00001195 3C1C
                                                <1>
1004 00001197 7509
                                                <1>
1005 00001199 66B800A6
                                                <1>
                                         <1>
1006 0000119D E9BC010000
1007
                                                <1> K35B:
                                                                       al, 83
                                                                                                    ; TEST FOR DELETE KEY
                                                <1> cmp
1008 000011A2 3C53
                                                                        al, 83
short K37C
| Short K37C | Short K37C | Short K37C | Short K37C | Short K37C | Short K37C | Short K32A | Sho
                                                                                                    ; HANDLE WITH OTHER EDIT KEYS
                                                                                                  ; TEST FOR KEYPAD /
                                                                                                 ; NOT THERE, NO OTHER EO SPECIALS
                                                               mov ax, 0A400h ; SPECIAL CODE
                                                                                                    ; BUFFER FILL
                                                              cmp al, 59
; TEST FOR FUNCTION KEYS (F1)
                                                              jb short K37B
                                                                                                       ; NO FN, HANDLE W/OTHER EXTENDED
                                                               cmp al, 68
                                                                                                     ; IN KEYPAD REGION?
                                                              ja short K32A;
ja K26
                                                                                                    ; IF SO, IGNORE
                                                               add ah, 45
                                                                                                     ; CONVERT TO PSEUDO SCAN CODE
                                                <1> K37A:
1022
                                                al, 0
jmp K57
<1> K37B:
                                                <1>
                                                                                                    ; ASCII CODE OF ZERO
1023 000011C6 B000
1024 000011C8 E991010000
                                                                                                        ; PUT IT IN THE BUFFER
1025
                                                               mov al, 0F0h
                                                <1>
1026 000011CD B0F0
                                                                                                     ; USE SPECIAL ASCII CODE
1027 000011CF E98A010000
                                                                       K57
                                                <1>
                                                                                                          ; PUT IT IN THE BUFFER
1028
                                                <1> K37C:
                                                               add al, 80
mov ah, al
                                                                                                    ; CONVERT SCAN CODE (EDIT KEYS)
                                                <1> add
1029 000011D4 0450
                                                                                                    ; (SCAN CODE NOT IN AH FOR INSERT)
1030 000011D6 88C4
                                                <1>
                                                                       short K37A
1031 000011D8 EBEC
                                                <1>
                                                                                                             ; PUT IT IN THE BUFFER
                                                               qmj
1032
                                                <1>
1033
                                                 <1>
                                                               ;---- NOT IN ALTERNATE SHIFT
                                                                                                  ; NOT-ALT-SHIFT
1034
                                                 <1> K38:
                                                                                                  ; BL STILL HAS SHIFT FLAGS
1035
                                                 <1>
                                                                test bl, CTL SHIFT
1036 000011DA F6C304
                                                 <1>
                                                                                                     ; ARE WE IN CONTROL SHIFT?
                                                               ;jnz short K38A ; YES, START PROCESSING
1037
                                                 <1>
                                                               jz
                                                                                                        ; NOT-CTL-SHIFT
1038 000011DD 0F84B0000000
                                                 <1>
                                                                          K44
1039
                                                 <1>
1040
                                                 <1>
                                                               ;---- CONTROL SHIFT, TEST SPECIAL CHARACTERS
1041
                                                 <1>
                                                               ;---- TEST FOR BREAK
1042
                                                 <1> K38A:
                                                               cmp al, SCROLL_KEY ; TEST FOR BREAK
                                                <1>
1043 000011E3 3C46
                                                               jne short K39 ; JUMP, NO-BREAK test bh, KBX ; IS THIS
1044 000011E5 7531
                                                <1>
                                                        test bh, KBX; IS THIS THE ENHANCED KEYBOARD?

jz short K38B; NO, BREAK IS VALID

test bh, LC_EO; YES, WAS LAST CODE AN EO?

jz short K39; NO-RPEAK TEST FOR DAUGE
1045 000011E7 F6C710
                                                <1>
1046 000011EA 7405
                                                <1>
1047 000011EC F6C702
                                                <1>
1048 000011EF 7427
                                                 <1>
                                                                        short K39
                                                                                                    ; NO-BREAK, TEST FOR PAUSE
                                                               jz
                                                <1> K38B:
1049
                                            <1>
1050 000011F1 8B1D[FA5E0000]
                                                               mov
                                                                       ebx, [BUFFER_HEAD] ; RESET BUFFER TO EMPTY
1051 000011F7 891D[FE5E0000]
                                                 <1>
                                                               mov
                                                                         [BUFFER_TAIL], ebx
                                                               mov byte [BIOS BREAK], 80h ; TURN ON BIOS BREAK BIT
1052 000011FD C605[EC5E0000]80
                                                 <1>
                                                 <1>
                                                               ;---- ENABLE KEYBOARD
1054
                                                               mov al, ENA KBD ; ENABLE KEYBOARD
1055 00001204 BOAE
                                                <1>
1056 00001206 E8BF010000
                                                               call SHIP_IT
                                               <1>
                                                                                                          ; EXECUTE ENABLE
1057
                                                <1>
                                                               ; CTRL+BREAK code here !!!
1058
                                                <1>
                                                                                   ; BREAK INTERRUPT VECTOR
1059
                                                <1>
                                                               ;INT 1BH
                                                                ; 17/10/2015
1060
                                                <1>
1061 0000120B E836520000
                                                 <1>
                                                               call ctrlbrk; control+break subroutine
                                                <1>
1062
                                                               sub ax, ax ; PUT OUT DUMMY CHARACTER jmp K57 ; BUFFER_FILL
1063 00001210 6629C0
                                                <1>
1064 00001213 E946010000
                                                <1>
1065
                                                <1>
                                                               ;---- TEST FOR PAUSE
1066
                                                <1>
                                                              ; NO_BREAK

test bh, KBX ; IS THIS THE ENHANCED KEY

jnz short K41 ; YES, THEN THIS CAN'T BE PAUSE

cmp al, NUM_KEY ; LOOK FOR PAUSE KEY

jne short K41 ; NO-PAUSE
1067
                                                <1> K39:
                                                                                                             ; IS THIS THE ENHANCED KEYBOARD?
1068 00001218 F6C710
                                                <1>
1069 0000121B 7537
                                                <1>
1070 0000121D 3C45
                                                <1>
                                                              jne
1071 0000121F 7533
                                                <1>
                                                <1> K39P:
1073 00001221 800D[EE5E0000]08
                                                                        byte [KB FLAG 1], HOLD STATE; TURN ON THE HOLD FLAG
                                               <1>
                                                 <1>
                                                               ;
```

```
<1> ;---- ENABLE KEYBOARD
<1> mov al, ENA_KBD ; ENABLE KEYBOARD
<1> call SHIP_IT ; EXECUTE ENABLE
1075
1076 00001228 BOAE
1077 0000122A E89B010000
                                 <1> K39A:
                                        mov al, EOI ; END OF INTERRUPT TO CONTROL PORT out 20h, al ;out INTA00, al ; ALLOW FURTHER KEYSTROKE INTERRUPTS
1079 0000122F B020
                                  <1>
1080 00001231 E620
                                  <1>
                                 <1>
mov al, [CRT_MODE_SET] ; GET THE VALUE OF THE CURRENT MODE
1088
                                 <1>
                                                                      ; PAUSE-LOOP
1089
                                  <1> K40:
                                            test byte [KB_FLAG_1], HOLD_STATE; CHECK HOLD STATE FLAG
1090 00001246 F605[EE5E0000]08 <1>
1091 0000124D 75F7
                                            jnz short K40 ; LOOP UNTIL FLAG TURNED OFF
                                 <1>
1092
                                 <1>
                                            jmp
1093 0000124F E977FEFFFF
                                 <1>
                                                      K27
                                                                             ; INTERRUPT_RETURN_NO_EOI
1094
                                 <1>
                                            ;---- TEST SPECIAL CASE KEY 55
1095
                                 <1>
                                            ; NO-PAUSE cmp al, 55 ; TEST FOR */PRTSC KEY
                                 <1> K41:
                             ; TEST FOR */PRTSC KEY

ine short K42; NOT-KEY-55

test bh, KBX; IS THIS THE ENHANCED KEYBOARD?

iz short K41A; NO, CTL-PRTSC IS VALID

test bh, LC_E0; YES, WAS LAST CODE AN E0?

iz short K42B; NO, TRANSLATE TO A FUNCTION

K41A.
1097 00001254 3C37
1098 00001256 7513
1099 00001258 F6C710
1100 0000125B 7405
1101 0000125D F6C702
1102 00001260 7421
<1> K41A:
                                jmp K57
                                                                             ; BUFFER_FILL
                                 <1>
                                            ;
                                            ;---- SET UP TO TRANSLATE CONTROL SHIFT
                                 ; NOT-KEY-55

<1> cmp al, 15 ; IS IT THE T

<1> je short K42B
1108
                             1109 0000126B 3C0F
1110 0000126D 7414
1111 0000126F 3C35
1112 00001271 750E
1113 00001273 F6C702
1114 00001276 7409
                           1115 00001278 66B80095
1116 0000127C E9DD000000
                                           jmp K57
                                                                     ; BUFFER FILL
                                 <1> K42A:
1117
                                 <1> ;;mov ebx, K8 ; SET UP TO TRANSLATE CTL
<1> cmp al, 59 ; IS IT IN CHARACTER TABLE?
<1> ;jb short K45F ; YES, GO TRANSLATE CHA
1118
1119 00001281 3C3B
1120
                                                                         ; YES, GO TRANSLATE CHAR
                                            ;;jb K56; 20/02/2015
1121
                                 <1>
                                           ;;jmp K64 ; 20/02/2015
1122
                                 <1>
                                  <1> K42B:
1123
1124 00001283 BB[E45D0000]
1125 00001288 0F82AE000000
1126 0000128E E9B9000000
                                                  ebx, K8
                                                                      ; SET UP TO TRANSLATE CTL
                                 <1> mov
                                            jb K56 ;; 20/02/2015
                                 <1>
                                 <1>
                                            jmp K64
1127
                                 <1>
                                                  ; NOT-CTL-SHIFT
1128
                                 <1>
                                            ;---- NOT IN CONTROL SHIFT
                                           ; NOT-CTL-SHIFT

cmp al, 55 ; PRINT SCREEN KEY?

jne short K45 ; NOT PRINT SCREEN

test bh, KBX ; IS THIS ENHANCED KEYBOARD?

jz short K44A ; NO, TEST FOR SHIFT STATE

test bh, LC_EO ; YES, LAST CODE A MARKER?

jnz short K44B ; YES, IS PRINT SCREEN

jmp short K45C ; NO, TRANSLATE TO '*' CHARROTTE
                                 <1> K44:
1129
                             1130 00001293 3C37
1131 00001295 7528
1132 00001297 F6C710
1133 0000129A 7407
1134 0000129C F6C702
                              <1> jnz 
<1> jmp 
<1> K44A:
1135 0000129F 7507
1136 000012A1 EB41
1137
                                 <1>
1138 000012A3 F6C303
                                            test bl, LEFT_SHIFT+RIGHT_SHIFT; NOT 101 KBD, SHIFT KEY DOWN?
1139 000012A6 743C
                                 <1>
                                            jz short K45C ; NO, TRANSLATE TO '*' CHARACTER
1140
                                 <1>
                                            ;---- ISSUE INTERRUPT TO INDICATE PRINT SCREEN FUNCTION
                                 <1>
1141
1142
                                 <1> K44B:
                                       mov al, ENA_KBD ; INSURE KEYBOARD IS ENABLED
call SHIP_IT ; EXECUTE ENABLE
mov al, EOI ; END OF CURRENT INTERRUPT
out 20h, al; out INTA00, al ; SO FURTHER THINGS CAN HAPPEN
; Print Screen !!! ; ISSUE PRINT SCREEN INTERRUPT (INT 0...; PUSH BP ; SAVE POINTER
; INT 5H ; ISSUE PRINT SCREEN INTERRUPT
                                 <1> mov
1143 000012A8 B0AE
1144 000012AA E81B010000
                                 <1>
1145 000012AF B020
                                 <1>
1146 000012B1 E620
                                  <1>
                                           ; Print Screen !!! ; ISSUE PRINT SCREEN INTERRUPT (INT 05h)
                                  <1>
1147
1148
                                  <1>
                                 1149
1150
1151 000012B3 8025[F05E0000]FC
1152 000012BA E90CFEFFFF
1153
                                  <1>
                                            ;---- HANDLE IN-CORE KEYS
1154
                                  <1>
                                                           ; NOT-PRINT-SCREEN
1155
                                  <1> K45:
                                                                       ; TEST FOR IN-CORE AREA
                                                   al, 58
1156 000012BF 3C3A
                                  <1>
                                            cmp
1157 000012C1 7734
                             <1>
                                                  short K46
                                                                  ; JUMP IF NOT
1158 000012C3 3C35
                                 <1>
                                                                     ; IS THIS THE '/' KEY?
                                            cmp al, 53
                                                                    ; NO, JUMP
1159 000012C5 7505
                                 <1>
                                                  short K45A
                                            ine
1160 000012C7 F6C702
                                                                      ; WAS THE LAST CODE THE MARKER?
                                 <1>
                                            test bh, LC E0
1161 000012CA 7518
                                            jnz short K45C
                                 <1>
                                                                      ; YES, TRANSLATE TO CHARACTER
1162
                                 <1> K45A:
                                                                             ; LENGHT OF SEARCH
                                                  ecx, 26
1163 000012CC B91A000000
                                 <1>
1164 000012D1 BF[BA5D0000]
                                 <1>
                                            mov edi, K30+10
                                                                    ; POINT TO TABLE OF A-Z CHARS
1165 000012D6 F2AE
                                 <1>
                                            repne scasb
                                                                      ; IS THIS A LETTER KEY?
                                                  ; 20/02/2015
1166
                                  <1>
1167 000012D8 7505
                                 <1>
                                                                           ; NO, SYMBOL KEY
                                            jne
                                                  short K45B
                                 <1>
1168
                                            test bl, CAPS_STATE
1169 000012DA F6C340
                                 <1>
                                                                             ; ARE WE IN CAPS LOCK?
                                                  short K45D
                                                                      ; TEST FOR SURE
1170 000012DD 750C
                                 <1>
                                            jnz
                                 <1> K45B:
                                            test bl, LEFT SHIFT+RIGHT SHIFT; ARE WE IN SHIFT STATE?
1172 000012DF F6C303
                                 <1>
                                                               ; YES, UPPERCASE
1173 000012E2 750C
                                  <1>
                                                  short K45E
1174
                                                                       ; NO, LOWERCASE
                                  <1>
1175
                                  <1> K45C:
                                 <1>
1176 000012E4 BB[3C5E0000]
                                            mov
                                                  ebx, K10
                                                                      ; TRANSLATE TO LOWERCASE LETTERS
1177 000012E9 EB51
                                 <1>
                                            jmp
                                                  short K56
                                  <1> K45D:
                                                                       ; ALMOST-CAPS-STATE
1179 000012EB F6C303
                                  <1>
                                            test bl, LEFT_SHIFT+RIGHT_SHIFT; CL ON. IS SHIFT ON, TOO?
```

```
1180 000012EE 75F4
                                <1>
                                                                  ; SHIFTED TEMP OUT OF CAPS STATE
                                          jnz
                                                short K45C
                                 <1> K45E:
                                                                   ; TRANSLATE TO UPPER CASE LETTERS
1182 000012F0 BB[945E0000]
                                                ebx, K11
                                <1>
                                          mov
1183 000012F5 EB45
                                <1> K45F: jmp
                                                short K56
1184
                                <1>
                                          ;---- TEST FOR KEYS F1 - F10
1185
                                <1>
                                          ; NOT IN-CORE AREA cmp al, 68 ; TEST FOR F1 - F10
                                <1> K46:
1186
1187 000012F7 3C44
                                <1>
                                                                  ; JUMP IF NOT
1188
                                <1>
                                                short K47
                                          ;ja
                                                                   ; YES, GO DO FN KEY PROCESS
                                          ;jmp short K53
1189
                                <1>
1190 000012F9 7635
                                <1>
                                          jna short K53
1191
                                <1>
                                          ;---- HANDLE THE NUMERIC PAD KEYS
1192
                                <1>
                                                        ; NOT F1 - F10
1193
                                <1> K47:
                                                           ; TEST NUMPAD KEYS ; JUMP IF NOT
                                          cmp al, 83
1194 000012FB 3C53
                                <1>
                                                short K52
1195 000012FD 772D
                                <1>
                                          jа
                                <1>
                                          ;---- KEYPAD KEYS, MUST TEST NUM LOCK FOR DETERMINATION
1197
                                <1>
1198
                                <1> K48:
                                                al , 74
1199 000012FF 3C4A
                                                                          ; SPECIAL CASE FOR MINUS
                                <1>
                                          cmp
                                         je short K402

cmp al , 78 ; SFECTION ; SPECTION ; SPECTION ; GO TRANSLATE ; GO TRANSLATE test bh, LC_EO ; IS THIS ONE OFTHE NEW KEYS? short K49 ; YES, TRANSLATE TO BASE STATE
                                                               ; GO TRANSLATE
1200 00001301 74ED
                                <1>
                                          je short K45E
1201 00001303 3C4E
                                <1>
                                                                    ; SPECIAL CASE FOR PLUS
1202 00001305 74E9
                                <1>
1203 00001307 F6C702
                                <1>
                                        jnz short K49
                                                                   ; YES, TRANSLATE TO BASE STATE
1204 0000130A 750A
                                <1>
1205
                                <1>
                                          test bl, NUM_STATE ; ARE WE IN NUM LOCK jnz short K50 ; TEST FOR SURE
1206 0000130C F6C320
                                <1>
1207 0000130F 7514
                                <1>
1208 00001311 F6C303
                                <1>
                                          test bl, LEFT_SHIFT+RIGHT_SHIFT; ARE WE IN SHIFT STATE?
                                          ;jnz short K51 ; IF SHIFTED, REALLY NUM STATE
1209
                                <1>
1210 00001314 75DA
                                <1>
                                          jnz short K45E
1211
                                <1>
                                          ;---- BASE CASE FOR KEYPAD
1212
                                <1>
1213
                                <1> K49:
                                                                ; SPECIAL CASE FOR BASE STATE 5 ; CONTINUE IF NOT KEYPAD 5 ; SPECIAL ASCII CODE
1214 00001316 3C4C
                                <1>
                                                al, 76
                                          cmp
1215 00001318 7504
                                <1>
                                                short K49A
                                          jne
1216 0000131A B0F0
                                <1>
                                                al, 0F0h
                                          mov
1217 0000131C EB40
                                <1>
                                                short K57
                                                                  ; BUFFER FILL
                                          jmp
1218
                                <1> K49A:
1219 0000131E BB[3C5E0000]
                                                ebx, K10
                                                                   ; BASE CASE TABLE
                                <1>
                                          mov
                                          jmp short K64
                                                                  ; CONVERT TO PSEUDO SCAN
1220 00001323 EB27
                                <1>
1221
                                <1>
                                          ;---- MIGHT BE NUM LOCK, TEST SHIFT STATUS
1222
                                <1>
1223
                                <1> K50:
                                                                   ; ALMOST-NUM-STATE
                                            test bl, LEFT_SHIFT+RIGHT_SHIFT
1224 00001325 F6C303
                                <1>
1225 00001328 75EC
                                <1>
                                           jnz short K49 ; SHIFTED TEMP OUT OF NUM STATE
                                <1> K51: jmp short K45E
1226 0000132A EBC4
                                                                   ; REALLY NUM STATE
                                <1>
1227
1228
                                <1>
                                          ;---- TEST FOR THE NEW KEYS ON WT KEYBOARDS
                                <1> K52:
                                                         ; NOT A NUMPAD KEY
1229
                                                                  ; IS IT THE NEW WT KEY?
1230 0000132C 3C56
                                <1>
                                          cmp al, 86
                                                            ; JUMP IF NOT
1231
                                <1>
                                          ;jne short K53
                                          ;jmp short K45B
1232
                                <1>
                                                                   ; HANDLE WITH REST OF LETTER KEYS
                                          je short K45B
1233 0000132E 74AF
                                <1>
1234
                                <1>
1235
                                <1>
                                          ;---- MUST BE F11 OR F12
                                                                    ; F1 - F10 COME HERE, TOO
                                <1> K53:
                                          test bl, LEFT_SHIFT+RIGHT_SHIFT ; TEST SHIFT STATE
1237 00001330 F6C303
                                <1>
1238 00001333 74E1
                                <1>
                                                short K49 ; JUMP, LOWER CASE PSEUDO SC'S
                                                ; 20/02/2015
                                <1>
1240 00001335 BB[945E0000]
                                <1>
                                          mov ebx, K11
                                                                   ; UPPER CASE PSEUDO SCAN CODES
1241 0000133A EB10
                                <1>
                                          jmp short K64
                                                                   ; TRANSLATE SCAN
1242
                                <1>
1243
                                <1>
                                          ;---- TRANSLATE THE CHARACTER
                                                           ; TRANSLATE-CHAR
1244
                                 <1> K56:
                                                                   ; CONVERT ORIGIN
1245 0000133C FEC8
                                 <1>
                                          dec
1246 0000133E D7
                                <1>
                                          xlat
                                                                    ; CONVERT THE SCAN CODE TO ASCII
1247 0000133F F605[F05E0000]02
                                          test byte [KB_FLAG_3], LC_EO ; IS THIS A NEW KEY?
                                <1>
1248 00001346 7416
                                <1>
                                                short K57 ; NO, GO FILL BUFFER
                                          jz
                                                ah, MC E0
                                                                   ; YES, PUT SPECIAL MARKER IN AH
1249 00001348 B4E0
                                <1>
                                          mov
1250 0000134A EB12
                                                                   ; PUT IT INTO THE BUFFER
                                <1>
                                          jmp short K57
1251
                                 <1>
                                          ;---- TRANSLATE SCAN FOR PSEUDO SCAN CODES
1252
                                <1>
                                                   ; TRANSLATE-SCAN-ORGD
1253
                                <1> K64:
                                                                   ; CONVERT ORIGIN
1254 0000134C FEC8
                                <1>
                                                al
                                          dec
1255 0000134E D7
                                <1>
                                                 xlat
                                                                            ; CTL TABLE SCAN
                                                ah, al
al, 0
                                                                  ; PUT VALUE INTO AH ; ZERO ASCII CODE
1256 0000134F 88C4
                                <1>
                                          mov
                                          mov
1257 00001351 B000
                                <1>
1258 00001353 F605[F05E0000]02
                                <1>
                                          test byte [KB_FLAG_3], LC_EO ; IS THIS A NEW KEY?
                                                short K57; NO, GO FILL BUFFER al, MC_EO; YES, PUT SPECIAL MARKER IN AL
1259 0000135A 7402
                                          jz
                                <1>
1260 0000135C B0E0
                                 <1>
                                          mov
1261
                                 <1>
                                          ;---- PUT CHARACTER INTO BUFFER
1262
                                 <1>
                                 <1> K57:
                                                                  ; BUFFER FILL
                                                                 ; IS THIS AN IGNORE CHAR
1264 0000135E 3CFF
                                 <1>
                                          cmp al, -1
                                                                 ; YES, DO NOTHING WITH IT ; YES, DO NOTHING WITH IT
                                           je short K59;
1265
                                <1>
1266 00001360 0F8459FDFFFF
                                          je K26
cmp ah, -1
                                <1>
                                                                 ; LOOK FOR -1 PSEUDO SCAN
1267 00001366 80FCFF
                                <1>
                                                                  ; NEAR_INTERRUPT_RETURN
; INTERRUPT_RETURN
                                          ;jne short K61
1268
                                <1>
1269 00001369 0F8450FDFFFF
                                <1>
                                          je K26
                                 <1> ; K59:
1270
                                                                  ; NEAR_INTERRUPT_RETURN
                                                                   ; INTERRUPT RETURN
1271
                                 <1>;
                                          jmp K26
1272
                                 <1>
1273
                                <1> _K60: ; 29/01/2016
1274 0000136F 80FC68
                                <1>
                                          cmp ah, 68h
                                                            ; ALT + F1 key
1275 00001372 721F
                                                short K61
                                <1>
                                          jb
1276 00001374 80FC6F
                                <1>
                                                ah, 6Fh ; ALT + F8 key
                                          cmp
1277 00001377 771A
                                <1>
                                                short K61
                                          jа
1278
                                <1>
1279 00001379 8A1D[EE580100]
                                                bl, [ACTIVE PAGE]
                                <1>
                                          mov
                                                bl, 68h
1280 0000137F 80C368
                                <1>
                                          add
1281 00001382 38E3
                                <1>
                                          cmp
                                                bl, ah
1282 00001384 740D
                                <1>
                                          jе
                                                short K61
                                          push ax
1283 00001386 6650
                                <1>
                                                al, ah
1284 00001388 88E0
                                <1>
                                          mov
```

```
sub
1285 0000138A 2C68
                                <1>
                                               al, 68h
1286 0000138C E8F4050000
                                          call set_active_page
                                <1>
1287 00001391 6658
                                          pop
                                                ax
; NOT-CAPS-STATE
1288
                                <1> K61:
                                                ebx, [BUFFER_TAIL] ; GET THE END POINTER TO THE BUFFER
                                       ;;cli ; TURN OFF INTERRUPTS
;;mov al, EOI ; END OF INTERRUPT COMMAND
;;out INTA00, al ; SEND COMMAND TO INTERRUPT CONTROL PORT
;MOV AL, ENA_KBD ; INSURE KEYBOARD IS ENABLED
;CALL SHIP_IT ; EXECUTE ENABLE
;MOV AX, 9102H ; MOVE IN POST CODE & TYPE
;INT 15H ; PERFORM OTHER FUNCTION
;;and byte [KB_ETAC_C]
1299
                                 <1>
1300
                                 <1>
1301
                                 <1>
1302
                                 <1>
1303
                                 <1>
1304
                                 <1>
                                          ;; and byte [KB_FLAG_3], \sim (LC_E0+LC_E1) ; RESET LAST CHAR H.C. FLAG
1305
                                 <1>
                                        ; JMP K27A ; INTERRUPT_RETURN
1306
                                 <1>
                                          ;;jmp K27
1307
                                 <1>
1308
                                 <1>
                                          ;---- BUFFER IS FULL SOUND THE BEEPER
1309
                                <1> K62:
1310 000013B6 B020
                                <1>
                                                al, EOI
                                                                         ; ENABLE INTERRUPT CONTROLLER CHIP
                                          mov
1311 000013B8 E620
                                <1>
                                          out INTA00, al
1312 000013BA 66B9A602
                                                                   ; DIVISOR FOR 1760 HZ
                               <1>
                                               bl, 4
                                          mov cx, 678
                                                               ; SHORT BEEP COUNT (1/16 + 1/64 DELAY)
; GO TO COMMON BEEP HANDLER
; EXIT
1313 000013BE B304
                                <1>
                                          mov
                                <1>
1314 000013C0 E8E5090000
                                          call beep
1315 000013C5 E901FDFFFF
                                <1>
                                                                  ; EXIT
1316
                                 <1>
                                 <1> SHIP_IT:
1317
                                     1318
                                 <1>
1319
                                 <1>
                                          ; SHIP IT
1320
                                 <1>
                                                THIS ROUTINES HANDLES TRANSMISSION OF COMMAND AND DATA BYTES
1321
                                 <1>
                                               TO THE KEYBOARD CONTROLLER.
1322
                                 <1>
1323
                                 <1>
                                                                   ; SAVE DATA TO SEND
1324 000013CA 6650
                                 <1>
                                          push ax
1325
                                 <1>
1326
                                 <1>
                                          ;---- WAIT FOR COMMAND TO ACCEPTED
                                          cli ; DISABLE INTERRUPTS TILL DATA SENT ; xor ecx, ecx ; CLEAR TIMEOUT COUNTER mov ecx, 10000h
1327 000013CC FA
                                <1>
                                          cli
1328
                                <1>
1329 000013CD B900000100
                                <1>
                                          mov ecx, 10000h
1330
                                <1> S10:
                                                                    ; READ KEYBOARD CONTROLLER STATUS
                                          in al, STATUS PORT
1331 000013D2 E464
                                <1>
1332 000013D4 A802
                                          test al, INPT_BUF_FULL ; CHECK FOR ITS INPUT BUFFER BUSY
                                <1>
1333 000013D6 E0FA
                                                                   ; WAIT FOR COMMAND TO BE ACCEPTED
                                <1>
                                          loopnz S10
                                <1>
1335 000013D8 6658
                                <1>
                                                                  ; GET DATA TO SEND
1336 000013DA E664
                                <1>
                                          out
                                                STATUS PORT, al
                                                                   ; SEND TO KEYBOARD CONTROLLER
1337 000013DC FB
                                                                   ; ENABLE INTERRUPTS AGAIN
                                <1>
                                          sti
1338 000013DD C3
                                <1>
                                                                   ; RETURN TO CALLER
1339
                                 <1>
1340
                                 <1> SND_DATA:
                                        ; -----
1341
                                 <1>
1342
                                 <1>
                                          ; SND_DATA
                                1343
                                                THIS ROUTINES HANDLES TRANSMISSION OF COMMAND AND DATA BYTES
1344
                                                TO THE KEYBOARD AND RECEIPT OF ACKNOWLEDGEMENTS. IT ALSO
1345
                                <1>
                                        ; HANDLES ANY RETRIES IF REQUIRED
1346
                                 <1>
1347
                                <1>
1348 000013DE 6650
                                <1>
                                                                  ; SAVE REGISTERS
                                         push ax
1349 000013E0 6653
                                <1>
                                          push bx
1350 000013E2 51
                                <1>
                                          push
                                                ecx
                                               bh, al
bl, 3
                                                              ; SAVE TRANSMITTED BYTE FOR RETRIES ; LOAD RETRY COUNT
1351 000013E3 88C7
                                <1>
                                          mov
1352 000013E5 B303
                                <1>
                                          mov
                                <1> SD0:
1353
                                                                   ; DISABLE INTERRUPTS
1354 000013E7 FA
                                <1>
                                          and byte [KB_FLAG_2], ~(KB_FE+KB_FA); CLEAR ACK AND RESEND FLAGS
1355 000013E8 8025[EF5E0000]CF <1>
1356
                                <1>
                                          ;---- WAIT FOR COMMAND TO BE ACCEPTED
1357
                                <1>
1358 000013EF B900000100
                                <1>
                                          mov ecx, 10000h ; MAXIMUM WAIT COUNT
                                <1> SD5:
1359
                                                al, STATUS_PORT
1360 000013F4 E464
                                <1>
                                          in
                                                                          ; READ KEYBOARD PROCESSOR STATUS PORT
                                          test al, INPT BUF FULL ; CHECK FOR ANY PENDING COMMAND
1361 000013F6 A802
                                <1>
1362 000013F8 E0FA
                                <1>
                                          loopnz SD5
                                                             ; WAIT FOR COMMAND TO BE ACCEPTED
1363
                                <1>
1364 000013FA 88F8
                                                 al, bh
                                                                  ; REESTABLISH BYTE TO TRANSMIT
                                <1>
                                          mov
                                                PORT_A, al ; SEND BYTE
1365 000013FC E660
                                 <1>
                                          out
                                                                 ; ENABLE INTERRUPTS
1366 000013FE FB
                                 <1>
                                          sti
1367
                                 <1>
                                          ;mov cx, 01A00h
                                                                   ; LOAD COUNT FOR 10 ms+
1368 000013FF B9FFFF0000
                                 <1>
                                                 ecx, OFFFFh
                                 <1> SD1:
1369
1370 00001404 F605[EF5E0000]30
                                                byte [KB_FLAG_2], KB_FE+KB_FA; SEE IF EITHER BIT SET
                                 <1>
                                                               ; IF SET, SOMETHING RECEIVED GO PROCESS
1371 0000140B 750F
                                 <1>
                                          jnz short SD3
                                                                   ; OTHERWISE WAIT
1372 0000140D E2F5
                                 <1>
                                          loop SD1
1373
                                 <1> SD2:
                                                                  ; DECREMENT RETRY COUNT
1374 0000140F FECB
                                <1>
                                                bl
                                          dec
                                                short SD0
1375 00001411 75D4
                                                                  ; RETRY TRANSMISSION
                                <1>
                                          jnz
1376 00001413 800D[EF5E0000]80
                                                byte [KB FLAG 2], KB ERR; TURN ON TRANSMIT ERROR FLAG
                                <1>
                                          or
1377 0000141A EB09
                                                short SD4 ; RETRIES EXHAUSTED FORGET TRANSMISSION
                                <1>
                                          jmp
1378
                                 <1> SD3:
1379 0000141C F605[EF5E0000]10
                                <1>
                                          test byte [KB_FLAG_2], KB_FA; SEE IF THIS IS AN ACKNOWLEDGE
1380 00001423 74EA
                                <1>
                                          jz
                                                 short SD2
                                                           ; IF NOT, GO RESEND
                                <1> SD4:
1382 00001425 59
                                <1>
                                                                   ; RESTORE REGISTERS
                                          pop
                                                 ecx
1383 00001426 665B
                                <1>
                                          pop
                                                bx
1384 00001428 6658
                                <1>
                                          pop
                                                ax
1385 0000142A C3
                                                                   ; RETURN, GOOD TRANSMISSION
                                <1>
                                          retn
1386
                                 <1>
                                 <1> SND_LED:
1387
1388
                                 <1>
1389
                                 <1>
                                          ; SND LED
```

```
1390
                               <1>
                                              THIS ROUTINES TURNS ON THE MODE INDICATORS.
1391
1392
                               <1>
1393
                               <1>
1394 0000142B FA
                               <1>
                                                                ; TURN OFF INTERRUPTS
                                        cli
                                             byte [KB_FLAG_2], KB_PR_LED ; CHECK FOR MODE INDICATOR UPDATE
1395 0000142C F605[EF5E0000]40
                               <1>
                                        test
                                              short SL1 ; DON'T UPDATE AGAIN IF UPDATE UNDERWAY
1396 00001433 755F
                               <1>
1397
                               <1>
1398 00001435 800D[EF5E0000]40
                              <1>
                                              byte [KB_FLAG_2], KB_PR_LED; TURN ON UPDATE IN PROCESS
                                              al, EOI ; END OF INTERRUPT COMMAND
20h, al ;out INTA00, al ; SEND COMMAND TO INTERRUPT CONTROL PORT
                              <1>
1399 0000143C B020
                                             al, EOI
                                        mov
1400 0000143E E620
                              <1>
                                        out
1401 00001440 EB11
                               <1>
                                              short SLO ; GO SEND MODE INDICATOR COMMAND
                                        jmp
                              <1> SND LED1:
1402
1403 00001442 FA
                              <1>
                                                                ; TURN OFF INTERRUPTS
                                      cli
1404 00001443 F605[EF5E0000]40 <1>
                                        test byte [KB_FLAG_2], KB_PR_LED; CHECK FOR MODE INDICATOR UPDATE
                                                         ; DON'T UPDATE AGAIN IF UPDATE UNDERWAY
                                              short SL1
1405 0000144A 7548
                               <1>
                                        jnz
                               <1>
                                        ;
1407 0000144C 800D[EF5E0000]40
                              <1>
                                              byte [KB_FLAG_2], KB_PR_LED; TURN ON UPDATE IN PROCESS
                                        or
                               <1> SL0:
1408
                                       mov al, LED_CMD ; LED CMD BYTE call SND_DATA ; SEND DATA TO KEYBOARD
1409 00001453 B0ED
                              <1> mov
1410 00001455 E884FFFFF
                              <1>
1411 0000145A FA
                               <1>
                                       cli
                              <1>
<1>
1412 0000145B E836000000
                                             MAKE LED
                                                               ; GO FORM INDICATOR DATA BYTE
                                       call
1413 00001460 8025[EF5E0000]F8 <1> and
                                             byte [KB_FLAG_2], 0F8h ; ~KB_LEDS ; CLEAR MODE INDICATOR BITS
1414 00001467 0805[EF5E0000]
                                    or
                                              [KB_FLAG_2], al ; SAVE PRESENT INDICATORS FOR NEXT TIME
                               <1>
                                             byte [KB_FLAG_2], KB_ERR; TRANSMIT ERROR DETECTED
1415 0000146D F605[EF5E0000]80
                              <1>
                                        test
1416 00001474 750F
                              <1>
                                       jnz short SL2
                                                              ; IF SO, BYPASS SECOND BYTE TRANSMISSION
1417
                               <1>
1418 00001476 E863FFFFFF
1419 0000147B FA
                                                         ; SEND DATA TO KEYBOARD ; TURN OFF INTERRUPTS
                               <1>
                                       call SND_DATA
1419 0000147B FA
                              <1>
                                        cli
1420 0000147C F605[EF5E0000]80 <1>
                                        test byte [KB_FLAG_2], KB_ERR; TRANSMIT ERROR DETECTED
1421 00001483 7408
                               <1>
                                        jz
                                              short SL3
                                                              ; IF NOT, DON'T SEND AN ENABLE COMMAND
1422
                               <1> SL2:
                                        mov al, KB_ENABLE ; GET KEYBOARD CSA ENABLE COMMAND call SND_DATA ; SEND DATA TO KEYBOARD
1423 00001485 B0F4
                              <1>
1424 00001487 E852FFFFF
                              <1>
                                        cli
1425 0000148C FA
                               <1>
                                                                ; TURN OFF INTERRUPTS
                              <1> SL3:
1427 0000148D 8025[EF5E0000]3F
                                              byte [KB_FLAG_2], ~(KB_PR_LED+KB_ERR) ; TURN OFF MODE INDICATOR
                              <1>
1428
                               <1> SL1:
                                                                ; UPDATE AND TRANSMIT ERROR FLAG
1429 00001494 FB
                                                                ; ENABLE INTERRUPTS
                               <1>
1430 00001495 C3
                               <1>
                                                                ; RETURN TO CALLER
1431
                               <1>
                               <1> MAKE_LED:
1432
1433
                               <1>
1434
                               <1>
                                        : MAKE LED
                                        ; THIS ROUTINES FORMS THE DATA BYTE NECESSARY TO TURN ON/OFF
1435
                               <1>
1436
                               <1>
                                              THE MODE INDICATORS.
                                      ;-----
1437
                               <1>
1438
                               <1>
                                    1439
                               <1>
1440 00001496 A0[ED5E0000]
                               <1>
                                       and al, CAPS_STATE+NUM_STATE+SCROLL_STATE; ISOLATE INDICATORS; mov cl, 4; SHIFT COUNT; rol al, cl; SHIFT BITS OVER TO TURN ON INDICATORS
1441 0000149B 2470
                               <1>
1442
                               <1>
                                                                ; SHIFT BITS OVER TO TURN ON INDICATORS
1443
                              <1>
                                       rol al, 4; 20/02/2015
and al, 07h
1444 0000149D C0C004
                               <1>
1445 000014A0 2407
                               <1>
                                                                      ; MAKE SURE ONLY MODE BITS ON
                                        ;pop cx
1446
                               <1>
                                                                ; RETURN TO CALLER
1447 000014A2 C3
                               <1>
                                       retn
1448
                               <1>
1449
                               <1> ; % include 'kybdata.s' ; KEYBOARD DATA
1450
                               <1>
1451
1452
                               <1>; /// End Of KEYBOARD FUNCTIONS ///
1940
1941
                                   %include 'video.s'; 07/03/2015
                               1
  2
                               <1>; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.0 - video.s
  3
                               <1> ; Last Update: 09/12/2017
  4
                               <1> ; -----
  5
  6
                               <1> ; Beginning: 16/01/2016
                               <1> ; --
                               <1>; Assembler: NASM version 2.11 (trdos386.s)
  8
                               <1> ; ------
  9
 10
                               <1>; Turkish Rational DOS
                               <1>; Operating System Project v2.0 by ERDOGAN TAN (Beginning: 04/01/2016)
 11
 12
 13
                               <1>; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
                               <1>; video.inc (13/08/2015)
 14
 15
                               <1>; Derived from 'IBM PC-AT' BIOS source code (1985)
 16
                               <1> ; ********************************
 18
 19
                               <1>; Retro UNIX 386 v1 Kernel - VIDEO.INC
 20
                               <1> ; Last Modification: 13/08/2015
                                               (Video Data is in 'VIDATA.INC')
 21
                               <1>;
 22
                               <1>; /////// VIDEO (CGA) FUNCTIONS //////////
 23
 24
 25
                               <1>; 16/01/2016 (32 bit modifications, TRDOS386 - TRDOS v2.0, video.s)
 26
                               <1>; INT 31H (TRDOS 386) = INT 10H (IBM PC/AT REAL MODE)
 27
                               <1> ; IBM PC-AT BIOS Source Code
 28
 29
                               <1>; TITLE VIDEO1 --- 06/10/85 VIDEO DISPLAY BIOS
 30
                               <1>
 31
                               <1> _int10h:
                                        ; 23/03/2016
                               <1>
                                        ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
 33
                               <1>
 34 000014A3 9C
                               <1>
                                        pushfd
 35 000014A4 0E
                               <1>
                                        push cs
 36 000014A5 E851000000
                                        call VIDEO_IO_1
                               <1>
 37 000014AA C3
                               <1>
                                        retn
 38
                               <1>
 39
                               <1> ;--- INT 10 H -------
 40
                               <1> ; VIDEO IO
```

```
THE FOLLOWING FUNCTIONS ARE PROVIDED:
                                 <1>;
                                <1>;
 43
                                         (AH) = 00H SET MODE (AL) CONTAINS MODE VALUE
 44
                                 <1>;
                                         (AL) = 00H 40X25 BW MODE (POWER ON DEFAULT)
 45
                                 <1>;
 46
                                 <1>;
                                                (AL) = 01H \quad 40X25 \quad COLOR
                                          (AL) = 01H  40X25  COLO 
(AL) = 02H  80X25  BW 
                                 <1>;
                                <1>;
                                               (AL) = 03H 80X25 COLOR
 48
 49
                                <1>;
                                                             GRAPHICS MODES
                                          (AL) = 04H 320X200 COLOR (AL) = 05H 320X200 BW MODE
 50
                                <1> ;
 51
 52
                                 <1>;
                                                (AL) = 06H  640X200 BW MODE
                                               (AL) = 07H 80X25 MONOCHROME (USED INTERNAL TO VIDEO ONLY):
                                <1>;
 53
 54
                                 <1>;
                                                *** NOTES -BW MODES OPERATE SAME AS COLOR MODES, BUT COLOR :
 55
                                                 BURST IS NOT ENABLED
                                 <1>;
                                <1>;
                                                         -CURSOR IS NOT DISPLAYED IN GRAPHICS MODE
 56
                                       (AH) = 01H SET CURSOR TYPE
                                         (CH) = BITS 4-0 = START LINE FOR CURSOR
                                <1>;
 58
                                <1>;
                                                       ** HARDWARE WILL ALWAYS CAUSE BLINK
 59
                                                      ** SETTING BIT 5 OR 6 WILL CAUSE ERRATIC BLINKING
                                <1>;
 60
                                                       OR NO CURSOR AT ALL
                                <1>;
 61
                                                 (CL) = BITS 4-0 = END LINE FOR CURSOR
 62
                                 <1>;
                                        (AH) = 02H SET CURSOR POSITION
 63
                                <1>;
                                        (DH, DL) = ROW, COLUMN (00H, 00H) IS UPPER LEFT
                                <1>;
                                                (BH) = A PAGE NUMBER (MUST BE 00H FOR GRAPHICS MODES)
 65
                                         (BH) = PAGE NUMBER (MUST BE 00H FOR GRAPHICS MODES)
ON EXIT (DH, DL) = ROW, COLUMN OF CURRENT CURSOR
                                <1>;
 66
 67
                                <1>;
 68
                                                        (CH,CL) = CURSOR MODE CURRENTLY SET
 69
                                 <1>;
                                        (AH) = 04H READ LIGHT PEN POSITION
                                <1>;
 70
                                <1> ;
                                         ON EXIT:
 71
                                                 (AH) = 00H -- LIGHT PEN SWITCH NOT DOWN/NOT TRIGGERED
                                 <1>;
 72
                                                 (AH) = 01H -- VALID LIGHT PEN VALUE IN REGISTERS
                                <1>;
 73
                                <1>;
 74
                                                       (DH, DL) = ROW, COLUMN OF CHARACTER LP POSITION
                                <1>;
                                                        (CH) = RASTER LINE (0-199)
 75
                                <1> ;
                                                      (BX) = PIXEL COLUMN (0-319,639)
 76
                                         (AH)= 05H SELECT ACTIVE DISPLAY PAGE (VALID ONLY FOR ALPHA MODES)
 77
 78
                                         (AL) = NEW PAGE VALUE (0-7 \text{ FOR MODES } 0\&1, 0-3 \text{ FOR MODES } 2\&3)
                                 <1>;
                                         (AH) = 06H SCROLL ACTIVE PAGE UP
 79
                                 <1>;
                                         (AL) = NUMBER OF LINES. ( LINES BLANKED AT BOTTOM OF WINDOW )
 80
                                 <1>;
                                                       (AL) = 00H MEANS BLANK ENTIRE WINDOW :
 81
                                 <1>;
                                         (CH,CL) = ROW, COLOFIN C. [
(DH,DL) = ROW, COLUMN OF LOWER RIGHT CORNER OF SCALE
(BH) = ATTRIBUTE TO BE USED ON BLANK LINE

COPOLI. ACTIVE PAGE DOWN

THE PROPERTY OF W
                                <1>;
 83
                                       (AH) = 07H SCROLL ACTIVE PAGE DOWN
 85
                                 <1>;
                                         (AL) = NUMBER OF LINES, INPUT LINES BLANKED AT TOP OF WINDOW
 86
                                 <1>;
                                                   (AL) = 00H MEANS BLANK ENTIRE WINDOW :
                                <1>;
                                                (CH,CL) = ROW, COLUMN OF UPPER LEFT CORNER OF SCROLL
 88
                                                 (DH, DL) = ROW, COLUMN OF LOWER RIGHT CORNER OF SCROLL
 89
                                 <1>;
                                                (BH) = ATTRIBUTE TO BE USED ON BLANK LINE
 90
                                <1> ;
 91
 92
                                 <1>;
                                       CHARACTER HANDLING ROUTINES
 93
                                <1>;
                                         (AH) = 08H READ ATTRIBUTE/CHARACTER AT CURRENT CURSOR POSITION
                                 <1>;
 94
 95
                                 <1>;
                                               (BH) = DISPLAY PAGE (VALID FOR ALPHA MODES ONLY) :
 96
                                 <1>;
                                                ON EXIT:
                                <1>;
                                                (AL) = CHAR READ
                                                (AH) = ATTRIBUTE OF CHARACTER READ (ALPHA MODES ONLY)
 98
                                 <1>;
 99
                                         (AH) = 09H WRITE ATTRIBUTE/CHARACTER AT CURRENT CURSOR POSITION
                                 <1>;
                                               (BH) = DISPLAY PAGE (VALID FOR ALPHA MODES ONLY) :
100
                                 <1>;
101
                                 <1>;
                                                (CX) = COUNT OF CHARACTERS TO WRITE
                                <1>;
102
                                                 (AL) = CHAR TO WRITE
                                <1>;
                                                (BL) = ATTRIBUTE OF CHARACTER (ALPHA)/COLOR OF CHAR (GRAPHICS)
103
                                                       SEE NOTE ON WRITE DOT FOR BIT 7 OF BL = 1. :
                                 <1>;
                                       (AH) = 0AH WRITE CHARACTER ONLY AT CURRENT CURSOR POSITION
105
                                 <1>;
                                         (BH) = DISPLAY PAGE (VALID FOR ALPHA MODES ONLY)
                                <1>;
106
107
                                 <1>;
                                                (CX) = COUNT OF CHARACTERS TO WRITE
                                               (AL) = CHAR TO WRITE
                                <1>;
108
                                                      NOTE: USE FUNCTION (AH) = 09H IN GRAPHICS MODES
109
                                 <1>;
                                        FOR READ/WRITE CHARACTER INTERFACE WHILE IN GRAPHICS MODE, THE
110
                                <1> ;
                                                CHARACTERS ARE FORMED FROM A CHARACTER GENERATOR IMAGE
111
                                 <1>;
112
                                                MAINTAINED IN THE SYSTEM ROM. ONLY THE 1ST 128 CHARS
                                                ARE CONTAINED THERE. TO READ/WRITE THE SECOND 128 CHARS,
113
                                 <1>;
114
                                 <1>;
                                                 THE USER MUST INITIALIZE THE POINTER AT INTERRUPT 1FH
                                                 (LOCATION 0007CH) TO POINT TO THE 1K BYTE TABLE CONTAINING:
115
                                 <1>;
                                <1>;
                                                THE CODE POINTS FOR THE SECOND 128 CHARS (128-255). :
116
                                          FOR WRITE CHARACTER INTERFACE IN GRAPHICS MODE, THE REPLICATION FACTOR :
                                                CONTAINED IN (CX) ON ENTRY WILL PRODUCE VALID RESULTS ONLY:
118
                                 <1>;
                                                 FOR CHARACTERS CONTAINED ON THE SAME ROW. CONTINUATION TO :
119
                                 <1>;
                                <1>;
                                                SUCCEEDING LINES WILL NOT PRODUCE CORRECTLY.
120
                                <1>;
121
                                 <1>;
                                         GRAPHICS INTERFACE
                                                   SET COLOR PALETTE
123
                                 <1>;
                                          (AH) = OBH
                                                 (BH) = PALETTE COLOR ID BEING SET (0-127)
124
125
                                 <1>;
                                                 (BL) = COLOR VALUE TO BE USED WITH THAT COLOR ID
                                <1>;
126
                                                       NOTE: FOR THE CURRENT COLOR CARD, THIS ENTRY POINT HAS
127
                                 <1>;
                                                              MEANING ONLY FOR 320X200 GRAPHICS.
128
                                <1>;
                                                       COLOR ID = 0 SELECTS THE BACKGROUND COLOR (0-15)
                                <1>;
129
                                                       COLOR ID = 1 SELECTS THE PALETTE TO BE USED:
130
                                 <1>;
                                                              0 = GREEN(1)/RED(2)/YELLOW(3)
                                <1>;
                                                               1 = \text{CYAN}(1) / \text{MAGENTA}(2) / \text{WHITE}(3)
131
                                 <1>;
132
                                                        IN 40X25 OR 80X25 ALPHA MODES, THE VALUE SET FOR
                                <1>;
133
                                                               PALETTE COLOR 0 INDICATES THE BORDER COLOR :
                                <1>;
134
                                                               TO BE USED (VALUES 0-31, WHERE 16-31 SELECT :
135
                                 <1>;
                                                               THE HIGH INTENSITY BACKGROUND SET.
136
                                 <1>;
                                         (AH) = OCH
                                                       WRITE DOT
                                                (DX) = ROW NUMBER
137
                                 <1>;
                                <1>;
                                                 (CX) = COLUMN NUMBER
138
139
                                 <1>;
                                                 (AL) = COLOR VALUE
140
                                                      IF BIT 7 OF AL = 1, THEN THE COLOR VALUE IS EXCLUSIVE
                                 <1>;
                                                       ORED WITH THE CURRENT CONTENTS OF THE DOT
141
                                <1>;
142
                                 <1>;
                                         (AH) = ODH
                                                    READ DOT
                                <1>;
                                                (DX) = ROW NUMBER
143
                                                 (CX) = COLUMN NUMBER
                                 <1>;
144
145
                                <1>;
                                                 (AL) = RETURNS THE DOT READ
```

THESE ROUTINES PROVIDE THE CRT DISPLAY INTERFACE

41

<1>;

```
<1>;
146
147
                                                <1>;
                                                            ASCII TELETYPE ROUTINE FOR OUTPUT
                                                <1> ;
148
149
                                                <1>;
                                                             (AH) = 0EH WRITE TELETYPE TO ACTIVE PAGE
                                                <1>;
                                                                      (AL) = CHAR TO WRITE
150
151
                                                <1>;
                                                                        (BL) = FOREGROUND COLOR IN GRAPHICS MODE
                                                             NOTE -- SCREEN WIDTH IS CONTROLLED BY PREVIOUS MODE SET (AH) = 0FH CURRENT VIDEO STATE
152
                                                <1>;
153
                                                <1>;
                                                <1>;
154
                                                                       RETURNS THE CURRENT VIDEO STATE
                                                                       (AL) = MODE CURRENTLY SET ( SEE (AH) = 00H FOR EXPLANATION)
155
                                                <1>;
156
                                                <1>;
                                                                        (AH) = NUMBER OR CHARACTER COLUMNS ON SCREEN
                                                                        (BH) = CURRENT ACTIVE DISPLAY PAGE
157
                                                <1>;
                                                <1>;
                                                             (AH) = 10H RESERVED
158
159
                                                <1>;
                                                             (AH) = 11H RESERVED
                                                             (AH) - 1

(AH) = 12H
                                                                                 RESERVED
                                                <1>;
160
161
                                                <1>;
                                                             (AH) = 13H
                                                                                 WRITE STRING
                                                                                ES:BP - POINTER TO STRING TO BE WRITTEN
162
                                                <1>;
                                                                              CX - LENGTH OF CHARACTER STRING TO WRITTEN

DX - CURSOR POSITION FOR STRING TO BE WRITTEN:

BH - PAGE NUMBER:
                                                <1>;
163
164
                                                <1>;
165
                                                <1>;
                                                                     (AL) = 00H WRITE CHARACTER STRING
166
                                                <1>;
                                                                                         - ATTRIBUTE
167
                                                <1>;
                                                                                _{\mathrm{BL}}
                                                                                 STRING IS <CHAR, CHAR, ..., CHAR>
                                                <1>;
168
169
                                                <1>;
                                                                               CURSOR NOT MOVED
                                                                     (AL) = 01H WRITE CHARACTER STRING AND MOVE CURSOR
170
                                                <1>;
                                                                       BL
171
                                                <1>;
                                                                                          - ATTRIBUTE
172
                                                <1>;
                                                                                 STRING IS <CHAR, CHAR, ..., CHAR>
                                                                                 CURSOR MOVED
173
                                                <1>;
174
                                                <1>;
                                                                       (AL) = 02H
                                                                                        WRITE CHARACTER AND ATTRIBUTE STRING
                                                                                          (VALID FOR ALPHA MODES ONLY)
175
                                                <1>;
                                                                                 STRING IS <CHAR, ATTR, CHAR, ATTR .. , CHAR, ATTR>
176
                                                <1>;
177
                                                <1>;
                                                                                 CURSOR IS NOT MOVED
                                                                       (AL) = 03H WRITE CHARACTER AND ATTRIBUTE STRING AND MOVE CURSOR
178
                                                <1>;
179
                                                <1>;
                                                                                           (VALID FOR ALPHA MODES ONLY)
180
                                                <1>;
                                                                                 STRING IS <CHAR, ATTR, CHAR, ATTR .. , CHAR, ATTR>
181
                                                <1>;
                                                                                 CURSOR IS MOVED
                                                                         NOTE: CARRIAGE RETURN, LINE FEED, BACKSPACE, AND BELL ARE:
182
                                                                                  TREATED AS COMMANDS RATHER THAN PRINTABLE CHARACTERS.
                                                <1>;
183
                                                <1>;
184
                                                              BX,CX,DX,SI,DI,BP,SP,DS,ES,SS PRESERVED DURING CALLS EXCEPT FOR
                                                <1>;
185
                                                              BX,CX,DX RETURN VALUES ON FUNCTIONS 03H,04H,0DH AND 0FH. ON ALL CALLS
186
                                                <1>;
187
                                                <1>;
                                                              AX IS MODIFIED.
                                                <1> ;---
188
                                               <1>
189
190 000014AB [58150000]
                                               <1> M1: dd
                                                                      SET MODE
                                                                                         ; TABLE OF ROUTINES WITHIN VIDEO I/O
191 000014AF [C0180000]
                                               <1>
                                                              dd
                                                                       SET_CTYPE
                                                                       SET CPOS
192 000014B3 [F4180000]
                                               <1>
                                                              dd
193 000014B7 [1C190000]
                                                                       READ CURSOR
                                               <1>
                                                              dd
194
                                               <1>
                                                              ; dd
                                                                      VIDEO_RETURN ; READ_LPEN
195 000014BB [41150000]
                                              <1>
                                                                       set mode ncm ; Set mode without clearing video memory
                                                              dd
196 000014BF [62190000]
                                                              dd
                                              <1>
                                                                       ACT_DISP_PAGE
197 000014C3 [F9190000]
                                               <1>
                                                              dd
                                                                       SCROLL UP
198 000014C7 [1D1B0000]
                                                                       SCROLL DOWN
                                               <1>
                                                              dd
199 000014CB [9E1B0000]
                                              <1>
                                                                       READ AC CURRENT
200 000014CF [F61B0000]
                                               <1>
                                                                       WRITE AC CURRENT
                                                              dd
201 000014D3 [1C1C0000]
                                               <1>
                                                              dd
                                                                       WRITE_C_CURRENT
                                           dd dd dd (1) dd 
202 000014D7 [42250000]
                                                                       SET COLOR
203 000014DB [AD250000]
                                                                       WRITE DOT
204 000014DF [78250000]
                                                                       READ DOT
205 000014E3 [9E1C0000]
                                                                       WRITE TTY
206 000014E7 [29150000]
                                                                       VIDEO_STATE
207 000014EB [F82E0000]
                                                                       vga pal funcs ; 10/08/2016 (TRDOS 386)
                                                                       font_setup ; 10/07/2016 (TRDOS 386)
208 000014EF [AE2A0000]
209 000014F3 [5D150000]
                                                                       VIDEO_RETURN ; RESERVED
210 000014F7 [0B1E0000]
                                                <1>
                                                              dd
                                                                       WRITE_STRING ; 23/06/2016 (TRDOS 386)
                                                <1> M1L EQU
211
                                                                      $ - M1
212
                                                <1>
                                                <1> ; 14/01/2017
213
214
                                                <1> ; 02/01/2017
215
                                                <1>; 04/07/2016
                                                <1> ; 12/05/2016
216
                                                <1>; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
217
                                                <1> int31h: ; Video BIOS
218
219
                                                <1>
                                                <1> ; BH = Video page number
220
221
                                                <1>; BL = Color/Attribute
222
                                                <1> ; AH = Function number
                                                <1> ; AL = Character
223
224
                                                <1>
225
                                                <1> VIDEO IO 1:
                                                                                                     ; INTERRUPTS BACK ON
                                                        ;stī
226
                                                <1>
                                                                                                     ; SET DIRECTION FORWARD
227 000014FB FC
                                                <1>
                                                              cld
228 000014FC 80FC14
                                                                                                   ; TEST FOR WITHIN TABLE RANGE
                                              <1>
                                                                        ah, M1L/4
                                                              cmp
229 000014FF 7327
                                               <1>
                                                                                                    ; BRANCH TO EXIT IF NOT A VALID COMMAND
                                                                        short M4
                                               <1>
230
231 00001501 06
                                              <1>
                                                              push es
232 00001502 1E
                                              <1>
                                                                                                   ; SAVE WORK AND PARAMETER REGISTERS
                                                              push ds
                                              <1>
233 00001503 52
                                                              push edx
234 00001504 51
                                              <1>
                                                              push
                                                                       ecx
235 00001505 53
                                             <1>
                                                              push ebx
236 00001506 56
                                              <1>
                                                              push esi
237 00001507 57
                                               <1>
                                                              push edi
238 00001508 55
                                              <1>
                                                              push ebp
239
                                              <1>
                                                                       si, KDATA ; POINT DS: TO DATA SEGMENT
240 00001509 66BE1000
                                              <1>
                                                             mov
241 0000150D 8EDE
                                              <1>
                                                              mov
                                                                       ds, si
242 0000150F 8EC6
                                              <1>
                                                             mov
                                                                       es, si
                                                              mov edi, 0B8000h ; GET offset FOR COLOR CARD
243 00001511 BF00800B00
                                              <1>
244 00001516 A3[48650100]
                                               <1>
                                                              mov
                                                                       [video_eax], eax ; 12/05/2016
245
                                              <1>
                                                             : 23/03/2016
                                              <1>
                                                                                                           ; TIMES 2 FOR WORD TABLE LOOKUP
246 0000151B C0E402
                                                           shl ah, 2 ; dword
                                                                                                          ; MOVE OFFSET INTO LOOK UP REGISTER (SI) ; MOVE CURRENT MODE INTO (AH) REGISTER
247 0000151E 0FB6F4
                                              <1>
                                                             movzx esi, ah
                                                              ;mov ah, [CRT_MODE]
248
                                              <1>
249
                                               <1>
250
                                                <1>
                                                             ;;15/01/2017
```

```
251
                                <1>
                                        ; 14/01/2017
252
                                <1>
                                         ; 02/01/2017
253
                                         ;;mov byte [intflg], 31h ; video interrupt
                                <1>
254 00001521 FB
                                <1>
                                          sti
255
                                <1>
256
                                <1>
257 00001522 FFA6[AB140000]
                                          JMP dword [esi+M1]
                                                                         ; GO TO SELECTED FUNCTION
                                <1>
258
                                <1>
259
                                <1> M4:
                                                                    ; COMMAND NOT VALID
260 00001528 CF
                                                                    ; DO NOTHING IF NOT IN VALID RANGE
                                <1>
                                          iretd
261
                                <1>
                                <1> VIDEO STATE:
262
                                     ; 26/06/2016
263
                                <1>
                                         ; 12/05/2016
264
                                <1>
                                       ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
265
                                <1>
266
                                <1>
                                <1> ;------
267
                                <1> ; VIDEO STATE
268
                                 <1> ; RETURNS THE CURRENT VIDEO STATE IN AX
269
270
                                <1> ; AH = NUMBER OF COLUMNS ON THE SCREEN
271
                                <1> ; AL = CURRENT VIDEO MODE
                                <1> ; BH = CURRENT ACTIVE PAGE
273
                                274
                                <1>
                                       mov ah, [CRT_COLS] ; GET NUMBER OF COLUMNS
mov al, [CRT_MODE] ; CURRENT MODE
; movzx esi, al
275 00001529 8A25[245F0000]
                                <1>
276 0000152F A0[225F0000]
                                <1>
                                <1>
                                     ;mov ah, [esi+M6]; BH = active page
mov bh, [ACTIVE_F
cli ; 02/01/2017
278
                                <1>
279
                                <1>
280 00001534 8A3D[EE580100]
                                         mov bh, [ACTIVE PAGE] ; GET CURRENT ACTIVE PAGE
                               <1>
281 0000153A FA
                                <1>
                                         cli ; 02/01/2017
                                                      ; RECOVER REGISTERS
282 0000153B 5D
                                <1>
                                                ebp
                                         pop
283 0000153C 5F
                               <1>
                                          pop
                                                edi
284 0000153D 5E
                               <1>
                                         pop
                                                esi
                                               ecx
                                                ecx ; DISCARD SAVED BX short M15 ; RETURN TO CALLER
285 0000153E 59
                                <1>
                                          pop
286 0000153F EB26
                                <1>
                                          jmp
                                <1>
288
                                <1> set_mode_ncm:
                                      ; 04/07/2016 - TRDOS 386 (TRDOS v2.0)
289
                                <1>
290
                                         ; set mode without clearing the video memory
                                <1>
                                     ; (ony for graphics modes)
cmp al, 7; IBM PC CGA modes
jna short SET_MODE; normal function (clear)
; do not clear memory
mov [noclearmem], al; > 0
call _set_mode
291
                                <1>
292 00001541 3C07
                                <1>
293 00001543 7613
                                <1>
294
                                <1>
295 00001545 A2[57650100]
                                <1>
296 0000154A E81F000000
                                <1>
                                       mov byte [noclearmem], 0
297 0000154F C605[57650100]00
                               <1>
298 00001556 EB05
                                <1>
                                         jmp short VIDEO_RETURN
299
                                <1>
                                        ; 10/08/2016
300
                                <1>
                                       ; 08/08/2016
301
                                <1>
                                        ; 30/07/2016
; 29/07/2016
302
                                <1>
303
                                <1>
                                        ; 27/07/2016
304
                                <1>
                                        ; 26/07/2016
; 25/07/2016
; 23/07/2016
305
                                <1>
306
                                <1>
307
                                <1>
                                       ; 18/07/2016
; 02/07/2016
; 26/06/2016
308
                                <1>
309
                                <1>
310
                                <1>
                                       ; 24/06/2016
; 29/05/2016 - TRDOS 386 (TRDOS v2.0)
311
                                <1>
312
                                <1>
313
                                <1> SET MODE:
314
                                <1> ; For 32 bit TRDOS and Retro UNIX 386:
                                         ; valid video mode: 03h only!
315
                                <1>
316
                                <1>
                                                (VGA modes will be selected with another routine)
317
                                <1>
                                <1>
                                          ; set_txt_mode ; 80*25 (16 fore colors, 8 back colors)
318
319
                                <1>
                                <1> ;-----
320
321
                                <1> ; SET MODE
                                <1>; THIS ROUTINE INITIALIZES THE ATTACHMENT TO
322
323
                                         THE SELECTED MODE, THE SCREEN IS BLANKED.
                                <1>;
                                <1> ; INPUT
324
                                <1>; (AL) - MODE SELECTED (RANGE 0-7)
325
                                <1> ; OUTPUT
326
327
                                <1>; NONE
328
                                329
                                <1>
330 00001558 E811000000
                                          call _set_mode ; 24/06/2016 (set_txt_mode)
                                <1>
331
                                <1>
                                <1> ; 12/05/2016
                                <1> ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
333
334
                                <1>
335
                                                NORMAL RETURN FROM ALL VIDEO RETURNS
                                <1> ;----
336
                                <1>
                                <1> VIDEO_RETURN:
338 0000155D A1[48650100]
                                       mov eax, [video_eax] ; 12/05/2016
                                <1>
                                <1> _video_return:
339
                                       cli ; 02/01/2017
340 00001562 FA
                                <1>
341 00001563 5D
                                <1>
                                          pop ebp
342 00001564 5F
                                <1>
                                                edi
                                         pop
343 00001565 5E
                                <1>
                                         pop
                                                esi
344 00001566 5B
                                         pop ebx
                                <1>
345
                                <1> M15: ; VIDEO_RETURN_C
346
                                <1>
                                          ;;15/01/2017
347
                                <1>
                                          ; 02/01/2017
                                         ;;mov byte [intflg], 0
                                <1>
348
349
                                <1>
350 00001567 59
                                <1>
                                         pop
                                                ecx
                                         pop
351 00001568 5A
                                <1>
                                               edx
352 00001569 1F
                                <1>
                                          pop
                                               ds
                                              es
                                                      ; RECOVER SEGMENTS
353 0000156A 07
                                <1>
                                          pop
                                          iretd
                                                     ; ALL DONE
354 0000156B CF
                                <1>
355
                                <1>
```

```
356
                                 <1> set_txt_mode:
                                      ; 29/07/2016
357
                                 <1>
                                           ; 27/06/2016
358
                                 <1>
359 0000156C B003
                                 <1>
                                          mov al, 3
                                 <1>
360
361
                                 <1> ; 10/08/2016
                                 <1>; 08/08/2016
362
                                 <1> ; 30/07/2016
363
364
                                 <1> ; 29/07/2016
                                 <1> ; 27/07/2016
365
                                 <1> ; 26/07/2016
366
                                 <1> ; 25/07/2016
367
                                 <1> ; 23/07/2016
368
369
                                 <1> ; 18/07/2016
370
                                 <1>; 07/07/2016
                                 <1>; 04/07/2016
371
                                 <1>; 03/07/2016
372
373
                                 <1> ; 02/07/2016
374
                                 <1> ; 26/06/2016
                                 <1> ; 24/06/2016 (set_txt_mode -> _set_mode)
375
                                 <1> ; 17/06/2016
376
377
                                 <1>; 29/05/2016
                                 <1> ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
378
                                 <1> _set_mode:
379
                                           ; 24/06/2016
380
                                 <1>
381 0000156E 3805[225F0000]
                                 <1>
                                           cmp
                                                 [CRT_MODE], al ; current mode = requested mode ?
382 00001574 750D
                                 <1>
                                                 short _sm_0
                                           jne
383 00001576 3C03
                                                               ; text, 80*25 color, default mode
                                 <1>
                                           cmp
                                                  al, 3
384
                                 <1>
                                                               ; for TRDOS 386 MainProg
                                                     short _sm_2 ; multiscreen is only for mode 3
385 00001578 755F
                                 <1>
                                            jne
386
                                 <1>
                                           ; If ' set mode' procedure is called for video mode 3
387
                                 <1>
                                                 while video mode is 3, video page will be cleared
388
                                 <1>
389
                                 <1>
                                                 and cursor position of video page will be reset.
390
                                 <1>
                                           ; 29/07/2016
391
                                 <1>
392 0000157A 800D[55650100]80
                                           or byte [p crt mode], 80h; clear page indicator
                                 <1>
393 00001581 EB5B
                                            jmp
                                 <1>
                                                     short _sm_3
394
                                 <1>
                                     sm 0:
395 00001583 803D[225F0000]03
                                                 byte [CRT MODE], 3
                                 <1>
                                           cmp
                                            jne
396 0000158A 7534
                                 <1>
                                                    short _sm_1
397
                                 <1>
398
                                           ; If '_set_mode' procedure is called for a video mode
                                 <1>
399
                                 <1>
                                                 except video mode 3, while current video mode
                                                 is 3; all video pages of mode 3 will be copied
400
                                 <1>
                                           ;
401
                                 <1>
                                                 to 98000h address as backup, before mode change.
402
                                 <1>
                                 <1> _sm_save_pm:
403
                                          ; \overline{03}/07/2016
404
                                 <1>
                                 <1>
                                           ; save video pages
406 0000158C BE00800B00
                                 <1>
                                                 esi, 0B8000h
407 00001591 BF00800900
                                 <1>
                                           mov
                                                  edi, 98000h ; 30/07/2016
                                                 ecx, (0B8000h-0B0000h)/4
408 00001596 B900200000
                                 <1>
                                           mov
409 0000159B F3A5
                                 <1>
                                                  movsd
                                           rep
410
                                 <1>
411 0000159D C605[55650100]03
                                 <1>
                                           mov
                                                  byte [p_crt_mode], 3 ; previous mode, backup sign
412
                                 <1>
                                                cl, [ACTIVE PAGE]
                                           ; mov
413
                                 <1>
                                           ;mov
                                                 [p_crt_page], cl
414
                                 <1>
415
                                 <1>
                                           ; save cursor positions
416 000015A4 BE[DE580100]
                                 <1>
                                           mov
                                                 esi, CURSOR_POSN
417 000015A9 BF[5A650100]
                                 <1>
                                           mov
                                                  edi, cursor pposn
                                                                      ; cursor positions backup
418 000015AE B104
                                 <1>
                                           mov
                                                 cl, 4
419 000015B0 F3A5
                                 <1>
                                                 movsd
420
                                 <1>
                                           ; 29/07/2016
421
                                 <1>
                                           ;mov [ACTIVE PAGE], cl; 0
                                 <1>
423 000015B2 860D[EE580100]
                                 <1>
                                           xchg cl, [ACTIVE_PAGE]
424 000015B8 880D[56650100]
                                 <1>
                                                  [p_crt_page], cl
                                                                      ; previous page (for mode 3)
425
                                 <1>
                                           ; [ACTIVE PAGE] = 0
426 000015BE EB19
                                 <1>
                                           jmp
                                                 short sm 2
427
                                 <1>
                                 <1> _sm_1:
428
                                                  al, 3
429 000015C0 3C03
                                 <1>
                                                               ; text, 80*25 color, default mode
                                                               ; for TRDOS 386 MainProg
                                 <1>
430
431 000015C2 7515
                                 <1>
                                           jne
                                                  short _sm_2 ; multiscreen is only for mode 3
432
                                 <1>
                                           ; If '_set_mode' procedure is called for video mode 3
433
                                 <1>
434
                                 <1>
                                                 while video mode is not 3 and if there is video
435
                                                 page backup for video mode 3, all (of 8) mode 3
                                 <1>
                                           ;
436
                                 <1>
                                                 video pages will be restored from 98000h.
437
                                 <1>
438 000015C4 803D[55650100]03
                                                  byte [p_crt_mode], 3 ; previous mode, backup sign
                                                  short _sm_2; there is no (multiscreen) video pages
439 000015CB 750C
                                 <1>
440
                                 <1>
                                                         ; to be restored
441 000015CD 8A0D[56650100]
                                 <1>
                                           mov
                                                  cl, [p_crt_page]
                                                  [ACTIVE PAGE], cl
442 000015D3 880D[EE580100]
                                 <1>
                                           mov
                                 <1>
443
                                 <1> _sm_2:
444
445 000015D9 A2[225F0000]
                                 <1>
                                                  [CRT_MODE], al ; save mode in global variable
                                      mov
                                 <1> _sm_3:
446
                                          ; 30/07/2016
447
                                 <1>
448
                                 <1>
                                           ; 26/07/2016
                                          ; 25/07/2016
449
                                 <1>
450
                                 <1>
                                          ; set_mode_vga:
451
                                 <1>
                                           ; 18/07/2016
                                           ; 14/07/2016
452
                                 <1>
                                          ; 09/07/2016
453
                                 <1>
                                           ; 04/07/2016
454
                                 <1>
                                           ; 03/07/2016 (TRDOS 386 = TRDOS v2.0)
455
                                 <1>
456
                                 <1>
                                           ; /// video mode 13h ///
457
                                 <1>
                                           ; derived from 'Plex86/Bochs VGABios' source code
458
                                 <1>
                                           ; vgabios-0.7a (2011)
                                           ; by the LGPL VGABios developers Team (2001-2008)
459
                                 <1>
                                 <1>
                                           ; 'vgabios.c', 'vgatables.h'
460
```

```
<1>
461
                               <1>
                                        ; Oracle VirtualBox 5.0.24 VGABios Source Code
                                        ; ('vgabios.c', 'vgatables.h', 'vgafonts.h', 'vgarom.asm')
463
                               <1>
464
                               <1>
465 000015DE 88C4
                               <1>
                                        mov
                                              ah, al
466 000015E0 B910000000
                               <1>
                                        mov
                                               ecx, vga_mode_count
467 000015E5 BE[3E5F0000]
                              <1>
                                        mov esi, vga modes
468 000015EA 31DB
                               <1>
                                        xor
                                              ebx, ebx
469
                               <1> _sm_4:
                              <1>
470 000015EC AC
                                        lodsb
471 000015ED 38C4
                              <1>
                                         cmp ah, al
472 000015EF 740C
                                               short _sm_5
                               <1>
                                         jе
473 000015F1 FEC3
                              <1>
                                        inc bl
474 000015F3 E2F7
                              <1>
                                        loop _sm_4
475
                               <1>
                                        ; UNIMPLEMENTED VIDEO MODE !
476
                               <1>
477 000015F5 31C0
                              <1>
                                        xor eax, eax
478 000015F7 A3[48650100]
                                         mov [video_eax], eax; 0
                               <1>
479 000015FC C3
                               <1>
                                         retn
480
                               <1>
                                              eBX POINTS TO CORRECT ROW OF INITIALIZATION TABLE
481
                               <1> ;----
                               <1>
                               <1> _sm_5:
                                               ; 25/07/2016
483
484 000015FD 89DE
                               <1>
                                              esi, ebx
485 000015FF 81C6[8E5F0000]
                               <1>
                                        add
                                              esi, vga_memmodel
486 00001605 8A06
                               <1>
                                        mov
                                               al, [esi]
487 00001607 A2[6E650100]
                               <1>
                                              [VGA_MTYPE], al
                                        mov
488
                               <1>
489 0000160C 89DF
                               <1>
                                        mov
                                               edi, ebx
490 0000160E 81C7[9E5F0000]
                                        add
                                              edi, vga_dac_s
                              <1>
491 00001614 C0E302
                               <1>
                                        shl
                                               bl, 2 ; byte -> dword
492 00001617 81C3[4E5F0000]
                               <1>
                                        add
                                              ebx, vga_mode_tbl_ptr
493
                               <1>
494
                               <1>
                                        ;mov
                                              dword [VGA BASE], 0B8000h
495
                               <1>
                                        ;cmp
                                              ah, ODh ; [CRT_MODE]
496
                               <1>
                                         ;jb
                                              short M9
                               <1>
                                              dword [VGA_BASE], 0A0000h
                                        ; mov
                               <1>;M9:
498
499 0000161D 8B33
                               <1>
                                               esi, [ebx]
500 0000161F 89F3
                              <1>
                                              ebx, esi
                                        mov
                                              esi, vga_p_cm_pos ; ebx + 20
ax, [esi] ; get the cursor mode from the table
501 00001621 83C614
                              <1>
                                        add
502 00001624 668B06
                               <1>
                                        mov
                                             [CURSOR_MODE], ax; save cursor mode (initial value)
503 00001627 66A3[3B5F0000]
                              <1>
                                        mov
                                        ; al = 6, ah = 7
504
                               <1>
                                        ; al = 0Dh, ah = 0Eh ; 25/07/2016 call cursor_shape_fix
505
                               <1>
506 0000162D E83B020000
                               <1>
                                        ; al = 14, ah = 15 (If [CHAR_HEIGHT] = 16)
                               <1>
508 00001632 668906
                               <1>
                                        mov [esi], ax
509
                               <1>
                                        push esi; *
510 00001635 56
                               <1>
511
                               <1>
512 00001636 8A25[295F0000]
                               <1>
                                               ah, [VGA MODESET CTL]
                                        mov
513 0000163C 80E408
                              <1>
                                        and
                                              ah, 8 ; default palette loading ?
                                              short _sm_6
514 0000163F 7524
                               <1>
                                        jnz
515 00001641 66BAC603
                               <1>
                                              dx, 3C6h ; VGAREG_PEL_MASK (DAC mask register)
                                        mov
516 00001645 B0FF
                              <1>
                                        mov
                                              al, OFFh ; PEL mask
                                    mov
out
517 00001647 EE
                              <1>
                                              dx, al
                                              ah, [edi] ; DAC model (selection number)
518 00001648 8A27
                               <1>
                                     mov
                                        call load_dac_palette
519 0000164A E8ED0F0000
                               <1>
520
                               <1>
                                        ; ecx = 0
521 0000164F F605[295F0000]02 <1>
                                      test byte [VGA_MODESET_CTL], 2 ; gray scale summing
                                      jz
522 00001656 740D
                                              short _sm_6
                               <1>
523 00001658 53
                               <1>
                                        push ebx
                                    sub ebx, ebx; sub bl, bl
524 00001659 29DB
                               <1>
                                        mov cx, 256
525 0000165B 66B90001
                               <1>
526 0000165F E82B100000
                               <1>
                                         call gray_scale_summing
527 00001664 5B
                              <1>
                                        pop
                                              ebx
                               <1> _sm_6:
528
                                     ; Reset Attribute Ctl flip-flop
529
                               <1>
530 00001665 66BADA03
                              <1>
                                        mov dx, 3DAh; VGAREG_ACTL_RESET
531 00001669 EC
                              <1>
                                              al, dx
                                        ; Set Attribute Ctl
532
                               <1>
533 0000166A 89DE
                              <1>
                                        mov esi, ebx; addr of params tbl for selected mode
534 0000166C 83C623
                              <1>
                                              esi, 35 ; actl regs
535 0000166F 30E4
                               <1>
                                        xor
                                              ah, ah ; 0
                                              dx, 3C0h; VGAREG_ACTL_ADDRESS
536 00001671 66BAC003
                              <1>
                                        mov
                               <1> _sm_7:
                                      mov
538 00001675 88E0
                               <1>
                                              al, ah
539 00001677 EE
                               <1>
                                              dx, al ; index
                                         out
540 00001678 AC
                               <1>
                                        lodsb
                                        ; DX = 3C0h = VGAREG_ACTL_WRITE_DATA
541
                               <1>
                                        out
542 00001679 EE
                               <1>
                                               dx, al ; value
543 0000167A FEC4
                              <1>
                                         inc
544 0000167C 80FC14
                              <1>
                                              ah, 20 ; number of actl registers
                                         cmp
545 0000167F 72F4
                               <1>
                                         jЬ
                                               short _sm_7
546
                              <1>
547 00001681 88E0
                              <1>
                                              al, ah ; 20
                                        mov
548 00001683 EE
                              <1>
                                              dx, al ; index
                                        out
549 00001684 28C0
                              <1>
                                               al, al ; 0
                                         sub
550 00001686 EE
                              <1>
                                              dx, al ; value
                                        out
551
                              <1>
552
                               <1>
                                        ; Set Sequencer Ctl
553 00001687 89DE
                              <1>
                                              esi, ebx; addr of params tbl for selected mode
                                        mov
554 00001689 83C605
                              <1>
                                              esi, 5 ; sequ regs
                               <1>
                                        ;
556 0000168C 66BAC403
                              <1>
                                        mov
                                               dx, 3C4h ; VGAREG_SEQU_ADDRESS
557 00001690 EE
                              <1>
                                              dx, al ; 0
                                        out
558 00001691 6642
                                               dx ; 3C5h ; VGAREG_SEQU_DATA
                              <1>
                                        inc
559 00001693 B003
                              <1>
                                        mov
                                               al, 3
560 00001695 EE
                              <1>
                                        out
                                              dx, al
561 00001696 B401
                              <1>
                                        mov
                                              ah, 1
                              <1> _sm_8:
<1> _mov
563 00001698 88E0
                                        mov
                                              al, ah
                              <1>
                                         ;mov dx, 3C4h; VGAREG_SEQU_ADDRESS
564
565 0000169A 664A
                               <1>
                                         dec dx
```

```
566 0000169C EE
                               <1>
                                       out dx, al; index
567 0000169D AC
                               <1>
                                         lodsb
                           <1>
<1>
568 0000169E 6642
                                         inc dx; 3C5h; VGAREG_SEQU_DATA
569 000016A0 EE
                                         out dx, al
                              <1>
<1>
570 000016A1 80FC04
                                         cmp
                                               ah, 4 ; number of sequ regs
571 000016A4 7304
                                         jnb
                                               short _sm_9
                              <1> inc ah <1> jmp sho
572 000016A6 FEC4
573 000016A8 EBEE
                                               short _sm_8
574
                               <1> _sm_9:
                                     ; Set Grafx Ctl
                              <1>
575
                             <1>
576 000016AA 89DE
                                         mov esi, ebx; addr of params tbl for selected mode
577 000016AC 83C637
                               <1>
                                         add
                                               esi, 55 ; grdc regs
                               <1>
578 000016AF 30E4
                                               ah, ah ; 0
                                         xor
579
                               <1> _sm_10:
                               <1>
580 000016B1 88E0
                                    mov
                                               al, ah
                               <1>
581 000016B3 66BACE03
                                         mov
                                               dx, 3CEh ; VGAREG_GRDC_ADDRESS
582 000016B7 EE
                              <1>
                                         out
                                               dx, al
583 000016B8 AC
                             <1>
                                      lodsb
584 000016B9 6642
                               <1>
                                         inc
                                               dx ; 3CFh ; VGAREG_GRDC_DATA
585 000016BB EE
                              <1>
                                        out
                                               dx, al
586 000016BC FEC4
                              <1>
                                     inc ah
                                       cmp
jb
;
587 000016BE 80FC09
                               <1>
                                               ah, 9 ; number of grdc regs
588 000016C1 72EE
                               <1>
                                               short _sm_10
                               <1>
                                      ; Disable CRTC write protection
590
                               <1>
591 000016C3 66BAD403
                               <1>
                                        mov dx, 3D4h ; VGAREG_VGA_CRTC_ADDRESS
                               <1>
                                        ;mov al, 11h
                                       ;our dx, al;inc dx;sub al, al
593
                               <1>
594
                               <1>
595
                               <1>
596
                               <1>
                                         ;out dx, al
                                        mov ax, 11h
out dx, ax
597 000016C7 66B81100
                               <1>
                                               ax, 11h
                               <1>
598 000016CB 66EF
                           599 000016CD 89DE
                                       mov esi, ebx; addr of params tbl for selected mode
600 000016CF 83C60A
601
602
                               <1> _sm_11:
603 000016D2 88E0
                               <1> mov
                                               al, ah
604
                               <1>
                                         ; dx = 3D4h = VGAREG_VGA_CRTC_ADDRESS
                                     out dx, al; index
605 000016D4 EE
                           <1>    out    dx, al; Index
<1>    lodsb
<1>    inc    dx ; VGAREG_VGA_CRTC_ADDRESS + 1
<1>    out    dx, al; value
<1>    cmp    ah, 24; number of crtc registers - 1
<1>    jnb    short _sm_12
<1>    inc    ah
<1>    dec    dx; 3D4h
<1>    jmp    short _sm_11
<1>    sm 12:
                              <1>
                           <1>
<1>
606 000016D5 AC
607 000016D6 6642
608 000016D8 EE
609 000016D9 80FC18
610 000016DC 7306
611 000016DE FEC4
612 000016E0 664A
613 000016E2 EBEE
614
                               <1> _sm_12:
                              <1> ; Set the misc register
<1> mov dx, 3CCh; VGAREG
615
                             <1>
                                         mov dx, 3CCh; VGAREG_READ_MISC_OUTPUT
616 000016E4 66BACC03
617 000016E8 8A4309
                               <1>
                                         mov
                                               al, [ebx+9]; misc reg
618 000016EB EE
                               <1>
                                        out dx, al
619
                               <1>
                                     ; Enable video
mov dx, 3C0h
mov al, 20h
620
                               <1>
621 000016EC 66BAC003
                               <1>
                                         mov dx, 3C0h; VGAREG_ACTL_ADDRESS
622 000016F0 B020
                              <1>
623 000016F2 EE
                               <1>
                                         out dx, al ; set bit 5 to 1
                                        mov dx, 3DAh ; VGAREG_ACTL_RESET
in al, dx
624 000016F3 66BADA03
                               <1>
625 000016F7 EC
                               <1>
626
                               <1>
                                       cmp byte [noclearmem], 0
627 000016F8 803D[57650100]00
                               <1>
628 000016FF 7740
                               <1>
                                         ja short _sm_15
629
                               <1>
                                         ; 29/07/2016
630
                               <1>
632 00001703 B900400000
                               <1>
                                         xor eax, eax
                                       mov
                               <1>
                                               ecx, 4000h; 16K words (32K)
                                     cmp byte [VGA_MTYPE], 2 ; CTEXT, MTEXT, CGA
633 00001708 803D[6E650100]02 <1>
                                     ja short _sm_14
mov edi, 0B8000h
635 00001711 BF00800B00
634 0000170F 7715
                               <1>
                                               short _sm_14
                                                              ; no ? (0A0000h)
                               <1>
                               636 00001716 7409
638 00001718 A3[6A650100]
                                         mov [VGA_INT43H], eax ; 0 ; default font
639 0000171D 66B82007
                               <1> _sm_13:
640
641 00001721 F366AB
                                      rep
                               <1>
                                               stosw
642 00001724 EB1B
                               <1>
                                               short sm 15
                                         jmp
643
                               <1>
                               <1> sm 14:
644
645 00001726 BF00000A00
                               <1>
                                     mov
                                             edi, 0A0000h
646
                               <1>
                                         ; ecx = 16384 \text{ dwords} (64K)
647 0000172B 66BAC403
                                               dx, 3C4h ; VGAREG_SEQU_ADDRESS
                                <1>
                                         mov
648 0000172F B002
                               <1>
                                         mov
                                               al, 2
649 00001731 EE
                                <1>
                                         out dx, al
                                         ;mov dx, 3C5h; VGAREG_SEQU_DATA inc dx
                               <1>
650
651 00001732 6642
                               <1>
                                               al, dx ; mmask
652 00001734 EC
                               <1>
                                         in
                             <1>
653 00001735 6650
                                         push ax
654 00001737 B00F
                               <1>
                                               al, OFh ; all planes
                                         mov
655 00001739 EE
                              <1>
                                         out
                                               dx, al
                               <1>
                                               al, al ; 0
656 0000173A 30C0
                                         xor
657 0000173C F3AB
                               <1>
                                         rep
                                               stosd ; ecx = 163684 (64K)
658 0000173E 6658
                               <1>
                                         pop
                                               ax
                                               dx, al ; mmask
659 00001740 EE
                               <1>
                               <1> _sm_15:
660
661
                               <1>
                                         ; ebx = addr of params tbl for selected mode
                                         ; 10/08/2016
                               <1>
                                         mov ax, [ebx]; num of columns, 'twidth'
663 00001741 668B03
                               <1>
664 00001744 A2[245F0000]
                               <1>
                                                [CRT_COLS], al
                                         ;; 26/07/2016
                               <1>
666
                               <1>
                                         ;; CRTC_ADDRESS = 3D4h (always)
667
                                <1>
                                         ;mov ah, [ebx+1]; num of rows, 'theightm1'
668 00001749 FEC4
                                         inc ah ; 09/07/2016
                               <1>
                                         mov [VGA ROWS], ah
669 0000174B 8825[2A5F0000]
                               <1>
                                <1>
                                         ; 10/08/2016
670
```

```
671 00001751 8A4302
                                 <1>
                                         mov al, [ebx+2]
672 00001754 A2[265F0000]
                                 <1>
                                          mov
                                                 [CHAR HEIGHT], al
                                          ; 29/07/2016
673
                                 <1>
                                 <1>
                                          ; length of regen buffer in bytes
675 00001759 668B4B03
                                 <1>
                                          mov cx, [ebx+3]; 'slength l'
676 0000175D 66890D[58650100]
                                <1>
                                           mov
                                                 [CRT_LEN], cx
677
                                <1>
                                          ; 27/07/2016
678
                                 <1>
679 00001764 30E4
                                 <1>
                                           xor
680 00001766 A0[EE580100]
                                                 al, [ACTIVE PAGE]; may be > 0 for mode 3
                                 <1>
                                          mov
681
                                 <1>
                                           ;mul word [CRT_LEN] ; 4096 for mode 3
682 0000176B 66F7E1
                                                 cx ; 29/07/2016
                                 <1>
                                          mul
683 0000176E 66A3[DC580100]
                                 <1>
                                                 [CRT_START], ax
                                          mov
                                 <1>
685 00001774 B060
                                 <1>
                                                 al, 60h
                                          mov
686 00001776 803D[57650100]00
                                <1>
                                           cmp
                                                 byte [noclearmem], 0
                                                 short _sm_16
687 0000177D 7602
                                 <1>
                                           jna
688 0000177F 0480
                                 <1>
                                                 al, 80h
                                           add
689
                                 <1> _sm_16:
690 00001781 A2[275F0000]
                                 <1>
                                                 [VGA VIDEO CTL], al
                                          mov
691 00001786 C605[285F0000]F9
                                <1>
                                           mov
                                                 byte [VGA_SWITCHES], 0F9h
692 0000178D 8025[295F0000]7F
                                 <1>
                                           and
                                                 byte [VGA_MODESET_CTL], 7Fh
693
                                 <1>
694 00001794 5E
                                 <1>
                                          pop esi; *
695
                                 <1>
                                          ; 26/07/2016
696
                                 <1>
                                 <1>
                                          ; 07/07/2016
698 00001795 668B0D[3B5F0000]
                                                 cx, [CURSOR_MODE] ; restore cursor mode (initial value)
                                 <1>
                                          mov
699 0000179C 66870E
                                           xchg cx, [esi] ; cl = start line, ch = end line
                                 <1>
                                                         ; reset to initial value
700
                                 <1>
                                           xchg \, ch, cl \,; ch = start line, cl = end line
701 0000179F 86E9
                                 <1>
702 000017A1 66890D[3B5F0000]
                                 <1>
                                          mov
                                                 [CURSOR_MODE], cx; save (fixed) cursor mode
703
                                 <1>
704
                                 <1>
                                           ; 27/07/2016
                                           cmp byte [VGA_MTYPE], 2 ; CTEXT, MTEXT
705 000017A8 803D[6E650100]02
                                 <1>
706 000017AF 7317
                                 <1>
                                           jnb
                                                short _sm_17
707
                                 <1>
                                           ; Set cursor shape
708
                                 <1>
                                           ;mov cx, 0607h
;call _set_ctype
709
                                 <1>
710
                                 <1>
711
                                 <1>
712
                                 <1>
                                          ; 29/07/2016
713 000017B1 B40A
                                           mov ah, 10; 6845 register for cursor set
                                 <1>
714 000017B3 E8C4050000
                                 <1>
                                           call m16 ; output cx register
715
                                 <1>
                                          ; 25/07/2016
716
                                 <1>
                                          cmp byte [CRT_MODE], 03h
jne short _sm_17
717 000017B8 803D[225F0000]03
                                <1>
718 000017BF 7507
                                 <1>
719
                                 <1>
                                          ; 26/07/2016
720
                                <1>
721 000017C1 A0[EE580100]
                                <1>
                                           mov al, [ACTIVE_PAGE]
722 000017C6 EB0C
                                 <1>
                                           jmp
                                                 short _sm_18
                                <1> _sm_17:
723
724
                                <1>
                                       ; Set cursor pos for page 0..7
725 000017C8 6629C0
                                <1>
                                           sub
                                                ax, ax; eax = 0
726 000017CB BF[DE580100]
                                 <1>
                                           mov
                                                 edi, CURSOR_POSN
727 000017D0 AB
                                 <1>
                                           stosd
728 000017D1 AB
                                 <1>
                                           stosd
729 000017D2 AB
                                 <1>
                                           stosd
730 000017D3 AB
                                 <1>
                                           stosd
731
                                 <1>
                                          ;; Set active page 0
732
                                 <1>
                                           ;mov [ACTIVE_PAGE], al ; 0
733
                                 <1> _sm_18:
734
                                 <1>
                                          ; 29/07/2016
735 000017D4 803D[6E650100]02
                                 <1>
                                           cmp byte [VGA_MTYPE], 2 ; CTEXT, MTEXT
736 000017DB 0F8386000000
                                 <1>
                                           jnb _sm_23
737
                                 <1>
738
                                           ;cmp byte [CHAR_HEIGHT], 16
                                 <1>
739
                                 <1>
                                           ;je short _sm_19
740
                                 <1>
                                           ;; copy and activate 8x16 font
741
                                 <1>
742
                                 <1>
                                           ; 26/07/2016
743
                                 <1>
744 000017E1 B004
                                 <1>
                                           mov al, 04h
745
                                          ; sub bl, bl
                                 <1>
                                           ; AX = 1104H ; Load ROM 8x16 Character Set
746
                                 <1>
747
                                 <1>
                                           ; (BL = font block to load (EGA: 0-3; VGA: 0-7))
748 000017E3 E83A150000
                                 <1>
                                           call load_text_8_16_pat
749
                                 <1>
750
                                           ; video_func_1103h:
                                 <1>
751
                                 <1>
                                           ; biosfn set text block specifier:
                                           ; BL = font block selector code
752
                                 <1>
753
                                           ; NOTE: TRDOS 386 only uses and sets font block 0
                                 <1>
754
                                 <1>
                                           ; (It is as BL = 0 for TRDOS 386)
                                                dx, 3C4h; VGAREG_SEQU_ADDRESS
755 000017E8 66BAC403
                                 <1>
                                           mov
756
                                 <1>
                                           ;mov
                                                ah, bl
757 000017EC 28E4
                                 <1>
                                                ah, ah ; 0
                                           sub
758 000017EE B003
                                 <1>
                                           mov
                                                 al, 03h
759 000017F0 66EF
                                 <1>
                                                dx, ax
                                           out
760
                                 <1> _sm_19:
                                           ; 29/07/2016
761
                                 <1>
762
                                 <1>
                                           ; 26/07/2016
763
                                 <1>
                                           ; 24/06/2016
                                           ;mov edi, 0B8000h
764
                                 <1>
                                           ;mov cx, 4000h ; 16K words (32K)
765
                                 <1>
766
                                 <1>
767 000017F2 30C0
                                 <1>
                                           xor
                                                al, al
768 000017F4 3805[55650100]
                                 <1>
                                            cmp
                                                    byte [p_crt_mode], al ; 0
                                                     short \underline{sm}_2\overline{0}; 3h, 80h or 83h
769 000017FA 7707
                                 <1>
                                             jа
770
                                 <1>
                                           ; 30/07/2016
771
                                 <1>
772
                                 <1>
                                          ; 24/06/2016
773
                                           ; TRDOS 386 (TRDOS v2) 'set mode' modification
                                 <1>
774
                                 <1>
                                           ; (for multiscreen feature):
775
                                 <1>
                                           ; If '_set_mode' procedure is called for video mode 3
```

```
<1>
                                                while video mode is 3, video page will be cleared
776
777
                                <1>
                                                and cursor position of video page will be reset.
778
                                          ; If '_set_mode' procedure is called for a video mode
                                <1>
779
                                <1>
                                                except video mode 3, while current video mode
780
                                <1>
                                                is 3; all video pages of mode 3 will be copied
781
                                <1>
                                                to 98000h address as backup, before mode change.
                                          ; If ' set mode' procedure is called for video mode 3
782
                                <1>
                                                \stackrel{-}{\text{while}} video mode is not 3 and if there is video
783
                                <1>
784
                                <1>
                                                page backup for video mode 3, all (of 8) mode 3
785
                                                video pages will be restored from 98000h.
                                <1>
                                          ;
786
                                <1>
787 000017FC A2[EE580100]
                                                [ACTIVE PAGE], al; 0
                                <1>
                                          mov
                                          ;mov ax, 0720h
788
                                <1>
789
                                <1>
                                          ;;mov cx, 4000h; 16K words (32K)
                                          ;;mov edi, 0B8000h
790
                                <1>
791
                                <1>
                                          ;rep stosw
                                          ; sub al, al
                                <1>
793 00001801 EB64
                                <1>
                                               short _sm_23
                                          jmp
794
                                <1> _sm_20:
795
                                <1>
                                         ; Previous video mode is 3
796
                                <1>
                                          ; New video mode is 3 while current video mode is not 3
797
                                <1>
                                          ; (multi screen) video pages will be restored from OB0000h
                                <1>
799 00001803 OFB61D[EE580100]
                                <1>
                                          movzx ebx, byte [ACTIVE_PAGE]
                                          shl bl, 1; * 2
800 0000180A D0E3
                                <1>
801 0000180C 81C3[DE580100]
                                <1>
                                          add
                                               ebx, CURSOR_POSN
                                <1>
                                          ; 29/07/2016
803
                                <1>
804 00001812 F605[55650100]7F
                                          test byte [p_crt_mode], 7Fh; 83h or 3h
                                <1>
805 00001819 7427
                                                short _sm_21 ; do not restore video pages
                                <1>
                                          jz
806
                                <1>
                                <1>
                                          ;; restore video pages
                                          mov esi, 98000h; 30/07/2016
808 0000181B BE00800900
                                <1>
809 00001820 BF00800B00
                                <1>
                                                 edi, 0B8000h
810 00001825 66B90020
                                <1>
                                                cx, 2000h; 8K dwords (32K)
                                          mov
811 00001829 F3A5
                                <1>
                                          rep
                                                movsd
812
                                <1>
813
                                <1>
                                         ; restore cursor positions
814 0000182B BE[5A650100]
                                <1>
                                          mov esi, cursor_pposn
815 00001830 BF[DE580100]
                                                edi, CURSOR POSN
                                <1>
                                          mov
                                <1>
                                          ;mov ecx, 4; restore all cursor positions (16 bytes)
817 00001835 B104
                                <1>
                                          mov
                                                cl, 4
818 00001837 F3A5
                                <1>
                                          rep
                                               movsd
819
                                <1>
820 00001839 F605[55650100]80
                                <1>
                                           test byte [p_crt_mode], 80h
821 00001840 7420
                                <1>
                                          jΖ
                                               short _sm_22 ; do not clear current video pages
                                <1> _sm_21:
                                     ; clear video page
823
                                <1>
824 00001842 668B0D[58650100]
                                          mov cx, [CRT LEN]; 4096
                                <1>
825 00001849 66D1E9
                                                cx, 1 ; 2072
                                <1>
                                          shr
826 0000184C 66B82007
                                <1>
                                          mov ax, 0720h
827 00001850 BF00800B00
                                <1>
                                         mov
                                                edi, OB8000h ; [crt_base]
828 00001855 66033D[DC580100]
                                         add
                                <1>
                                                di, [CRT_START]
                                                stosw ; FILL THE REGEN BUFFER WITH BLANKS
829 0000185C F366AB
                                <1>
                                        rep
830
                                <1>
                                       mov
831 0000185F 66890B
                                <1>
                                                [ebx], cx ; reset cursor position
                                <1> _sm_22:
833 00001862 A2[55650100]
                                <1>
                                         mov
                                                 [p_crt_mode], al; 0
834
                                <1> _sm_23:
                                      ; al = video page number
835
                                <1>
836
                                <1>
                                          ; [CRT_LEN] = length of regen buffer in bytes
837 00001867 E81E010000
                                <1>
                                          call _set_active_page
                                <1>
838
839
                                <1> ;----
                                                 NORMAL RETURN FROM ALL VIDEO RETURNS
840 0000186C C3
                                <1>
                                        retn
841
                                <1>
842
                                <1> cursor shape fix:
                                        ; 07/07/2016
843
                                <1>
844
                                <1>
                                          ; (Cursor start and cursor end line values -6,7-
845
                                <1>
                                          ; will be fixed depending on character height)
846
                                <1>
                                          ; derived from 'Plex86/Bochs VGABios' source code
847
                                <1>
                                          ; vgabios-0.7a (2011)
848
                                <1>
                                         ; by the LGPL VGABios developers Team (2001-2008)
849
                                <1>
850
                                <1>
                                         ; 'vgabios.c', ' biosfn_set_cursor_shape (CH,CL)'
851
                                <1>
852
                                <1>
                                         ; INPUT ->
853
                                <1>
                                        ;
                                                AL = cursor start line (=6)
                                                AH = cursor end line (=7)
854
                                <1>
855
                                          ; OUTPUT ->
                                <1>
                                                AL = cursor start line (=14)
856
                                <1>
857
                                 <1>
                                                 AH = cursor end line (=15)
858
                                <1>
859
                                <1>
                                          ;; if((modeset_ctl&0x01)&&(cheight>8)&&(CL<8)&&(CH<0x20))
860
                                <1>
                                          ;test byte [VGA MODESET CTL], 1 ; VGA active
861
                                <1>
862
                                <1>
                                          ;jz short csf 3
863 0000186D 803D[265F0000]08
                                                byte [CHAR_HEIGHT], 8
                                <1>
                                          cmp
                                                short csf_3
864 00001874 7649
                                <1>
                                          jna
865 00001876 80FC08
                                <1>
                                                ah, 8
                                          cmp
                                                short csf 3
866 00001879 7344
                                <1>
                                          jnb
                                                al, 20h
867 0000187B 3C20
                                <1>
                                          cmp
868 0000187D 7340
                                <1>
                                                short csf 3
                                          jnb
                                <1>
870 0000187F 6650
                                <1>
                                          push
                                                ax
871
                                <1>
                                          ; {
                                          ; if (CL!=(CH+1))
872
                                <1>
873 00001881 FEC0
                                <1>
                                          inc
                                                al
874 00001883 38C4
                                <1>
                                          cmp
                                                ah, al ; ah != al + 1
875 00001885 740F
                                <1>
                                                short csf 1
                                          iе
                                          ; CH = ((CH+1) * cheight / 8) -1;
876
                                <1>
877 00001887 8A25[265F0000]
                                <1>
                                          mov
                                                ah, [CHAR_HEIGHT]
878 0000188D F6E4
                                <1>
                                          mul
                                                ah
879 0000188F C0E803
                                <1>
                                                al, 3 ; / 8
                                          shr
880 00001892 FEC8
                                <1>
                                          dec al ; - 1
```

```
881 00001894 EB0E
                                <1>
                                        jmp short csf_2
                                <1> csf 1:
883
                                         ; }
                                <1>
884
                                <1>
                                         ; else
                                                     ; ah = al + 1
885
                                <1>
                                         ; {
                                               ah ; ah = ah + 1
886 00001896 FEC4
                                <1>
                                         inc
                                     ; CH = ((CL+1) ^ cnerg...

mov al, [CHAR_HEIGHT]

mul ah

shr al, 3; / 8

al. 2; - 2
                                         ; CH = ((CL+1) * cheight / 8) - 2;
                               <1>
888 00001898 A0[265F0000]
                               <1>
889 0000189D F6E4
                                <1>
890 0000189F C0E803
                               <1>
                               <1>
891 000018A2 2C02
                               <1>
892
                                         ; al = 14 (if [CHAR\_HEIGHT] = 16)
                               <1> csf_2:
893
894 000018A4 880424
                               <1>
                                        mov
                                               [esp], al
895 000018A7 8A642401
                               <1>
                                         mov
                                               ah, [esp+1]
                                         ; CL = ((CL+1) * cheight / 8) - 1;
896
                               <1>
                                     ; CL = ((C
inc ah
897 000018AB FEC4
                               <1>
898 000018AD A0[265F0000]
899 000018B2 F6E4
                                      mov al, [CHAR_HEIGHT]
                               <1>
                               <1>
                                         mul
                                               ah
                                        shr al, 3 ; / 8
900 000018B4 C0E803
                               <1>
                                      dec al ; - 1
901 000018B7 FEC8
                               <1>
                                       mov [esp+1], al
902 000018B9 88442401
                               <1>
903
                               <1>
                                         ; ah = 15 (if [CHAR HEIGHT] = 16)
                               <1>
905 000018BD 6658
                                <1>
                                         pop
                                                ax
                                <1> csf_3:
906
907 000018BF C3
                                <1>
908
                                <1>
909
                                <1> SET_CTYPE:
                                     ; 12/09/2016
910
                                <1>
911
                                <1>
                                         ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
                                     cmp byte [CRT_MODE], 7
ja VIDEO_RETURN; 12/09/2016
call _set_ctype
jmp VIDEO_RETURN
912 000018C0 803D[225F0000]07
                                <1>
913 000018C7 0F8790FCFFFF
                                <1>
914 000018CD E805000000
                                <1>
915 000018D2 E986FCFFFF
                                <1>
916
                                <1>
917
                                <1> _set_ctype:
                                       -; 02/09/2014 (Retro UNIX 386 v1)
918
                                <1>
919
                                <1>
920
                                <1>
                                         ; VIDEO.ASM - 06/10/85 VIDEO DISPLAY BIOS
921
                                <1>
922
                                <1>
                                         ; (CH) = BITS 4-0 = START LINE FOR CURSOR
                                         ; ** HARDWARE WILL ALWAYS CAUSE BLINK
923
                                <1>
                                         ; ** SETTING BIT 5 OR 6 WILL CAUSE ERRATIC BLINKING
924
                                <1>
                                        ; OR NO CURSOR AT ALL
925
                                <1>
926
                                <1>
                                         ; (CL) = BITS 4-0 = END LINE FOR CURSOR
927
                                <1>
                                928
                                <1> ; SET CTYPE
929
                                <1>; THIS ROUTINE SETS THE CURSOR VALUE
930
931
                                <1> ; INPUT
932
                                <1>;
                                         (CX) HAS CURSOR VALUE CH-START LINE, CL-STOP LINE
933
                                <1> : OUTPUT
934
                                <1> ; NONE
935
                                <1> ;-----
936
                                <1>
937
                                <1>
                                         ; 07/07/2016
                                        ; Fixing cursor start and stop line depending on
938
                                <1>
939
                                <1>
                                         ; current character height (=16)
                                         ; (Note: Default/initial values are 6 and 7.
940
                                <1>
941
                                <1>
                                        ; If set values are 6 (start) & 7 (stop) and
                                        ; [CHAR_HEIGHT] = 16 :
942
                                <1>
943
                                <1>
                                         ; After fixing, start line will be 14, stop line
                                <1>
                                        ; will be 15.)
944
                                        mov ax, cx xchg al, ah
945 000018D7 6689C8
                                <1>
946 000018DA 86C4
                                <1>
                                <1>
                                         ; AL = start line, AH = stop line
948 000018DC E88CFFFFFF
                               <1>
                                        call cursor_shape_fix
949
                                <1>
                                         ; AL = start line (fixed), AH = stop line (fixed)
                                         mov cx, ax
950 000018E1 6689C1
                               <1>
951 000018E4 86E9
                               <1>
                                         xchg ch, cl
952
                                <1>
                                         ; CH = start line (fixed), CL = stop line (fixed)
953
                                <1>
954 000018E6 B40A
                                <1>
                                        mov ah, 10; 6845 register for cursor set
                                       mov
955 000018E8 66890D[3B5F0000]
                                <1>
                                               [CURSOR_MODE], cx ; save in data area
956
                                <1>
                                         ;call m16 ; output cx register
                                <1>
                                         ;retn
958 000018EF E988040000
                                <1>
                                          jmp
                                                   m16
959
                                <1>
960
                                <1> SET_CPOS:
                                      ; 12/09/2016
961
                                <1>
                                          ; 07/07/2016
962
                                <1>
                                          ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
963
                                <1>
964 000018F4 80FF07
                                <1>
                                               bh, 7 ; video page > 7 ; 07/07/2016
965 000018F7 0F8760FCFFFF
                                <1>
                                                VIDEO_RETURN
                                          jа
966
                                <1>
967 000018FD 803D[225F0000]07
                                <1>
                                                byte [CRT MODE], 7
                                          cmp
968 00001904 770A
                                <1>
                                                short vga_set_cpos ; 12/09/2016
                                          jа
                                                _set_cpos
969 00001906 E846040000
                                <1>
                                          call
970 0000190B E94DFCFFFF
                                                 VIDEO RETURN
                                <1>
                                           jmp
971
                                <1>
                                <1> vga_set_cpos:
<1> ; 12/09/2016
972
973
                                         ; 09/07/2016
974
                                <1>
975
                                <1>
                                         ; set cursor position
976
                                <1>
                                         ; NOTE: Hardware cursor position will not be set
977
                                <1>
                                         ; in any VGA modes (>7)
978
                                <1>
                                         ; But, cursor position will be saved into
                                             [CURSOR POSN].
979
                                <1>
                                         ; TRDOS 386 (TRDOS v2.0) uses only one page
980
                                <1>
                                         ; (page 0) for all graphics modes.
981
                                <1>
982
                                <1>
983 00001910 668915[DE580100]
                                <1>
                                         mov [CURSOR POSN], dx ; save cursor pos for pg 0
984
                                <1>
                                         ; 04/08/2016
                                          ;mov bh, [ACTIVE_PAGE] ; = 0
985
                                <1>
```

```
986
                              <1>
                                      ;call _set_cpos
                                      jmp VIDEO_RETURN
 987 00001917 E941FCFFFF
                              <1>
988
                              <1>
989
                              <1> READ_CURSOR:
                                    ; 12/09/2016
990
                              <1>
991
                              <1>
                                      ; 07/07/2016
                                    ; 12/05/2016
                                     ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
993
                              <1>
994
                              <1>
                                     ; VIDEO.ASM - 06/10/85 VIDEO DISPLAY BIOS
                              <1>
995
996
                              <1>
997
                              998
                              <1> ; READ CURSOR
                                     THIS ROUTINE READS THE CURRENT CURSOR VALUE FROM THE
999
                              <1>;
1000
                              <1>;
                                      845, FORMATS IT, AND SENDS IT BACK TO THE CALLER
                              <1> ; INPUT
1001
                              <1>; BH - PAGE OF CURSOR
1002
                              <1> ; OUTPUT
1003
                                    DX - ROW, COLUMN OF THE CURRENT CURSOR POSITION
1004
                              <1>;
                                      CX - CURRENT CURSOR MODE
1005
                              <1>;
1006
                              <1> ;------
1007
                              <1>
1008
                              <1>
                                      ; BH = Video page number (0 to 7)
1009
                              <1>
                                    ; 07/07/2016
1010
                              <1>
                                    cmp bh, 7; video page > 7 (invalid)
jna short read_cursor_1
1011 0000191C 80FF07
                             <1>
1012 0000191F 7606
                             <1>
                             1013
1014 00001921 31C9
1015 00001923 31D2
1016 00001925 EB15
1017
                             <1> read cursor 1:
                             1018
1019 00001927 803D[225F0000]07 <1>
                                  ja
1020 0000192E 7727
                                            short vga_get_cpos
                             <1>
1021
                             <1>
1024
                             <1> read_cursor_2:
1025 0000193C 5D
                                  pop ebp
                             <1>
1026 0000193D 5F
                             <1>
                                           edi
                                      pop
                                    pop
1027 0000193E 5E
                             <1>
                                            esi
1028 0000193F 5B
                             <1>
                                      pop
                                            ebx
                                  pop eax
pop eax
1029 00001940 58
                             <1>
                                            eax ; DISCARD SAVED CX AND DX
1030 00001941 58
                             <1>
1031 00001942 A1[48650100]
                             <1>
                                      mov
                                            eax, [video_eax] ; 12/05/2016
                             <1>
                                      ;;15/01/2017
1033
                                      ;;mov byte [intflg], 0 ; 07/01/2017
                             <1>
1034 00001947 1F
                             <1>
                                      pop ds
1035 00001948 07
                             <1>
                                           es
                                      pop
1036 00001949 CF
                             <1>
                                      iretd
1037
                              <1>
1038
                              <1> get_cpos:
                                   ; 12/05/2016
1039
                              <1>
1040
                              <1>
                                      ; 16/01/2016
                                     ; BH = Video page number (0 to 7)
1041
                              <1>
1042
                              <1>
                                    shl bh, 1; WORD OFFSET
1043 0000194A D0E7
                             <1>
1044 0000194C 0FB6F7
                              <1>
                                      movzx esi, bh
1045 0000194F 0FB796[DE580100]
                                      movzx edx, word [esi+CURSOR POSN]
                             <1>
1046 00001956 C3
                              <1>
1047
                              <1>
1048
                              <1> vga_get_cpos:
                                  ; 12/09/2016
1049
                              <1>
                                   ; get cursor postcon (vgc,
movzx edx, word [CURSOR_POSN] ; cursor pos
xor ecx, ecx ; Cursor Mode = 0 (invalid)
1050
                              <1>
1051 00001957 0FB715[DE580100]
                              <1>
                                      movzx edx, word [CURSOR_POSN] ; cursor pos for pg 0
1052 0000195E 31C9
                              <1>
1053 00001960 EBDA
                              <1>
1054
                              <1>
1055
                              <1> ACT_DISP_PAGE:
                                  ; 07/07/2016
1056
                              <1>
1057
                              <1>
                                      ; 26/06/2016
                                      ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
1058
                              <1>
1059
                              <1>
                                    ; VIDEO.ASM - 06/10/85 VIDEO DISPLAY BIOS
1060
                              <1>
1061
                              <1>
1062
1063
                              <1> ; ACT DISP PAGE
                              1064
1065
                              <1> ; INPUT
1066
                              <1> ;
1067
                                      AL HAS THE NEW ACTIVE DISPLAY PAGE
1068
                              <1> ; OUTPUT
1069
                              <1>; THE 6845 IS RESET TO DISPLAY THAT PAGE
1070
                              <1>;-
                                      ; 07/07/2016
1071
                              <1>
                                      cmp al, 7; > 7 = invalid video page number
1072 00001962 3C07
                              <1>
1073 00001964 0F87F3FBFFFF
                              <1>
                                          VIDEO RETURN
                                      jа
1074 0000196A 803D[225F0000]03
                             <1>
                                       cmp byte [CRT_MODE], 3
                                      je short adp_1
1075 00001971 7408
                              <1>
                                      and al, al
1076 00001973 20C0
                              <1>
1077 00001975 0F85E2FBFFFF
                              <1>
                                       jnz VIDEO_RETURN
1078
                             <1>
                                      ; sub al, al; 0; force to page 0
1079
                              <1> adp_1:
1080 0000197B E805000000
                              <1>
                                      call set_active_page
1081 00001980 E9D8FBFFFF
                              <1>
                                        jmp VIDEO_RETURN
                              <1>
1083
                              <1> set_active_page: ; tty_sw
                                    ; 09<del>/</del>12/2017
1084
                              <1>
1085
                                      ; 26/07/2016
                              <1>
1086
                                     ; 26/06/2016
                              <1>
1087
                              <1>
                                      ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
                                      ; 30/06/2015
1088
                              <1>
1089
                              <1>
                                     ; 04/03/2014 (act_disp_page --> tty_sw)
1090
                              <1>
                                     ; 10/12/2013
```

```
1091
                                       ; 04/12/2013
                               <1>
1092
                                <1>
                               <1> ;
<1> mov [ACTIVE_PAGE], al ; save active page value ; [ptty]
1093 00001985 A2[EE580100]
1094
                               <1> _set_active_page:
                                      ; 27/06/2015
1095
                               <1>
1096 0000198A 0FB6D8
                               <1>
                                         movzx ebx, al
sub ah, ah; 09/12/2017
mul word [CRT_LEN]; get saved length of regen buffer
                                                         ; display page times regen length
                              <1> ; 10/12/2013
<1> mov [CRT_START], ax ; save start address for later
<1> mov cx, ax ; start address to cx
                                        ; 10/12/2013
1102
1103 00001996 66A3[DC580100]
1104 0000199C 6689C1
1105
                               <1> _M16:
1106
                               <1>
                                        ;sar cx, 1
1107 0000199F 66D1E9
                                        shr cx, 1; divide by 2 for 6845 handling
                              <1>
1108 000019A2 B40C
                                              ah, 12; 6845 register for start address
                                        mov
                               <1>
1109 000019A4 E8D3030000
                               <1>
                                        call m16
                                    ;sal bx, 1
1110
                               <1>
                              1111
1112 000019A9 D0E3
1113 000019AB 81C3[DE580100]
1114 000019B1 668B13
                                      ; 16/01/2016
1115
                               <1>
1116
                               <1>
                                        ;call m18
                                       ;retn
1117
                               <1>
1118 000019B4 E9AF030000
                               <1>
                                       jmp
                                              m18
1119
                                <1>
1120
                               <1> position:
                                       ; 24/06/2016
1121
                                <1>
                                       ; 12/05/2016 - TRDOS 386 (TRDOS v2.0)
; 27/06/2015
1122
                                <1>
1123
                                <1>
                                      ; 02/09/2014
1124
                                <1>
                                      ; 30/08/2014 (Retro UNIX 386 v1)
; 04/12/2013 (Retro UNIX 8086 v1)
1125
                                <1>
1126
                                <1>
1127
                                      ; VIDEO.ASM - 06/10/85 VIDEO DISPLAY BIOS
1128
                                <1>
1129
                                <1>
                                <1> ;-----
1130
1131
                                <1>; POSITION
                                      THIS SERVICE ROUTINE CALCULATES THE REGEN BUFFER ADDRESS
1132
                                <1>;
                                <1>;
1133
                                        OF A CHARACTER IN THE ALPHA MODE
1134
                                <1> ; INPUT
1135
                                <1>; AX = ROW, COLUMN POSITION
1136
                                <1> ; OUTPUT
1137
                                <1>; AX = OFFSET OF CHAR POSITION IN REGEN BUFFER
                                <1> ;-----
1138
1139
                                <1>
1140
                                       ; DX = ROW, COLUMN POSITION
                               <1>
                                    movzx eax, byte [CRT_COLS]; 24/06/2016 mul dh ; row value
1141 000019B9 0FB605[245F0000] <1>
                                        mul dh ; row value xor dh, dh ; 0
1142 000019C0 F6E6
                               <1>
1143 000019C2 30F6
                               <1>
                                       add ax, dx ; add column value to the result
1144 000019C4 6601D0
                              <1>
1145 000019C7 66D1E0
                               <1>
                                        shl ax, 1 ; * 2 for attribute bytes
1146
                               <1>
                                              ; EAX = AX = OFFSET OF CHAR POSITION IN REGEN BUFFER
1147 000019CA C3
                               <1>
1148
                               <1>
1149
                               <1> find_position:
                               <1> ; 24/06/2016
1150
1151
                               <1>
                                       ; 12/05/2016 - TRDOS 386 (TRDOS v2.0)
                                       ; 27/06/2015
; 07/09/2014
1152
                                <1>
1153
                                <1>
                                       ; 02/09/2014
1154
                                <1>
                                       ; 30/08/2014 (Retro UNIX 386 v1)
1155
                                <1>
1156
                               <1>
                                        ; VIDEO.ASM - 06/10/85 VIDEO DISPLAY BIOS
                               <1>
1158 000019CB 0FB6CF
                               <1>
                                       movzx ecx, bh ; video page number
1159 000019CE 89CE
                               <1>
                                        mov esi, ecx
                                         shl si, 1
1160 000019D0 66D1E6
                               <1>
1161 000019D3 668B96[DE580100] <1>
                                         mov dx, [esi+CURSOR_POSN]
1162 000019DA 740C
                                         jz short p
xor si, si
                               <1>
                                              short p21
1163 000019DC 6631F6
                               <1>
1164
                               <1> p20:
                                              si, [CRT_LEN] ; 24/06/2016
1165 000019DF 660335[58650100]
                               <1>
                                         add
                                         ; add si, 80*25*2; add length of buffer for one page
1166
                               <1>
1167 000019E6 E2F7
                               <1>
                                         loop p20
1168
                               <1> p21:
                                         and dx, dx jz short p22
1169 000019E8 6621D2
                               <1>
1170 000019EB 7407
                               <1>
                                         call position; determine location in regen in page
1171 000019ED E8C7FFFFF
                               <1>
                                         add esi, eax; add location to start of regen page
1172 000019F2 01C6
                                <1>
1173
                                <1> p22:
1174
                                               dx, [addr 6845]; get base address of active display
                                <1>
1175
                                <1>
                                               dx, 03D4h; I/O address of color card
                                         ; mov
1176
                                <1>
                                         ;add
                                               dx, 6 ; point at status port
1177 000019F4 66BADA03
                                <1>
                                              dx, 03DAh ; status port
                                         mov
                                <1>
1178
                                         ; cx = 0
1179 000019F8 C3
                                <1>
                                         retn
1180
                                <1>
1181
                                <1> SCROLL_UP:
1182
                                <1>
                                      ; 07/07/2016
1183
                                <1>
                                         ; 26/06/2016
                                        ; 12/05/2016
                                <1>
1184
                                        ; 30/01/2016
1185
                                <1>
                                        ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
1186
                                <1>
1187
                                <1>
                                        ; 07/09/2014
                                       ; 02/09/2014
                                <1>
1188
1189
                                <1>
                                        ; 01/09/2014 (Retro UNIX 386 v1 - beginning)
                                        ; 04/04/2014
1190
                                <1>
1191
                                <1>
                                        ; 04/12/2013
1192
                                <1>
                                         ; VIDEO.ASM - 06/10/85 VIDEO DISPLAY BIOS
1193
                                <1>
1194
                                <1>
1195
                                <1> ;-----
```

```
<1> ; SCROLL UP
1196
1197
                                          THIS ROUTINE MOVES A BLOCK OF CHARACTERS UP
1198
                                <1>;
                                         ON THE SCREEN
1199
                                <1> ; INPUT
                                       (AH) = CURRENT CRT MODE
                                <1>;
1200
1201
                                <1>;
                                          (AL) = NUMBER OF ROWS TO SCROLL
                                         (CX) = ROW/COLUMN OF UPPER LEFT CORNER
1202
                                <1>;
                                       (DX) = ROW/COLUMN OF LOWER RIGHT CORNER
1203
                                <1>;
1204
                                <1>;
                                         (BH) = ATTRIBUTE TO BE USED ON BLANKED LINE
                                <1>;
                                         (DS) = DATA SEGMENT
1205
1206
                                <1>;
                                       (ES) = REGEN BUFFER SEGMENT
1207
                                <1> ; OUTPUT
1208
                                <1>;
                                         NONE -- THE REGEN BUFFER IS MODIFIED
1209
                                <1> ;-----
1210
                                <1>
1211
                                <1>
                                         ; 07/07/2016
1212 000019F9 38F5
                                       cmp ch, dh
                                <1>
                                       ja VIDEO_RETURN
cmp cl, dl
ja VIDEO_RETURN
                                               VIDEO RETURN
1213 000019FB 0F875CFBFFFF
                                <1>
1214 00001A01 38D1
                                <1>
1215 00001A03 0F8754FBFFFF
                                <1>
1216
                                <1>
                                        call _scroll_up
jmp VIDEO_RETURN
1217 00001A09 E805000000
                                <1>
1218 00001A0E E94AFBFFFF
                                <1>
                                <1>
                                <1> _scroll_up: ; from 'write_tty'
1220
1221
                                <1>
1222
                                <1>
                                         ; cl = left upper column
                                        ; ch = left upper row
1223
                                <1>
                                        ; dl = right lower column
1224
                                <1>
                                         ; dh = right lower row
1225
                                <1>
1226
                                <1>
1227
                                <1>
                                         ; al = line count
1228
                                <1>
                                         ; bl = attribute to be used on blanked line
1229
                                <1>
                                         ; bh = video page number (0 to 7)
1230
                                <1>
1231 00001A13 E896000000
                                <1>
                                          call test_line_count; 16/01/2016
                                <1>
1233 00001A18 8A25[225F0000]
                                               ah, [CRT_MODE] ; current video mode
                                <1>
                                          mov
1234
                                <1>
                                               ah, 4
                                          ;cmp
1235
                                          ;jb short n0
                                <1>
1236
                                          ;cmp byte [CRT_MODE], 4
                                <1>
1237 00001A1E 80FC04
                                          cmp ah, 4; 0\overline{7}/07/2016
                                <1>
1238 00001A21 0F8320050000
                                <1>
                                                GRAPHICS_UP ; 26/06/2016
                                          jnb
1239
                                <1>
                                          ;cmp ah, 7 ; TEST FOR BW CARD
1240
                                <1>
                                          ;jne GRAPHICS_UP
1241
                                <1>
1242
                                <1> n0:
                                         ; 07/07/2016
1243
                                <1>
1244 00001A27 80FF07
                                <1>
                                          cmp
                                               bh, 7 ; video page number
1245 00001A2A 7606
                                               short n1
                                <1>
                                          jna
1246 00001A2C 8A3D[EE580100]
                            <1>
                                               bh, [ACTIVE_PAGE]
1247
                                <1> n1:
1248 00001A32 88DC
                                               ah, bl ; attribute
                                <1>
                                          mov
1249 00001A34 6650
                                <1>
                                         push ax ; *
1250
                                <1>
                                         ;mov esi, [CRT_BASE]
                                     mov esi, OB8000h
cmp bh, [ACTIVE_PAGE]
jne short n2
1251 00001A36 BE00800B00
                                <1>
1252 00001A3B 3A3D[EE580100] <1>
1253 00001A41 750B
                                <1>
1254
                                <1>
1255 00001A43 66A1[DC580100]
                                                   ax, [CRT_START]
                               <1>
                                          mov
1256 00001A49 6601C6
                               <1>
                                          add
                                                   si, ax
1257 00001A4C EB11
                                          jmp
                                <1>
                                                   short n4
1258
                                <1> n2:
1259 00001A4E 20FF
                               <1>
                                          and
                                                 bh, bh
                                          jz short n4 mov al, bh
1260 00001A50 740D
                                <1>
1261 00001A52 88F8
                                <1>
                                <1> n3:
1263 00001A54 660335[58650100] <1>
                                          add si, [CRT_LEN]
1264 00001A5B FEC8
                                <1>
                                           dec al
1265 00001A5D 75F5
                                <1>
                                          jnz short n3
1266
                                <1> n4:
1267 00001A5F E85D000000
                                <1>
                                          call scroll_position; 16/01/2016
1268 00001A64 7420
                                <1>
                                          jz short n6
1269
                                <1>
                                          add esi, ecx; from address for scroll
1270 00001A66 01CE
                                <1>
                                          mov ch, dh ; #rows in block
1271 00001A68 88F5
                                <1>
1272 00001A6A 28C5
                                <1>
                                          sub ch, al; #rows to be moved
1273
                                <1> n5:
1274 00001A6C E894000000
                                <1>
                                          call n10 ; 16/01/2016
1275
                                <1>
1276 00001A71 51
                                <1>
                                            push ecx
1277 00001A72 0FB60D[245F0000]
                                <1>
                                          movzx ecx, byte [CRT_COLS]
                                          add cl, cl
1278 00001A79 00C9
                                <1>
1279 00001A7B 01CE
                                <1>
                                          add esi, ecx ; next line
                                           add edi, ecx
1280 00001A7D 01CF
                                <1>
1281 00001A7F 59
                                <1>
                                          pop ecx
                                <1>
                                                      ; count of lines to move
1283 00001A80 FECD
                                <1>
                                               ch
                                          dec
1284 00001A82 75E8
                                <1>
                                               short n5 ; row loop
                                          jnz
1285
                                <1>
                                          ; ch = 0
1286 00001A84 88C6
                                <1>
                                          mov
                                               dh, al ; #rows
1287
                                <1> n6:
1288
                                <1>
                                          ; attribute in ah
1289 00001A86 B020
                                <1>
                                          mov al, ''
                                                             ; fill with blanks
1290
                                <1> n7:
                                          call n11; 16/01/2016
1291 00001A88 E885000000
                                <1>
                                <1>
                                               cl, [CRT_COLS]
1293 00001A8D 8A0D[245F0000]
                                <1>
                                          mov
1294 00001A93 00C9
                                <1>
                                          add
                                               cl, cl
1295 00001A95 01CF
                                <1>
                                          add edi, ecx
1296
                                <1>
1297 00001A97 FECE
                                <1>
                                          dec
                                                dh
1298 00001A99 75ED
                                <1>
                                          jnz
                                               short n7
1299
                                <1> n16:
1300 00001A9B 3A3D[EE580100]
                                <1>
                                               bh, [ACTIVE_PAGE]
                                          cmp
```

```
1301 00001AA1 750A
                               <1>
                                         jne
                                              short n8
                               <1>
1303
                               <1>
                                              byte [CRT_MODE], 7 ; is this the black and white card
                                        ;cmp
1304
                               <1>
                                                                ; if so, skip the mode reset
1305
                               <1>
                                               al, [CRT MODE SET] ; get the value of mode set
1306 00001AA3 A0[235F0000]
                               <1>
                                         mov
1307 00001AA8 66BAD803
                                               dx, 03D8h; always set color card port
                               <1>
                                         mov
1308 00001AAC EE
                               <1>
                                         out
                                              dx, al
1309
                               <1> n8:
1310 00001AAD C3
                               <1>
                                         retn
1311
                               <1>
1312
                               <1> test line count:
                                    ; 12/05/2016
1313
                               <1>
1314
                               <1>
                                        ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
                                        ; 07/09/2014 (scroll_up)
1315
                               <1>
1316 00001AAE 08C0
                               <1>
                                        or al, al
                                      jz
1317 00001AB0 740E
                              <1>
                                              short al_set2
                                      push dx
1318 00001AB2 6652
                               <1>
1319 00001AB4 28EE
                               <1>
                                        sub
                                              dh, ch ; subtract upper row from lower row number
1320 00001AB6 FEC6
                              <1>
                                        inc dh ; adjust difference by 1
1321 00001AB8 38C6
                                         cmp dh, al ; line count = amount of rows in window?
                              <1>
1322 00001ABA 7502
                                              short al set1 ; if not the we're all set
                               <1>
                                         jne
1323 00001ABC 30C0
                               <1>
                                              al, al; otherwise set al to zero
                                        xor
1324
                               <1> al_set1:
                               <1> pop
1325 00001ABE 665A
                                              dx
1326
                               <1> al_set2:
1327 00001AC0 C3
                               <1>
                                        retn
1328
                               <1>
1329
                               <1> scroll_position:
1330
                               <1> ; 26/06/2016
                                       ; 30/01/2016
1331
                               <1>
                                        ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
1332
                               <1>
                                        ; 07/09/2014 (scroll_up)
1333
                               <1>
1334
                               <1>
1335 00001AC1 6652
                               <1>
                                        push dx
1336 00001AC3 6689CA
                                              dx, cx; now, upper left position in DX
                               <1>
                                        mov
1337 00001AC6 E8EEFEFFFF
                                        call position
                              <1>
1338 00001ACB 01C6
                               <1>
                                        add esi, eax
1339 00001ACD 89F7
                               <1>
                                        mov
                                              edi, esi
                                        pop dx ; lower right position in DX
1340 00001ACF 665A
                              <1>
                                     sub dx, cx
1341 00001AD1 6629CA
                              <1>
                                              dh  ; dh = #rows
dl  ; dl = #cols
1342 00001AD4 FEC6
                               <1>
                                        inc
1343 00001AD6 FEC2
                                                    ; dl = \#cols in block
                              <1>
                                        inc
1344 00001AD8 59
                                        pop ecx ; return address
                              <1>
                                        pop ax ; *; al = line count, ah = attribute
push ecx ; return address
1345 00001AD9 6658
                               <1>
1346 00001ADB 51
                               <1>
1347 00001ADC 0FB7C8
                                        movzx ecx, ax
                              <1>
1348 00001ADF 8A25[245F0000]
                              <1>
                                        mov ah, [CRT_COLS]
                                              ah ; determine offset to from address
1349 00001AE5 F6E4
                               <1>
                                        mul
1350 00001AE7 6601C0
                               <1>
                                        add ax, ax ; *2 for attribute byte
1351
                               <1>
1352 00001AEA 6650
                               <1>
                                         push ax
                                                   ; offset
1353 00001AEC 6652
                               <1>
                                         push dx
1354
                               <1>
1355
                               <1>
                                        ; 04/04/2014
1356 00001AEE 66BADA03
                               <1>
                                         mov dx, 3DAh ; guaranteed to be color card here
                                                ; wait_display_enable al, dx ; get port
                               <1> n9:
1358 00001AF2 EC
                               <1>
                                         in
1359 00001AF3 A808
                               <1>
                                         test al, RVRT; wait for vertical retrace
1360 00001AF5 74FB
                              <1>
                                         jz short n9 ; wait_display_enable
1361 00001AF7 B025
                              <1>
                                        mov al, 25h
1362 00001AF9 B2D8
                               <1>
                                        mov
                                              dl, 0D8h; address control port
                              <1>
1363 00001AFB EE
                                              dx, al; turn off video during vertical retrace
                                         out
                                        pop dx ; #rows, #cols
1364 00001AFC 665A
                              <1>
                                              pop ax
                                                         ; offset
1365 00001AFE 6658
                               <1>
1366 00001B00 6691
                               <1>
                                         xchg ax, cx;
1367
                               <1>
                                        ; ecx = offset, al = line count, ah = attribute
1368
                               <1>
1369 00001B02 08C0
                               <1>
                                         or
                                              al, al
1370 00001B04 C3
                               <1>
                                         retn
1371
                               <1> n10:
1372
                               <1>
                                         ; Move rows
1373 00001B05 88D1
                               <1>
                                         mov cl, dl; get # of cols to move
1374 00001B07 56
                                         push esi
                               <1>
                                        push edi ; save start address
1375 00001B08 57
                               <1>
                               <1> n10r:
1376
1377 00001B09 66A5
                               <1>
                                                    ; move that line on screen
                                        movsw
1378 00001B0B FEC9
                               <1>
                                         dec cl
                                         jnz short n10r
1379 00001B0D 75FA
                               <1>
1380 00001B0F 5F
                               <1>
                                         pop edi
1381 00001B10 5E
                               <1>
                                         pop esi ; recover addresses
1382 00001B11 C3
                               <1>
                                         retn
1383
                               <1> n11:
                                         ; Clear rows
1384
                               <1>
                                                   ; dh = #rows
                               <1>
1385
1386 00001B12 88D1
                                           mov cl, dl; get # of cols to clear
                               <1>
                                          push edi ; save address
1387 00001B14 57
                               <1>
1388
                               <1> n11r:
1389 00001B15 66AB
                               <1>
                                                         ; store fill character
1390 00001B17 FEC9
                               <1>
                                         dec cl
1391 00001B19 75FA
                               <1>
                                         jnz
                                                  short n11r
1392 00001B1B 5F
                               <1>
                                          pop
                                                  edi ; recover address
1393 00001B1C C3
                               <1>
                                         retn
1394
                               <1>
1395
                               <1> SCROLL DOWN:
                                        ; 07/07/2016
1396
                               <1>
1397
                               <1>
                                        ; 27/06/2016
1398
                               <1>
                                       ; 26/06/2016
1399
                               <1>
                                        ; 12/05/2016
                                        ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
1400
                               <1>
1401
                               <1>
                                       ; VIDEO.ASM - 06/10/85 VIDEO DISPLAY BIOS
1402
                                <1>
1403
                               <1>
1404
                               <1> ;-----
1405
                               <1> ; SCROLL DOWN
```

```
<1>;
1406
                                          THIS ROUTINE MOVES THE CHARACTERS WITHIN A DEFINED
1407
                                 <1>;
                                          BLOCK DOWN ON THE SCREEN, FILLING THE TOP LINES
                                <1>;
1408
                                          WITH A DEFINED CHARACTER
1409
                                <1> ; INPUT
1410
                                <1>;
                                          (AH) = CURRENT CRT MODE
1411
                                <1>;
                                          (AL) = NUMBER OF LINES TO SCROLL
                                          (CX) = UPPER LEFT CORNER OF RECION
1412
                                <1>;
                                          (DX) = LOWER RIGHT CORNER OF REGION
1413
                                <1>;
1414
                                <1>;
                                          (BH) = FILL CHARACTER
                                <1>;
                                          (DS) = DATA SEGMENT
1415
1416
                                <1>;
                                          (ES) = REGEN SEGMENT
1417
                                <1> ; OUTPUT
1418
                                <1>;
                                         NONE -- SCREEN IS SCROLLED
1419
                                <1> ;-----
1420
                                <1>
1421
                                <1>
                                          ; 07/07/2016
1422 00001B1D 38F5
                                <1>
                                         cmp ch, dh
1423 00001B1F 0F8738FAFFFF
                                                VIDEO RETURN
                                <1>
                                       ja
1424 00001B25 38D1
                                <1>
                                               cl, dl
                                          cmp
1425 00001B27 0F8730FAFFFF
                                         ja VIDEO_RETURN
                                <1>
1426
                                <1>
                                         call _scroll_down
jmp VIDEO_RETURN
1427 00001B2D E805000000
                                <1>
1428 00001B32 E926FAFFFF
                                <1>
                                <1>
1430
                                <1> _scroll_down: ; 27/06/2016
1431
                                <1>
1432
                                <1>
                                          ; cl = left upper column
                                <1>
                                         ; ch = left upper row
1433
1434
                                <1>
                                         ; dl = right lower column
                                         ; dh = right lower row
1435
                                <1>
1436
                                <1>
1437
                                <1>
                                         ; al = line count
                                          ; bl = attribute to be used on blanked line
1438
                                <1>
1439
                                <1>
                                          ; bh = video page number (0 to 7)
1440
                                <1>
1441
                                <1>
                                          ; !!!!
1442 00001B37 FD
                                                      ; DIRECTION FOR SCROLL DOWN
                                <1>
1443
                                <1>
                                          ; !!!!
1444 00001B38 E871FFFFFF
                                <1>
                                          call test_line_count; 16/01/2016
1445
                                <1>
1446 00001B3D 8A25[225F0000]
                                <1>
                                          mov ah, [CRT_MODE] ; current video mode
                                          ;cmp ah, 4
1447
                                <1>
                                               short n0
1448
                                <1>
                                          ;jb
                                          ;cmp byte [CRT MODE], 4
1449
                                <1>
                                          cmp ah, 4; 0\overline{7}/07/2016
1450 00001B43 80FC04
                                <1>
1451 00001B46 0F83DF070000
                                <1>
                                          jnb
                                                 GRAPHICS_DOWN ; 26/06/2016
                                <1>
                                          ;cmp ah, 7 ; TEST FOR BW CARD
1453
                                <1>
1454
                                <1>
                                          ;jne GRAPHICS_DOWN
                                <1> _n0:
1455
                                          ; 07/07/2016
1456
                                <1>
                                          cmp bh, 7; video page number
1457 00001B4C 80FF07
                                <1>
1458 00001B4F 7606
                                <1>
                                          jna
                                                short n12
1459 00001B51 8A3D[EE580100]
                               <1>
                                               bh, [ACTIVE_PAGE]
                               ;
<1> n12:
1460
1461
                                                      ; CONTINUE_DOWN
                                          mov
1462 00001B57 88DC
                                               ah, bl
                               <1>
                                          push ax ; * ; save attribute in ah
1463 00001B59 6650
                               <1>
1464 00001B5B 6689D0
                                <1>
                                                ax, dx; LOWER RIGHT CORNER
                                          mov
1465 00001B5E E85EFFFFFF
                               <1>
                                          call scroll_position ; GET REGEN LOCATION
1466 00001B63 741F
                               <1>
                                               short n14
1467 00001B65 29CE
                                <1>
                                          sub
                                               esi, ecx ; SI IS FROM ADDRESS
1468 00001B67 88F5
                                               ch, dh ; #rows in block
                               <1>
                                          mov
1469 00001B69 28C5
                                <1>
                                          sub ch, al; #rows to be moved
1470
                                <1> n13:
1471 00001B6B E895FFFFFF
                                <1>
                                          call n10 ; MOVE ONE ROW
                               <1>
1473 00001B70 51
                                          push ecx
                                <1>
1474 00001B71 8A0D[245F0000]
                                <1>
                                               cl, [CRT_COLS]
                                          add cl, cl
1475 00001B77 00C9
                                <1>
                                          sub esi, ecx ; next line
1476 00001B79 29CE
                                <1>
1477 00001B7B 29CF
                                          sub edi, ecx
                                <1>
1478 00001B7D 59
                                <1>
                                           pop ecx
1479
                                <1>
                                          dec ch ; count of li
jnz short n13 ; row loop
                                                     ; count of lines to move
1480 00001B7E FECD
                                <1>
1481 00001B80 75E9
                                <1>
                                <1>
                                          ; ch = 0
1483 00001B82 88C6
                                          mov dh, al ; #rows
                                <1>
1484
                                <1> n14:
1485
                                          ; attribute in ah
                                <1>
                                                            ; fill with blanks
                                          mov al, ''
1486 00001B84 B020
                                <1>
                                 <1> n15:
                                               n11 ; 16/01/2016
1488 00001B86 E887FFFFF
                                <1>
                                          call
                                <1>
1490 00001B8B 8A0D[245F0000]
                                                cl, [CRT_COLS]
                                <1>
                                          mov
1491 00001B91 00C9
                                <1>
                                          add
                                                cl, cl
1492 00001B93 29CF
                                <1>
                                           sub edi, ecx
1493
                                <1>
1494 00001B95 FECE
                                <1>
                                          dec
                                                dh
1495 00001B97 75ED
                                <1>
                                                short n15
                                          jnz
1496
                                <1>
                                 <1>
1497 00001B99 E9FDFEFFFF
                                          qmŗ
                                                n16 ; 27/06/2016
1498
                                <1>
1499
                                <1>;
                                                bh, [ACTIVE_PAGE]
1500
                                <1>;
                                                short n16
                                          jne
1501
                                <1>;
                                                byte [CRT MODE], 7 ; is this the black and white card
1502
                                 <1>;
                                          ; cmp
                                <1>;
1503
                                                                 ; if so, skip the mode reset
                                          ;je
                                                short n16
1504
                                 <1>;
                                                al, [CRT MODE SET] ; get the value of mode set
1505
                                <1>;
                                          mov
1506
                                <1>;
                                          mov
                                                dx, 03D8h; always set color card port
1507
                                 <1>;
                                          out
                                                dx, al
1508
                                <1> ;n16:
                                <1> ;
1509
                                          ; !!!!
                                <1>;
1510
                                                      ; Clear direction flag !
                                          cld
```

```
<1>;
1511
                                     ; !!!!
1512
                               <1>;
1513
                               <1>
1514
                               <1> READ_AC_CURRENT:
1515
                               <1> , 08/07/2016
1516
                               <1>
                                       ; 26/06/2016
                                     ; 12/05/2016
1517
                               <1>
                                     ; 18/01/2016
1518
                               <1>
                                       ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
1519
                               <1>
1520
                               <1>
                                      ; VIDEO.ASM - 06/10/85 VIDEO DISPLAY BIOS
1521
                               <1>
1522
                               <1>
                                     ;
; 08/07/2016
1523
                               <1>
1524 00001B9E 803D[225F0000]07
                              <1>
                                       cmp byte [CRT_MODE], 7; 6!?
                                       jna short read_ac_c
xor eax, eax
1525 00001BA5 7607
                              <1>
1526 00001BA7 31C0
                              <1> xor eax, eax
<1> jmp _video_return
1527 00001BA9 E9B4F9FFFF
                              <1> read_ac_c: <1> call
1528
1529 00001BAE E805000000
                                              _read_ac_current
1530
                              <1>
                                       ; 12/05/2016
                              <1>
1531
                                       ;jmp VIDEO_RETURN
1532 00001BB3 E9AAF9FFFF
                              <1>
                                      jmp _video_return
1533
                              <1>
1534
                               <1> ;-----
                               <1> ; READ AC CURRENT
1535
1536
                               <1>; THIS ROUTINE READS THE ATTRIBUTE AND CHARACTER AT THE CURRENT
1537
                               <1>;
                                        CURSOR POSITION AND RETURNS THEM TO THE CALLER
1538
                               <1> ; INPUT
                               <1>; (AH) = CURRENT CRT MODE
<1>; (BH) = DISPLAY PAGE ( ALPHA MODES ONLY )
1539
1540
                                                                                              :
1541
                               <1>;
                                       (DS) = DATA SEGMENT
                                       (ES) = REGEN SEGMENT
1542
                               <1>;
                               <1>; OUTPUT
1543
1544
                               <1>; (AL) = CHARACTER READ
1545
                               <1>;
                                       (AH) = ATTRIBUTE READ
1546
                               1547
1548
                               <1> _read_ac_current:
                                    ; 26/06/2016
1549
                               <1>
                                       ; 12/05/2016
1550
                               <1>
1551
                               <1>
                                      ; 18/01/2016
1552
                               <1>
1553
                               <1>
                                       ;mov ah, [CRT_MODE] ; current video mode
1554
                               <1>
                                       ;cmp ah, 4
                                       ;jb
                                             short p10
1555
                               <1>
1556 00001BB8 803D[225F0000]04
                               <1>
                                        cmp
                                              byte [CRT_MODE], 4
1557 00001BBF 0F83BB080000
                                             GRAPHICS_READ ; 26/06/2016
                               <1>
                                        jnb
                               <1>
1558
                                        ;cmp ah, 7 ; TEST FOR BW CARD
1559
                               <1>
                                        ;jne GRAPHICS_READ
1560
                               <1>
                               <1> p10:
1561
1562 00001BC5 E801FEFFFF
                               <1>
                                        call find position; GET REGEN LOCATION AND PORT ADDRESS
1563
                              <1>
                                       ; esi = regen location
1564
                              <1>
1565
                              <1>
                                       ; dx = status port
1566
                              <1>
1567 00001BCA 8A25[225F0000]
                              <1>
                                       mov
                                             ah, [CRT_MODE]
                                        sub ah, 2
1568 00001BD0 80EC02
                              <1>
1569 00001BD3 D0EC
                              <1>
                                        shr
                                             ah, 1
1570 00001BD5 7515
                              <1>
                                             short p13
                                        jnz
1571
                              <1>
1572
                                        ; WAIT FOR HORIZONTAL RETRACE OR VERTICAL RETRACE IF COLOR 80
                              <1>
1573
                              <1> p11:
1574 00001BD7 FB
                              <1>
                                        sti
                                                   ; enable interrupts first
1575 00001BD8 3A3D[EE580100]
                              <1>
                                        cmp
                                              bh, [ACTIVE_PAGE]
1576 00001BDE 750C
                              <1>
                                        jne
                                             short p13
1577 00001BE0 FA
                              <1>
                                        cli
                                              ; block interrupts for single loop
1578 00001BE1 EC
                              <1>
                                             al, dx; get status from the adapter
                                       in
1579 00001BE2 A801
                              <1>
                                        test al, RHRZ; is horizontal retrace low
1580 00001BE4 75F1
                                        jnz short pl1 ; wait until it is
                              <1>
                              <1> p12:
1581
                                                   ; wait for either retrace high
1582 00001BE6 EC
                                             al, dx ; get status again
                              <1>
                                        in
1583 00001BE7 A809
                                        test al, RVRT+RHRZ; is horizontal or vertical retrace high
                              <1>
                                              short p12; wait until either retrace active
1584 00001BE9 74FB
                              <1>
                                        jz
1585 00001BEB FB
                              <1>
                                       sti
                              <1> p13:
1586
1587 00001BEC 81C600800B00
                              <1>
                                             esi, 0B8000h
1588 00001BF2 668B06
                               <1>
                                             ax, [esi]
1589
                               <1>
1590 00001BF5 C3
                               <1>
                                        retn ; 18/01/2016
1591
                               <1>
                               <1> WRITE AC CURRENT:
1592
                                      ; 08/07/2016
1593
1594
                               <1>
                                      ; 26/06/2016
                                      ; 24/06/2016
1595
                               <1>
1596
                               <1>
                                       ; 12/05/2016
                                       ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
1597
1598
                               <1>
                                      ; VIDEO.ASM - 06/10/85 VIDEO DISPLAY BIOS
1599
                               <1>
                               <1>
1600
1601
                               <1> ; WRITE AC CURRENT
1602
                                       THTS ROUTINE WRITES THE ATTRIBUTE AND CHARACTER
1603
                               <1>;
                               <1>;
1604
                                        AT THE CURRENT CURSOR POSITION
1605
                               <1> ; INPUT
1606
                               <1>;
                                        (AH) = CURRENT CRT MODE
                                       (BH) = DISPLAY PAGE
1607
                               <1>;
                               <1>;
                                       (CX) = COUNT OF CHARACTERS TO WRITE
1608
1609
                               <1>;
                                        (AL) = CHAR TO WRITE
                               <1>;
                                       (BL) = ATTRIBUTE OF CHAR TO WRITE
1610
                                     (DS) = DATA SEGMENT
(ES) = REGEN SEGMENT
1611
                               <1>;
1612
                               <1>;
                               <1> : OUTPUT
1613
1614
                               <1> ; DISPLAY REGEN BUFFER UPDATED
1615
                               <1> ;------
```

```
1616
                                <1>
1617
                                <1>
                                         ; 08/07/2016
                                       ; 08/0//2010
cmp byte [CRT_MODE], 7 ; 6!?
jna short write_ac_c
1618 00001BF6 803D[225F0000]07
                               <1>
1619 00001BFD 760A
                                <1>
1620
                                <1>
1621 00001BFF E8F20A0000
                               <1>
                                         call vga_write_char_attr
1622 00001C04 E954F9FFFF
                               <1>
                                         jmp VIDEO RETURN
1623
                               <1>
1624
                               <1> write_ac_c:
1625 00001C09 E834000000
                               <1>
                                         call _write_c_current
1626
                               <1>
1627 00001C0E 0FB6F7
                               <1>
                                         movzx esi, bh ; video page number (0 to 7)
1628 00001C11 889E[2B5F0000]
                               <1>
                                         mov [esi+chr_attrib], bl ; color/attribute
1629
                               <1>
1630 00001C17 E941F9FFFF
                               <1>
                                                 VIDEO RETURN
                                         jmp
1631
                                <1>
                                <1> WRITE C CURRENT:
1632
                                <1> ; 08/07/2016
1633
                                      ; 26/06/2016
; 12/05/2016
1634
                                <1>
1635
                                <1>
                                       ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
1636
                                <1>
1637
                                <1>
                                      ; ; VIDEO.ASM - 06/10/85 VIDEO DISPLAY BIOS ;
                                <1>
1638
1639
                                <1>
                                <1> ;------
1640
                                <1> ; WRITE_C_CURRENT
1641
1642
                                <1>; THIS ROUTINE WRITES THE CHARACTER AT
                                         THE CURRENT CURSOR POSITION, ATTRIBUTE UNCHANGED
                                <1>;
1643
1644
                                <1> ; INPUT
                                      (AH) = CURRENT CRT MODE
1645
                                <1> ;
1646
                                <1>;
                                       (BH) = DISPLAY PAGE
                                         (CX) = COUNT OF CHARACTERS TO WRITE
1647
                                <1>;
                                       (AL) = CHAR TO WRITE
                                <1>;
1648
                                       (DS) = DATA SEGMENT
(ES) = REGEN SEGMENT
1649
                                <1>;
1650
                                <1>;
1651
                                <1> ; OUTPUT
                                <1>; DISPLAY REGEN BUFFER UPDATED
1652
1653
                                <1> ;-----
1654
                                <1>
                                     cmp byte [CRT_MODE], 7; 6!? jna short write_c_c
1655
                                         ; 08/07/2016
                                <1>
1656 00001C1C 803D[225F0000]07
                               <1>
1657 00001C23 760A
                                <1>
1658
                                <1>
                               <1> call vga_write_char_only
<1> jmp VIDEO_RETURN
1659 00001C25 E8CC0A0000
1660 00001C2A E92EF9FFFF
                                <1>
1661
                               <1> write_c_c:
1662
                               <1> ; and bh, 7 ; video page number (<= 7)</pre>
1663
movzx esi, bh
                                       mov bl, [esi+chr_attrib]
1666
                               <1>
                                      call _write_c_current
jmp VIDEO_RETURN
1667 00001C38 E805000000
                                <1>
1668 00001C3D E91BF9FFFF
                                <1>
1669
                                <1>
1670
                                <1> _write_c_current: ; from 'write_tty'
                                     ; 26/06/2016
1671
                                <1>
1672
                                <1>
                                         ; 24/06/2016
                                       ; 12/05/2016
1673
                                <1>
                                        ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
; 30/08/2014 (Retro UNIX 386 v1)
1674
                                <1>
1675
                                <1>
                                        ; 18/01/2014
1676
                                <1>
1677
                                <1>
                                         ; 04/12/2013
1678
                                <1>
                                         ; VIDEO.ASM - 06/10/85 VIDEO DISPLAY BIOS
1679
                                <1>
1680
                                <1>
1681
                                <1>
                                         ;mov ah, [CRT_MODE] ; current video mode
1682
                                <1>
                                         ;cmp ah, 4
                                         ;jb short p40
1683
                                <1>
1684 00001C42 803D[225F0000]04
                                <1>
                                               byte [CRT_MODE], 4
                                         cmp
                                         jnb GRAPHICS_WRITE ; 26/06/2016
1685 00001C49 0F8381070000
                                <1>
1686
                                <1>
                                         ;cmp ah, 7 ; TEST FOR BW CARD
1687
                                <1>
                                         ;jne GRAPHICS_WRITE
1688
                                <1>
1689
                                <1> p40:
                                         ; al = character
1690
                                <1>
1691
                                <1>
                                         ; bl = color/attribute
1692
                                <1>
                                         ; bh = video page
1693
                                <1>
                                         ; cx = count of characters to write
                                         push dx
1694 00001C4F 6652
                                <1>
1695 00001C51 88DC
                                <1>
                                         mov ah, bl ; color/attribute (12/05/2016)
1696 00001C53 6650
                                <1>
                                         push ax ; save character & attribute/color
                                         push cx
call find_position ; get regen location and port address
1697 00001C55 6651
                                <1>
1698 00001C57 E86FFDFFFF
                               <1>
1699 00001C5C 6659
                                <1>
1700
                                <1>
                                         ; esi = regen location
1701
                                <1>
                                         ; dx = status port
1702
                               <1>
1703 00001C5E 81C600800B00
                               <1>
                                        add esi, 0B8000h; 30/08/2014 (crt_base)
1704
                                <1>
1705 00001C64 8A25[225F0000]
                               <1>
                                               ah, [CRT_MODE]
                                         mov
1706 00001C6A 80EC02
                               <1>
                                         sub
                                               ah, 2
1707 00001C6D D0EC
                                <1>
                                         shr
                                               ah, 1
1708 00001C6F 7519
                               <1>
                                               short p44 ; 26/06/2016
                                         jnz
1709
                               <1>
                                         ; WAIT FOR HORIZONTAL RETRACE OR VERTICAL RETRACE IF COLOR 80
1710
                                <1>
1711
                                <1> p41:
1712 00001C71 FB
                               <1>
                                                      ; enable interrupts first
1713 00001C72 3A3D[EE580100]
                                                bh, [ACTIVE_PAGE]
                               <1>
                                         cmp
1714 00001C78 7510
                               <1>
                                         jne
                                               short p44
1715 00001C7A FA
                               <1>
                                               ; block interrupts for single loop
                                         cli
1716 00001C7B EC
                                         in al, dx; get status from the adapter
                               <1>
                             <1>
<1>
                                         test al, RVRT; check for vertical retrace first jnz short p43; Do fast write now if vertical retrace
1717 00001C7C A808
1718 00001C7E 7509
                               <1>
1719 00001C80 A801
                                         test al, RHRZ; is horizontal retrace low
1720 00001C82 75ED
                               <1>
                                         jnz short p41; wait until it is
```

```
<1> p42:
                                                      ; wait for either retrace high
1721
1722 00001C84 EC
                                <1>
                                          in
                                                al, dx ; get status again
                                          test al, RVRT+RHRZ; is horizontal or vertical retrace high
1723 00001C85 A809
                                <1>
1724 00001C87 74FB
                                <1>
                                                short p42; wait until either retrace active
1725
                                <1> p43:
1726 00001C89 FB
                                <1>
                                <1> p44:
1727
1728 00001C8A 668B0424
                                                ax, [esp]; restore the character (al) & attribute (ah)
                                <1>
                                          mov
1729 00001C8E 668906
                                <1>
                                                [esi], ax
1730
                                <1>
1731 00001C91 6649
                                <1>
                                          dec
                                                CX
1732 00001C93 7404
                                <1>
                                          jг
                                                short p45
1733
                                <1>
1734 00001C95 46
                                <1>
                                          inc
                                                esi
1735 00001C96 46
                                <1>
                                          inc
                                                esi
1736 00001C97 EBD8
                                <1>
                                          jmp
                                                short p41
                                <1> p45:
1738 00001C99 6658
                                <1>
                                          pop
                                                ax
1739 00001C9B 665A
                                <1>
                                          pop
                                                dx
1740 00001C9D C3
                                <1>
                                          retn
1741
                                <1>
1742
                                <1>; 09/07/2016
1743
                                <1> ; 26/06/2016
1744
                                <1> ; 24/06/2016
1745
                                <1> ; 12/05/2016
                                <1> ; 18/01/2016
1746
1747
                                <1>; 16/01/2016 - TRDOS 386 (TRDOS v2.0)
1748
                                <1>; 30/06/2015
1749
                                <1> ; 27/06/2015
1750
                                <1> ; 11/03/2015
1751
                                <1>; 02/09/2014
1752
                                 <1>; 30/08/2014
                                <1> ; VIDEO FUNCTIONS
1753
1754
                                <1>; (write tty - Retro UNIX 8086 v1 - U9.ASM, 01/02/2014)
1755
                                <1>
                                <1> WRITE_TTY:
1756
1757
                                <1> -; 09/12/2017
                                        ; 09/07/2016
1758
                                <1>
                                        ; 01/07/2016
; 26/06/2016
1759
                                <1>
1760
                                <1>
                                        ; 24/06/2016
1761
                                <1>
                                        ; 13/05/2016
; 12/05/2016
1762
                                <1>
1763
                                <1>
1764
                                <1>
                                        ; 30/01/2016
                                        ; 18/01/2016
1765
                                <1>
                                         ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
1766
                                <1>
                                         ; 13/08/2015
1767
                                <1>
                                        ; 02/09/2014
                                <1>
1768
                                         ; 30/08/2014 (Retro UNIX 386 v1 - beginning)
1769
                                <1>
                                         ; 01/02/2014 (Retro UNIX 8086 v1 - last update)
1770
                                <1>
1771
                                <1>
                                        ; 03/12/2013 (Retro UNIX 8086 v1 - beginning)
1772
                                <1>
                                         ; (Modified registers: EAX, EBX, ECX, EDX, ESI, EDI)
1773
                                <1>
1774
                                <1>
                                         ; INPUT -> AL = Character to be written
1775
                                <1>
                                                   BL = Color (Forecolor, Backcolor)
                                        ;
1776
                                <1>
                                                   BH = Video Page (0 to 7)
1777
                                 <1>
                                          ; 09/07/2016
1778
                                <1>
1779 00001C9E 803D[225F0000]07
                                <1>
                                           cmp byte [CRT_MODE], 7
1780 00001CA5 760A
                                <1>
                                          jna short write_tty_cga
1781
                                <1>
1782 00001CA7 E8290D0000
                                <1>
                                          call vga write teletype
                                          jmp VIDEO_RETURN
1783 00001CAC E9ACF8FFFF
                                <1>
1784
                                <1>
                                <1> write_tty_cga:
1785
1786
                                <1>
                                       ; 13/05/2016
                                         ;call _write_tty
1787
                                <1>
1788
                                         ; 01/07/2016
                                <1>
1789 00001CB1 E818000000
                                <1>
                                          call _write_tty_m3
1790 00001CB6 E9A2F8FFFF
                                <1>
                                                VIDEO RETURN
                                          jmp
1791
                                <1>
                                                           ; VIDEO VERTICAL RETRACE BIT
1792
                                <1> RVRT equ
                                                00001000b
1793
                                <1> RHRZ equ
                                                             ; VIDEO HORIZONTAL RETRACE BIT
                                                00000001b
1794
                                <1>
1795
                                 <1> ; Derived from "WRITE_TTY" procedure of IBM "pc-at" rombios source code
                                <1>; (06/10/1985), 'video.asm', INT 10H, VIDEO_IO
1796
1797
                                <1> ; 06/10/85 VIDEO DISPLAY BIOS
1798
1799
                                 <1>;
1800
                                 1801
                                <1>;
                                 <1>;
                                        THIS INTERFACE PROVIDES A TELETYPE LIKE INTERFACE TO THE
1802
                                        VIDEO CARDS. THE INPUT CHARACTER IS WRITTEN TO THE CURRENT
1803
1804
                                       CURSOR POSITION, AND THE CURSOR IS MOVED TO THE NEXT POSITION.
1805
                                 <1>;
                                        IF THE CURSOR LEAVES THE LAST COLUMN OF THE FIELD, THE COLUMN
1806
                                 <1>;
                                        IS SET TO ZERO, AND THE ROW VALUE IS INCREMENTED. IF THE ROW
                                        ROW VALUE LEAVES THE FIELD, THE CURSOR IS PLACED ON THE LAST ROW,
1807
                                <1>;
                                        FIRST COLUMN, AND THE ENTIRE SCREEN IS SCROLLED UP ONE LINE.
1808
1809
                                <1>;
                                        WHEN THE SCREEN IS SCROLLED UP, THE ATTRIBUTE FOR FILLING THE
1810
                                        NEWLY BLANKED LINE IS READ FROM THE CURSOR POSITION ON THE PREVIOUS
                                        LINE BEFORE THE SCROLL, IN CHARACTER MODE. IN GRAPHICS MODE,
1811
                                 <1>;
                                        THE 0 COLOR IS USED.
1812
                                 <1>;
                                <1>;
                                        ENTRY --
1813
1814
                                 <1>;
                                        (AH) = CURRENT CRT MODE
                                          (AL) = CHARACTER TO BE WRITTEN
1815
                                 <1>;
                                          NOTE THAT BACK SPACE, CARRIAGE RETURN, DELL INC. _ HANDLED AS COMMANDS RATHER THAN AS DISPLAY GRAPHICS CHARACTERS
                                             NOTE THAT BACK SPACE, CARRIAGE RETURN, BELL AND LINE FEED ARE:
1816
                                <1>;
1817
                                 <1>;
                                         (BL) = FOREGROUND COLOR FOR CHAR WRITE IF CURRENTLY IN A GRAPHICS MODE :
1818
                                <1>;
                                <1>;
1819
                                        EXIT --
1820
                                <1>;
                                       ALL REGISTERS SAVED
                                <1> ;------
1821
1822
                                 <1>
                                <1>; 09/12/2017
1823
                                 <1>; 08/07/2016
1824
1825
                                 <1> ; 26/06/2016
```

```
<1> ; 24/06/2016
1826
                                 <1> _write_tty: ; 13/05/2016 <1> cli
1827
1828 00001CBB FA
1829
                                  <1>
1830
                                 <1>
                                           ; 01/09/2014
                                                 byte [CRT MODE], 3
1831 00001CBC 803D[225F0000]03
                                 <1>
                                           cmp
1832 00001CC3 7409
                                 <1>
                                                  short _write_tty_m3
                                           jе
1833
                                 <1>
1834
                                 <1> set_mode_3:
1835 00001CC5 53
                                           push ebx
                                 <1>
1836 00001CC6 50
                                 <1>
                                           push eax
1837 00001CC7 E8A2F8FFFF
                                                 _set_mode
                                 <1>
                                           call
1838 00001CCC 58
                                 <1>
                                           pop
                                                  eax
1839 00001CCD 5B
                                 <1>
                                                 ebx
                                           pop
1840
                                 <1>
                                 <1> _write_tty_m3: ; 24/06/2016 (m3 -> _write_tty_m3)
1841
1842 00001CCE 0FB6F7
                                           movzx esi, bh; 12/05/2016
                                 <1>
                                           shl si, 1
1843 00001CD1 66D1E6
                                 <1>
                                                 esi, CURSOR POSN
1844 00001CD4 81C6[DE580100]
                                 <1>
                                           add
1845 00001CDA 668B16
                                           mov dx, [esi]
                                 <1>
1846
                                 <1>
1847
                                 <1>
                                           ; dx now has the current cursor position
1848
                                 <1>
                                                 al, ODh
1849 00001CDD 3C0D
                                 <1>
                                           cmp
                                                               ; CR ; is it carriage return or control character
1850 00001CDF 7636
                                 <1>
                                                 short u8
                                           jbe
1851
                                 <1>
1852
                                 <1>
                                           ; write the char to the screen
1853
                                 <1> u0:
1854
                                  <1>
                                           ; al = character
1855
                                           ; bl = attribute/color
                                  <1>
1856
                                 <1>
                                           ; bh = video page number (0 to 7)
1857
                                  <1>
                                                 cx, 1 ; 24/06/2016
1858 00001CE1 66B90100
                                 <1>
                                           mov
1859
                                 <1>
                                           ; cx = count of characters to write
1860
                                 <1>
1861 00001CE5 E858FFFFFF
                                 <1>
                                           call _write_c_current; 16/01/2015
1862
                                 <1>
1863
                                 <1>
                                           ; position the cursor for next char
                                                            ; next column
1864 00001CEA FEC2
                                 <1>
                                            inc dl
                                                 dl, [CRT_COLS] ; test for column overflow
1865 00001CEC 3A15[245F0000]
                                 <1>
                                           cmp
                                            jne _set_cpos
mov dl, 0
1866 00001CF2 755D
                                 <1>
                                                             ; column = 0
1867 00001CF4 B200
                                 <1>
                                           mov
                                                               ; (line feed found)
1868
                                 <1> u10:
                                                 dh, 25-1
                                                              ; check for last row
1869 00001CF6 80FE18
                                 <1>
1870 00001CF9 7218
                                 <1>
                                           jb
                                                  short u6
1871
                                 <1>
1872
                                           ; scroll required
                                 <1>
1873
                                 <1> u1:
1874
                                 <1>
                                            ; SET CURSOR POSITION (04/12/2013)
1875 00001CFB E851000000
                                 <1>
                                           call _set_cpos
1876
                                 <1>
1877
                                  <1>
                                           ; determine value to fill with during scroll
1878
                                 <1> u2:
1879
                                 <1>
                                           ; bh = video page number
1880
                                 <1>
1881 00001D00 E8B3FEFFFF
                                 <1>
                                           call _read_ac_current ; 18/01/2016
1882
                                  <1>
1883
                                 <1>
                                           ; al = character, ah = attribute
1884
                                  <1>
                                           ; bh = video page number
1885
                                  <1> u3:
1886
                                  <1>
                                           ;;mov ax, 0601h
                                                             ; scroll one line
                                           ;;sub cx, cx ; upper left corner
;;mov dh, 25-1 ; lower right row
1887
                                  <1>
1888
                                  <1>
1889
                                  <1>
                                           ;;;mov dl, [CRT_COLS]
                                           ;mov dl, 80 - ; lower right column
1890
                                  <1>
1891
                                  <1>
                                           ;;dec dl
1892
                                  <1>
                                           ;;mov dl, 79
1893
                                 <1>
1894
                                  <1>
                                           ;;call scroll_up
                                                             ; 04/12/2013
                                           ;;; 11/03/2015
1895
                                  <1>
1896
                                  <1>
                                           ; 02/09/2014
1897
                                  <1>
                                           ;;;mov cx, [crt_ulc] ; Upper left corner (0000h)
                                           ;;;mov dx, [crt_lrc]; Lower right corner (184Fh)
1898
                                 <1>
                                           ; 11/03/2015
1899
                                 <1>
1900 00001D05 6629C9
                                 <1>
                                           sub
                                                 CX, CX
                                                 dx, 184Fh; dl = 79 (column), dh = 24 (row)
1901 00001D08 66BA4F18
                                 <1>
                                           mov
                                 <1>
1903 00001D0C B001
                                 <1>
                                           mov
                                                 al, 1
                                                             ; scroll 1 line up
1904
                                  <1>
                                                  ; ah = attribute
1905
                                                 bl, al ; 12/05/2016
                                 <1>
                                           ; mov
1906 00001D0E E900FDFFFF
                                                 _scroll_up ; 16/01/2016
                                 <1>
                                           jmp
                                  <1>; u4:
1908
                                  <1>
                                           ;;int 10h
                                                               ; video-call return
1909
                                  <1>
                                                               ; scroll up the screen
1910
                                  <1>
                                                               ; tty return
1911
                                  <1> ;u5:
1912
                                  <1>
                                           ;retn
                                                               ; return to the caller
1913
                                 <1>
                                                               ; set-cursor-inc
1914
                                 <1> u6:
1915 00001D13 FEC6
                                 <1>
                                           inc dh
                                                               ; next row
1916
                                 <1>
                                                               ; set cursor
                                 <1> ;u7:
1917
                                           ;;mov ah, 02h
1918
                                 <1>
                                           ;;jmp short u4
1919
                                 <1>
                                                               ; establish the new cursor
1920
                                 <1>
                                           ;call _set_cpos
1921
                                 <1>
                                           ;jmp short u5
1922 00001D15 EB3A
                                 <1>
                                                   _set_cpos
                                           jmp
1923
                                 <1>
1924
                                 <1>
                                           ; check for control characters
1925
                                 <1> u8:
1926 00001D17 7436
                                 <1>
                                           jе
                                                  short u9
1927 00001D19 3C0A
                                 <1>
                                                 al, OAh
                                                                     ; is it a line feed (OAh)
                                           cmp
1928 00001D1B 74D9
                                                  short u10
                                 <1>
                                           jе
1929 00001D1D 3C07
                                 <1>
                                                 al, 07h
                                                               ; is it a bell
                                           cmp
1930 00001D1F 747A
                                 <1>
                                                  short u11
                                           jе
```

```
1931 00001D21 3C08
                                       cmp al, 08h
                               <1>
                                                                  ; is it a backspace
                                    ; jne short u0
je short bs ; 12/12/2013
; 12/12/2013 (tab stop)
1932
                                <1>
1933 00001D23 7422
                               <1>
1934
                               <1>
1935 00001D25 3C09
                               <1>
                                        cmp al, 09h ; is it a tab stop
1936 00001D27 75B8
                               <1>
                                         jne
                                               short u0
1937 00001D29 88D0
                                         mov al, dl
                               <1>
1938
                               <1>
                                         ;cbw
1939 00001D2B 30E4
                               <1>
                                         xor
                                               ah, ah ; 09/12/2017
                                         mov
1940 00001D2D B108
                               <1>
                                              cl, 8
1941 00001D2F F6F1
                                         div cl
                               <1>
1942 00001D31 28E1
                               <1>
                                         sub cl, ah
                               <1> ts:
1943
1944
                               <1>
                                         ; 02/09/2014
                                         ; 01/09/2014
1945
                               <1>
1946 00001D33 B020
                               <1>
                                        mov
                                              al, 20h
                               <1> tsloop:
1947
1948 00001D35 6651
                               <1> push cx
1949 00001D37 6650
                               <1>
                                         push ax
1950
                               <1>
                                         ;mov bh, [ACTIVE PAGE]
                         <1>
                                         call _write_tty_m3 ; 24/06/2016 (m3 -> _write_tty_m3)
1951 00001D39 E890FFFFF
                                              ax ; ah = attribute/color
1952 00001D3E 6658
                               <1>
                                         pop
1953 00001D40 6659
                               <1>
                                         pop
                                               CX
1954 00001D42 FEC9
                               <1>
                                         dec cl
1955 00001D44 75EF
                               <1>
                                         jnz short tsloop
1956 00001D46 C3
                               <1>
                                         retn
                               <1> bs:
                                         ; back space found
1958
                               <1>
1959
                               <1>
1960 00001D47 08D2
                                         or
                               <1>
                                               dl, dl
                                                               ; is it already at start of line
                                         or dl, dl ; is it ; je short u7 ; set_cursor
1961
                               <1>
1962 00001D49 7406
                               <1>
                                               short _set_cpos
                                         jΖ
1963 00001D4B 664A
                                                                  ; no -- just move it back
                               <1>
                                         dec
                                              dx
1964
                               <1>
                                         jmp short u7;
1965 00001D4D EB02
                               <1>
                                         jmp short _set_cpos
1966
                               <1>
                                         ; carriage return found
1967
                               <1>
1968
                               <1> u9:
                                         mov dl, 0;jmp short u7
                                                          ; move to first column
1969 00001D4F B200
                                <1>
1970
                                <1>
1971
                                <1>
                                         ;jmp short _set_cpos ; 30/01/2016
1972
                                <1>
                                         ; line feed found
1973
                                <1>
1974
                                <1> ;u10:
                                <1>; cmp dh, 25-1 ; bottom of screen
<1>; jne short u6 ; no, just set the cursor
<1>; jmp u1 ; yes, scroll the screen
1975
1976
1977
1978
                                <1>
                                <1> _set_cpos:
1979
                                     -; 12/05/2016 - TRDOS 386 (TRDOS v2.0)
                                <1> -
1980
1981
                                <1>
                                       ; 27/06/2015
                                      ; 01/09/2014
; 30/08/2014 (Retro UNIX 386 v1)
1982
                                <1>
                                <1>
1983
1984
                                <1>
                                       ; 04/12/2013 - 12/12/2013 (Retro UNIX 8086 v1)
1985
                                <1>
1986
                                <1>
                                      ; VIDEO.ASM - 06/10/85 VIDEO DISPLAY BIOS
1987
                                       ;
1988
                                <1>
1989
                                <1> ;-----
1990
                                <1> ; SET_CPOS
1991
                                <1> ; THIS ROUTINE SETS THE CURRENT CURSOR POSITION TO THE
                                         NEW X-Y VALUES PASSED
1992
                                <1>;
1993
                                <1> ; INPUT
                                      DX - ROW, COLUMN OF NEW CURSOR
1994
                                <1>;
1995
                                <1>;
                                         BH - DISPLAY PAGE OF CURSOR
1996
                                <1> ; OUTPUT
1997
                                <1>; CURSOR ID SET AT 6845 IF DISPLAY PAGE IS CURRENT DISPLAY
1998
                                <1> ;
1999
                               <1>
                                         mov esi, CURSOR_POSN
2000 00001D51 BE[DE580100]
                                      movzx eax, bh ; BH = video page number
or al, al
jz short _set_cpos_0
2001 00001D56 0FB6C7
                               <1>
2002
                                <1>;
                               <1> ;
2003
                                       shl al, 1; word offset add esi, eax
                               <1>
2004 00001D59 D0E0
                                                 esi, eax
2005 00001D5B 01C6
                               <1>
2006
                               <1> ;_set_cpos_0:
                               cmp [ACTIVE_PAGE], bh
cl> jne short m17
cl> ;call m10
2007 00001D5D 668916
                               <1> mov [esi], dx ; save the pointer
2008 00001D60 383D[EE580100]
                               <1>
2009 00001D66 7532
                                         ;call m18 ; CURSOR SET
2010
                                <1> ;m17:
2011
                                                    ; SET_CPOS_RETURN
                                     ; 01/09/2014
2012
                                <1>
2013
                                <1>;
2014
                                <1>
                                               ; DX = row/column
2015
                                <1> m18:
2016 00001D68 E84CFCFFFF
                                <1>
                                         call position; determine location in regen buffer
                                         mov
                                              cx, [CRT_START]
2017 00001D6D 668B0D[DC580100]
                               <1>
2018 00001D74 6601C1
                               <1>
                                               cx, ax; add char position in regen buffer
                                         add
2019
                                <1>
                                                    ; to the start address (offset) for this page
                                               cx, 1 ; divide by 2 for char only count
2020 00001D77 66D1E9
                               <1>
2021 00001D7A B40E
                                <1>
                                         mov ah, 14; register number for cursor
                                <1>
2022
                                         ; call m16 ; output value to the 6845
2023
                                <1>
                                         ;retn
2024
                                <1>
                                         ;---- THIS ROUTINE OUTPUTS THE CX REGISTER
2025
                                <1>
2026
                                <1>
                                               TO THE 6845 REGISTERS NAMED IN (AH)
2027
                                <1> m16:
2028 00001D7C FA
                                <1>
                                         cli
2029
                                <1>
                                         ;mov dx, [addr_6845] ; address register
2030 00001D7D 66BAD403
                                               dx, 03D4h ; I/O address of color card
                               <1>
                                         mov
2031 00001D81 88E0
                               <1>
                                         mov
                                               al, ah; get value
2032 00001D83 EE
                               <1>
                                         out
                                               dx, al; register set
                               <1>
2033 00001D84 6642
                                              dx ; data register
$+2 ; i/o delay
                                         inc
2034 00001D86 EB00
                               <1>
                                         jmp
2035 00001D88 88E8
                               <1>
                                         mov
                                               al, ch; data
```

```
2036 00001D8A EE
                                <1>
                                          out
                                                dx, al
2037 00001D8B 664A
                                <1>
                                          dec
                                                dx
2038 00001D8D 88E0
                                <1>
                                          mov
                                                al, ah
2039 00001D8F FEC0
                                <1>
                                          inc al ; point to other data register
2040 00001D91 EE
                                <1>
                                          out
                                                 dx, al; set for second register
2041 00001D92 6642
                                <1>
                                          inc
                                                 dx
2042 00001D94 EB00
                                <1>
                                          jmp
                                                $+2 ; i/o delay
2043 00001D96 88C8
                                <1>
                                                 al, cl; second data value
                                          mov
2044 00001D98 EE
                                <1>
                                           out
                                                 dx, al
2045 00001D99 FB
                                <1>
                                           sti
                                 <1> m17:
2046
2047 00001D9A C3
                                 <1>
                                           retn
2048
                                 <1>
                                 <1> beeper:
2049
                                       ; 04/08/2016
2050
                                 <1>
                                          ; 12/05/2016 - TRDOS 386 (TRDOS v2.0)
2051
                                 <1>
                                         ; 30/08/2014 (Retro UNIX 386 v1)
2052
                                 <1>
                                         ; 18/01/2014
2053
                                 <1>
                                        ; 03/12/2013
2054
                                 <1>
                                          ; bell found
2055
                                 <1>
2056
                                 <1> u11:
                                          sti
2057 00001D9B FB
                                 <1>
2058 00001D9C 3A3D[EE580100]
                                           cmp bh, [ACTIVE_PAGE]
                                 <1>
2059 00001DA2 7551
                                 <1>
                                          jne short u12 ; Do not sound the beep
2060
                                 <1>
                                                              ; if it is not written on the active page
                                 <1> beeper_gfx: ; 04/08/2016
2061
2062 00001DA4 66B93305
                                <1> mov cx, 1331 ; divisor for 896 hz tone
                                                            ; set count for 31/64 second for beep
                                          mov bl, 31
2063 00001DA8 B31F
                                 <1>
                                          2064
                                 <1>
2065
                                 <1>
2066
                                 <1>
                                          ;retn
2067
                                 <1>
                                                 040h ; 8254 TIMER - BASE ADDRESS
equ 061h ; PORT B READ/WRITE DIAGNOSTIC REGISTER
                                 <1> TIMER equ
                                                 040h
2068
2069
                                 <1> PORT B
                                 <1> GATE2 equ
                                                 00000001b ; TIMER 2 INPUT CATE CLOCK BIT 00000010b ; SPEAKER OUTPUT DATA ENABLE BIT
2070
                                 <1> SPK2 equ
2071
                                                00000010b
2072
                                 <1>
                                 <1> beep:
2073
                                       ; 07/02/2015
2074
                                 <1>
                                          ; 30/08/2014 (Retro UNIX 386 v1)
2075
                                 <1>
                                         ; 18/01/2014
2076
                                 <1>
                                         ; 03/12/2013
2077
                                 <1>
2078
                                 <1>
2079
                                 <1>
                                          ; TEST4.ASM - 06/10/85 POST AND BIOS UTILITY ROUTINES
2080
                                 <1>
                                          ; ROUTINE TO SOUND THE BEEPER USING TIMER 2 FOR TONE
2081
                                 <1>
2082
                                 <1>
2083
                                         ; ENTRY:
                                 <1>
                                          ; (BL) = DURATION COUNTER ( 1 FOR 1/64 SECOND )
; (CX) = FREQUENCY DIVISOR (1193180/FREQUENCY) (1331 FOR 886 HZ)
2084
                                 <1>
2085
                                 <1>
                                           ; EXIT:
2086
                                 <1>
                                           ; (AX),(BL),(CX) MODIFIED.
2087
                                 <1>
2088
                                 <1>
2089 00001DAA 9C
                                 <1>
                                           pushf ; 18/01/2014 ; save interrupt status
2090 00001DAB FA
                                 <1>
                                                             ; block interrupts during update
                                           cli
2091 00001DAC B0B6
                                <1>
                                                 al, 10110110b; select timer 2, lsb, msb binary
                                           mov
2092 00001DAE E643
                                <1>
                                           out TIMER+3, al ; write timer mode register
                                           jmp $+2 ; I/O delay
mov al, cl ; divisor for hz (low)
2093 00001DB0 EB00
                                <1>
2094 00001DB2 88C8
                                <1>
                                                TIMER+2,AL ; write timer 2 count - 1sb
2095 00001DB4 E642
                                <1>
                                           out
                                           jmp $+2 ; I/O delay
mov al, ch ; divisor for
2096 00001DB6 EB00
                                <1>
2097 00001DB8 88E8
                                          mov al, ch ; divisor for hz (high) out TIMER+2, al ; write timer 2 count - msb
                                <1>
2098 00001DBA E642
                                <1>
                                                al, PORT_B ; get current setting of port
ah, al ; save that setting
2099 00001DBC E461
                                <1>
                                          mov
2100 00001DBE 88C4
                                <1>
2101 00001DC0 0C03
                                <1>
                                           or
                                                 al, GATE2+SPK2 ; gate timer 2 and turn speaker on
                                           out PORT_B, al ; and restore interrupt status
2102 00001DC2 E661
                                <1>
                                           ; popf ; 18/01/2014
2103
                                 <1>
2104 00001DC4 FB
                                 <1>
2105
                                 <1> g7:
                                                              ; 1/64 second per count (bl)
                                <1>
                                                ecx, 1035
2106 00001DC5 B90B040000
                                           mov
                                                            ; delay count for 1/64 of a second
                                           call waitf
                                                             ; go to beep delay 1/64 count
2107 00001DCA E827000000
                                 <1>
2108 00001DCF FECB
                                           dec
                                <1>
                                                              ; (bl) length count expired?
                                                bl
2109 00001DD1 75F2
                                               short g7 ; no - continue beeping speaker
                                <1>
                                          jnz
2110
                                <1>
                                                              ; save interrupt status
                                          ;pushf
2111
                                <1>
2112 00001DD3 FA
                                <1>
                                          cli ; 18/01/2014 ; block interrupts during update
                                          in al, PORT B ; get current port value
2113 00001DD4 E461
                                <1>
                                          ;or al, not (GATE2+SPK2); isolate current speaker bits in case or al, ~(GATE2+SPK2)
2114
                                 <1>
                                                    al, \sim (GATE2+SPK2)
2115 00001DD6 0CFC
                                <1>
                                          and ah, al ; someone turned them off during beep
2116 00001DD8 20C4
                                 <1>
                                                              ; recover value of port
2117 00001DDA 88E0
                                 <1>
                                           mov
                                                 al, ah
                                                    al, not (GATE2+SPK2) ; force speaker data off
                                 <1>
2119 00001DDC 0CFC
                                 <1>
                                                 al, ~(GATE2+SPK2); isolate current speaker bits in case
2120 00001DDE E661
                                 <1>
                                                PORT_B, al ; and stop speaker timer
                                           out
                                                             ; restore interrupt flag state
2121
                                 <1>
                                           ;popf
2122 00001DE0 FB
                                 <1>
                                           sti
2123 00001DE1 B90B040000
                                 <1>
                                                 ecx, 1035 ; force 1/64 second delay (short)
                                           mov
                                                            ; minimum delay between all beeps
2124 00001DE6 E80B000000
                                 <1>
                                           call
                                                waitf
2125
                                 <1>
                                          ;pushf
                                                              ; save interrupt status
2126 00001DEB FA
                                                             ; block interrupts during update
                                 <1>
                                           cli
2127 00001DEC E461
                                 <1>
                                           in
                                                 al, PORT B
                                                              ; get current port value in case
2128 00001DEE 2403
                                 <1>
                                                al, GATE2+SPK2 ; someone turned them on
                                           and
2129 00001DF0 08E0
                                 <1>
                                                 al, ah ; recover value of port_b
                                           or
                                                             ; restore speaker status
2130 00001DF2 E661
                                 <1>
                                           out
                                                 PORT B, al
2131 00001DF4 9D
                                                              ; restore interrupt flag state
                                 <1>
                                           popf
                                 <1> u12:
2133 00001DF5 C3
                                 <1>
                                           retn
2134
                                 <1>
                                 <1> REFRESH BIT equ 00010000b ; REFRESH TEST BIT
2135
2136
                                 <1>
2137
                                 <1> WAITF:
2138
                                 <1> waitf:
                                      ; 30/08/2014 (Retro UNIX 386 v1)
2139
                                 <1>
2140
                                 <1>
                                          ; 03/12/2013
```

```
<1> ; ; push ax
2142
                                                              ; save work register (ah)
2143
                                   <1> ; waitf1:
                                            ; use timer 1 output bits
in al, PORT_B ; read current counter output status
and al, REFRESH_BIT ; mask for refresh determine bit
2144
                                   <1>
2145
                                   <1>;
2146
                                   <1>;
                                             cmp al, ah ; did it just change
2147
                                   <1>;
                                          je short waitf1 ; wait for a change in output line
                                   <1>;
2148
2149
                                   <1>;
                                            ;
                                          ;
mov ah, al ; save new lflag state
loop waitfl ; decrement half cycles till count end
2150
                                   <1>;
2151
                                   <1> ;
                                          ;
pop ax ; restore (ah)
retn ; return (cx)=0
2152
                                   <1>;
                                   <1>;
2153
2154
                                   <1>;
2155
                                   <1>
                                   <1>; 06/02/2015 (unix386.s <-- dsectrm2.s)
2156
                                   <1>; 17/12/2014 (dsectrm2.s)
2157
2158
                                   <1> ; WAITF
2159
                                   <1>; /// IBM PC-XT Model 286 System BIOS Source Code - Test 4 - 06/10/85 ///
2160
                                   2161
                                   <1>; FIXED TIME WAIT ROUTINE (HARDWARE CONTROLLED - NOT PROCESSOR)
2162
2163
                                   <1> : ENTRY:
                                   <1> ; (CX) = COUNT OF 15.085737 MICROSECOND INTERVALS TO WAIT
<1> ; MEMORY DEEDEST TIMED 1 OUTDIT USED 30 DEEDESTS.
2164
                                             MEMORY REFRESH TIMER 1 OUTPUT USED AS REFERENCE
2165
                                   <1>;
2166
                                   <1> ; EXIT:
                                   <1>; (CX) = 0
2167
                                                         AFTER (CX) TIME COUNT (PLUS OR MINUS 16 MICROSECONDS)
2168
2169
                                   <1> ;-----
2170
2171
                                   <1>; Refresh period: 30 micro seconds (15-80 us)
                                   <1> ; (16/12/2014 - AWARDBIOS 1999 - ATORGS.ASM, WAIT_REFRESH)
2172
2173
                                   <1>
                                            ; DELAY FOR (CX)*15
PUSH AX ; SAVE WORK REGISTER (AH)
; 16/12/2014
2174
                                   <1> ; WAITF:
                                                                             ; DELAY FOR (CX) *15.085737 US
                                  <1> PUSH AX <1> ; 16/12/2014
2175 00001DF6 6650
2176
                                  ;shr cx, 1
                                                                        ; convert to count of 30 micro seconds
2178 00001DF8 D1E9
2179
2180
                                   <1> ; WAITF1:
                                   <1>; IN AL, PORT_B ;061h ; READ CURRENT COUNTER OUTPUT STATUS
                                          AL, FORT_B ; U61n ; READ CURRENT COUNTER OUTPUT STATUS

AND AL, REFRESH_BIT ;00010000b ; MASK FOR REFRESH DETERMINE BIT

CMP AL, AH ; DID IT JUST CHANGE

JE short WAITF1 ; WAIT FOR A CHANGE IN OUTPUT LINE

MOV AH, AL ; SAVE NEW FLAG STATE

LOOP WAITF1 ; DECREMENT HALF CYCLES TILL COUNT END
2181
2182
                                   <1>;
2183
                                   <1>;
2184
                                   <1>;
2185
                                   <1>;
2186
                                   <1>;
2187
                                            ;
                                           ; 17/12/2014
                                   <1>
<1>
2188
2189
                                   <1>
                                            ; Modification from 'WAIT_REFRESH' procedure of AWARD BIOS - 1999
2190
2191
                                   <1>
2192
                                   <1>; WAIT REFRESH: Uses port 61, bit 4 to have CPU speed independent waiting.
                                   <1>; INPUT: CX = number of refresh periods to wait
2193
2194
                                  <1>;
                                                     (refresh periods = 1 per 30 microseconds on most machines)
2195
                                  <1> WR_STATE_0:
                                 <1> IN AL, PORT_B ; IN AL, SYS1 <1> TEST AL, 010H
2196 00001DFA E461
2197 00001DFC A810

<1> TEST AL,010H
<1> JZ SHORT WR_STATE_0
<1> WR_STATE_1:
<1> IN AL,PORT_B
<1> TEST AL,010H
<1> JNZ SHORT WR_STATE_1
<1> LOOP WR_STATE_1
<1> ;
<1> POP AX
<1> RETn
<1>
2198 00001DFE 74FA
2199
2200 00001E00 E461
                                                                       ; IN AL, SYS1
2201 00001E02 A810
2202 00001E04 75FA
2203 00001E06 E2F2
2204
                                                                ; RESTORE (AH)
2205 00001E08 6658
2206 00001E0A C3
                                                                        ; (CX) = 0
2207
                                   <1>
2208
                                   <1>; 09/07/2016
2209
                                   <1>; 01/07/2016
2210
                                   <1> ; 24/06/2016
2211
                                   <1>; 23/06/2016 - TRDOS 386 (TRDOS v2.0)
                                   2212
2213
                                   <1>; WRITE STRING
2214
2215
                                   <1>; THIS ROUTINE WRITES A STRING OF CHARACTERS TO THE CRT.
2216
                                   <1> ; INPUT
2217
                                   <1>;
                                            (AL) = WRITE STRING COMMAND 0 - 3
                                   <1>;
                                             (BH) = DISPLAY PAGE (ACTIVE PAGE)
2218
2219
                                             (CX) = COUNT OF CHARACTERS TO WRITE, IF (CX) = 0 THEN RETURN
                                   <1>;
2220
                                             (DX) = CURSOR POSITION FOR START OF STRING WRITE
                                   <1>;
                                             (BL) = ATTRIBUTE OF CHARACTER TO WRITE IF (AL) = 0 OR (AL) = 1
2221
                                   <1>;
                                             (eBP) = SOURCE STRING OFFSET
2222
                                   <1>;
                                   <1> ; OUTPUT
2223
2224
2225
                                   <1> ;-----
2226
                                   <1>
2227
                                   <1>; AL = 00h: Assign all characters the attribute in BL; do not update cursor
2228
                                   <1>; AL = 01h: Assign all characters the attribute in BL; update cursor
                                   <1> ; AL = 02h: Use attributes in string; do not update cursor
2229
2230
                                   <1> ; AL = 03h: Use attributes in string; update cursor
2231
                                   <1>
2232
                                   <1> WRITE STRING:
                                           ; 12/09/2016
2233
                                   <1>
                                             ; 09/07/2016
2234
                                   <1>
2235
                                   <1>
                                            ;cmp byte [CRT MODE], 7 ; 6?!
                                                                   ; not a valid function for VGA modes
2236
                                   <1>
                                             ;ja
                                                   VIDEO_RETURN
2237
                                   <1>
2238 00001E0B A2[54650100]
                                   <1>
                                             mov
                                                    [w_str_cmd], al
                                                                              ; save (AL) command
                                                                        ; TEST FOR INVALID WRITE STRING OPTION
2239 00001E10 3C04
                                   <1>
                                             CMP
                                                    AL, 4
                                                   VIDEO_RETURN
                                                                        ; IF OPTION INVALID THEN RETURN
2240 00001E12 0F8345F7FFFF
                                  <1>
                                             JNB
2241
                                  <1>
                                             ; JCXZ VIDEO RETURN
2242
                                   <1>
                                                                              ; IF ZERO LENGTH STRING THEN RETURN
2243
                                   <1>
                                                                      ; 01/07/2016
2244 00001E18 67E35E
                                   <1>
                                                       P55
2245
                                   <1>
```

2141

```
2246
                                 <1>
2247
                                 <1>
                                          ; 01/07/2016
2248
                                 <1>
                                          ;and ecx, OFFFFh
                                        ; ECX = byte count
2249
                                <1>

<1> ;push ecx
<1> mov esi, ebp; user buffer
<1> mov edi, Cluster_Buffer; system buffer
<1> call transfer_from_user_buffer
<1> ;pon ecx
2250
2251 00001E1B 89EE
2252 00001E1D BF00000700
                                      ;pop ecx
jc VIDEO_RETURN
; ecx = transfer (byte) count = character count
mov ebp, Cluster_Buffer
. 10/00/2017
2253 00001E22 E820CA0000
2254
                                <1>
2255 00001E27 0F8230F7FFFF
                                <1>
2256
                                <1>
2257 00001E2D BD00000700
                                 <1>
2258
                                <1>
                                          ; 12/09/2016
                                        cmp byte [CRT MODE], 7; 6?!
2259 00001E32 803D[225F0000]07 <1>
2260 00001E39 0F879F000000
                                <1>
                                        jа
                                               vga_write_string
2261
                                 <1>
2262 00001E3F 0FB6F7
                                                                          ; GET CURRENT CURSOR PAGE
                                 <1>
                                          movzx esi, bh
                                          SAL SI,1
: ****
                                                                  ; CONVERT TO PAGE OFFSET (SI= PAGE)
2263 00001E42 66D1E6
                                <1>
                                          ; ****
2264
                                 <1>
2265 00001E45 66FFB6[DE580100]
                                <1>
                                          PUSH word [eSI+CURSOR_POSN] ; SAVE CURRENT CURSOR POSITION IN STACK
2266
                                 <1>
                                                                  ; SET NEW CURSOR POSITION
2267
                                 <1>
                                          ;MOV AX,0200H
2268
                                <1>
                                          ;INT
                                                10H
2269
                                <1> P50next:
                                <1> push ebx; ****
2270 00001E4C 53
                                          push ecx; ***
2271 00001E4D 51
                                <1>
                                          push esi; **
2272 00001E4E 56
                               <1>
                         2273 00001E4F 52
2274 00001E50 E8FCFEFFFF
                                          call _set_cpos
2275
2276 00001E55 8A4500
                               <1> MOV
                                                AL, [eBP] ; GET CHARACTER FROM INPUT STRING
2277 00001E58 45
                                <1>
                                          INC
                                                eBP
                                                                    ; BUMP POINTER TO CHARACTER
2278
                                <1>
2279
                                <1> ;----
                                                 TEST FOR SPECIAL CHARACTER'S
2280
                                <1>
                           2281 00001E59 3C08
                                          CMP
                                                AL, 08H
                                                                           ; IS IT A BACKSPACE
                                               short P51 ; BACK_SPACE
AL, ODh; CR ; IS IT CARRIAGE RETURN
short P51 ; CAR_RET
AL, OAh; LF ; IS IT A LINE FEED
short P51 ; LINE_FEED
AL, O7h ; IS IT A BELL
2282 00001E5B 740C
                                          JE
2283 00001E5D 3C0D
                                          CMP
2284 00001E5F 7408
                                          JE
2285 00001E61 3C0A
                                          CMP
2286 00001E63 7404
                                          JE
2287 00001E65 3C07
                                <1>
                                          CMP
                                <1>
                                                short P52
                                                                  ; IF NOT THEN DO WRITE CHARACTER
2288 00001E67 7515
                                          JNE
2289
                                <1> P51:
                                          ;MOV AH,0EH ; TTY_CHARACTER_WRITE
;INT 10H ; WRITE TTY_CHARACTER
2290
                                <1>
2291
                                <1>
                                          ;INT
                                                10H
                                                                    ; WRITE TTY CHARACTER TO THE CRT
                                <1>
2293 00001E69 E860FEFFFF
                                          call
                                <1>
                                                 _write_tty_m3
2294
                                <1>
2295 00001E6E 5A
                                <1>
                                                 edx ; *
                                          pop
                                                 esi ; **
2296 00001E6F 5E
                                <1>
                                          pop
2297
                                 <1>
2298 00001E70 668B96[DE580100]
                                                 DX, [eSI+CURSOR_POSN] ; GET CURRENT CURSOR POSITION
                               <1>
                                          MOV
2299 00001E77 EB46
                                <1>
                                          JMP
                                                 SHORT P54 ; SET CURSOR POSITION AND CONTINUE
2300
                                <1> P55:
2301 00001E79 E9DFF6FFFF
                                <1>
                                          JMP
                                                 VIDEO_RETURN
                                <1> P52:
2303 00001E7E 66B90100
                                                 CX, 1
                                          MOV
                                                                   ; SET CHARACTER WRITE AMOUNT TO ONE
                                <1>
2304 00001E82 803D[54650100]02
                                <1>
                                           CMP
                                                 byte [w_str_cmd], 2; IS THE ATTRIBUTE IN THE STRING
2305 00001E89 7204
                                                 short P53 ; IF NOT THEN SKIP
                                <1>
                                          JB
2306 00001E8B 8A5D00
                                <1>
                                          MOV
                                                 BL, [eBP]
                                                                   ; ELSE GET NEW ATTRIBUTE
2307 00001E8E 45
                                <1>
                                          INC
                                                 eBP
                                                                   ; BUMP STRING POINTER
                                <1> P53:
2308
                                          ; MOV AH, 09H
2309
                                <1>
                                                                  ; GOT CHARACTER
                                          ;INT
2310
                                 <1>
                                                                    ; WRITE CHARACTER TO THE CRT
2311
                                <1>
2312 00001E8F E8AEFDFFFF
                                <1>
                                          call
                                                _write_c_current
2313
                                <1>
2314 00001E94 5A
                                <1>
                                                 edx ; *
                                          pop
2315
                                <1>
2316 00001E95 0FB6F7
                                          movzx esi, bh; video page number (0 to 7)
                                <1>
2317 00001E98 889E[2B5F0000]
                                <1>
                                                [esi+chr_attrib], bl ; color/attribute
                                          mov
2318
                                <1>
                                                                  ; INCREMENT COLUMN COUNTER
                                          INC DL
2319 00001E9E FEC2
                                <1>
2320 00001EA0 3A15[245F0000]
                                <1>
                                          CMP
                                                DL, [CRT_COLS]
                                                                         ; IF COLS ARE WITHIN RANGE FOR THIS MODE
                                                                ; IF COLS ARE WITHIN ; THEN GO TO COLUMNS SET
2321 00001EA6 7217
                                                 short P54
                                <1>
                                          JB
2322 00001EA8 FEC6
                                <1>
                                          INC
                                                DH
                                                                   ; BUMP ROW COUNTER BY ONE
                                          SUB DL, DL
CMP DH, 25
JB short P54
                                                                  ; SET COLUMN COUNTER TO ZERO
                                <1>
2323 00001EAA 28D2
2324 00001EAC 80FE19
                                <1>
                                                                   ; IF ROWS ARE LESS THAN 25 THEN
2325 00001EAF 720E
                                <1>
                                                                   ; GO TO ROWS_COLUMNS_SET
2326
                                <1>
2327 00001EB1 66B80A0E
                                                 AX,0E0AH
                                                             ; ELSE SCROLL SCREEN
                                 <1>
                                          MOV
2328
                                 <1>
                                          ;INT
                                                10H
                                                                    ; RESET ROW COUNTER TO 24
2329
                                 <1>
2330 00001EB5 E814FEFFFF
                                 <1>
                                          call _write_tty_m3
2331
                                <1>
2332 00001EBA 66BA0018
                                <1>
                                                dx, 1800h
                                                                    ; Column = 0, Row = 24
                                          mov
2333 00001EBE 5E
                                <1>
                                          pop esi; **
2334
                                <1> P54:
                                                                   ; ROW_COLUMNS SET
2335
                                 <1>
                                           ;MOV AX,0200H
                                                                  ; SET NEW CURSOR POSITION COMMAND
2336
                                 <1>
                                                                    ; ESTABLISH NEW CURSOR POSITION
2337
                                 <1>
                                          ;INT
                                                10H
2338
                                <1>
2339 00001EBF 59
                                <1>
                                                 ecx ; ***
2340 00001EC0 5B
                                                ebx ; ****
                                <1>
                                          pop
2341
                                <1>
                                          ;LOOP P50
                                                                    ; DO IT ONCE MORE UNTIL (CX) = ZERO
                                <1>
2343 00001EC1 6649
                                <1>
                                          dec
                                                CX
2344 00001EC3 7587
                                 <1>
                                          jnz
                                                short P50next
2345
                                <1>
                                                 DX ; ****
                                                                   ; RESTORE OLD CURSOR COORDINATES
2346 00001EC5 665A
                                 <1>
                                          POP
2347
                                 <1>
2348 00001EC7 F605[54650100]01
                                <1>
                                          test byte [w_str_cmd], 1; IF CURSOR WAS NOT TO BE MOVED
2349 00001ECE 0F8589F6FFFF
                                 <1>
                                                VIDEO_RETURN ; THEN EXIT WITHOUT RESETTING OLD VALUE
2350
                                 <1>
```

```
2351
                                                                      ; ELSE RESTORE OLD CURSOR POSITION
                                  <1>
                                           ;MOV AX,0200H
2352
                                  <1>
                                           ;INT 10H
2353
                                                                      ; DONE - EXIT WRITE STRING
                                  <1>
2354 00001ED4 E878FEFFFF
                                  <1>
                                            call
                                                   set cpos
2355 00001ED9 E97FF6FFFF
                                  <1>
                                            JMP
                                                                      ; RETURN TO CALLER
                                                  VIDEO RETURN
2356
                                  <1>
2357
                                  <1> vga_write_string:
                                          ; 1\overline{2}/09/2016 - TRDOS 386 (TRDOS v2.0)
2358
                                  <1>
2359
                                  <1>
2360
                                  <1>
                                           ; derived from 'Plex86/Bochs VGABios' source code
2361
                                  <1>
                                           ; vgabios-0.7a (2011)
2362
                                  <1>
                                           ; by the LGPL VGABios developers Team (2001-2008)
                                           ; 'vgabios.c', ' biosfn_write_string'
2363
                                  <1>
2364
                                  <1>
2365
                                  <1>
                                           ; INPUT
2366
                                  <1>
                                                  (AL) = WRITE STRING COMMAND 0 - 3
                                                  (BH) = DISPLAY PAGE (ACTIVE PAGE)
2367
                                                  (CX) = COUNT OF CHARACTERS TO WRITE, IF (CX) = 0 THEN RETURN
2368
                                  <1>
                                                  (DX) = CURSOR POSITION FOR START OF STRING WRITE
2369
                                  <1>
2370
                                  <1>
                                                  (BL) = ATTRIBUTE OF CHARACTER TO WRITE IF (AL) = 0 OR
                                                                                                              (AL) = 1 :
2371
                                  <1>
                                                  (eBP) = SOURCE STRING OFFSET
                                           ; OUTPUT
2372
                                  <1>
2373
                                  <1>
                                            ;
                                                 NONE
2374
                                  <1>
2375
                                  <1>
2376
                                  <1>
                                           ; AL = 00h: Assign all characters the attribute in BL; do not update cursor
2377
                                  <1>
                                           ; AL = 01h: Assign all characters the attribute in BL; update cursor
2378
                                  <1>
                                           ; AL = 02h: Use attributes in string; do not update cursor
2379
                                  <1>
                                           ; AL = 03h: Use attributes in string; update cursor
2380
                                  <1>
2381
                                  <1>
                                            ; biosfn_write_string(GET_AL(),GET_BH(),GET_BL(),CX,GET_DH(),GET_DL(),ES,BP);
2382
                                  <1>
                                            ; static void biosfn_write_string (flag,page,attr,count,row,col,seg,offset)
2383
                                  <1>
2384
                                  <1>
                                           ; // Read curs info for the page
                                           ; biosfn_get_cursor_pos(page, &dummy, &oldcurs);
2385
                                  <1>
2386
                                  <1>
                                            ; bh = video page = 0
2387
                                  <1>
                                           ;movzx esi, word [CURSOR_POSN] ; current cursor position for video page 0
2388
                                  <1>
2389
                                  <1>
                                           ; // if row=0xff special case : use current cursor position
2390
                                  <1>
                                           ; if(row==0xff)
2391
                                  <1>
                                           ; {col=oldcurs&0x00ff;
2392
                                  <1>
                                               row=(oldcurs&0xff00)>>8;
                                           ;
2393
                                  <1>
                                            ; }
2394
                                  <1>
2395
                                  <1>
                                           ;mov al, [w_str_cmd]
2396
                                  <1>
2397 00001EDE 80FEFF
                                  <1>
                                                  dh, OFFh
                                            cmp
2398 00001EE1 7407
                                 <1>
                                                  short vga_wstr_1 ; user current cursor position
                                            jе
2399
                                  <1> vga_wstr_0:
2400
                                 <1>
                                           ; set cursor position
2401 00001EE3 668915[DE580100]
                                  <1>
                                            mov
                                                 [CURSOR_POSN], dx ; save cursor pos for pg 0
2402
                                  <1> vga wstr 1:
2403 00001EEA 66FF35[DE580100]
                                  <1>
                                           push word [CURSOR_POSN] ; *
2404
                                  <1>
2405
                                  <1>
                                           ; ebp = string offset in system buffer (user buffer was copied to)
2406
                                  <1>
2407
                                  <1>
                                           ; while (count--!=0)
2408
                                  <1>
                                           ; {
2409
                                  <1>
                                               car=read_byte(seg,offset++);
                                            ; if ((flag \& 0 \times 02)! = 0)
2410
                                  <1>
2411
                                  <1>
                                               attr=read_byte(seg,offset++);
2412
                                  <1>
                                                biosfn_write_teletype(car,page,attr,WITH_ATTR);
                                           ;
2413
                                  <1>
                                           ; }
2414
                                  <1>
2415
                                  <1>
                                           ;push eax ; **
2416
                                  <1>
                                            ;test al, 2
2417 00001EF1 F605[54650100]02
                                  <1>
                                            test byte [w_str_cmd], 2
2418 00001EF8 751D
                                 <1>
                                            jnz
                                                 short vga_wstr_3
2419 00001EFA 881D[EF580100]
                                 <1>
                                                  [ccolor], bl
2420
                                 <1> vga_wstr_2:
2421 00001F00 51
                                 <1>
2422 00001F01 8A4500
                                 <1>
                                           mov
                                                  al, [ebp]
2423 00001F04 E8CC0A0000
                                                  vga write teletype
                                 <1>
                                           call
2424 00001F09 59
                                 <1>
                                           pop
                                                  ecx
2425 00001F0A 6649
                                 <1>
                                           dec
                                                  CX
2426 00001F0C 741E
                                 <1>
                                            jz
                                                  short vga_wstr_4
2427 00001F0E 45
                                 <1>
                                                  ebp
                                           inc
2428 00001F0F 8A1D[EF580100]
                                 <1>
                                           mov
                                                  bl, [ccolor]
2429 00001F15 EBE9
                                 <1>
                                                  short vga wstr 2
                                            jmp
2430
                                 <1> vga_wstr_3:
2431 00001F17 51
                                  <1>
                                           push ecx
                                                  al, [ebp]
2432 00001F18 8A4500
                                  <1>
                                            mov
2433 00001F1B 45
                                 <1>
                                            inc
                                                  ebp
2434 00001F1C 8A5D00
                                  <1>
                                                  bl, [ebp]
2435 00001F1F E8B10A0000
                                 <1>
                                            call
                                                  vga write teletype
2436 00001F24 59
                                 <1>
                                                  ecx
                                            pop
2437 00001F25 6649
                                 <1>
                                            dec
                                                  CX
2438 00001F27 7403
                                 <1>
                                                  short vga wstr 4
                                            jz
2439 00001F29 45
                                 <1>
                                            inc
                                                  ebp
2440 00001F2A EBEB
                                 <1>
                                            jmp
                                                  short vga_wstr_3
                                  <1> vga_wstr_4:
2441
2442
                                  <1>
                                           ; // Set back curs pos
                                            ; if((flag&0x01)==0)
2443
                                  <1>
2444
                                  <1>
                                           ; biosfn_set_cursor_pos(page,oldcurs);
2445
                                  <1>
                                           ; }
                                                 eax ; **
2446
                                  <1>
                                           ;pop
                                           pop dx; word [CURSOR POSN]; *
2447 00001F2C 665A
                                  <1>
2448
                                  <1>
                                           ;test al, 1
2449 00001F2E F605[54650100]01
                                  <1>
                                            test byte [w_str_cmd], 1
2450 00001F35 0F8522F6FFFF
                                  <1>
                                                  VIDEO RETURN
                                            jnz
2451 00001F3B 668915[DE580100]
                                  <1>
                                           mov
                                                 [CURSOR_POSN], dx
2452 00001F42 E916F6FFFF
                                  <1>
                                           JMP
                                                 VIDEO RETURN
2453
                                  <1>
2454
                                  <1> ; 07/07/2016
2455
                                  <1> ; 27/06/2016 - TRDOS 386 (TRDOS v2.0)
```

```
2456
                                <1>; VIDEO1.ASM - 24/03/1985 (IBM PC-AT BIOS source code)
 2457
                                <1> ; SCROLL UP
 2458
 2459
                                <1> ; THIS ROUTINE SCROLLS UP THE INFORMATION ON THE CRT
2460
                                <1> ; ENTRY ---
2461
                                <1> ; CH,CL = UPPER LEFT CORNER OF REGION TO SCROLL
                                <1> ; DH, DL = LOWER RIGHT CORNER OF REGION TO SCROLL
 2462
                                <1>; BOTH OF THE ABOVE ARE IN CHARACTER POSITIONS
2463
2464
                                <1> ; BH = FILL VALUE FOR BLANKED LINES
                                <1>; AL = # LINES TO SCROLL (AL=0 MEANS BLANK THE ENTIRE FIELD)
2465
2466
                                <1> ; DS = DATA SEGMENT
                                <1> ; ES = REGEN SEGMENT
 2467
                                <1> ; EXIT --
2468
2469
                                <1> ; NOTHING, THE SCREEN IS SCROLLED
2470
                                <1> ;------
2471
                                <1>
                                       ; cl = upper left column
 2472
                                     ; ch = upper left row
; dl = lower rigth column
; dh = lower right row
2473
                                <1>
2474
                                <1>
2475
                               <1>
2476
                               <1>
                                    ; al = line count (AL=0 means blank entire fields)
; bl = fill value for blanked lines
; bh = unused
 2477
                                <1>
2478
                               <1>
 2479
                               <1>
2480
                               <1>
                               <1> GRAPHICS_UP:
 2481
                               <1> ; 07/07/2016
 2482
                        2483
 2484 00001F47 80FC07
 2485 00001F4A 7766
 2486
 2487
                                              bh, al ; save line count in BH
AX, CX ; GET UPPER LEFT POSITION INTO AX REG
 2488 00001F4C 88C7
 2489 00001F4E 6689C8
 2490
                               <1>
                               <1> ;----
2491
                                              USE CHARACTER SUBROUTINE FOR POSITIONING
                               <1> ;----
                                              ADDRESS RETURNED IS MULTIPLIED BY 2 FROM CORRECT VALUE
2492
2493
                               <1>
                                        CALL GRAPH POSN
 2494 00001F51 E8D9050000
                               <1>
 2495 00001F56 0FB7F8
                                      MOVzx eDI, ĀX
                              <1>
                                                                     ; SAVE RESULT AS DESTINATION ADDRESS
2496
                               <1>
                               <1> ;---- DETERMINE SIZE OF WINDOW
 2497
                        <1>
 2498
 2499 00001F59 6629CA
 2500 00001F5C 6681C20101
 2501 00001F61 C0E602
 2502
                                            DETERMINE CRT MODE
                               <1> ;----
 2503
 2504
                               <1>
                                    CMP byte [CRT_MODE], 6 ; TEST FOR MEDIUM RES

JNC short R7_ ; FIND_SOURCE
 2505 00001F64 803D[225F0000]06 <1>
 2506 00001F6B 7305
                              <1>
 2507
                              <1>
<1> ;----
                         <1>;---- MEDION 132
<1> SAL DL, 1
<1> SAL DI, 1
2508
                                                              ; # COLUMNS * 2, SINCE 2 BYTES/CHAR
; OFFSET *2 SINCE 2 BYTES/CHAR
 2509 00001F6D D0E2
 2510 00001F6F 66D1E7
2511
2524
                               <1> ;----
                                              LOOP THROUGH, MOVING ONE ROW AT A TIME, BOTH EVEN AND ODD FIELDS
                        ; ROW LOOP
2525
                                                                    ; MOVE ONE ROW
 2526 00001F8A E812040000
                                                              ; MOVE TO NEXT ROW
 2527 00001F8F 6681EEB01F
 2528 00001F94 6681EFB01F
                                                             ; NUMBER OF ROWS TO MOVE
 2529 00001F99 FECC
 2530 00001F9B 75ED
                                                                ; CONTINUE TILL ALL MOVED
                               <1>
2531
                               <1> ;---- FILL IN THE VACATED LINE(S)
2532
 2533
                               <1> _R9:
                                                                       ; CLEAR ENTRY
 2534 00001F9D 88D8
                               <1>
                                    mov al, bl
                                                                ; attribute to fill with
                              <1> R10:

<1> CALL R18

<1> SUB DI, 2000h-80

<1> dec bh
2535
                                                                      ; CLEAR THAT ROW
 2536 00001F9F E819040000
                                                           ; POINT TO NEXT LINE
 2537 00001FA4 6681EFB01F
 2538 00001FA9 FECF
                                                                ; number of lines to fill
 2539 00001FAB 75F2
                               <1>
                                         JNZ
                                                  short _R10_
                                                                 ; CLEAR LOOP
 2540 00001FAD C3
                                                                ; EVERYYHING DONE
                               <1>
                                        retn
 2541
                               <1>
                                                                        ; BLANK FIELD
 2542
                               <1> R11:
 2543 00001FAE 88F7
                               <1>
                                        mov bh, dh
                                                                ; set blank count to everything in field
 2544 00001FB0 EBEB
                               <1>
                                         JMP short R9
                                                                 ; CLEAR THE FIELD
 2545
                               <1>
2546
                               <1> vga_graphics_up:
                                      ; 08/08/2016
 2547
                                <1>
                                        ; 07/08/2016
 2548
                               <1>
 2549
                                <1>
                                        ; 04/08/2016
                                       ; 01/08/2016
 2550
                                <1>
2551
                                <1>
                                        ; 31/07/2016
 2552
                                <1>
                                        ; 07/07/2016 - TRDOS 386 (TRDOS v2.0)
 2553
                                <1>
 2554
                                <1>
                                        ; derived from 'Plex86/Bochs VGABios' source code
                                        ; vgabios-0.7a (2011)
2555
                                <1>
                                        ; by the LGPL VGABios developers Team (2001-2008)
2556
                                <1>
 2557
                                <1>
                                        ; 'vgabios.c', 'biosfn scroll'
 2558
                                <1>
 2559
                                <1>
2560
                                <1>
                                        ; cl = upper left column
```

```
2561
                                                 <1>
                                                             ; ch = upper left row
2562
                                                  <1>
                                                               ; dl = lower rigth column
                                                               ; dh = lower right row
2563
                                                  <1>
2564
                                                  <1>
                                                  <1>
                                                               ; al = line count (AL=0 means blank entire fields)
2565
                                                                ; bl = fill value for blanked lines
2566
                                                 <1>
                                                               ; bh = unused
2567
                                                 <1>
                                                 <1>
2568
2569
                                                 <1>
                                                               ; ah = [CRT_MODE], current video mode
2570
                                                 <1>
2571 00001FB2 88C7
                                                <1>
                                                               mov bh, al; 31/07/2016
2572 00001FB4 BE[465F0000]
                                                                        esi, vga_g_modes
                                                 <1>
                                                               mov
                                                               mov
2573 00001FB9 89F7
                                                <1>
                                                                        edi, esi
2574 00001FBB 83C708
                                                <1>
                                                                add edi, vga_g_mode_count
2575
                                                <1> vga_g_up_0:
2576 00001FBE AC
                                                <1>
                                                               lodsb
2577 00001FBF 38E0
                                                               cmp al, ah; [CRT MODE]
                                               <1>
                                               <1>
                                                                        short vga_g_up_1
2578 00001FC1 7405
                                                               je
                                                               cmp esi, edi
jb short vga_g_up_0
2579 00001FC3 39FE
                                                <1>
2580 00001FC5 72F7
                                               <1>
                                                        ; xor bh, bh; \overline{31/07/2016}) retn; nothing to do
2581
                                                <1>
2582 00001FC7 C3
                                                <1>
                                                <1> vga_g_up_1:
2583
2584 00001FC8 88F8
                                                <1> mov al, bh; 31/07/2016
2585 00001FCA 83C64F
                                                               add esi, vga_g_memmodel - (vga_g_modes + 1)
                                                <1>
2586
                                                 <1>
                                                               ; [ESI] = VGA memory model number (LINEAR8, PLANAR4, PLANAR1)
2587
                                                 <1>
2588
                                                 <1>
                                                             ; if(rlr>=nbrows)rlr=nbrows-1;
2589
                                                 <1>
                                                               ; if(clr>=nbcols)clr=nbcols-1;
2590
                                                 <1>
                                                               ; if (nblines>nbrows) nblines=0;
2591
                                                 <1>
                                                             ; cols=clr-cul+1;
2592
                                                  <1>
2593 00001FCD 3A35[2A5F0000]
                                                 <1>
                                                              cmp dh, [VGA_ROWS]
2594 00001FD3 7208
                                                 <1>
                                                            jb
                                                                         short vga_g_up_2
2595 00001FD5 8A35[2A5F0000]
                                                <1>
                                                         mov
dec
                                                                         dh, [VGA_ROWS]
2596 00001FDB FECE
                                                 <1>
                                                 <1> vga_g_up_2:
2598 00001FDD 3A15[245F0000]
                                                <1> cmp dl, [CRT_COLS] ; = [VGA_COLS]
2599 00001FE3 7208
                                                <1>
                                                                jb
                                                                          short vga_g_up_3
                                               <1> mov dl, <1> dec dl
                                                                         dl, [CRT_COLS]
2600 00001FE5 8A15[245F0000]
2601 00001FEB FECA
                                                <1> vga_g_up_3:
2602
                                               <1>
2603 00001FED 3A05[2A5F0000]
                                                               cmp al, [VGA_ROWS]
                                                                        short vga_g_up_4
2604 00001FF3 7602
                                                <1>
2605 00001FF5 28C0
                                                <1>
                                                               sub
                                                                         al, al ; \overline{0}
                                                <1> vga_g_up_4:
2606
                                               2607 00001FF7 88D7
2608 00001FF9 28CF
                                                <1>
                                                               sub
                                                                        bh, cl ; cul
2609 00001FFB FEC7
                                                <1>
                                                                        bh ; cols = clr-cul+1
                                                <1>
                                            and al, al; nblines = 0
al, al; nblines =
2611 00001FFD 20C0
2612 00001FFF 755D
2613 00002001 20ED
2614 00002003 7559
2615 00002005 20C9
                                                             jnz
2616 00002007 7555
                                                <1>
                                                                        short vga_g_up_6
                                                <1>
2618 00002009 6650
                                                               push ax
                                                <1>
2619 0000200B A0[2A5F0000]
                                                <1>
                                                               mov
                                                                         al, [VGA_ROWS]
2620 00002010 FEC8
                                                <1>
                                                               dec
                                                                        al
2621 00002012 38C6
                                                <1>
                                                                cmp dh, al ; rlr = nbrows-1
2622 00002014 7546
                                                <1>
                                                                jne
                                                                        short vga_g_up_5
2623 00002016 A0[245F0000]
                                                <1>
                                                                mov al, [CRT_COLS] ; = VGA_COLS
                                                                dec al
2624 0000201B FEC8
                                                <1>
2625 0000201D 38C2
                                                <1>
                                                                        dl, al ; clr = nbcols-1
                                                                cmp
2626 0000201F 753B
                                                <1>
                                                                jne
                                                                         short vga_g_up_5
                                                               pop ax
2627 00002021 6658
                                                <1>
2628
                                                <1>
2629 00002023 66B80502
                                                <1>
                                                                        ax, 0205h
                                                               mov
2630 00002027 66BACE03
                                                <1>
                                                                        dx, 3CEh ; VGAREG_GRDC_ADDRESS
                                                               mov
                                                                         dx, ax
2631 0000202B 66EF
                                                <1>
                                                                out
                                                                         al, [VGA_ROWS]
2632 0000202D A0[2A5F0000]
                                                 <1>
                                                               mov
2633 00002032 8A25[245F0000]
                                                                         ah, [CRT_COLS] ; = [VGA_COLS]
                                                <1>
                                                               mov
2634 00002038 F6E4
                                                <1>
                                                             mul ah
                                                            movzx edx, ax
2635 0000203A 0FB7D0
                                                 <1>
                                                               ; 08/08/2016
2636
                                                 <1>
                                                            movzx eax, byte [CHAR_HEIGHT]
2637 0000203D 0FB605[265F0000] <1>
2638 00002044 F7E2
                                                 <1>
                                                            mul edx
2639
                                                 <1>
                                                               ; eax = byte count
2640 00002046 89C1
                                                 <1>
                                                               mov ecx, eax
                                                                ;; 07/08/2016
2641
                                                  <1>
                                                                ;shl dx, 3; * 8; * [CHAR_HEIGHT]
2642
                                                  <1>
2643
                                                 <1>
                                                                         ecx, edx
                                                                ; mov
2644 00002048 88D8
                                                 <1>
                                                                          al, bl ; fill value for blanked lines
                                                                          edi, 0A0000h
2645 0000204A BF00000A00
                                                 <1>
                                                                mov
2646 0000204F F3AA
                                                <1>
                                                                rep
                                                                         stosb
                                                 <1>
2648 00002051 66B80500
                                                <1>
                                                               mov
                                                                          ax, 5
2649 00002055 66BACE03
                                                 <1>
                                                                          dx, 3CEh ; VGAREG_GRDC_ADDRESS
                                                                mov
                                                                         dx, ax ; 0005h
2650 00002059 66EF
                                                 <1>
                                                                out
2651
                                                 <1>
2652 0000205B C3
                                                 <1>
                                                                retn
2653
                                                 <1>
                                                 <1> vga g up 5:
2654
2655 0000205C 6658
                                                 <1>
                                                               pop ax
2656
                                                 <1>
                                                  <1> vga_g_up_6:
2657
                                                               ; [ESI] = VGA memory model number for current video mode
2658
                                                 <1>
2659
                                                 <1>
2660
                                                 <1>
                                                                  ; LINEAR8 equ 5
                                                                 ; PLANAR4 equ 4
2661
                                                 <1>
                                                                ; PLANAR1 equ 3
2662
                                                 <1>
2663
                                                 <1>
2664 0000205E 803E04
                                                                cmp byte [esi], PLANAR4
                                                 <1>
2665 00002061 7424
                                                 <1>
                                                                         short vga_g_up_planar
                                                                jе
```

```
2666 00002063 803E03
                                 <1>
                                           cmp
                                                byte [esi], PLANAR1
2667 00002066 741F
                                 <1>
                                           jе
                                                 short vga_g_up_planar
                                 <1> vga_g_up_linear8:
2668
                                         ; 07/07/2016 (TEMPORARY)
2669
                                 <1>
2670
                                 <1>
2671
                                 <1>
                                          ; cl = upper left column ; cul
2672
                                 <1>
                                          ; ch = upper left row ; rul
                                 <1>
2673
                                         ; dl = lower rigth column ; clr
2674
                                 <1>
                                          ; dh = lower right row ; rlr
2675
                                 <1>
2676
                                 <1> vga_g_up_10:
                                         ; { for (i=rul; i<=rlr; i++)
2677
                                 <1>
2678
                                 <1>
                                          ; if((i+nblines>rlr)||(nblines==0))
2679 00002068 08C0
                                <1>
                                                 al, al
                                          jz
2680 0000206A 7414
                                <1>
                                                 short vga_g_up_12
2681 0000206C 88C4
                                <1>
                                          mov
                                                 ah, al
2682 0000206E 00EC
                                                ah, ch ; i+nblines
                                <1>
                                          add
                                 <1>
                                                short vga_g_up_12
2683
                                          ;jc
2684 00002070 38F4
                                 <1>
                                                ah, dh
                                           cmp
2685 00002072 770C
                                <1>
                                                short vga_g_up_12
                                          jа
2686
                                <1>
                                          ; else
2687
                                 <1>
                                          ; vgamem_copy_pl4(cul,i+nblines,i,cols,nbcols,cheight);
2688 00002074 E8F2000000
                                          call vgamem_copy_18
                                <1>
                                 <1> vga_g_up_l1:
2689
2690 00002079 FEC5
                                          inc ch
                                 <1>
2691 0000207B 38F5
                                <1>
                                           cmp
                                                ch, dh
2692 0000207D 76E9
                                <1>
                                           jna short vga_g_up_10
2693 0000207F C3
                                <1>
                                          retn
                                <1> vga_g_up_12:
2694
                                          ; vgamem_fill_pl4(cul,i,cols,nbcols,cheight,attr);
2695
                                <1>
2696 00002080 E850010000
                                <1>
                                           call vgamem_fill_18
2697 00002085 EBF2
                                 <1>
                                          jmp short vga_g_up_l1
2698
                                 <1>
2699
                                 <1> vga_g_up_planar:
                                       ; cl = upper left column ; cul
2700
                                 <1>
2701
                                 <1>
                                          ; ch = upper left row ; rul
                                          ; dl = lower rigth column ; clr
2702
                                 <1>
2703
                                 <1>
                                         ; dh = lower right row ; rlr
2704
                                 <1> vga_g_up_pl0:
                                      ; { for (i=rul; i<=rlr; i++)
2705
                                 <1>
                                          ; if((i+nblines>rlr)||(nblines==0))
2706
                                 <1>
2707 00002087 2000
                                 <1>
                                          and
                                                al, al
2708 00002089 7414
                                                 short vga_g_up_pl2
                                <1>
                                          jz
                                                ah, al
2709 0000208B 88C4
                                <1>
                                          mov
2710 0000208D 00EC
                                <1>
                                          add
                                                ah, ch ; i+nblines
                                                 short vga_g_up_pl2
2711
                                <1>
                                          ;jc
                                                ah, dh
2712 0000208F 38F4
                                <1>
                                          cmp
2713 00002091 770C
                                <1>
                                          jа
                                                short vga_g_up_pl2
                                          ; else
2714
                                 <1>
2715
                                <1>
                                          ; vgamem_copy_pl4(cul,i+nblines,i,cols,nbcols,cheight);
2716 00002093 E80E000000
                                <1>
                                           call vgamem_copy_pl4
2717
                                 <1> vga_g_up_pl1:
2718 00002098 FEC5
                                 <1>
                                          inc ch
2719 0000209A 38F5
                                <1>
                                                ch, dh
                                           cmp
2720 0000209C 76E9
                                <1>
                                           jna
                                                short vga_g_up_pl0
2721 0000209E C3
                                <1>
                                           retn
                                <1> vga_g_up_pl2:
                                          ; vgamem_fill_pl4(cul,i,cols,nbcols,cheight,attr);
2723
                                 <1>
2724 0000209F E870000000
                                 <1>
                                           call vgamem_fill_pl4
2725 000020A4 EBF2
                                 <1>
                                                short vga_g_up_pl1
                                           jmp
2726
                                 <1>
                                 <1> vgamem_copy_pl4:
2727
2728
                                 <1>
                                         ; 08/08/2016
2729
                                 <1>
                                           ; 07/08/2016
2730
                                 <1>
                                          ; 07/07/2016 - TRDOS 386 (TRDOS v2.0)
2731
                                 <1>
2732
                                 <1>
                                          ; derived from 'Plex86/Bochs VGABios' source code
2733
                                 <1>
                                          ; vgabios-0.7a (2011)
2734
                                 <1>
                                          ; by the LGPL VGABios developers Team (2001-2008)
                                          ; 'vgabios.c', 'vgamem_copy_pl4'
2735
                                 <1>
2736
                                 <1>
                                          ; vgamem_copy_pl4(xstart,ysrc,ydest,cols,nbcols,cheight)
2737
                                 <1>
                                           ; cl = xstart, ah = ysrc (i+nblines), ch = ydest (i),
2738
                                 <1>
2739
                                 <1>
                                          ; bh = cols, [CRT_COLS] = nbcols, [CHAR_HEIGHT] = cheight
2740
                                 <1>
2741
                                 <1>
                                           ; src=ysrc*cheight*nbcols+xstart;
2742
                                 <1>
                                           ; dest=ydest*cheight*nbcols+xstart;
2743
                                 <1>
2744 000020A6 52
                                 <1>
                                           push edx
2745 000020A7 50
                                 <1>
                                           push eax
2746
                                 <1>
                                           ; outw(VGAREG_GRDC_ADDRESS, 0x0105)
2747
                                 <1>
2748 000020A8 66B80501
                                <1>
                                           mov
                                                ax, 0105h
2749 000020AC 66BACE03
                                 <1>
                                                 dx, 3CEh ; VGAREG_GRDC_ADDRESS
2750 000020B0 66EF
                                 <1>
                                                dx, ax
                                           out
2751
                                 <1>
2752
                                 <1>
                                          ; 07/08/2016
2753
                                 <1>
                                          ;mov ah, [esp+1]
2754
                                 <1>
                                          ;movzx edx, ah ; ysrc
                                          movzx edx, byte [esp+1]
2755 000020B2 0FB6542401
                                 <1>
                                          ; 08/08/2016
2756
                                 <1>
                                          movzx eax, byte [CHAR HEIGHT]
2757 000020B7 0FB605[265F0000]
                                 <1>
                                          mov ah, [CRT_COLS]; nbcols
mul ah
2758 000020BE 8A25[245F0000]
                                 <1>
2759 000020C4 F6E4
                                 <1>
                                          ;; 07/08/2016
2760
                                 <1>
2761
                                           ;movzx eax, byte [CRT_COLS]
                                 <1>
                                          ;shl ax, 3; * 8; * [CHAR HEIGHT]
                                 <1>
                                           push eax ; cheight * nbcols
2763 000020C6 50
                                <1>
2764 000020C7 F7E2
                                <1>
                                           mul edx; * ysrc
                                          ; eax = ysrc * cheight * nbcols
2765
                                <1>
2766
                                <1>
                                          ; edx = 0
2767 000020C9 88CA
                                <1>
                                          mov dl, cl ; edx = xstart
2768 000020CB 01D0
                                                eax, edx
                                <1>
                                          add
2769 000020CD 89C6
                                <1>
                                           mov esi, eax; src
2770 000020CF 88EA
                                 <1>
                                          mov dl, ch ; ydest
```

```
2771 000020D1 58
                                <1>
                                                eax ; cheight * nbcols
                                          pop
2772 000020D2 F7E2
                                <1>
                                          mul
                                                edx
                                          ; eax = ydest * cheight * nbcols
2773
                                <1>
2774 000020D4 88CA
                                <1>
                                          mov dl, cl ; edx = xstart
                                          add eax, edx
mov edi, eax; dest
2775 000020D6 01D0
                                <1>
2776 000020D8 89C7
                                <1>
2777
                                          ; esi = src
                                <1>
                                         ; edi = dest
2778
                                 <1>
2779
                                 <1>
                                          ; for(i=0;i<cheight;i++)</pre>
2780
                                 <1>
                                          ; {
                                          ; memcpyb(0xa000,dest+i*nbcols,0xa000,src+i*nbcols,cols);
2781
                                 <1>
2782
                                 <1>
                                          ; }
2783 000020DA 51
                                 <1>
                                           push ecx
2784 000020DB B900000A00
                                <1>
                                          mov ecx, 0A0000h
                                          add esi, ecx add edi, ecx
2785 000020E0 01CE
                                 <1>
2786 000020E2 01CF
                                <1>
                                       ; 08/08/2016
mov dh, [CHAR_HEIGHT]
2787
                                <1>
2788 000020E4 8A35[265F0000]
                                <1>
2789
                                 <1>
                                          ;; 07/08/2016
2790
                                <1>
                                          ;mov dh, 8; 07/08/2016
2791 000020EA 28D2
                                           sub dl, dl; i
                                <1>
                                 <1> vgamem_copy_pl4_0:
2792
2793 000020EC 56
                                <1>
                                          push esi
2794 000020ED 57
                                <1>
                                          push edi
2795 000020EE 0FB605[245F0000] <1>
                                          movzx eax, byte [CRT_COLS]
2796 000020F5 F6E2
                                 <1>
                                          mul dl
2797
                                <1>
                                          ; eax = i * nbcols
                                <1>
2798 000020F7 01C7
                                        add edi, eax ; dest+i*nbcols
2799 000020F9 01C6
                                <1>
                                          add
                                                 esi, eax
2800 000020FB 0FB6CF
                                <1>
                                          movzx ecx, bh; cols
2801 000020FE F3A4
                                <1>
                                          rep movsb
2802 00002100 5F
                                <1>
                                          pop
                                                 edi
2803 00002101 5E
                                <1>
                                           pop
                                                 esi
2804 00002102 FECE
                              <1>
2805 00002104 75E6
                                <1>
                                                 short vgamem_copy_pl4_0
                                           jnz
2806
                                <1> vgamem_copy_pl4_1:
2807 00002106 59
                                <1>
                                          pop
                                                ecx
2808
                                <1>
2809
                                 <1>
                                           ; outw(VGAREG_GRDC_ADDRESS, 0x0005);
2810 00002107 66B80500
                                           mov ax, 00\overline{05}h
                                <1>
                                                 dx, 3CEh ; VGAREG_GRDC_ADDRESS
2811 0000210B 66BACE03
                                <1>
                                           mov
2812 0000210F 66EF
                                <1>
                                           out
                                                 dx, ax
2813
                                <1>
2814 00002111 58
                                <1>
                                           pop
                                                 eax
2815 00002112 5A
                                 <1>
                                                 edx
                                           pop
2816
                                 <1>
2817 00002113 C3
                                 <1>
                                           retn
2818
                                 <1>
2819
                                 <1> vgamem_fill_pl4:
                                         ; 08/<del>0</del>8/2016
2820
                                 <1>
2821
                                 <1>
                                          ; 07/08/2016
2822
                                 <1>
                                          ; 04/08/2016
                                          ; 07/07/2016 - TRDOS 386 (TRDOS v2.0)
2823
                                 <1>
2824
                                 <1>
                                         ; derived from 'Plex86/Bochs VGABios' source code
2825
                                 <1>
2826
                                 <1>
                                           ; vgabios-0.7a (2011)
2827
                                 <1>
                                          ; by the LGPL VGABios developers Team (2001-2008)
                                           ; 'vgabios.c', 'vgamem_fill_pl4'
                                 <1>
2828
2829
                                 <1>
                                 <1>
                                          ; vgamem_fill_pl4(xstart,ystart,cols,nbcols,cheight,attr)
2830
2831
                                 <1>
                                           ; cl = xstart, edi = ch = ystart, bh = cols,
2832
                                 <1>
                                           ; [CRT_COLS] = nbcols, [CHAR_HEIGHT] = cheight, attr = 0
2833
                                 <1>
2834
                                 <1>
                                           ; dest=ystart*cheight*nbcols+xstart;
2835 00002114 52
                                 <1>
                                           push edx
2836 00002115 50
                                 <1>
                                           push eax
2837
                                <1>
2838
                                          ; outw(VGAREG_GRDC_ADDRESS, 0x0205)
                                 <1>
2839 00002116 66B80502
                                 <1>
                                           mov ax, 0205h
2840 0000211A 66BACE03
                                 <1>
                                                dx, 3CEh ; VGAREG_GRDC_ADDRESS
                                           mov
2841 0000211E 66EF
                                 <1>
                                           out dx, ax
2842
                                 <1>
2843
                                                 ; 08/08/2016
                                 <1>
                                          movzx eax, byte [CHAR_HEIGHT]
2844 00002120 0FB605[265F0000]
                                 <1>
2845 00002127 F6E5
                                 <1>
                                          mul ch
                                           ;; 07/08/2016
2846
                                 <1>
2847
                                 <1>
                                          ;movzx eax, ch
                                           ;shl ax, 3; * 8; * [CHAR_HEIGHT]
2848
                                 <1>
2849 00002129 0FB615[245F0000]
                                 <1>
                                           movzx edx, byte [CRT_COLS] ; = [VGA_COLS]
2850 00002130 F7E2
                                 <1>
                                           mul edx
2851
                                 <1>
                                           ; edx = 0
2852 00002132 88CA
                                 <1>
                                           mov dl, cl
2853 00002134 01D0
                                 <1>
                                           add
                                                eax, edx
2854 00002136 89C7
                                 <1>
                                           mov edi, eax
                                           ; edi = dest
2855
                                 <1>
2856
                                 <1>
                                           ; for(i=0;i<cheight;i++)</pre>
2857
                                 <1>
                                          ; {
2858
                                 <1>
                                          ; memsetb(0xa000,dest+i*nbcols,attr,cols);
2859
                                 <1>
2860 00002138 81C700000A00
                                 <1>
                                                edi, 0A0000h
                                           add
2861 0000213E 51
                                 <1>
                                          push ecx
                                          ; 08/08/2016
2862
                                 <1>
2863 0000213F 8A35[265F0000]
                                           mov dh, [CHAR HEIGHT]
                                 <1>
                                          ;; 07/08/2016
                                 <1>
2864
                                          ;mov dh, 8; 07/08/2016
sub dl, dl; i
2865
                                 <1>
2866 00002145 28D2
                                <1>
                                <1> vgamem_fill_pl4_0:
2868 00002147 57
                                <1>
                                           push edi
2869 00002148 0FB605[245F0000] <1>
                                           movzx eax, byte [CRT_COLS]
2870 0000214F F6E2
                                          mul dl
                                <1>
                                       ; eax = i * nbcols
2871
                                <1>
                                          add edi, eax; dest+i*nbcols mov al, bl; attr; 04/08/2016
2872 00002151 01C7
                                <1>
                                <1>
2873 00002153 88D8
                                <1>
2874 00002155 OFB6CF
                                        movzx ecx, bh ; cols
2875 00002158 F3AA
                                <1>
                                        rep stosb
```

```
2876 0000215A 5F
                                 <1>
                                                 edi
                                           pop
2877 0000215B 75EA
                                 <1>
                                           jnz
                                                 short vgamem_fill_pl4_0
2878
                                <1> vgamem_fill_pl4_1:
2879 0000215D 59
                                <1>
                                          pop
2880
                                 <1>
2881
                                 <1>
                                           ; outw(VGAREG_GRDC_ADDRESS, 0x0005);
2882 0000215E 66B80500
                                <1>
                                           mov ax, 0005h
                                <1>
                                                 dx, 3CEh ; VGAREG_GRDC_ADDRESS
2883 00002162 66BACE03
                                           mov
2884 00002166 66EF
                                <1>
                                                 dx, ax
                                <1>
2886 00002168 58
                                <1>
                                                  eax
                                           pop
2887 00002169 5A
                                 <1>
                                                 edx
                                           pop
2888
                                 <1>
2889 0000216A C3
                                 <1>
                                           retn
2890
                                 <1>
2891
                                 <1> vgamem_copy_18:
2892
                                 <1>
                                          ; 08/08/2016
                                          ; 07/08/2016
2893
                                 <1>
                                          ; 06/08/2016
2894
                                 <1>
2895
                                 <1>
                                          ; 07/07/2016 - TRDOS 386 (TRDOS v2.0)
2896
                                 <1>
                                         ; TEMPORARY
2897
                                 <1>
2898
                                 <1>
2899
                                 <1>
                                          ; derived from 'Plex86/Bochs VGABios' source code
                                 <1>
                                         ; vgabios-0.7a (2011)
2900
2901
                                 <1>
                                           ; by the LGPL VGABios developers Team (2001-2008)
                                           ; 'vgabios.c', 'vgamem_copy_pl4'
2902
                                 <1>
2903
                                 <1>
2904
                                 <1>
                                          ; vgamem_copy_pl4(xstart,ysrc,ydest,cols,nbcols,cheight)
                                           ; cl = xstart, ah = ysrc (i+nblines), ch = ydest (i),
2905
                                 <1>
2906
                                 <1>
                                           ; bh = cols, [CRT_COLS] = nbcols, [CHAR_HEIGHT] = cheight
2907
                                 <1>
2908
                                 <1>
                                           ; src=ysrc*cheight*nbcols+xstart;
2909
                                 <1>
                                           ; dest=ydest*cheight*nbcols+xstart;
2910
                                 <1>
                                           push edx
2911 0000216B 52
                                 <1>
2912 0000216C 50
                                 <1>
                                           push eax
2913
                                 <1>
2914
                                 <1>
                                           ; outw(VGAREG_GRDC_ADDRESS, 0x0105)
                                           ; mov ax, 01\overline{05}h
2915
                                 <1>
2916
                                 <1>
                                           ;mov dx, 3CEh ; VGAREG_GRDC_ADDRESS
2917
                                 <1>
                                           ;out dx, ax
2918
                                 <1>
2919
                                 <1>
                                           ;mov ah, [esp+1]
2920
                                 <1>
                                           movzx edx, ah ; ysrc
2921 0000216D 0FB6D4
                                 <1>
2922
                                 <1>
                                          ; 08/08/2016
2923 00002170 0FB605[265F0000]
                                           movzx eax, byte [CHAR_HEIGHT]
                                 <1>
                                          mov ah, [CRT_COLS]; nbcols mul ah
2924 00002177 8A25[245F0000]
                                 <1>
2925 0000217D F6E4
                                 <1>
                                          ;; 07/08/2016
2926
                                 <1>
2927
                                 <1>
                                          ; movzx eax, byte [CRT_COLS]
                                           ;shl ax, 3; * 8; * [CHAR_HEIGHT]
2928
                                 <1>
                                          push eax ; cheight * nbcols
2929 0000217F 50
                                <1>
2930 00002180 F7E2
                                <1>
                                          mul edx ; * ysrc
2931
                                <1>
                                          ; eax = ysrc * cheight * nbcols
                                          ; edx = 0
                                <1>
2933 00002182 88CA
                                <1>
                                           mov dl, cl ; edx = xstart
2934 00002184 01D0
                                 <1>
                                           add
                                                 eax, edx
                                          mov esi, eax; src
2935 00002186 89C6
                                <1>
2936 00002188 66C1E603
                                <1>
                                          shl si, 3; * 8; 06/08/2016
2937 0000218C 88EA
                                <1>
                                          mov
                                                 dl, ch ; ydest
2938 0000218E 58
                                <1>
                                                 eax ; cheight * nbcols
                                          pop
2939 0000218F F7E2
                                <1>
                                          mul edx
2940
                                <1>
                                          ; eax = ydest * cheight * nbcols
2941 00002191 88CA
                                <1>
                                           mov dl, cl ; edx = xstart
2942 00002193 01D0
                                <1>
                                          add eax, edx
                                          mov edi, eax; dest
shl di, 3; * 8; 06/08/2016
2943 00002195 89C7
                                <1>
2944 00002197 66C1E703
                                 <1>
2945
                                 <1>
                                          ; esi = src
                                          ; edi = dest
2946
                                 <1>
2947
                                 <1>
                                           ; for(i=0;i<cheight;i++)</pre>
2948
                                 <1>
                                           ; {
2949
                                 <1>
                                          ; memcpyb(0xa000,dest+i*nbcols,0xa000,src+i*nbcols,cols);
2950
                                 <1>
                                          ; }
2951 0000219B 51
                                           push ecx
                                 <1>
2952 0000219C B900000A00
                                 <1>
                                           mov ecx, 0A0000h
                                           add esi, ecx
2953 000021A1 01CE
                                 <1>
2954 000021A3 01CF
                                 <1>
                                           add
                                                  edi, ecx
2955
                                           ; 08/08/2016
                                 <1>
2956 000021A5 8A35[265F0000]
                                 <1>
                                           mov dh, [CHAR_HEIGHT]
                                 <1>
                                           ;; 07/08/2016
                                <1>
                                           ;mov dh, 8 ; 07/08/2016
2959 000021AB 28D2
                                 <1>
                                           sub dl, dl; i
                                 <1> vgamem_copy_18_0:
2960
2961 000021AD 56
                                          push esi
                                <1>
2962 000021AE 57
                                          push edi
                                <1>
2963 000021AF 0FB605[245F0000] <1>
                                       movzx eax, byte [CRT_COLS]
                                      mul dl

; eax = i * nbcols

shl ax, 3; * 8; 06/08/2016

add edi, eax; dest+i*nbcols

add esi, eax
2964 000021B6 F6E2
                                <1>
                                <1>
                       <1>
<1>
2966 000021B8 66C1E003
2967 000021BC 01C7
2968 000021BE 01C6
                                <1>
                            movzx ecx, bh; cols
2969 000021C0 0FB6CF
                                      shl cx, 3; * 8; 06/08/2016 rep movsb
                               <1>
<1>
2970 000021C3 66C1E103
2971 000021C7 F3A4
                                        pop
2972 000021C9 5F
                               <1>
                                                 edi
2973 000021CA 5E
                              <1>
<1>
                                       bob
                                                 esi
2974 000021CB FEC2
                                           inc
                                                 dl ; 06/08/2016
                                <1>
2975 000021CD FECE
                                           dec
                                                dh
2976 000021CF 75DC
                                           jnz short vgamem_copy_18_0
                               <1>
                                <1> vgamem_copy_18_1:
2977
2978 000021D1 59
                                <1>
                                           pop ecx
2979
                                 <1>
2980
                                 <1>
                                           ;; outw(VGAREG_GRDC_ADDRESS, 0x0005);
```

```
2981
                                     <1>
                                                ;mov ax, 0005h
2982
                                     <1>
                                                ;mov dx, 3CEh ; VGAREG GRDC ADDRESS
2983
                                     <1>
                                                ;out dx, ax
2984
                                     <1>
2985 000021D2 58
                                     <1>
                                                pop
                                                       eax
2986 000021D3 5A
                                     <1>
                                                       edx
                                                pop
                                     <1>
2988 000021D4 C3
                                     <1>
                                                retn
2989
                                     <1>
                                     <1> vgamem_fill_18:
2990
2991
                                     <1>
                                           ; 08/08/2016
2992
                                                ; 07/08/2016
                                     <1>
                                               ; 06/08/2016
2993
                                     <1>
2994
                                     <1>
                                              ; 04/08/2016
                                              ; 07/07/2016 - TRDOS 386 (TRDOS v2.0)
2995
                                     <1>
2996
                                     <1>
                                                ; TEMPORARY
2997
                                     <1>
                                     <1>
2998
                                                ; derived from 'Plex86/Bochs VGABios' source code
2999
                                     <1>
3000
                                     <1>
                                                ; vgabios-0.7a (2011)
3001
                                     <1>
                                              ; by the LGPL VGABios developers Team (2001-2008)
                                                ; 'vgabios.c', 'vgamem_fill_pl4'
3002
                                     <1>
                                     <1>
3003
3004
                                     <1>
                                               ; vgamem_fill_pl4(xstart,ystart,cols,nbcols,cheight,attr)
3005
                                     <1>
                                                ; cl = xstart, edi = ch = ystart, bh = cols,
3006
                                     <1>
                                                ; [CRT_COLS] = nbcols, [CHAR_HEIGHT] = cheight, attr = 0
3007
                                     <1>
3008
                                     <1>
                                                ; dest=ystart*cheight*nbcols+xstart;
3009 000021D5 52
                                     <1>
                                                push edx
3010 000021D6 50
                                     <1>
                                                push eax
3011
                                     <1>
3012
                                                ;; outw (VGAREG GRDC ADDRESS, 0x0205)
                                     <1>
3013
                                     <1>
                                                ;mov ax, 0205h
3014
                                     <1>
                                                ;mov dx, 3CEh ; VGAREG_GRDC_ADDRESS
                                                ;out dx, ax
3015
                                     <1>
3016
                                     <1>
                                                ; 08/08/2016
                                     <1>
3018 000021D7 0FB605[265F0000]
                                     <1>
                                                movzx eax, byte [CHAR_HEIGHT]
3019 000021DE F6E5
                                     <1>
                                                mul ch
                                                ;; 07/08/2016
3020
                                     <1>
3021
                                     <1>
                                                ;movzx eax, ch
                                                ;shl ax, 3; * 8; * [CHAR HEIGHT]
3022
                                     <1>
3023 000021E0 0FB615[245F0000] <1>
                                                movzx edx, byte [CRT_COLS]; = [VGA_COLS]
3024 000021E7 F7E2
                                     <1>
                                                mul edx
                                                ; edx = 0
3025
                                     <1>
3026 000021E9 88CA
                              <1>
<1>
                                                mov dl, cl
3027 000021EB 01D0
                                               add eax, edx
                                             mov edi, eax
shl di, 3; * 8; 06/08/2016; edi = dest
3028 000021ED 89C7
                                    <1>
3029 000021EF 66C1E703
                                    <1>
                                                ; edi = dest
3030
                                     <1>
3031
                                                ; for(i=0;i<cheight;i++)</pre>
                                     <1>
                                                ; {
3032
                                     <1>
                                           add edi, 0A0000h
push ecx
; 08/08/2016
mov
                                                ; memsetb(0xa000,dest+i*nbcols,attr,cols);
3033
                                     <1>
3034
                                     <1>
3035 000021F3 81C700000A00
                                     <1>
3036 000021F9 51
                                     <1>
                                     <1>
3038 000021FA 8A35[265F0000]
                                     <1>
                                                mov dh, [CHAR_HEIGHT]
3039
                                     <1>
                                                ;; 07/08/2016
3040
                                     <1>
                                                ;mov dh, 8 ; 07/08/2016
                                               sub dl, dl ; i
3041 00002200 28D2
                                    <1>
                                    <1> vgamem_fill_18_0:
<1> push edi
3043 00002202 57
| Movzx eax, byte [CRT_COLS] | 3045 0000220A F6E2 | <1> mul dl | dl | | 3046 | <1> ; eax = i * nbcols | shl ax, 3 ; * 8 ; 06/08/2016 | 3048 00002210 01C7 | <1> add edi, eax ; dest+i*nbcols | 3049 00002212 88D8 | <1> mov al, bl; attr; 04/08/2016 | 3050 00002214 0FB6CF | <1> movzx ecx, bh; cols | 3050 00002217 66C1E103 | <1> shl cx, 3 ; * 8 ; 06/08/2016 | 3052 00002218 F3AA | <1> rep stosb
3044 00002203 0FB605[245F0000] <1>
                                                movzx eax, byte [CRT_COLS]
3053 0000221D 5F
                                    <1>
                                                pop
                                                       edi
3054 0000221E FEC2
                                    <1>
                                                inc dl ; 06/08/2016
3055 00002220 FECE
                                     <1>
                                                dec
                                                       dh
3056 00002222 75DE
                                     <1>
                                                jnz
                                                       short vgamem_fill_18_0
3057
                                     <1> vgamem_fill_18_1:
3058 00002224 59
                                     <1>
                                                pop ecx
3059
                                     <1>
3060
                                     <1>
                                                ;; outw(VGAREG_GRDC_ADDRESS, 0x0005);
3061
                                     <1>
                                                ;mov ax, 0005h
                                                        dx, 3CEh ; VGAREG_GRDC_ADDRESS
3062
                                      <1>
                                                ;mov
3063
                                     <1>
                                                ;out
                                                        dx, ax
                                      <1>
3065 00002225 58
                                     <1>
                                                pop
                                                        eax
3066 00002226 5A
                                     <1>
                                                        edx
                                                pop
3067
                                      <1>
3068 00002227 C3
                                     <1>
                                                retn
3069
                                     <1>
3070
                                     <1> vga graphics down:
3071
                                                ; 08/08/2016
                                     <1>
3072
                                                ; 07/08/2016
                                      <1>
                                                ; 31/07/2016
3073
                                     <1>
3074
                                     <1>
                                                ; 07/07/2016 - TRDOS 386 (TRDOS v2.0)
3075
                                      <1>
                                                ; derived from 'Plex86/Bochs VGABios' source code
3076
                                     <1>
3077
                                      <1>
                                                ; vgabios-0.7a (2011)
3078
                                     <1>
                                                ; by the LGPL VGABios developers Team (2001-2008)
                                                ; 'vgabios.c', 'biosfn scroll'
3079
                                     <1>
3080
                                     <1>
3081
                                     <1>
3082
                                      <1>
                                                ; cl = upper left column
                                                ; ch = upper left row
3083
                                     <1>
3084
                                      <1>
                                                ; dl = lower rigth column
3085
                                      <1>
                                                ; dh = lower right row
```

```
3086
                                 <1>
3087
                                 <1>
                                          ; al = line count (AL=0 means blank entire fields)
                                          ; bl = fill value for blanked lines
3088
                                 <1>
3089
                                 <1>
                                          ; bh = unused
3090
                                 <1>
3091
                                 <1>
                                          ; ah = [CRT_MODE], current video mode
3092
                                 <1>
3093 00002228 FC
                                                  ; !!! Clear direction flag !!!
                                <1>
3094
                                <1>
3095 00002229 88C7
                                <1>
                                          mov bh, al; 31/07/2016
3096
                                <1>
3097 0000222B BE[3E5F0000]
                                <1>
                                          mov
                                                esi, vga modes
3098 00002230 89F7
                                <1>
                                                edi, esi
                                          mov
3099 00002232 83C710
                                <1>
                                          add edi, vga_mode_count
3100
                                <1> vga_g_down_0:
3101 00002235 AC
                                <1>
                                          lodsb
                                          cmp al, ah; [CRT MODE]
3102 00002236 38E0
                               <1>
                               <1>
                                                short vga_g_down_1
3103 00002238 7405
                                          je
                                         cmp esi, edi
jb short vga_g_down_0
3104 0000223A 39FE
                                <1>
3105 0000223C 72F7
                               <1>
                                      ; xor bh, bh; 31/07/2016 retn; nothing to do
3106
                               <1>
3107 0000223E C3
                                <1>
                               <1> vga_g_down_1:
3108
3109 0000223F 88F8
                                <1> mov al, bh; 31/07/2016
3110 00002241 83C64F
                                <1>
                                          add esi, vga_memmodel - (vga_modes + 1)
3111
                                <1>
                                          ; [ESI] = VGA memory model number (LINEAR8, PLANAR4, PLANAR1)
3112
                                <1>
3113
                                <1>
                                          ; if(rlr>=nbrows)rlr=nbrows-1;
3114
                                 <1>
                                          ; if(clr>=nbcols)clr=nbcols-1;
3115
                                 <1>
                                          ; if (nblines>nbrows) nblines=0;
3116
                                 <1>
                                          ; cols=clr-cul+1;
3117
                                 <1>
3118 00002244 3A35[2A5F0000]
                                          cmp dh, [VGA_ROWS]
                                <1>
3119 0000224A 7208
                                <1>
                                        jb short vga_g_down_2
3120 0000224C 8A35[2A5F0000]
                                <1>
                                                dh, [VGA_ROWS]
                                      dec
                                          mov
                                               dh
3121 00002252 FECE
                                <1>
3122
                                <1> vga_g_down_2:
3123 00002254 3A15[245F0000]
                                <1> cmp dl, [CRT_COLS] ; = [VGA_COLS]
                                                short vga g down 3
3124 0000225A 7208
                                <1>
                                          jb
                                                dl, [CRT_COLS]
3125 0000225C 8A15[245F0000]
                               <1> mov dl, <1> dec dl
3126 00002262 FECA
                                <1> vga_g_down_3:
3127
                               <1> cmp al, [VGA_ROWS]
<1> jna short vga_g_down_4
3128 00002264 3A05[2A5F0000]
3129 0000226A 7602
3130 0000226C 28C0
                                <1>
                                         sub
                                                al, al ; \overline{0}
3131
                                <1> vga_g_down_4:
                               <1> mov bh, dh; clr
3132 0000226E 88F7
3133 00002270 28CF
                                <1>
                                          sub bh, cl; cul
3134 00002272 FEC7
                                <1>
                                          inc
                                                bh ; cols = clr-cul+1
                               <1>
                                     and al, al; nblines = 0
jnz short vga_g_down_6
and ch. ch : nul = 0
                             <1>
3136 00002274 20C0
3137 00002276 755B
                                <1>
                                               ch, ch; \overline{rul} = 0
3138 00002278 20ED
                               <1>
                                          and
3139 0000227A 7557
                               <1>
                                                short vga_g_down_6
                                          jnz
                                          and
3140 0000227C 20C9
                                <1>
                                                cl, cl; cul = 0
3141 0000227E 7553
                                <1>
                                          jnz
                                                short vga_g_down_6
                                <1>
                                          push ax
3143 00002280 6650
                                <1>
3144 00002282 A0[2A5F0000]
                                <1>
                                          mov
                                                al, [VGA_ROWS]
3145 00002287 FEC8
                                <1>
                                          dec
                                                al
3146 00002289 38C6
                                <1>
                                          cmp dh, al ; rlr = nbrows-1
3147 0000228B 7544
                                <1>
                                          jne
                                                short vga_g_down_5
3148 0000228D A0[245F0000]
                                <1>
                                          mov al, [CRT_COLS] ; = VGA_COLS
                                          dec al
3149 00002292 FEC8
                                <1>
3150 00002294 38C2
                                <1>
                                                dl, al ; clr = nbcols-1
                                          cmp
3151 00002296 7539
                                <1>
                                          jne
                                                short vga_g_down_5
                                          pop ax
3152 00002298 6658
                                <1>
3153
                                <1>
3154 0000229A 66B80502
                                <1>
                                          mov
                                                ax, 0205h
3155 0000229E 66BACE03
                                <1>
                                                dx, 3CEh ; VGAREG_GRDC_ADDRESS
                                          mov
                                                dx, ax
3156 000022A2 66EF
                                <1>
                                          out
                                                al, [VGA_ROWS]
3157 000022A4 A0[2A5F0000]
                                <1>
                                          mov
3158 000022A9 8A25[245F0000]
                                                ah, [CRT_COLS] ; = [VGA_COLS]
                                <1>
                                          mov
3159 000022AF F6E4
                                <1>
                                        mul ah
3160 000022B1 0FB7D0
                                <1>
                                         movzx edx, ax
3161
                                <1>
                                          ; 08/08/2016
3162 000022B4 0FB605[265F0000] <1>
                                          movzx eax, byte [CHAR_HEIGHT]
3163 000022BB F7E2
                                <1>
                                          mul edx
                                          ; eax = byte count
3164
                                 <1>
3165 000022BD 89C1
                                 <1>
                                          mov ecx, eax
                                          ;; 07/08/2016
3166
                                 <1>
                                          ;shl dx, 3; * 8; * [CHAR_HEIGHT]
3167
                                 <1>
3168
                                <1>
                                                ecx, edx
                                          ; mov
3169 000022BF 88D8
                                 <1>
                                                 al, bl ; fill value for blanked lines
3170 000022C1 BF00000A00
                                                 edi, 0A0000h
                                <1>
                                          mov
3171 000022C6 F3AA
                                <1>
                                          rep
                                                 stosb
                                <1>
3173 000022C8 B005
                                <1>
                                          mov
                                                 al, 5
3174 000022CA 66BACE03
                                <1>
                                                 dx, 3CEh ; VGAREG_GRDC_ADDRESS
                                          mov
                                                dx, ax ; 0005h
3175 000022CE 66EF
                                <1>
                                          out
3176
                                <1>
3177 000022D0 C3
                                 <1>
                                          retn
3178
                                <1>
3179
                                 <1> vga_g_down_5:
3180 000022D1 6658
                                <1>
                                          pop ax
3181
                                <1>
3182
                                 <1> vga_g_down_6:
                                          ; [ESI] = VGA memory model number for current video mode
3183
                                 <1>
3184
                                 <1>
3185
                                 <1>
                                            ; LINEAR8 equ 5
3186
                                 <1>
                                           ; PLANAR4 equ 4
                                          ; PLANAR1 equ 3
3187
                                 <1>
3188
                                <1>
3189 000022D3 803E04
                                 <1>
                                          cmp byte [esi], PLANAR4
3190 000022D6 742C
                                <1>
                                                short vga_g_down_planar
                                          jе
```

```
3191 000022D8 803E03
                               <1>
                                         cmp byte [esi], PLANAR1
3192 000022DB 7427
                               <1>
                                         jе
                                               short vga_g_down_planar
                                <1> vga_g_down_linear8:
3193
                                       ; 07/07/2016 (TEMPORARY)
3194
                               <1>
3195
                                <1>
                                       ; cl = upper left column ; cul
; ch = upper left row ; rul
3196
                                <1>
3197
                                <1>
                                       ; dl = lower rigth column ; clr
                                <1>
3198
3199
                                <1>
                                       ; dh = lower right row ; rlr
                                <1>
3200
3201
                                <1> vga_g_down_10:
                                       ;{for(i=rlr;i>=rul;i--)
3202
                                <1>
3203
                               <1>
                                         ; if((i<rul+nblines)||(nblines==0))</pre>
3204 000022DD 08C0
                               <1>
                                      jz
                                               short vga_g_down_12
3205 000022DF 741C
                               <1>
3206 000022E1 88C4
                               <1>
                                         mov
                                               ah, al
                                      add ah, ch
3207 000022E3 00EC
                               <1>
                                      ;jc short vga_g_down_12
3208
                               <1>
                                         xchg ch, dh
3209 000022E5 86EE
                               <1>
3210 000022E7 38E5
                               <1>
                                        cmp ch, ah
                                     jb short vga_g_down_iz
mov ah, ch
sub ah, al; ah = i - nblines
3211 000022E9 7212
                               <1>
3212 000022EB 88EC
                               <1>
3213 000022ED 28C4
                               <1>
3214
                               <1>
                                        ; vgamem_copy_pl4(cul,i,i-nblines,cols,nbcols,cheight);
3215
                               <1>
3216 000022EF E877FEFFFF
                               <1>
                                         call vgamem_copy_18
                               <1> vga_g_down_l1:
                               <1>
3218 000022F4 86F5
                                         xchg dh, ch
3219 000022F6 FECE
                               <1>
                                               dh
                                         dec
3220 000022F8 38EE
                               <1>
                                         cmp
                                               dh, ch
3221 000022FA 73E1
                               <1>
                                         jnb short vga_g_down_10
3222 000022FC C3
                               <1>
                                        retn
3223
                               <1>
3224
                               <1> vga_g_down_12:
                                     ; vgamem_fill_pl4(cul,i,cols,nbcols,cheight,attr);
3225
                               <1>
3226 000022FD E8D3FEFFFF
                               <1>
                                         call vgamem_fill_18
3227 00002302 EBF0
                               <1>
                                         jmp short vga_g_down_l1
3228
                                <1>
3229
                                <1> vga_g_down_planar:
                                     ; cl = upper left column ; cul
3230
                                <1>
                                         ; ch = upper left row ; rul
3231
                                <1>
3232
                                <1>
                                         ; dl = lower rigth column ; clr
                                       ; dh = lower right row ; rlr
3233
                                <1>
3234
                                <1> vga_g_down_pl0:
                                     ;{for(i=rlr;i>=rul;i--)
3235
                                <1>
3236
                               <1>
                                         ; if((i<rul+nblines)||(nblines==0))</pre>
3237 00002304 08C0
                               <1>
                                         or al, al
3238 00002306 741C
                                     jz
mov
                               <1>
                                               short vga_g_down_pl2
3239 00002308 88C4
                               <1>
                                        mov
                                               ah, al
                                       add ah, ch
3240 0000230A 00EC
                               <1>
3241
                               <1>
                                        ;jc short vga_g_down_pl2
3242 0000230C 86EE
                               <1>
                                         xchg ch, dh
3243 0000230E 38E5
                                        cmp ch, ah
                               <1>
                                      mov ah, ch
sub ah, al; ah = i - nblines; else
3244 00002310 7212
                               <1>
3245 00002312 88EC
                               <1>
3246 00002314 28C4
                               <1>
                               <1>
                                        ; vgamem_copy_pl4(cul,i,i-nblines,cols,nbcols,cheight);
3248
                               <1>
3249 00002316 E88BFDFFFF
                               <1>
                                         call vgamem_copy_pl4
                               <1> vga_g_down_pl1:
3250
                                     xchg dh, ch
3251 0000231B 86F5
                               <1>
3252 0000231D FECE
                               <1>
                                         dec
                                               dh
                               <1>
3253 0000231F 38EE
                                         cmp
                                               dh, ch
                                       jnb
3254 00002321 73E1
                               <1>
                                               short vga_g_down_pl0
3255 00002323 C3
                               <1>
                                        retn
3256
                               <1>
3257
                               <1> vga_g_down_pl2:
                                     ; vgamem_fill_pl4(cul,i,cols,nbcols,cheight,attr);
3258
                               <1>
3259 00002324 E8EBFDFFFF
                               <1>
                                         call vgamem_fill_pl4
                                         jmp short vga_g_down_pl1
3260 00002329 EBF0
                                <1>
3261
                                <1>
                                <1>; 07/07/2016
3262
                                <1>; 27/06/2016 - TRDOS 386 (TRDOS v2.0)
3263
                                <1>; VIDEO1.ASM - 24/03/1985 (IBM PC-AT BIOS source code)
3264
3265
                                <1> ;------
                                <1> ; SCROLL DOWN
3266
3267
                                <1>; THIS ROUTINE SCROLLS DOWN THE INFORMATION ON THE CRT
                                <1> ; ENTRY --
3268
3269
                                <1> ; CH,CL = UPPER LEFT CORNER OF REGION TO SCROLL
3270
                                <1> ; DH, DL = LOWER RIGHT CORNER OF REGION TO SCROLL
                                <1> ; BOTH OF THE ABOVE ARE IN CHARACTER POSITIONS
3271
3272
                                <1> ; BH = FILL VALUE FOR BLANKED LINES
                                <1> ; AL = \# LINES TO SCROLL (AL=0 MEANS BLANK THE ENTIRE FIELD)
3273
3274
                                <1>; DS = DATA SEGMENT
                                <1> ; ES = REGEN SEGMENT
3275
                                <1> ; EXIT --
3276
3277
                                <1>; NOTHING, THE SCREEN IS SCROLLED
3278
                                3279
                                <1>
3280
                                <1>
                                         ; cl = upper left column
                                         ; ch = upper left row
3281
                                <1>
3282
                                <1>
                                         ; dl = lower rigth column
                                         ; dh = lower right row
                                <1>
3283
3284
                                <1>
3285
                                <1>
                                         ; al = line count (AL=0 means blank entire fields)
3286
                                <1>
                                         ; bl = fill value for blanked lines
3287
                                <1>
                                         ; bh = unused
3288
                                <1>
3289
                                <1> GRAPHICS DOWN:
                                       ; \overline{0}7/07/2016
3290
                                <1>
                                         ;AH = Current video mode, [CRT MODE]
                                         , SID ; SET DIRECTION cmp ah, 7
3291
                                <1>
3292
                                <1>
3293 0000232B 80FC07
                                <1>
3294 0000232E 0F87F4FEFFFF
                                <1>
                                         ja vga_graphics_down
                                <1>
                                         ;je _n0
```

```
3296
                                               <1>
                                                          MOV bh, al ; save line count in BH MOV AX, DX ; GET LOWER RIGHT POSIT
 3297 00002334 88C7
                                       <1>
<1>
                                              <1>
 3298 00002336 6689D0
                                                                                                ; GET LOWER RIGHT POSITION INTO AX REG
 3299
                                              <1>
 3300
                                               <1> ;----
                                                                      USE CHARACTER SUBROUTINE FOR POSITIONING
                                               <1> ;----
 3301
                                                                      ADDRESS RETURNED IS MULTIPLIED BY 2 FROM CORRECT VALUE
 3302
                                              <1>
                                                        CALL GRAPH_POSN
MOVzx eDI, AX
 3303 00002339 E8F1010000 <1>
3304 0000233E 0FB7F8 <1>
                                                                                                        ; SAVE RESULT AS DESTINATION ADDRESS
                                             <1>
 3305
                                              <1> ;---- DETERMINE SIZE OF WINDOW
 3306
 3307
                                              <1>
 3311
                                               <1>
                                                                                                ; AND EVEN/ODD ROWS
 3312
                                               <1>
 3313
                                                                     DETERMINE CRT MODE
                                               <1> ;----
 3316 00002353 7307
3317
                                             <1> ;----
                                                                     MEDIUM RES DOWN
                                         <1> SAL DL, 1
<1> SAL DI, 1
<1> SAL DI, 1
<1> INC DI
 3319 00002355 D0E2
                                                                                               ; # COLUMNS * 2, SINCE 2 BYTES/CHAR
                                                                                            ; OFFSET *2 SINCE 2 BYTES/CHAR
 3320 00002357 66D1E7
 3321 0000235A 6647
                                                                                                ; POINT TO LAST BYTE
                                              <1>
                                                                      DETERMINE THE SOURCE ADDRESS IN THE BUFFER
 3323
                                               <1> ;----
 3324
                                               <1>
3325
                                              <1> _R12:
                                                                                                          ; FIND_SOURCE_DOWN
                                                                     LOOP THROUGH, MOVING ONE ROW AT A TIME, BOTH EVEN AND ODD FIELDS
 3337
                                               <1> ;----
 3338
                                               <1>
                                                            CALL R17; ROW_LOOP_DOWN; MOVE ONE ROW

SUB SI, 2000h+80; MOVE TO NEXT ROW
 3339
                                               <1> _R13:
 3346
                                                                                                    ; CLEAR ENTRY DOWN
                                        <1>
<1>
<1`
                                                      R16:

mov bh, dh

mov short R14
                                              <1> _R16: mc
                                                                                                            ; BLANK FIELD DOWN
 3358 0000239D 88F7
                                                                                                ; set blank count to everything in field
                                                                                                        ; CLEAR THE FIELD
 3359 0000239F EBEB
                                              <1>
                                                             JMP short R14
 3360
                                               <1>
                                               <1>; 27/06/2016 - TRDOS 386 (TRDOS v2.0)
 3361
                                                <1>; VIDEO1.ASM - 24/03/1985 (IBM PC-AT BIOS source code)
 3362
 3363
                                                <1>
 3364
                                               <1> ;----
                                                                     ROUTINE TO MOVE ONE ROW OF INFORMATION
 3365
                                               <1>
 3366
                                               <1> _R17:
                                                       MOVzx ecx, DL
 3367 000023A1 0FB6CA
                                               <1>
                                                                                                        ; NUMBER OF BYTES IN THE ROW
                                       3368 000023A4 56
 3369 000023A5 57
                                                                                            ; SAVE POINTERS
 3370 000023A6 F3A4
                                                                                              ; MOVE THE EVEN FIELD
 3371 000023A8 5F
 3372 000023A9 5E
 3373 000023AA 6681C60020
 3374 000023AF 6681C70020
                                                                                              ; POINT TO THE ODD FIELD
 3375 000023B4 56
                                                                                 ; SAVE THE POINTERS : COUNT BACK
                                                       PUSH eDI
 3376 000023B5 57
                                               <1>
                                                                                            ; COUNT BACK
 3377 000023B6 88D1
                                                <1>
                                                             MOV
                                                                      CL, DL
                                        <1>
 3378 000023B8 F3A4
                                                                                                ; MOVE THE ODD FIELD
                                                            REP
                                                                      MOVSB
 3379 000023BA 5F
                                              <1>
 3380 000023BB 5E
                                               <1>
                                                             POP
                                                                                                ; POINTERS BACK
                                                                     eSI
 3381 000023BC C3
                                               <1>
                                                            RETn
                                                                                                 ; RETURN TO CALLER
                                               <1>
                                               <1> ;---- CLEAR A SINGLE ROW
 3383
 3384
                                               <1>
                                              <1> _R18:
 3385
                                          <1> -
 3386 000023BD 0FB6CA
                                                            MOVzx ecx, DL
                                                                                                  ; NUMBER OF BYTES IN FIELD
                                                                                          ; NUMBER
; SAVE POINTER
 3387 000023C0 57
                                              <1>
                                                             PUSH eDI
                                              <1>
                                                                                                ; STORE THE NEW VALUE
| STORE THE NEW VALUE | STOSE | STORE THE NEW VALUE | STOSE | STORE THE NEW VALUE | STOR
                                                            REP STOSB
 3388 000023C1 F3AA
 3396
                                               <1>
 3397
                                               <1> ; 04/07/2016
 3398
                                               <1> ; 01/07/2016
 3399
                                               <1>; 30/06/2016 - TRDOS 386 (TRDOS v2.0)
 3400
                                                <1>; VIDEO1.ASM - 24/03/1985 (IBM PC-AT BIOS source code)
```

```
3401
                                    <1> ;------
 3402
                                    <1> ; GRAPHICS WRITE
 3403
                                    <1>; THIS ROUTINE WRITES THE ASCII CHARACTER TO THE CURRENT
 3404
                                    <1> ; POSITION ON THE SCREEN.
 3405
                                    <1> ; ENTRY --
 3406
                                    <1> ; AL = CHARACTER TO WRITE
                                    <1> ; BL = COLOR ATTRIBUTE TO BE USED FOR FOREGROUND COLOR
 3407
                                    <1> ; IF BIT 7 IS SET, THE CHAR IS XOR'D INTO THE REGEN BUFFER <1> ; (0 IS USED FOR THE BACKGROUND COLOR)
 3408
 3409
                                    <1>; CX = NUMBER OF CHARS TO WRITE
 3410
 3411
                                    <1> ; DS = DATA SEGMENT
                                    <1> ; ES = REGEN SEGMENT
 3412
                                    <1> ; EXIT --
 3413
 3414
                                    <1> ; NOTHING IS RETURNED
 3415
                                    <1>;
                                    <1> ; GRAPHICS READ
 3416
                                    <1>; THIS ROUTINE READS THE ASCII CHARACTER AT THE CURRENT CURSOR
 3417
                                    <1>; POSITION ON THE SCREEN BY MATCHING THE DOTS ON THE SCREEN TO THE
 3418
                                    <1>; CHARACTER GENERATOR CODE POINTS
 3419
 3420
                                    <1> ; ENTRY --
                                    <1> ; NONE (0 IS ASSUMED AS THE BACKGROUND COLOR)
 3421
 3422
                                    <1> ; EXIT --
                                    <1>; AL = CHARACTER READ AT THAT POSITION (0 RETURNED IF NONE FOUND)
 3423
 3424
                                    <1>; FOR BOTH ROUTINES, THE IMAGES USED TO FORM CHARS ARE CONTAINED IN ROM
 3425
                                    <1>; FOR THE 1ST 128 CHARS. TO ACCESS CHARS IN THE SECOND HALF, THE USER
 3426
 3427
                                    <1>; MUST INITIALIZE THE VECTOR AT INTERRUPT 1FH (LOCATION 0007CH) TO
                                    <1>; POINT TO THE USER SUPPLIED TABLE OF GRAPHIC IMAGES (8X8 BOXES).
 3428
 3429
                                    <1> ; FAILURE TO DO SO WILL CAUSE IN STRANGE RESULTS
 3430
                                    <1> ;------
 3431
                                   <1>
 3432
                                   <1> GRAPHICS WRITE:
                                   <1> and eax, OFFh ; ZERO TO HIGH OF CODE POINT
 3433 000023D0 25FF000000
 3434 000023D5 50
                                   <1>
                                             PUSH eAX
                                                                        ; SAVE CODE POINT VALUE
 3435
                                   <1>
 3436
                                   <1> ;----
                                                    DETERMINE POSITION IN REGEN BUFFER TO PUT CODE POINTS
                                   <1>
                                                                        ; FIND LOCATION IN REGEN BUFFER
 3438 000023D6 E84D010000
                                   <1>
                                             CALL S26
 3439 000023DB 89C7
                                   <1>
                                             VOM
                                                    eDI, eAX
                                                                         ; REGEN POINTER IN DI
 3440
                                   <1>
                                                     DETERMINE REGION TO GET CODE POINTS FROM
 3441
                                   <1> ;----
 3442
                                   <1>
 3443 000023DD 58
                                                                        ; RECOVER CODE POINT
                                   <1>
                                             POP
                                                    eAX
 3444
                                   <1>
                                         MOV
 3445 000023DE BE[3C2D0100]
                                   <1>
                                                    esi, crt_char_gen ; offset of images
 3446
                                   <1>
                                                    DETERMINE GRAPHICS MODE IN OPERATION
 3447
                                   <1> ;----
                                 3448
3449 000023E3 66C1E003
                                                                   ; DETERMINE MODE
                                                    AX, 3
                                                                        ; MULTIPLY CODE POINT VALUE BY 8
                                                                        ; SI HAS OFFSET OF DESIRED CODES
                                                    eSI, eAX
 3451
 3452 000023E9 803D[225F0000]06 <1>
                                                    byte [CRT MODE], 6
                                                               ; TEST FOR MEDIUM RESOLUTION MODE
 3453 000023F0 7231
                                                    short S6
 3454
                                   <1>
 3455
                                   <1> ;----
                                                    HIGH RESOLUTION MODE
 3456
                                   <1>
 3457 000023F2 81C700800B00
                                   <1>
                                              add
                                                    edi, 0B8000h
 3458
                                   <1> S1:
                                                                        ; HIGH_CHAR
 3459 000023F8 57
                                   <1>
                                              PUSH
                                                                        ; SAVE REGEN POINTER
                                                    eDI
 3460 000023F9 56
                                              PUSH
                                                                        ; SAVE CODE POINTER
                                   <1>
                                                   eSI
 3461 000023FA B604
                                  <1>
                                              MOV
                                                    DH, 4
                                                                       ; NUMBER OF TIMES THROUGH LOOP
                                   <1> S2:
                                  <1>
<1>
 3463 000023FC AC
                                             LODSB
                                                                        ; GET BYTE FROM CODE POINTS
 3464 000023FD F6C380
                                              TEST BL, 80H
                                                                               ; SHOULD WE USE THE FUNCTION
                                              JNZ short S5
 3465 00002400 7515
                                   <1>
                                                                        ; TO PUT CHAR IN
 3466 00002402 AA
                                   <1>
                                              STOSB
                                                                        ; STORE IN REGEN BUFFER
 3467 00002403 AC
                                   <1>
                                             LODSB
                                   <1> S4:
 3468
 3469 00002404 8887FF1F0000
                                   <1>
                                              MOV
                                                    [eDI+2000H-1], AL ; STORE IN SECOND HALF
 3470 0000240A 83C74F
                                                    eDI, 79 ; MOVE TO NEXT ROW IN REGEN
                                  <1>
                                              ADD
                                                                        ; DONE WITH LOOP
 3471 0000240D FECE
                                  <1>
                                              DEC
                                                   DH
                                                    short S2
 3472 0000240F 75EB
                                   <1>
                                             JNZ
 3473 00002411 5E
                                   <1>
                                             POP
                                                    eSI
                                  <1>
 3474 00002412 5F
                                             POP eDI
                                                                       ; RECOVER REGEN POINTER
 3475 00002413 47
                                                                        ; POINT TO NEXT CHAR POSITION
                                   <1>
                                             INC
                                                    eDI
                                             LOOP S1
 3476 00002414 E2E2
                                   <1>
                                                                         ; MORE CHARS TO WRITE
 3477 00002416 C3
                                   <1>
                                             retn
 3478
                                   <1>
                                    <1> S5:
 3479
 3480 00002417 3207
                                              XOR AL, [eDI]
                                                                        ; EXCLUSIVE OR WITH CURRENT
                                   <1>
 3481 00002419 AA
                                   <1>
                                              STOSB
                                                                        ; STORE THE CODE POINT
                                                                       ; AGAIN FOR ODD FIELD
 3482 0000241A AC
                                    <1>
                                              LODSB
 3483 0000241B 3287FF1F0000
                                                    AL, [eDI+2000H-1]
                                   <1>
                                              XOR
 3484 00002421 EBE1
                                   <1>
                                                                        ; BACK TO MAINSTREAM
                                                    short S4
 3485
                                   <1>
                                                    MEDIUM RESOLUTION WRITE
 3486
                                   <1> ;----
                                                                ; MED_RES_WRITE
                                  <1> S6:
                                                   DL, BL ; SAVE HIGH COLOR BIT
DI, 1 ; OFFSET*2 SINCE 2 BYTES/CHAR
; EXPAND BL TO FULL WORD OF COLOR
BL, 3 ; ISOLATE THE COLOR BITS ( LOW 2 BITS )
AL, 055H ; GET BIT CONVERSION MULTIPLIER
BL ; EXPAND 2 COLOR BITS TO 4 REPLICATIONS
BL, AL ; PLACE BACK IN WORK REGISTER
BH, AL ; EXPAND TO 8 REPLICATIONS OF COLOR BITS
edi, 0B8000h
                              <1>
<1>
<1>
 3488 00002423 88DA
                                             MOV
3489 00002425 66D1E7
                                             SAL
                                                                                 ; MED CHAR
                                                               ; SAVE REGEN POINTER
; SAVE THE CODE POINTER
; NUMBER OF LOOPS
                                             PUSH eDI
                                             PUSH eSI
                                <1>
 3500 0000243B B604
                                                    DH, 4
                                             VOM
                                  <1> S8:
 3501
                                             LODSB ; GET CODE POINT

CALL S21 ; DOUBLE UP ALL THE BITS

AND AX, BX ; CONVERT TO FOREGROUND COLOR ( 0 BACK )

XCHG AH, AL ; SWAP HIGH/LOW BYTES FOR WORD MOVE
3502 0000243D AC

3503 0000243E E8B3000000 <1>

3504 00002443 6621D8 <1>

3504 00002445 86E0 <1>
```

```
3506 00002448 F6C280
                                      TEST DL, 80H
                           <1>
                                                                    ; IS THIS XOR FUNCTION
                                       JZ short S9
XOR AX, [eDI]
3507 0000244B 7403
                              <1>
                                                              ; NO, STORE IT IN AS IS
                             <1>
3508 0000244D 663307
                                                              ; DO FUNCTION WITH LOW/HIGH
3509
                             <1> S9:
3510 00002450 668907
                                       VOM
                                                              ; STORE FIRST BYTE HIGH, SECOND LOW
                              <1>
                                            [eDI], AX
3511 00002453 AC
                              <1>
                                       LODSB
                                                              ; GET CODE POINT
                             3512 00002454 E89D000000
3513 00002459 6621D8
3514 0000245C 86E0
3515 0000245E F6C280
3516 00002461 7407
3517 00002463 66338700200000
                                                               ; FUNCTION WITH FIRST HALF LOW
                             <1> _S10:
3518
                             [eDI+2000H], AX ; STORE SECOND PORTION HIGH
3519 0000246A 66898700200000
3520 00002471 6683C750
                                                           ; POINT TO NEXT LOCATION
                                            DI, 80
3521 00002475 FECE
                                            short S8
3522 00002477 75C4
                                                            ; KEEP GOING
                                                             ; RECOVER CODE POINTER
3523 00002479 5E
3524 0000247A 5F
                                                              ; RECOVER REGEN POINTER
                                                              ; POINT TO NEXT CHAR POSITION
3525 0000247B 47
3526 0000247C 47
3527 0000247D E2BA
                                                              ; MORE TO WRITE
                                      LOOP S7
3528 0000247F C3
                              <1>
                                      retn
3529
                              <1>
                              <1> ; 04/07/2016
3530
                              <1> ; 01/07/2016
3531
3532
                              <1>; 30/06/2016 - TRDOS 386 (TRDOS v2.0)
                              <1>; VIDEO1.ASM - 24/03/1985 (IBM PC-AT BIOS source code)
3533
3534
                              <1> ;-----
                              <1> ; GRAPHICS READ
3535
3536
                              <1> ;-----
3537
                              <1> GRAPHICS READ:
                             3538 00002480 E8A3000000
3539 00002485 89C6
3540 00002487 81C600800B00
3541 0000248D 83EC08
3542 00002490 89E5
3543
                             <1>;----
                              <1>
3544
                                             DETERMINE GRAPHICS MODES
dh, 4 ; number of passes ; 01/07/2016
                                            byte [CRT_MODE], 6
                                            short S12 ; MEDIUM RESOLUTION
3549
                              <1> ;----
                                             HIGH RESOLUTION READ
                              <1>;----
3550
                                             GET VALUES FROM REGEN BUFFER AND CONVERT TO CODE POINT
                              <1> ; MOV
3551
                                            DH, 4
                                                             ; NUMBER OF PASSES
                              <1> S11:
3552
AL, [eSI] ; GET FIRST BYTE
[eBP], AL ; SAVE IN STORAGE AREA
eBP ; NEXT LOCATION
AL, [eSI+2000H] ; GET LOWER REGION BYTE
[eBP], AL ; ADJUST AND STORE
                                            eBP
                                            eSI, 80
                                            ; POINTER I
; LOOP CONTROL
short S11 ; DO IT SOME MORE
SHORT S14 : CO MARGINE
                                                                    ; POINTER INTO REGEN
3562 000024B4 EB1D
                              <1>
                                                              ; GO MATCH THE SAVED CODE POINTS
3563
                              <1>
3564
                              <1> ;----
                                             MEDIUM RESOLUTION READ
                              <1> S12:
3565
3566 000024B6 66D1E6
                              <1>
                                       SAL SI, 1
                                                             ; OFFSET*2 SINCE 2 BYTES/CHAR
                                                     ; NUMBER OF PASSES
3567
                              <1>
                                       ; MOV DH, 4
                                            ; GET BYTES FROM REGEN INTO SINGLE SAVE
eSI, 2000H-2; GO TO LOWER REGION
s23; GET BUILD TO THE
3568
                              <1> S13:
                             <1> CALL S23
<1> ADD eSI,
<1> CALL S23
<1> SUB eSI,
<1> DEC DH
3569 000024B9 E84D000000
3570 000024BE 81C6FE1F0000
3571 000024C4 E842000000
                                                               ; ADJUST POINTER BACK INTO UPPER
                                            eSI, 2000H-80+2
3572 000024C9 81EEB21F0000
3573 000024CF FECE
                                      DEC
                                            DH
                              <1>
                                            short S13 ; KEEP GOING UNTIL ALL 8 DONE
3574 000024D1 75E6
                              <1>
                                       JNZ
3575
                              <1>
                                             SAVE AREA HAS CHARACTER IN IT, MATCH IT
3576
                              <1> ;----
                              <1> S14:
                                                ; FIND CHAR
                                             eDI, CRT_CHAR_GEN ; ESTABLISH ADDRESSING
3578 000024D3 BF[3C2D0100]
                                       MOV
                              <1>
3579 000024D8 83ED08
                              <1>
                                       SUB
                                            eBP, 8 ; ADJUST POINTER TO START OF SAVE AREA
3580 000024DB 89EE
                              <1>
                                       MOV
                                            eSI, eBP
                              <1> S15:
3581
3582 000024DD 66B80001
                              <1>
                                            ax, 256
                                                                    ; NUMBER TO TEST AGAINST
                                       mov
3583
                              <1> S16:
                                                              ; SAVE SAVE AREA POINTER
                              <1>
3584 000024E1 56
                                       PUSH eSI
                                                              ; SAVE CODE POINTER
3585 000024E2 57
                                       PUSH eDI
                              <1>
                                                        ; SAVE CODE POINTER
; NUMBER OF WORDS TO MATCH
                                       ;MOV eCX, 4
3586
                              <1>
                                                            ; COMPARE THE 8 BYTES AS WORDS
                                       ; REPE CMPSW
3587
                              <1>
                                       cmpsd
3588 000024E3 A7
                              <1>
                                                              ; compare first 4 bytes
3589 000024E4 7501
                              <1>
                                            short S17
                                       jne
3590 000024E6 A7
                              <1>
                                       cmpsd
                                                              ; compare last 4 bytes
3591
                              <1> S17:
3592 000024E7 5F
                             <1>
                                             eDI
                                                              ; RECOVER THE POINTERS
3593 000024E8 5E
                              <1>
                                       POP
                                            eSI
3594
                              <1>
                                       ;JZ
                                            short S18
                                                              ; IF ZERO FLAG SET, THEN MATCH OCCURRED
3595 000024E9 7407
                             <1>
                                             short S18
                                       jе
3596
                              <1>
                                                             ; NO MATCH, MOVE ON TO NEXT
3597 000024EB 83C708
                              <1>
                                       ADD
                                            eDI, 8
                                                              ; NEXT CODE POINT
3598 000024EE 6648
                                                              ; LOOP CONTROL
                              <1>
                                       dec
                                            ax
3599 000024F0 75EF
                              <1>
                                       JNZ
                                            short S16
                                                              ; DO ALL OF THEM
3600
                              <1>
                              <1> ;----
                                             CHARACTER IS FOUND ( AL=0 IF NOT FOUND )
3601
                              <1> S18:
3602
                                       ADD
3603 000024F2 83C408
                                                              ; READJUST THE STACK, THROW AWAY SAVE
                                            eSP, 8
                              <1>
3604 000024F5 C3
                              <1>
                                       retn
                                                              ; ALL DONE
3605
                              <1>
                              <1>; 30/06/2016 - TRDOS 386 (TRDOS v2.0)
3606
3607
                              <1>; VIDEO1.ASM - 24/03/1985 (IBM PC-AT BIOS source code)
3608
                              <1> :------
3609
                              <1> ; EXPAND BYTE
3610
                              <1> ; THIS ROUTINE TAKES THE BYTE IN AL AND DOUBLES ALL
```

```
<1>; OF THE BITS, TURNING THE 8 BITS INTO 16 BITS.
3611
3612
                                  <1> ; THE RESULT IS LEFT IN AX
3613
                                  <1> ;-----
                                  <1> S21:
3614
                                                               ; SAVE REGISTER
                                           PUSH CX;MOV CX, 8
3615 000024F6 6651
                                  <1>
3616
                                  <1>
                                                                      ; SHIFT COUNT REGISTER FOR ONE BYTE
3617 000024F8 B108
                                 <1>
                                           mov cl, 8
                               <1> S22:
<1> ROR AL,1
<1> RCR BP,1
<1> SAR BP,1
<1> ;LOOP S22
<1> dec cl
<1> jnz short S22
<1> XCHG AX, BP
<1> POP CX
<1> RETn
                                 <1> S22:
3618
                                                          ; SHIFT BITS, LOW BIT INTO CARRY FLAG
3619 000024FA D0C8
                                                                     ; MOVE CARRY FLAG (LOW BIT INTO RESULTS
3620 000024FC 66D1DD
3621 000024FF 66D1FD
                                                                    ; SIGN EXTEND HIGH BIT (DOUBLE IT)
3622
                                                                     ; REPEAT FOR ALL 8 BITS
3623 00002502 FEC9
3624 00002504 75F4
                                                                     ; MOVE RESULTS TO PARAMETER REGISTER
3625 00002506 6695
3626 00002508 6659
                                                                      ; RECOVER REGISTER
3627 0000250A C3
                                                                      ; ALL DONE
3628
3629
                                  <1>; 01/07/2016 - TRDOS 386 (TRDOS v2.0)
                                  <1>; VIDEO1.ASM - 24/03/1985 (IBM PC-AT BIOS source code)
3630
3631
                                  <1> ;-----
                                  <1> ; MED READ BYTE
3632
                                  <1>; THIS ROUTINE WILL TAKE 2 BYTES FROM THE REGEN BUFFER,
3633
                                  <1> ; COMPARE AGAINST THE CURRENT FOREGROUND COLOR, AND PLACE
3634
                                  <1>; THE CORRESPONDING ON/OFF BIT PATTERN INTO THE CURRENT
3635
                                  <1> ; POSITION IN THE SAVE AREA
3636
3637
                                  <1> ; ENTRY --
                                  <1> ; SI,DS = POINTER TO REGEN AREA OF INTEREST
3638
3639
                                  <1> ; BX = EXPANDED FOREGROUND COLOR
                                  <1> ; BP = POINTER TO SAVE AREA
3640
3641
                                  <1> ; EXIT --
3642
                                  <1> ; SI AND BP ARE INCREMENTED
                                  <1> ;-----
3643
3644
                                 <1> S23:
                                                                    ; GET FIRST BYTE AND SECOND BYTES
                                 <1>
3645 0000250B 66AD
                                           LODSW
                                           XCHG AL, AH
                                           XCHG AL, AH ; SWAP FOR COMPARE

MOV CX, 0C000H ; 2 BIT MASK TO TEST THE ENTRIES

MOV DI 0 . PESULT PECISTER
3646 0000250D 86C4
                                 <1>
                              <1>
3647 0000250F 66B900C0
                                 <1>
                                           MOV DL, 0
                                                                    ; RESULT REGISTER
3648 00002513 B200
                                           TEST AX, CX ; IS THIS SECTION BACKCROUND?

JZ short S25 ; IF ZERO, IT IS BACKG
3649
                                 <1> S24:
                                 <1>
3650 00002515 6685C8
3651 00002518 7401
                                                                             ; IF ZERO, IT IS BACKGROUND (CARRY=0)
                                 <1>
                                 <1> JZ <1> STC
                                                                     ; WASN'T, SO SET CARRY
3652 0000251A F9
                                 <1> S25:
3653
                          <1> S23.
<1> RCL DL, 1
<1> SHR CX, 2
<1> JNC short S24
<1> MOV [eBP], DL
<1> INC eBP
                                                                    ; MOVE THAT BIT INTO THE RESULT
3654 0000251B D0D2
                                                                   ; MOVE THE MASK TO THE RIGHT BY 2 BITS
3655 0000251D 66C1E902
                                                 short S24
[eBP], DL
3656 00002521 73F2
                                                                     ; DO IT AGAIN IF MASK DIDN'T FALL OUT
                                                                    ; STORE RESULT IN SAVE AREA
3657 00002523 885500
                                                                     ; ADJUST POINTER
3658 00002526 45
                                                                      ; ALL DONE
3659 00002527 C3
                                 <1>
                                           RETn
3660
                                  <1>
3661
                                  <1>; 30/06/2016 - TRDOS 386 (TRDOS v2.0)
3662
                                  <1> ; VIDEO1.ASM - 24/03/1985 (IBM PC-AT BIOS source code)
                                  <1> ;------
3663
3664
                                  <1>; V4 POSITION
3665
                                  <1>; THIS ROUTINE TAKES THE CURSOR POSITION CONTAINED IN
                                  <1> ; THE MEMORY LOCATION, AND CONVERTS IT INTO AN OFFSET
3666
                                  <1>; INTO THE REGEN BUFFER, ASSUMING ONE BYTE/CHAR.
3667
                                  <1> ; FOR MEDIUM RESOLUTION GRAPHICS, THE NUMBER MUST
3668
3669
                                  <1>; BE DOUBLED.
3670
                                  <1> ; ENTRY -- NO REGISTERS, MEMORY LOCATION @CURSOR_POSN IS USED
3671
                                  <1> ; EXIT--
3672
                                  <1> ; AX CONTAINS OFFSET INTO REGEN BUFFER
3673
                                  <1> ;------
3674
                                  <1> S26:
3675 00002528 0FB705[DE580100]
                                 <1> movzx eax, word [CURSOR POSN] ; GET CURRENT CURSOR
                                PUSH eBX ; SAVE REGISTER

1> movzx ebx, al ; SAVE A COPY OF CURRENT CURSOR

1> MOV AL, [CRT_COLS] ; GET BYTES PER COLUMN

1> MUL AH ; MULTIPLY BY ROWS

1> SHL AX, 2 ; MULTIPLY * 4 SINCE 4 ROWS/BYTE

1> ADD eAX, eBX ; DETERMINE OFFSET

1> POP eBX ; RECOVER POINTEP

1> RETN
3676
3677 0000252F 53
3678 00002530 0FB6D8
3679 00002533 A0[245F0000]
3680 00002538 F6E4
3681 0000253A 66C1E002
3682 0000253E 01D8
3683 00002540 5B
3684 00002541 C3
3685
                                  <1>
                                  <1> ; 09/07/2016
3686
3687
                                  <1>; 01/07/2016 - TRDOS 386 (TRDOS v2.0)
                                  <1> ; VIDEO1.ASM - 24/03/1985 (IBM PC-AT BIOS source code)
3688
3689
                                  <1> ;-----
3690
                                  <1> ; SET COLOR
                                  ^-THIS ROUTINE WILL ESTABLISH THE BACKGROUND COLOR, THE OVERSCAN COLOR,
3691
                                  <1>;
                                            AND THE FOREGROUND COLOR SET FOR MEDIUM RESOLUTION GRAPHICS
3692
                                  <1> ; INPUT
3693
3694
                                           (BH) HAS COLOR ID
                                               IF BH=0, THE BACKGROUND COLOR VALUE IS SET
3695
                                  <1>;
3696
                                  <1>;
                                                    FROM THE LOW BITS OF BL (0-31)
3697
                                  <1>;
                                                  IF BH=1, THE PALETTE SELECTION IS MADE
                                  <1>;
3698
                                                      BASED ON THE LOW BIT OF BL:
                                  <1>;
3699
                                                                0 = GREEN, RED, YELLOW FOR COLORS 1,2,3
                                                               1 = BLUE, CYAN, MAGENTA FOR COLORS 1,2,3
3700
                                  <1>;
3701
                                  <1>;
                                           (BL) HAS THE COLOR VALUE TO BE USED
3702
                                  <1> ; OUTPUT
                                  <1>: THE COLOR SELECTION IS UPDATED
3703
3704
                                  <1> ;-----
3705
                                  <1> SET COLOR:
                                           cmp byte [CRT_MODE], 7 ; 09/07/2016
3706 00002542 803D[225F0000]07
                                  <1>
3707 00002549 0F870EF0FFFF
                                  <1>
                                            ja VIDEO_RETURN ; nothing to do for VGA modes
3708
                                  <1>
                                           ;MOV DX, [ADDR 6845]
                                                                           ; I/O PORT FOR PALETTE
3709
                                  <1>
3710
                                                 dx, 3D4h
                                  <1>
                                           ;mov
                                                                     ; OVERSCAN PORT
3711
                                 <1>
                                           ;ADD DX,5
3712 0000254F 66BAD903
                                 <1>
                                           mov
                                                  dx, 3D9h
                                 <1>
3713 00002553 A0[255F0000]
                                           MOV
                                                  AL, [CRT_PALETTE] ; GET THE CURRENT PALETTE VALUE
                                 <1>
3714 00002558 08FF
                                           OR
                                                  BH, BH ; IS THIS COLOR 0?
                                                  short M20
3715 0000255A 7512
                                 <1>
                                           JNZ
                                                                    ; OUTPUT COLOR 1
```

```
3716
                                 <1>
3717
                                 <1> ;----
                                                 HANDLE COLOR 0 BY SETTING THE BACKGROUND COLOR
                                AND AL, 0E0H
<1> AND BL, 01FH
<1> OP
3718
                                                                    ; TURN OFF LOW 5 BITS OF CURRENT
3719 0000255C 24E0
3720 0000255E 80E31F
                                                                    ; TURN OFF HIGH 3 BITS OF INPUT VALUE
3721 00002561 08D8
                                                                     ; PUT VALUE INTO REGISTER
                                <1> M19:
                                                                    ; OUTPUT THE PALETTE
3722
                                <1> OUT
                                                                     ; OUTPUT COLOR SELECTION TO 3D9 PORT
3723 00002563 EE
                                                 DX, AL
3724 00002564 A2[255F0000]
                                 <1>
                                           VOM
                                                  [CRT_PALETTE], AL ; SAVE THE COLOR VALUE
3725 00002569 E9EFEFFFF
                                      JMP
                                 <1>
                                                 VIDEO RETURN
3726
                                 <1>
3727
                                                  HANDLE COLOR 1 BY SELECTING THE PALETTE TO BE USED
                                 <1> ;----
3728
                                 <1>
3729
                                 <1> M20:
                                                                   ; TURN OFF PALETTE SELECT BIT
                                                 AL, ODFH
3730 0000256E 24DF
                                 <1> AND
                               <1> JNC <1> OR JMP
3731 00002570 D0EB
                                                 BL, 1
                                                                     ; TEST THE LOW ORDER BIT OF BL
                                          JNC short M19 ; ALREADY DONE OR AL, 20H ; TURN O
3732 00002572 73EF
3733 00002574 0C20
                                                                      ; TURN ON PALETTE SELECT BIT
3734 00002576 EBEB
                                                 short M19
                                                                    ; GO DO IT
3735
                                 <1>
                                 <1> ; 09/07/2016
3736
3737
                                 <1>; 01/07/2016 - TRDOS 386 (TRDOS v2.0)
                                 <1>; VIDEO1.ASM - 24/03/1985 (IBM PC-AT BIOS source code)
3738
3739
                                  <1>; READ DOT -- WRITE DOT
3740
3741
                                  <1> ; THESE ROUTINES WILL WRITE A DOT, OR READ THE
3742
                                  <1> ; DOT AT THE INDICATED LOCATION
3743
                                  <1> ; ENTRY --
3744
                                  <1>; DX = ROW (0-199)
                                                              (THE ACTUAL VALUE DEPENDS ON THE MODE)
                                  <1>; CX = COLUMN (0-639) ( THE VALUES ARE NOT RANGE CHECKED )
3745
3746
                                  <1>; AL = DOT VALUE TO WRITE (1,2 OR 4 BITS DEPENDING ON MODE,
                                        REQUIRED FOR WRITE DOT ONLY, RIGHT JUSTIFIED) BIT 7 OF AL = 1 INDICATES XOR THE VALUE INTO THE LOCATION
3747
                                  <1>;
                                 <1>;
3748
3749
                                 <1>; DS = DATA SEGMENT
3750
                                  <1>; ES = REGEN SEGMENT
3751
                                 <1>;
3752
                                  <1> ; EXIT
3753
                                 <1>; AL = DOT VALUE READ, RIGHT JUSTIFIED, READ ONLY
3754
                                  <1> ;-
3755
                                 <1>
                                 <1> READ DOT:
3756
                                 <1> ; 09/07/2016
3757
3758 00002578 8A25[225F0000]
                                          mov ah, [CRT_MODE] cmp ah, 7; 6!?
                                <1>
3759 0000257E 80FC07
                                 <1>
3760 00002581 760A
                                 <1>
                                         jna short read_dot_cga
3761
                                 <1>
                                        call vga_read_pixel
; al = pixel value
3762 00002583 E8CB030000
                                 <1>
3763
                                 <1>
3764 00002588 E9D5EFFFFF
                                 <1>
                                         jmp _video_return
3765
                                 <1>
3766
                                 <1> read_dot_cga:
                                       ;je VIDEO_RETURN; 7 cmp ah, 4; graphics?
3767
                                 <1>
3768 0000258D 80FC04
3769 00002590 0F82C7EFFFFF
                                <1>
                                <1>
                                                 VIDEO_RETURN ; no, text mode, nothing to do
                                 <1>
                                                                     ; DETERMINE BYTE POSITION OF DOT
3771 00002596 E855000000
                                <1>
                                          CALL R3
                                        MOV AL, [eSI]
                                                                 ; GET THE BYTE ; MASK OFF THE OTHER BITS IN THE BYTE
3772 0000259B 8A06
                                <1>
                                           AND AL, AH
                                <1>
3773 0000259D 20E0
3774 0000259F D2E0
                                <1>
                                           SHL
                                                 AL, CL
                                                                    ; LEFT JUSTIFY THE VALUE
                                       MOV CL, DH
ROL AL, CL
                                                                    ; GET NUMBER OF BITS IN RESULT
3775 000025A1 88F1
                                <1>
                                                                   ; RIGHT JUSTIFY THE RESULT
3776 000025A3 D2C0
                                <1>
                                                                  ; RETURN FROM VIDEO I/O
                                 <1>
                                          ;JMP VIDEO_RETURN
3778 000025A5 0FB6C0
                                 <1>
                                           movzx eax, al
3779 000025A8 E9B5EFFFFF
                                 <1>
                                           jmp
                                                 _video_return
3780
                                 <1>
3781
                                 <1>; 09/07/2016
                                 <1>; 01/07/2016 - TRDOS 386 (TRDOS v2.0)
3782
                                 <1>; VIDEO1.ASM - 24/03/1985 (IBM PC-AT BIOS source code)
3783
3784
                                 <1>
                                 <1> WRITE_DOT:
3785
                                       ; 09/07/2016
                                        mov ah, [CRT_MODE]
3786
                                 <1>
3787 000025AD 8A25[225F0000]
                                 <1>
3788 000025B3 80FC07
                                                 ah, 7 ; 6!?
                                 <1>
                                          cmp
3789 000025B6 760A
                                 <1>
                                         jna short write_dot_cga
3790
                                 <1>
                                       call vga_write_pixel
jmp VIDEO_RETURN
3791 000025B8 E805030000
                                 <1>
3792 000025BD E99BEFFFFF
                                 <1>
3793
                                 <1>
                                 <1> write dot cga:
3794
3795
                                 <1> ;je VIDEO_RETURN ; 7
3796 000025C2 80FC04
                                 <1>
                                           cmp
                                                 ah, 4 ; graphics ?
3797 000025C5 0F8292EFFFFF
                                 <1>
                                         jb
                                                 VIDEO_RETURN ; no, text mode, nothing to do
3798
                                 <1>
3799
                                 <1>
                                           ; PUSH AX
                                                                   ; SAVE DOT VALUE
                                                                   ; TWICE
3800 000025CB 6650
                                           PUSH AX
                                 <1>
                                           CALL R3
3801 000025CD E81E000000
3802 000025D2 D2E8
                                                                   ; DETERMINE BYTE POSITION OF THE DOT ; SHIFT TO SET UP THE BITS FOR OUTPUT
                                <1>
                                        SHR AL, CL
AND AL, AH
                                <1>
                                                                 ; STRIP OFF THE OTHER BITS
; GET THE CURRENT BYTE
; RECOVER XOR FLAG
; IS IT ON
                              <1>
<1>
3803 000025D4 20E0
3804 000025D6 8A0E
                                                 CL, [eSI]
                                           VOM
3805 000025D8 665B
                                <1>
                                           POP BX
                                           POP BX
TEST BL, 80H
3806 000025DA F6C380
                                <1>
                                                                  ; YES, XOR THE DOT
3807 000025DD 750D
                                <1>
                                           JNZ
                                                 short R2
                                                                    ; SET MASK TO REMOVE THE INDICATED BITS
3808 000025DF F6D4
                                <1>
                                           NOT
                                                 AΗ
                                           AND CL, AH
3809 000025E1 20E1
                                <1>
                                                                   ; OR IN THE NEW VALUE OF THOSE BITS
3810 000025E3 08C8
                                           OR
                                                 AL, CL
                                 <1>
                                 <1> R1:
                                                                     ; FINISH DOT
3811
3812 000025E5 8806
                                <1>
                                                 [eSI], AL
                                                                    ; RESTORE THE BYTE IN MEMORY
                                           ; POP AX
3813
                                 <1>
                                                                    ; RETURN FROM VIDEO I/O
3814 000025E7 E971EFFFFF
                                 <1>
                                           JMP
                                                  VIDEO_RETURN
                                                                     ; XOR DOT
                                 <1> R2:
3815
3816 000025EC 30C8
                                 <1>
                                           XOR
                                                 AL, CL
                                                                    ; EXCLUSIVE OR THE DOTS
                                                                   ; FINISH UP THE WRITING
3817 000025EE EBF5
                                 <1>
                                           JMP
                                                 short R1
3818
                                 <1>
3819
                                 <1>; 01/07/2016 - TRDOS 386 (TRDOS v2.0)
3820
                                 <1>; VIDEO1.ASM - 24/03/1985 (IBM PC-AT BIOS source code)
```

```
3821
                                    <1>
3822
                                    <1>; THIS SUBROUTINE DETERMINES THE REGEN BYTE LOCATION OF THE
3823
3824
                                    <1> ; INDICATED ROW COLUMN VALUE IN GRAPHICS MODE.
3825
                                    <1> ; ENTRY --
3826
                                    <1>; DX = ROW VALUE (0-199)
                                    <1>; CX = COLUMN VALUE (0-639)
3827
                                    <1> : EXIT --
3828
3829
                                    <1> ; SI = OFFSET INTO REGEN BUFFER FOR BYTE OF INTEREST
3830
                                    <1> ; AH = MASK TO STRIP OFF THE BITS OF INTEREST
3831
                                    <1> ; CL = BITS TO SHIFT TO RIGHT JUSTIFY THE MASK IN AH
3832
                                    <1>; DH = # BITS IN RESULT
                                    <1> ; BX = MODIFIED
3833
3834
                                    <1> ;-----
                                    <1> R3:
3835
3836
                                    <1>
                                                 DETERMINE 1ST BYTE IN INDICATED ROW BY MULTIPLYING ROW VALUE BY 40
                                    <1> ;----
3837
                                                    ( LOW BIT OF ROW DETERMINES EVEN/ODD, 80 BYTES/ROW )
3838
                                    <1> ;----
3839
                                    <1>
                                                                                 ; WILL SAVE AL AND AH DURING OPERATION
3840 000025F0 0FB7F0
                                   <1>
                                              movzx esi, ax

3841 000025F3 B028
3842 000025F5 F6E2
                                                                          ; AX= ADDRESS OF START OF INDICATED ROW
3843 000025F7 A808
                                                                          ; TEST FOR EVEN/ODD ROW CALCULATED
3844 000025F9 7404
                                                                         ; JUMP IF EVEN ROW
                                                                        ; OFFSET TO LOCATION OF ODD ROWS ADJUST
3845 000025FB 6605D81F
3846
                                                                          ; EVEN_ROW
                                   <1> XCHG SI, AX <1> add esi, 0B80
3847 000025FF 6696
                                                                          ; MOVE POINTER TO (SI) AND RECOVER (AX)
3848 00002601 81C600800B00
                                              add esi, 0B8000h
                                            MOV
3849 00002607 6689CA
                                   <1>
                                                     DX, CX
                                                                          ; COLUMN VALUE TO DX
3850
                                    <1>
3851
                                    <1> ;----
                                                     DETERMINE GRAPHICS MODE CURRENTLY IN EFFECT
3852
                                    <1>
                                    <1> ; SET UP THE REGISTERS ACCORDING TO THE MODE
3853
3854
                                    <1> ; CH = MASK FOR LOW OF COLUMN ADDRESS ( 7/3 FOR HIGH/MED RES )
3855
                                    <1> ; CL = # OF ADDRESS BITS IN COLUMN VALUE ( 3/2 FOR H/M )
3856
                                    <1> ; BL = MASK TO SELECT BITS FROM POINTED BYTE ( 80H/C0H FOR H/M )
                                    <1> ; BH = NUMBER OF VALID BITS IN POINTED BYTE ( 1/2 FOR H/M )
3857
3858
                                    <1>
MOV BX, 2C0H

SOOU UUUU260E 66B90203 <1> MOV CX, 302H

3861 00002612 803D[225F0000]06 <1> CMP byte [CRT_

3862 00002619 7208 <1> JC short R5

3863 0000261B 66BB8001 <1> MOV BX, 180H

3864 0000261F 66B90307
                                                     BX, 2C0H
                                                                          ; SET PARMS FOR MED RES
                                                     byte [CRT_MODE], 6
                                                                         ; HANDLE IF MED RES
                                                                          ; SET PARMS FOR HIGH RES
3865
                                    <1>
                                                     DETERMINE BIT OFFSET IN BYTE FROM COLUMN MASK
                                    <1> ;----
3866
                                    <1> R5:
3867
3868 00002623 20D5
                                              AND
                                                                          ; ADDRESS OF PEL WITHIN BYTE TO CH
                                   <1>
                                                     CH, DL
3869
                                   <1>
3870
                                   <1> ;----
                                                     DETERMINE BYTE OFFSET FOR THIS LOCATION IN COLUMN
                                   <1>
3871
3872 00002625 66D3EA
                                   <1>
                                              SHR
                                                     DX, CL
                                                                          ; SHIFT BY CORRECT AMOUNT
3873 00002628 6601D6
                                                                          ; INCREMENT THE POINTER
                                   <1>
                                              ADD
                                                     SI, DX
3874 0000262B 88FE
                                   <1>
                                              VOM
                                                     DH, BH
                                                                          ; GET THE # OF BITS IN RESULT TO DH
3875
                                    <1>
                                    <1> ;----
3876
                                                     MULTIPLY BH (VALID BITS IN BYTE) BY CH (BIT OFFSET)
3877
                                    <1>
3878 0000262D 28C9
                                                                        ; ZERO INTO STORAGE LOCATION
                                              SUB
                                    <1>
                                                     CL, CL
3879
                                    <1> R6:
                                  ; LEFT JUSTIFY VALUE IN AL (FOR WRITE)
3880 0000262F D0C8
                                                     AL, 1
                                              ROR
3881 00002631 00E9
                                              ADD
                                                     CL, CH
                                                                        ; ADD IN THE BIT OFFSET VALUE
                                                                         ; LOOP CONTROL
3882 00002633 FECF
                                              DEC
                                                     BH
                                                                         ; ON EXIT, CL HAS COUNT TO RESTORE BITS
3883 00002635 75F8
                                              JNZ
                                                     short R6
                                                                         ; GET MASK TO AH
3884 00002637 88DC
                                              MOV
                                                    AH, BL
                                                                         ; MOVE THE MASK TO CORRECT LOCATION
3885 00002639 D2EC
                                              SHR AH, CL
3886 0000263B C3
                                   <1>
                                              RETn
                                                                          ; RETURN WITH EVERYTHING SET UP
3887
                                    <1>
3888
                                    <1> load_dac_palette:
                                          ; 29/07/2016
3889
                                    <1>
                                              ; 23/07/2016
3890
                                    <1>
3891
                                    <1>
                                            ; 03/07/2016 (TRDOS 386 = TRDOS v2.0)
3892
                                    <1>
                                              ; (set mode vga)
                                              ; derived from 'Plex86/Bochs VGABios' source code
3893
                                    <1>
3894
                                    <1>
                                             ; vgabios-0.7a (2011)
3895
                                    <1>
                                              ; by the LGPL VGABios developers Team (2001-2008)
                                              ; 'vgabios.c', 'load_dac_palette'
3896
                                    <1>
3897
                                    <1>
3898
                                    <1>
                                             ; Oracle VirtualBox 5.0.24 VGABios Source Code
3899
                                    <1>
                                              ; ('vgabios.c', 'vgatables.h', 'vgafonts.h', 'vgarom.asm')
3900
                                    <1>
3901
                                    <1>
                                              ; INPUT -> AH = DAC selection number (3, 2 or 1)
                                              ; OUTPUT \rightarrow ECX = 0, AX = 0
3902
                                    <1>
3903
                                    <1>
                                              ; (Modifed registers: EAX, ECX, EDX, ESI)
                                    <1>
3905 0000263C 66BAC803
                                    <1>
                                                     dx, 3C8h ; VGAREG DAC WRITE ADDRESS
                                              mov
3906 00002640 28C0
                                   <1>
                                              sub
                                                     al, al ; 0
3907 00002642 EE
                                   <1>
                                                     dx, al; 0; color index, always 0 at the beginning
                                              out
                                   <1>
3908 00002643 6642
                                            inc
                                                     dx ; 3C9h ; VGAREG_DAC_DATA
3909 00002645 B900010000
                                   <1>
                                                     ecx, 256 ; always 256*3 values
                                              mov
3910
                                   <1>
                                              ;push esi
3911 0000264A 88E0
                                   <1>
                                            mov
                                                     al, ah
3912 0000264C B43F
                                   <1>
                                              mov
                                                     ah, 3Fh
                                                                 ; 3Fh except DAC selection number 3
3913 0000264E 3C02
                                   <1>
                                              cmp
                                                     al, 2
3914 00002650 7414
                                   <1>
                                                     short l_dac_p_2
                                              jе
3915 00002652 7719
                                   <1>
                                              jа
                                                     short l_dac_p_3
3916 00002654 20C0
                                   <1>
                                              and
                                                     al, al
                                                     short l_dac_p_1
3917 00002656 7507
                                   <1>
                                              jnz
3918
                                   <1> l_dac_p_0:
3919 00002658 BE[FC270100]
                                   <1>
                                              mov
                                                     esi, palette0
3920 0000265D EB15
                                   <1>
                                              qmr
                                                     short l_dac_p_4
3921
                                   <1> l_dac_p_1:
3922 0000265F BE[BC280100]
                                   <1>
                                                     esi, palettel
                                              mov
3923 00002664 EB0E
                                   <1>
                                              jmp
                                                     short l_dac_p_4
                                   <1> 1 dac_p_2:
3925 00002666 BE[7C290100]
                                   <1> mov
                                                     esi, palette2
```

```
3926 0000266B EB07
                               <1>
                                      jmp
                                              short l_dac_p_4
                               <1> 1 dac p 3:
                               <1>
3928 0000266D B4FF
                                               ah, OFFh ; dac registers
                                        mov
3929 0000266F BE[3C2A0100]
                              <1>
                                        mov
                                              esi, palette3
                               <1> 1_dac_p_4:
3930
3931 00002674 AC
                               <1>
                                        lodsb
3932 00002675 EE
                                        out dx, al ; Red
                              <1>
                           3933 00002676 AC
                                        lodsb
3934 00002677 EE
                                        out dx, al; Green
3935 00002678 AC
                                        lodsb
3936 00002679 EE
                                        out dx, al; Blue
3937 0000267A 20E4
                              <1>
                                        and
                                              ah, ah
                                        jz
                              <1>
3938 0000267C 7405
                                              short l_dac_p_5
3939 0000267E FECC
                              <1>
                                        dec
3940 00002680 E2F2
                               <1>
                                        loop l_dac_p_4
3941
                               <1>
                                        ;pop
                                              esi
3942 00002682 C3
                               <1>
                                        retn
                               <1> l_dac_p_5:
3943
                                    ; 29/07/2016
3944
                               <1>
3945 00002683 FEC9
                               <1>
                                        dec cl
                                              short l_dac_p_7
3946 00002685 7407
                               <1>
3947
                               <1>
                                      sub
3948 00002687 28C0
                              <1>
                                              al, al ; 0
3949
                              <1> l_dac_p_6:
                               <1>
                                        out
3950 00002689 EE
                                              dx, al ; outb(VGAREG_DAC_DATA,0);
3951 0000268A EE
                              <1>
                                         out
                                              dx, al
3952 0000268B EE
                              <1>
                                        out
                                              dx, al
3953 0000268C E2FB
                              <1>
                                        loop l_dac_p_6
3954
                               <1> l_dac_p_7:
3955
                               <1> ;pop
                                              esi
3956 0000268E C3
                               <1>
3957
                               <1>
3958
                               <1> gray_scale_summing:
                                    ; 03/07/2016 (TRDOS 386 = TRDOS v2.0)
3959
                               <1>
3960
                               <1>
                                        ; (set_mode_vga)
                                      ; derived from 'Plex86/Bochs VGABios' source code ; vgabios-0.7a (2011)
3961
                               <1>
3962
                                      ; by the LGPL VGABios developers Team (2001-2008)
3963
                               <1>
3964
                               <1>
                                        ; 'vgabios.c', 'biosfn_perform_gray_scale_summing'
3965
                               <1>
3966
                               <1>
                                       ; Oracle VirtualBox 5.0.24 VGABios Source Code
                                        ; ('vgabios.c', 'vgatables.h', 'vgafonts.h', 'vgarom.asm')
3967
                               <1>
                               <1>
3968
3969
                               <1>
3970
                               <1>
                                        ; INPUT -> EBX = Start address (color index <= 255)
3971
                               <1>
                                        ; ECX = Count (<= 256)
                                        ; OUTPUT \rightarrow (E) CX = 0
3972
                               <1>
3973
                                        ; (Modifed registers: EAX, ECX, EDX, EBX)
                               <1>
3974
                               <1>
3975 0000268F 66BADA03
                               <1>
                                        mov dx, 3DAh; VGAREG_ACTL_RESET
3976 00002693 EC
                               <1>
                                        in
                                              al, dx
                                              al, al ; 0
3977 00002694 30C0
                               <1>
                                        xor
                                        mov
                                              dx, 3C0h; VGAREG_ACTL_ADDRESS
3978 00002696 66BAC003
                               <1>
3979 0000269A EE
                               <1>
                                        out dx, al; clear bit 5
3980
                               <1>
                                                   ; (while loading palette registers)
3981
                               <1>
                                        ; set read address and switch to read mode
                               <1> g_s_s_1:
                                             dx, 3C7h ; VGAREG_DAC_READ_ADDRESS
3983 0000269B 66BAC703
                               <1>
                                        mov
3984 0000269F 88D8
                               <1>
                                              al, bl
                                        mov
3985 000026A1 EE
                              <1>
                                        out dx, al
3986
                              <1>
                                      ; get 6-bit wide RGB data values
3987
                               <1>
                                        ; intensity = (0.3*Red) + (0.59*Green) + (0.11*Blue)
                                        ; i = ((77*r + 151*g + 28*b) + 0x80) >> 8;
3988
                              <1>
3989 000026A2 66BAC903
                              <1>
                                      mov dx, 3C9h; VGAREG_DAC_DATA
                                        in
                                              al, dx ; red
3990 000026A6 EC
                               <1>
3991 000026A7 B44D
                               <1>
                                        mov
                                              ah, 77 ; 0.3* Red
3992 000026A9 F6E4
                              <1>
                                        mul
                                              ah
                                      push ax
3993 000026AB 6650
                              <1>
3994 000026AD EC
                               <1>
                                              al, dx ; green
                                        in
3995 000026AE B497
                              <1>
                                              ah, 151 ; 0.59 * Green
                                        mov
3996 000026B0 F6E4
                              <1>
                                        mul
                                              ah
3997 000026B2 6650
                               <1>
                                        push ax
3998 000026B4 EC
                                        in
                                              al, dx ; blue
                              <1>
3999 000026B5 B41C
                              <1>
                                              ah, 28 ; 0.11 * Blue
4000 000026B7 F6E4
                               <1>
                                        mul
                                              ah
4001 000026B9 665A
                              <1>
                                        pop
                                              dx
4002 000026BB 6601D0
                              <1>
                                        add
                                              ax, dx
4003 000026BE 665A
                               <1>
                                        pop
                                              dx
                                              ax, dx
4004 000026C0 6601D0
                               <1>
                                        add
4005 000026C3 66058000
                              <1>
                                        add
                                              ax, 80h
4006 000026C7 B03F
                               <1>
                                        mov
                                              al, 3Fh
4007 000026C9 38C4
                                               ah, al
                               <1>
                                        cmp
                           <1>
4008 000026CB 7602
                                         jna
                                              short g_s_s_2
4009 000026CD 88C4
                              <1>
                                              ah, al
                              <1> g_s_s_2:
<1> mov
4010
4011 000026CF 66BAC803
                                        mov
                                               dx, 3C8h ; VGAREG_DAC_WRITE_ADDRESS
                      <1>
                                              al, bl ; color index
4012 000026D3 88D8
                                        mov
4013 000026D5 EE
4014 000026D6 88E0
4015 000026D8 6642
4016 000026DA EE
4017 000026DB 88E0
4018 000026DD EE
4019 000026DE 88E0
4020 000026E0 EE
4021 000026E1 6649
4022 000026E3 7404
4023 000026E5 FEC3
4024 000026E7 EBB2
                          <1> g_s_s_3:
<1> mov
4025
4026 000026E9 66BADA03
                                              dx, 3DAh ; VGAREG_ACTL_RESET
                              <1>
<1>
4027 000026ED EC
                                        in
                                              al, dx
                           <1> .... <1> mov out
4028 000026EE B020
                                              al, 20h
                                              dx, 3C0h; VGAREG ACTL ADDRESS
4029 000026F0 66BAC003
4030 000026F4 EE
                              <1> out dx, al; 20h \rightarrow set bit 5
```

```
4031
                                 <1>
                                                          ; (after loading palette regs)
4032 000026F5 C3
                                 <1>
                                           retn
                                 <1>
4033
4034
                                 <1> vga_write_char_attr:
                                 <1> vga_write_char_only:
4035
                                           ; 08/07/2016 (TRDOS 386 = TRDOS v2.0)
4036
                                 <1>
4037
4038
                                 <1>
                                          ; derived from 'Plex86/Bochs VGABios' source code
4039
                                 <1>
                                           ; vgabios-0.7a (2011)
                                 <1>
                                           ; by the LGPL VGABios developers Team (2001-2008)
4040
4041
                                 <1>
                                           ; 'vgabios.c', 'biosfn_write_char_attr'
4042
                                  <1>
                                           ; 'biosfn_write_char_only'
4043
                                 <1>
4044
                                 <1>
                                          ; INPUT ->
                                           ; [CRT MODE] = current video mode (>7)
4045
                                 <1>
                                           ; CX = Count of characters to write
4046
                                 <1>
                                           ; AL = Character to write
4047
                                  <1>
                                 <1>
                                           ; BL = Color of character
4048
4049
                                 <1>
                                           ; OUTPUT ->
4050
                                 <1>
                                           ; Regen buffer updated
4051
                                 <1>
4052 000026F6 8A25[225F0000]
                                                  ah, [CRT MODE]
                                 <1>
                                           mov
                                                  dx, [CURSOR_POSN] ; cursor pos for page 0
4053 000026FC 668B15[DE580100]
                                 <1>
                                           mov
                                 <1>
4055 00002703 BE[3E5F0000]
                                 <1>
                                           mov
                                                  esi, vga_modes
4056 00002708 89F7
                                 <1>
                                           mov
                                                  edi, esi
4057 0000270A 83C710
                                 <1>
                                           add
                                                 edi, vga_mode_count
4058
                                 <1> vga_wca_0:
4059 0000270D AC
                                 <1>
                                           lodsb
4060 0000270E 38E0
                                           cmp al, ah; [CRT MODE]
                                <1>
4061 00002710 7405
                                <1>
                                           jе
                                                 short vga_wca_2
4062 00002712 39FE
                                 <1>
                                           cmp
                                                 esi, edi
4063 00002714 72F7
                                 <1>
                                           jb
                                                 short vga_wca_0
4064
                                 <1> vga_wca_1:
4065 00002716 C3
                                           retn ; nothing to do
                                 <1>
4066
                                 <1> vga_wca_2:
4067 00002717 83C64F
                                                 esi, vga memmodel - (vga modes + 1)
                                 <1>
                                           ; [ESI] = VGA memory model number (LINEAR8, PLANAR4, PLANAR1)
4068
                                 <1>
4069
                                 <1>
4070
                                 <1>
                                           ; biosfn_write_char_attr (car,page,attr,count)
4071
                                 <1>
                                           ; AL = car, page = 0, BL = attr, CX = count
                                           cmp byte [esi], PLANAR4
4072 0000271A 803E04
                                 <1>
4073 0000271D 741D
                                 <1>
                                           jе
                                                 short vga_wca_planar
                                           cmp byte [esi], PLANAR1
4074 0000271F 803E03
                                <1>
                                               short vga_wca_planar
4075 00002722 7418
                                 <1>
                                         jе
                                 <1> vga_wca_linear8:
4076
                                       ; while((count-->0) && (xcurs<nbcols))
4077
                                 <1>
4078
                                 <1>
                                           ; CX = count
4079 00002724 6621C9
                                 <1>
                                           and cx, cx
4080 00002727 74ED
                                           jz short vga_wca_1
                                 <1>
4081 00002729 3A15[245F0000]
                                 <1>
                                         cmp dl, [CRT_COLS]
                                         jnb
4082 0000272F 73E5
                                 <1>
                                                 short vga_wca_1
                                           ; write_gfx_char_lin(car,attr,xcurs,ycurs,nbcols);
4083
                                 <1>
4084
                                 <1>
                                          ; AL = car, BL = attr, DL = xcurs, DH = ycurs,
4085
                                 <1>
                                          ; [CRT_COLS] = nbcols
4086 00002731 E81E000000
                                 <1>
                                           call write_gfx_char_lin
4087 00002736 6649
                                 <1>
                                           dec cx; count
                                           inc dl ; xcurs
4088 00002738 FEC2
                                 <1>
4089 0000273A EBE8
                                 <1>
                                                 short vga_wca_linear8
                                           jmp
                                 <1> vga_wca_planar:
4090
                                         ; while((count-->0) && (xcurs<nbcols))
4091
                                 <1>
                                 <1>
                                           ; CX = count
                                           and cx, cx
4093 0000273C 6621C9
                                 <1>
4094 0000273F 74D5
                                 <1>
                                           jz
                                                 short vga_wca_1
                                          cmp dl, [CRT_COLS] jnb short vga_wca_1
4095 00002741 3A15[245F0000]
                                 <1>
4096 00002747 73CD
                                 <1>
4097
                                 <1>
                                           ; write_gfx_char_pl4(car,attr,xcurs,ycurs,nbcols,cheight);
4098
                                 <1>
                                           ; AL = car, BL = attr, DL = xcurs, DH = ycurs,
4099
                                 <1>
                                           ; [CRT_COLS] = nbcols, [CHAR_HEIGHT] = cheight
                                           call write_gfx_char_pl4
4100 00002749 E89D000000
                                 <1>
                                           dec cx; count
4101 0000274E 6649
                                 <1>
4102 00002750 FEC2
                                 <1>
                                           inc
                                                 dl ; xcurs
4103 00002752 EBE8
                                 <1>
                                           jmp
                                                 short vga_wca_planar
4104
                                 <1>
4105
                                 <1> write_gfx_char_lin:
                                         ; 08/08/\overline{2}016
4106
                                 <1>
4107
                                 <1>
                                           ; 31/07/2016
                                           ; 08/07/2016 (TRDOS 386 = TRDOS v2.0)
4108
                                 <1>
4109
                                 <1>
                                 <1>
                                           ; derived from 'Plex86/Bochs VGABios' source code
4110
4111
                                 <1>
                                           ; vgabios-0.7a (2011)
4112
                                  <1>
                                           ; by the LGPL VGABios developers Team (2001-2008)
                                           ; 'vgabios.c', 'write_gfx_char_lin'
4113
                                  <1>
                                  <1>
4114
4115
                                  <1>
                                           ; write_gfx_char_lin(car,attr,xcurs,ycurs,nbcols)
                                           ; INPUT ->
4116
                                 <1>
4117
                                  <1>
                                           ; AL = car, BL = attr, DL = xcurs, DH = ycurs,
4118
                                 <1>
                                           ; [CRT COLS] = nbcols
4119
                                 <1>
                                           ; OUTPUT ->
4120
                                 <1>
                                           ; Regen buffer updated
4121
                                 <1>
4122 00002754 51
                                 <1>
                                           push ecx
4123 00002755 53
                                 <1>
                                           push
                                                  ebx
4124 00002756 52
                                 <1>
                                                edx
4125 00002757 50
                                 <1>
                                           push eax
                                           ; addr=xcurs*8+ycurs*nbcols*64;
4126
                                 <1>
4127
                                 <1>
                                           ; 08/08/2016
4128 00002758 0FB6F0
                                 <1>
                                           movzx esi, al ; car
4129 0000275B 0FB6C6
                                 <1>
                                           movzx eax, dh ; ycurs
4130 0000275E 8A25[245F0000]
                                 <1>
                                           mov ah, [CRT_COLS]; nbcols
4131 00002764 F6E4
                                 <1>
                                           mul
                                                 ah
4132
                                 <1>
                                           ;shl ax, 6; * 64
4133 00002766 66C1E003
                                                 ax, 3 ; * 8
                                 <1>
                                           shl
4134
                                 <1>
                                           ;sub dh, dh
4135
                                 <1>
                                           ;shl dx, 3; xcurs * 8
```

```
;movzx edi, dx
movzx edi, dl
shl
4136
                                   <1>
4137 0000276A 0FB6FA
                                   <1>
4138 00002771 30F6
                                             shl di, 3 ; xcurs * 8
                                   <1>
4139 00002771 30F6
                                  <1>
                                          xor dh, dh
4140 00002773 8A15[265F0000] <1>
4141 00002779 66F7E2 <1>
                                             mov dl, [CHAR_HEIGHT]
mul dx
4141 00002779 66F7E2

    mul    dx
    ; eax = ycurs*nbcols*8*[CHAR_HEIGHT]
    add    edi, eax; addr
    add    edi, 0A0000h
    ;shl    si, 3; car * 8
    xor    ah, ah
    mov    al, [CHAR_HEIGHT]
    mul    si
    mov    ei    ax

4142
4143 0000277C 01C7
4144 0000277E 81C700000A00
4145
4146 00002784 30E4
4147 00002786 A0[265F0000]
4148 0000278B 66F7E6
                                          mov si, ax
4149 0000278E 6689C6
                                   <1>
                                           ;; esi = src = car * 8
4150
                                   <1>
                                              ; esi = src = car * [CHAR HEIGHT]
4151
                                   <1>
                                              ; i = 0
4152
                                   <1>
                                            ;add esi, vgafont8 ; fdata [src+i]
4153
                                   <1>
                                 mov eax, [VGA_INT41]
<1> cmp eax, vgafont16
<1> je short wgfxl
<1> cmp eax, vgafont14
<1> je short wgfxl_0
eax, vgafont14
<1> je short wgfxl_0
esi, vgafont8
<1> jmp
4154
                                   <1>
                                              ; 08/08/2016
                                           mov eax, [VGA_INT43H]
4155 00002791 A1[6A650100]
4156 00002796 3D[3C430100]
4157 0000279B 740F
                                              je short wgfxl 0
4157 0000279B 740F
4158 0000279D 3D[3C350100]
4159 000027A2 7408
4160 000027A4 81C6[3C2D0100]
4161 000027AA EB02
4162
                                   <1> wgfxl_0:
4163 000027AC 01C6
                                   <1>
                                              add esi, eax
4164
                                   <1> wgfxl_1:
4165 000027AE 28FF
                                   <1> sub bh, bh; i = 0
4166
                                   <1> wgfxl_2:
                                   <1> ; for(i=0;i<8;i++)
4167
                       4168 000027B0 57
                                              push edi ; addr
                                   <1>
4169 000027B1 0FB605[245F0000] <1>
4170 000027B8 F6E7
4171 000027BA 66C1E003
4173 000027BE 01C7
4174 000027C0 B180
                             4175
4176
                                           ; for (j=0; j<8; j++)
sub edx, edx; j = 0
4177 000027C2 29D2
4178
                                              mov al, [esi] ; al = fdata[src+i]
4179 000027C4 8A06
                                  <1>
<1>
4180 000027C6 20C8
                                              and al, cl; if (fdata[src+i] & mask)
                                                     short wgfxl_4; data = 0, zf = 1
4181 000027C8 7402
                                              jz
                                           mov al, bl; data = attr;
4182 000027CA 88D8
                                 <1>
                                   <1> wgfxl_4:
4183
                                   <1> ; write_byte(0xa000,dest+j,data);
4184
                                        stosb; dest + j (
;inc dl; j++
;cmp dl, 8
cmp dl, 7
jb short wgfxl_5
pop edi
; 08/00/1
4185 000027CC AA
                                   <1>
                                              stosb ; dest + j (+ 0A0000h)
4186
                                   <1>
4187
                                   <1>
4188 000027CD 80FA07
                                   <1>
                                        pop edi
; 08/08/2016
; cmp bh, 7
; jnb short wgfxl_6
inc bh; i++
cmp bh
4189 000027D0 720E
                                   <1>
4190 000027D2 5F
                                   <1>
4191
                                   <1>
4192
                                   <1>
4193
                                   <1>
4194 000027D3 FEC7
                                   <1>
                                             inc bh; i++
cmp bh, [CHAR_HEIGHT]
4195 000027D5 3A3D[265F0000]
                                   <1>
4196 000027DB 7309
                                   <1>
                                              jnb short wgfxl_6
4197 000027DD 46
                                   <1>
                                              inc
                                                    esi
4198 000027DE EBD0
                                   <1>
                                                     short wgfxl_2
                                              jmp
4199
                                   <1> wgfxl 5:
                                   <1>
4200 000027E0 D0E9
                                                     cl, 1 ; mask >>= 1;
                                              shr
4201 000027E2 FEC2
                                   <1>
                                              inc
                                                    dl ; j++
4202 000027E4 EBDE
                                   <1>
                                              jmp short wgfxl_3
4203
                                   <1> wgfxl_6:
                                   <1> pop
4204 000027E6 58
4205 000027E7 5A
                                   <1>
                                                     edx
                                              pop
4206 000027E8 5B
                                   <1>
                                              pop
                                                     ebx
4207 000027E9 59
                                              pop
                                   <1>
                                                     ecx
4208 000027EA C3
                                   <1>
                                              retn
4209
                                   <1>
4210
                                    <1> write_gfx_char_pl4:
                                           ; 08/08/2016
4211
                                    <1>
4212
                                    <1>
                                              ; 08/07/2016 (TRDOS 386 = TRDOS v2.0)
4213
                                    <1>
4214
                                    <1>
                                              ; derived from 'Plex86/Bochs VGABios' source code
4215
                                    <1>
                                              ; vgabios-0.7a (2011)
                                              ; by the LGPL VGABios developers Team (2001-2008)
4216
                                    <1>
                                              ; 'vgabios.c', 'write_gfx_char_pl4'
4217
                                    <1>
4218
                                    <1>
4219
                                    <1>
                                              ; write_gfx_char_pl4(car,attr,xcurs,ycurs,nbcols,cheight)
4220
                                    <1>
                                              ; INPUT ->
                                              ; AL = car, BL = attr, DL = xcurs, DH = ycurs,
4221
                                    <1>
4222
                                              ; [CRT_COLS] = nbcols, [CHAR_HEIGHT] = cheight
                                    <1>
                                              ; OUTPUT ->
4223
                                    <1>
4224
                                    <1>
                                              ; Regen buffer updated
4225
                                    <1>
4226 000027EB 51
                                    <1>
                                              push ecx
4227 000027EC 53
                                    <1>
                                              push ebx
4228 000027ED 52
                                    <1>
                                              push edx
4229 000027EE 50
                                    <1>
                                              push
                                    <1> wgfxpl f0:
4230
                                              ; switch(cheight)
4231
                                    <1>
                                              mov ah, [CHAR HEIGHT]
4232 000027EF 8A25[265F0000]
                                    <1>
4233 000027F5 80FC10
                                    <1>
                                                     ah, 16 ; case 16:
                                              cmp
4234 000027F8 7507
                                    <1>
                                              jne
                                                     short wgfxpl_f1
4235
                                   <1>
                                              ; fdata = &vgafont16;
4236 000027FA BE[3C430100]
                                   <1>
                                              mov esi, vgafont16
4237 000027FF EB13
                                    <1>
                                              jmp
                                                     short wgfxpl_f3
                                   <1> wgfxpl_f1:
4238
4239 00002801 80FC0E
                                    <1>
                                                      ah, 14 ; case 14:
                                              cmp
4240 00002804 7507
                                    <1>
                                                     short wgfxpl_f2
                                              jne
```

```
mo.
jmp
4241 00002806 BE[3C350100]
                                 <1>
                                               mov esi, vgafont14
4242 0000280B EB07
                                    <1>
                                                      short wgfxpl f3
                                    <1> wgfxpl f2:
4243
4244
                                    <1> ; default:
4245
                                    <1>
                                                ; fdata = &vgafont8;
4246 0000280D BE[3C2D0100]
                                    <1>
                                                mov esi, vgafont8
                                             mov ah, 8
4247 00002812 B408
                                    <1>
                                    <1> wgfxpl_f3:
4248
                                           ; al = car
4249
                                    <1>
                                          mul ah; ah = cheight
and eax, OFFFFh; car * cheight
; src = car * cheight;
add esi, eax; esi = fdata[src+i]
; addr=xcurs*8+ycurs*nbcols*64;
mov al, dh; ycurs
mov ah, [CRT_COLS]; nbcols
mul ah
; 08/08/2016
; shl ax, 6; * 64
4250 00002814 F6E4
                                    <1>
                                               mul ah ; ah = cheight
                             <1>
<1>
4251 00002816 25FFFF0000
                                    <1>
                                    <1>
4253 0000281B 01C6
4254
                                    <1>
4255 0000281D 88F0
                                    <1>
4256 0000281F 8A25[245F0000]
                                    <1>
4257 00002825 F6E4
                                    <1>
                                     <1>
4258
                                              ;shl ax, 6; * 64
shl ax, 3; * 8
4259
                                     <1>
4260 00002827 66C1E003
                                    <1>
                                            ;sub dh, dh; 0; shl dx, 3; xcurs * 8; movzx edi, dx
4261
                                    <1>
                                    <1>
4262
                                    <1>
4263
                                               ;movzx edi, dx
                          <1><1><1><1><1></1>
4264 0000282B 0FB6FA
                                               movzx edi, dl
                                               shl di, 3; xcurs * 8 xor dh, dh
4265 0000282E 66C1E703
4266 00002832 30F6
                                            mov dl, [CHAR_HEIGHT]
4267 00002834 8A15[265F0000] <1>
                                           mul dx
; eax = ycurs*nbcols*8;
add edi, eax; addr
add edi, 0A0000h
4268 0000283A 66F7E2
                                    <1>
4269
                                    <1>
                                               ; eax = ycurs*nbcols*8*[CHAR_HEIGHT]
4270 0000283D 01C7 <1>
4271 0000283F 81C700000A00 <1>
4272 <1>
                                    <1>
4272
                                               ;
                                          ; outw(VGAREG_SEQU_ADDRESS, UxUIUZ);
; outw(VGAREG_GRDC_ADDRESS, 0x0205);
mov dx, 3C4h; VGAREG_SEQU_ADDRESS
mov ax, 0F02h
out dx, ax
mov dx, 3CEh; VGAREG_GRDC_ADDRESS
mov ax, 0205h
out dx, ax
                                               ; outw(VGAREG_SEQU_ADDRESS, 0x0f02);
4273
                            <1>
4274
4275 00002845 66BAC403
4276 00002849 66B8020F
4277 0000284D 66EF
                            <1><1>
4278 0000284F 66BACE03
4279 00002853 66B80502
4280 00002857 66EF
                               cut dx, ax
<1>
;
<1>
mov dx, 3CEh; VGAREG_GRDC_i
<1>
test bl, 80h; if(attr&0x80)
<1>
jz short wgfxpl_f4; else
<1>
; outw(VGAREG_GRDC_ADDRESS, 0x
<1>
mov ax, 1803h
<1>
jmp short wgfxpl_f5
                                    <1>
4281
                                                       dx, 3CEh ; VGAREG_GRDC_ADDRESS
4282 00002859 66BACE03
4283 0000285D F6C380
4284 00002860 7406
4285
                                               ; outw(VGAREG_GRDC_ADDRESS, 0x1803);
4286 00002862 66B80318
4287 00002866 EB04
                                    <1> wgfxpl_f4:
4288
4289
                                    <1>
                                                ; outw(VGAREG_GRDC_ADDRESS, 0x0003);
                                                mov ax, 0003h
4290 00002868 66B80300
                                    <1>
                                    <1> wgfxpl f5:
4291
4292 0000286C 66EF
                                     <1>
                                                out
                                                       dx, ax
4293
                                    <1>
4294 0000286E 28FF
                                    <1>
                                                sub bh, bh; i = 0
4295
                                     <1> wgfxpl_0:
                                    <1> ; for(i=0;i<cheight;i++)
<1> push edi; addr
4296
4297 00002870 57
                                               push edi ; addr
                                    <1>
                                            movzx eax, byte [CRT_COLS]; nbcols
mul bh; nbcols*i
; dest=addr+i*nbcols
4298 00002871 0FB605[245F0000] <1>
4299 00002878 F6E7
                                    <1>
4300
                                    <1>
4301 0000287A 01C7
                                    <1>
                                              add edi, eax ; dest
4302 0000287C B580
                                    <1>
                                               mov
                                                      ch, 80h ; mask = 0x80;
4303
                                    <1>
                                                ; for(j=0;j<8;j++)
4304 0000287E 28C9
                                    <1>
                                                sub cl, cl; j = 0
4305
                                     <1> wgfxpl_1:
4306 00002880 D2ED
                                     <1>
                                               shr
                                                      ch, cl; mask=0x80>>j;
4307
                                     <1>
                                              ; outw(VGAREG_GRDC_ADDRESS, (mask << 8) | 0x08);</pre>
4308
                                     <1>
4309
                                     <1>
                                                ; read_byte(0xa000,dest);
4310
                                    <1>
                                                ;mov dx, 3CEh ; VGAREG_GRDC_ADDRESS
4311 00002882 88EC
                                    <1>
                                                mov ah, ch
4312 00002884 B008
                                     <1>
                                                mov
                                                       al, 8
                                                     dx, ax
4313 00002886 66EF
                                    <1>
                                                out
4314 00002888 8A07
                                    <1>
                                                mov al, [edi]; ? (io delay?)
4315
                                     <1>
4316 0000288A 28C0
                                    <1>
                                               sub
                                                      al, al ; attr = 0
4317
                                    <1>
                                                ; if (fdata[src+i] & mask)
4318 0000288C 842E
                                    <1>
                                                test byte [esi], ch
4319 0000288E 7404
                                     <1>
                                                       short wgfxpl_2; zf = 1
                                                jz
                                                ; write_byte(0xa000,dest,attr&0x0f);
                                    <1>
4320
4321 00002890 88D8
                                     <1>
                                                mov al, bl; attr;
                                                                      ; attr&0x0f
4322 00002892 240F
                                     <1>
                                                and
                                                       al, OFh
4323
                                     <1> wgfxpl_2:
                                               ; write_byte(0xa000,dest,0x00);
4324
                                     <1>
                                                mov [edi], al ; dest (+ 0A0000h)
4325 00002894 8807
                                     <1>
4326 00002896 FEC1
                                     <1>
                                                inc
                                                       cl ; j++
4327 00002898 80F908
                                    <1>
                                                cmp cl, 8
                                             jb
4328 0000289B 72E3
                                    <1>
                                                       short wgfxpl_1
                                             pop
4329 0000289D 5F
                                     <1>
                                                       edi
                                               ; 08/08/2016
4330
                                     <1>
4331
                                             ;cmp bh, 7
                                     <1>
                                             ;jnb short wgfxpl_3
inc bh; i++
4332
                                     <1>
                                                       bh ; i++
4333 0000289E FEC7
                                    <1>
4334 000028A0 3A3D[265F0000]
                                    <1>
                                                       bh, [CHAR_HEIGHT]
                                              cmp
                                             jnb
4335 000028A6 7303
                                     <1>
                                                       short wgfxpl_3
4336 000028A8 46
                                    <1>
                                                inc
                                                       esi
                                                       short wgfxpl 0
4337 000028A9 EBC5
                                    <1>
                                                jmp
4338
                                    <1> wgfxpl_3:
4339
                                    <1>
                                                ;mov
                                                       dx, 3CEh ; VGAREG_GRDC_ADDRESS
4340 000028AB 66B808FF
                                                       ax, 0FF08h
                                    <1>
                                                mov
4341 000028AF 66EF
                                    <1>
                                                out
                                                       dx, ax
                                   <1>
<1>
4342 000028B1 66B80500
                                                mov
                                                       ax, 0005h
4343 000028B5 66EF
                                               out
                                                       dx, ax
                                                       ax, 0003h
4344 000028B7 66B80300
                                    <1>
4345 000028BB 66EF
                                     <1>
                                               out
                                                       dx, ax
```

```
4346
                                <1>
4347 000028BD 58
                                <1>
                                                eax
                                         pop
4348 000028BE 5A
                                <1>
                                         pop
                                                edx
4349 000028BF 5B
                                <1>
                                         pop
                                                ebx
4350 000028C0 59
                                <1>
                                         pop
                                                ecx
4351 000028C1 C3
                                <1>
                                          retn
                                <1>
4353
                                <1> vga_write_pixel:
4354
                                <1>
                                         ; 09/07/2016 (TRDOS 386 = TRDOS v2.0)
4355
                                <1>
4356
                                <1>
                                         ; derived from 'Plex86/Bochs VGABios' source code
                                         ; vgabios-0.7a (2011)
4357
                                <1>
                                         ; by the LGPL VGABios developers Team (2001-2008)
4358
                                <1>
                                         ; 'vgabios.c', 'biosfn write pixel'
4359
                                <1>
4360
                                <1>
                                         ; INPUT ->
4361
                                <1>
                                               DX = row (0-239)
4362
                                <1>
                                         ;
                                               CX = column (0-799)
                                <1>
4363
4364
                                <1>
                                               AL = pixel value
                                         ;
                                               (AH = [CRT_MODE])
4365
                                <1>
                                         ; OUTPUT ->
4366
                                <1>
4367
                                <1>
                                               none
                                         ;
4368
                                <1>
4369 000028C2 88C3
                                <1>
                                               bl, al ; pixel value
                                         ;mov ah, [CRT_MODE]
4370
                                <1>
4371 000028C4 BE[3E5F0000]
                                <1>
                                         mov
                                                esi, vga_modes
4372 000028C9 89F7
                               <1>
                                         mov
                                               edi, esi
4373 000028CB 83C710
                               <1>
                                         add
                                               edi, vga_mode_count
                                <1> vga_wp_0:
4374
4375 000028CE AC
                               <1>
                                         lodsb
4376 000028CF 38E0
                               <1>
                                          cmp al, ah ; [CRT_MODE]
                                                short vga_wp_1
4377 000028D1 7405
                               <1>
                                         jе
4378 000028D3 39FE
                               <1>
                                               esi, edi
                                         cmp
4379 000028D5 72F7
                               <1>
                                               short vga_wp_0
4380 000028D7 C3
                               <1>
                                         retn ; nothing to do
4381
                               <1> vga_wp_1:
4382 000028D8 83C64F
                                               esi, vga memmodel - (vga modes + 1)
                               <1>
4383
                                <1>
                                         ; [ESI] = VGA memory model number (LINEAR8, PLANAR4, PLANAR1)
                                         mov edi, 0A0000h
4384 000028DB BF00000A00
                                <1>
4385
                               <1>
                                         ;
                                               byte [esi], PLANAR4
4386 000028E0 803E04
                               <1>
                                         cmp
4387 000028E3 741D
                                <1>
                                         jе
                                                short vga_wp_planar
4388 000028E5 803E03
                                               byte [esi], PLANAR1
                               <1>
                                         cmp
4389 000028E8 7418
                                       jе
                               <1>
                                               short vga_wp_planar
4390
                                <1> vga_wp_linear8:
                                         ; addr=CX+DX*(read_word(BIOSMEM_SEG,BIOSMEM_NB_COLS)*8);
4391
                                <1>
4392 000028EA 0FB605[245F0000] <1>
                                         movzx eax, byte [CRT_COLS] ; = [VGA_COLS] ; nbcols
4393 000028F1 66C1E003
                                         shl ax, 3; * 8
                               <1>
4394 000028F5 66F7E2
                                <1>
                                         mul
                                               dx
                                         push eax
4395 000028F8 50
                               <1>
                                         ;mov edi, 0A0000h
4396
                               <1>
4397 000028F9 6601CF
                               <1>
                                         add
                                               di, cx
4398 000028FC 58
                               <1>
                                         pop
                                               eax
4399 000028FD 01C7
                               <1>
                                         add
                                               edi, eax ; addr
                               <1>
                                         ; write_byte(0xa000,addr,AL);
4401 000028FF 881F
                               <1>
                                         mov
                                               [edi], bl
4402 00002901 C3
                               <1>
                                         retn
4403
                                <1> vga_wp_planar:
                                      ; addr = CX/8+DX*read_word(BIOSMEM_SEG,BIOSMEM_NB_COLS);
4404
                                <1>
4405 00002902 0FB7C1
                               <1>
                                         movzx eax, cx
4406 00002905 66C1E803
                               <1>
                                         shr ax, 3; CX/8
4407 00002909 50
                                <1>
                                         push eax
4408 0000290A 28E4
                                               ah, ah ; 0
                               <1>
                                         sub
4409 0000290C A0[245F0000]
                               <1>
                                         mov
                                               al, [CRT_COLS] ; = [VGA_COLS] ; nbcols
4410 00002911 66F7E2
                                <1>
                                         mul
                                               dx
4411
                                <1>
                                         ;mov
                                               edi, 0A0000h
4412 00002914 6601C7
                               <1>
                                         add di, ax
4413 00002917 58
                               <1>
                                         pop eax
4414 00002918 01C7
                               <1>
                                               edi, eax ; addr
                                         add
4415 0000291A 80E107
                               <1>
                                         and cl, 7
4416 0000291D B580
                               <1>
                                         mov
                                               ch, 80h ; mask
4417 0000291F D2ED
                                <1>
                                         shr
                                               ch, cl ; mask = 0x80 >> (CX \& 0x07);
4418
                                <1>
4419
                                <1>
                                         ; outw(VGAREG_GRDC_ADDRESS, (mask << 8) | 0x08);</pre>
4420 00002921 66BACE03
                                <1>
                                               dx, 3CEh ; VGAREG_GRDC_ADDRESS
                                         mov
4421 00002925 88EC
                                <1>
                                         mov
                                                ah, ch
4422 00002927 B008
                                <1>
                                         mov
                                               al, 8
4423 00002929 66EF
                                <1>
                                         out
                                               dx, ax
                                         ; outw(VGAREG GRDC ADDRESS, 0x0205);
4424
                                <1>
4425 0000292B 66B80502
                                <1>
                                         mov ax, 0205h
4426 0000292F 66EF
                                <1>
                                         out dx, ax
                                          ; data = read_byte(0xa000,addr);
4427
                                <1>
                                         mov al, [edi]; (delay?)
4428 00002931 8A07
                                <1>
4429
                                <1>
                                         ; if (AL & 0x80)
4430
                                <1>
                                         ; {
                                         ; outw(VGAREG_GRDC_ADDRESS, 0x1803);
4431
                                <1>
                                <1>
                                         ; }
4433 00002933 F6C380
                               <1>
                                        test bl, 80h
4434 00002936 7406
                               <1>
                                               short vga wp 2
                                         jΖ
4435 00002938 66B80318
                               <1>
                                               ax, 1803h
                                         mov
                                               dx, ax
4436 0000293C 66EF
                               <1>
                                         out
                                <1> vga wp 2:
                                         ; write byte(0xa000,addr,AL);
4438
                               <1>
4439 0000293E 881F
                               <1>
                                         mov [edi], bl
4440
                               <1>
                                              dx, 3CEh ; VGAREG_GRDC_ADDRESS
4441
                               <1>
                                         ;mov
4442 00002940 66B808FF
                             <1>
                                               ax, 0FF08h
                                         mov
4443 00002944 66EF
                               <1>
                                         out
                                               dx, ax
4444 00002946 66B80500
                               <1>
                                         mov
                                               ax, 0005h
4445 0000294A 66EF
                               <1>
                                         out
                                               dx, ax
4446 0000294C 66B80300
                               <1>
                                         mov
                                               ax, 0003h
4447 00002950 66EF
                               <1>
                                         out dx, ax
                               <1>
4448
4449 00002952 C3
                               <1>
                                         retn
4450
                                <1>
```

```
4451
                                 <1> vga_read_pixel:
                                       ; 09/07/2016 (TRDOS 386 = TRDOS v2.0)
4452
4453
                                 <1>
4454
                                 <1>
                                          ; derived from 'Plex86/Bochs VGABios' source code
4455
                                 <1>
                                          ; vgabios-0.7a (2011)
4456
                                 <1>
                                          ; by the LGPL VGABios developers Team (2001-2008)
                                          ; 'vgabios.c', 'biosfn read pixel'
4457
                                 <1>
                                 <1>
4458
4459
                                 <1>
                                          ; INPUT ->
                                 <1>
                                                DX = row (0-239)
4460
                                          ;
4461
                                 <1>
                                                 CX = column (0-799)
                                                 (AH = [CRT MODE])
4462
                                 <1>
                                          ; OUTPUT ->
4463
                                 <1>
4464
                                 <1>
                                                AL = pixel value
4465
                                 <1>
4466
                                 <1>
                                          ;mov ah, [CRT_MODE]
4467 00002953 BE[3E5F0000]
                                                 esi, vga modes
                                 <1>
                                          mov
4468 00002958 89F7
                                                 edi, esi
                                 <1>
                                          mov
                                                 edi, vga mode count
4469 0000295A 83C710
                                 <1>
                                          add
4470
                                <1> vga_rp_0:
4471 0000295D AC
                                <1>
                                         lodsb
4472 0000295E 38E0
                                <1>
                                          cmp al, ah; [CRT MODE]
4473 00002960 7405
                                                 short vga_rp_1
                                <1>
                                          jе
4474 00002962 39FE
                                <1>
                                          cmp
                                                esi, edi
4475 00002964 72F7
                                <1>
                                                short vga_rp_0
                                          jb
4476 00002966 C3
                                <1>
                                          retn ; nothing to do
4477
                                <1> vga_rp_1:
4478 00002967 83C64F
                                          add
                                <1>
                                                esi, vga_memmodel - (vga_modes + 1)
4479
                                 <1>
                                          ; [ESI] = VGA memory model number (LINEAR8, PLANAR4, PLANAR1)
4480 0000296A BF00000A00
                                                edi, 0A0000h
                                <1>
                                          mov
4481
                                 <1>
4482 0000296F 803E04
                                 <1>
                                                byte [esi], PLANAR4
                                          cmp
4483 00002972 741D
                                <1>
                                                 short vga_rp_planar
                                          jе
4484 00002974 803E03
                                <1>
                                        cmp
je
                                                byte [esi], PLANAR1
4485 00002977 7418
                                <1>
                                                short vga_rp_planar
4486
                                <1> vga_rp_linear8:
                                         ; addr=CX+DX* (read word(BIOSMEM SEG, BIOSMEM NB COLS)*8);
4487
                                <1>
4488 00002979 0FB605[245F0000] <1>
                                          movzx eax, byte [CRT_COLS] ; = [VGA_COLS] ; nbcols
4489 00002980 66C1E003
                                 <1>
                                          shl ax, 3; * 8
4490 00002984 66F7E2
                                <1>
                                          mul dx
4491 00002987 50
                                        push eax
                                <1>
                                          ;mov edi, 0A0000h
4492
                                <1>
4493 00002988 6601CF
                                          add
                                <1>
                                                di, cx
4494 0000298B 58
                                <1>
                                          pop
                                                eax
                                          add
4495 0000298C 01C7
                                <1>
                                                edi, eax ; addr
4496
                                <1>
                                          ; attr=read_byte(0xa000,addr);
4497 0000298E 8A07
                                <1>
                                          mov al, [edi] ; pixel value
4498 00002990 C3
                                <1>
                                         retn
4499
                                <1> vga_rp_planar:
                                      ; addr = CX/8+DX*read_word(BIOSMEM_SEG,BIOSMEM_NB_COLS);
4500
                                <1>
4501 00002991 0FB7C1
                                <1>
                                          movzx eax, cx
4502 00002994 66C1E803
                                <1>
                                          shr ax, 3; CX/8
4503 00002998 50
                                <1>
                                          push eax
4504 00002999 28E4
                                <1>
                                          sub ah, ah; 0
                                          mov
4505 0000299B A0[245F0000]
                                <1>
                                                al, [CRT_COLS] ; = [VGA_COLS] ; nbcols
4506 000029A0 66F7E2
                                <1>
                                          mul
                                          ;mov edi, 0A0000h
                                <1>
4508 000029A3 6601C7
                                          add di, ax
                                <1>
4509 000029A6 58
                                <1>
                                          pop eax
                                          add edi, eax ; addr
4510 000029A7 01C7
                                <1>
4511 000029A9 80E107
                                <1>
                                          and cl, 7
4512 000029AC B580
                                          mov ch, 80h; mask
shr ch, cl; mask = 0x80 >> (CX & 0x07);
                                <1>
4513 000029AE D2ED
                                <1>
4514
                                <1>
                                          ; attr = 0 \times 00;
                                          xor bl, bl; attr = bl = 0,
xor cl, cl; i = cl = 0
4515 000029B0 30DB
                                 <1>
4516 000029B2 30C9
                                 <1>
4517
                                 <1>
                                          ; for (i=0; i<4; i++)
4518
                                 <1>
                                                ; {
4519
                                 <1>
                                                 ; outw(VGAREG_GRDC_ADDRESS, (i << 8) | 0x04);</pre>
4520
                                 <1>
                                                 ; data = read byte(0xa000,addr) & mask;
4521
                                 <1>
                                                 ; if (data > 0) attr = (0x01 << i);
4522
                                 <1>
                                                 ; }
                                 <1> vga_rp_2:
4523
                                                ah, cl ; i << 8
4524 000029B4 88CC
                                 <1>
                                                al, 4 ; | 0x04
                                 <1>
4525 000029B6 B004
                                          mov
                                                 dx, 3CEh ; VGAREG_GRDC_ADDRESS
4526 000029B8 66BACE03
                                <1>
                                          mov
4527 000029BC 66EF
                                <1>
                                          out
                                                dx, ax
4528
                                <1>
                                          ; data = read_byte(0xa000,addr) & mask;
                                          mov al, [edi]
and al, ch; & mask
4529 000029BE 8A07
                                 <1>
4530 000029C0 20E8
                                 <1>
                                          ; if (data > 0) attr = (0x01 << i);
4531
                                 <1>
4532 000029C2 08C0
                                                 al, al
                                 <1>
                                          or
4533 000029C4 7408
                                <1>
                                          jΖ
                                                 short vga_rp_3; al = 0
4534 000029C6 B701
                                 <1>
                                          mov
                                                bh, 1
                                                bh, cl ; (0x01 << i)
4535 000029C8 D2E7
                                 <1>
                                          shl
                                                 bl, bh; attr |= (0x01 << i)
4536 000029CA 08FB
                                 <1>
                                          or
4537 000029CC 88D8
                                 <1>
                                                al, bl ; pixel value
                                          mov
4538
                                 <1> vga_rp_3:
4539 000029CE C3
                                 <1>
4540
                                 <1>
4541
                                 <1> vga_beeper:
                                        ; 04/08/2016 (TRDOS 386 = TRDOS v2.0)
4542
                                 <1>
                                          sti
4543 000029CF FB
                                 <1>
                                 <1>
                                          ; mov bh, [ACTIVE_PAGE]
4544
4545 000029D0 E9CFF3FFFF
                                 <1>
                                          jmp beeper_gfx
4546
                                 <1>
                                 <1> vga_write_teletype:
4547
                                         ; 09/12/2017
4548
                                 <1>
                                          ; 06/08/2016
4549
                                 <1>
                                          ; 04/08/2016
4550
                                 <1>
                                         ; 01/08/2016
4551
                                 <1>
                                         ; 31/07/2016
4552
                                 <1>
                                          ; 09/07/2016 (TRDOS 386 = TRDOS v2.0)
4553
                                 <1>
4554
                                 <1>
4555
                                 <1>
                                          : derived from 'Plex86/Bochs VGABios' source code
```

```
4556
                                   <1>
                                             ; vgabios-0.7a (2011)
4557
                                    <1>
                                             ; by the LGPL VGABios developers Team (2001-2008)
                                             ; 'vgabios.c', 'biosfn_write_teletype'
4558
                                   <1>
4559
                                   <1>
                                             ; 'biosfn_write_char_only'
4560
                                   <1>
                                             ; INPUT ->
4561
                                   <1>
                                             ; [CRT MODE] = current video mode (>7)
4562
                                   <1>
                                            ; AL = Character to write
                                   <1>
4563
4564
                                   <1>
                                             ; BL = Color of character
                                   <1>
                                             ; OUTPUT ->
4565
4566
                                   <1>
                                             ; Regen buffer updated
4567
                                   <1>
4568
                                   <1>
                                             ; biosfn_write_teletype (car, page, attr, flag)
4569
                                   <1>
                                             ; car = character (AL)
4570
                                   <1>
                                             ; page = 0
                                             ; attr = color (BL)
4571
                                   <1>
                                             ; 'flag' not used
4572
                                   <1>
4573
                                   <1>
4574 000029D5 8A25[225F0000]
                                                    ah, [CRT MODE]
                                   <1>
                                             mov
4575 000029DB 88C7
                                   <1>
                                             mov
                                                    bh, al ; character
4576 000029DD 668B15[DE580100] <1>
                                                    dx, [CURSOR_POSN] ; cursor pos for page 0
                                   <1>
4578 000029E4 BE[465F0000]
                                  <1>
                                                    esi, vga_g_modes
                                             mov
4579 000029E9 89F7
                                  <1>
                                                    edi, esi
                           <1> ada <1> vga_wtty_0: <1> lodsb <1> cmp <1> je <1> cmp
4580 000029EB 83C708
                                                    edi, vga_g_mode_count
4581
4582 000029EE AC
4583 000029EF 38E0
                                             cmp al, ah; [CRT_MODE]
                                <1> je <1> cmp <1> jb
                                                    short vga_wtty_2
4584 000029F1 7405
4585 000029F3 39FE
                                                    esi, edi
4586 000029F5 72F7
                                                    short vga_wtty_0
                                  <1> vga_wtty_1:
4587
                                  <1> retn ; nothing to do
4588 000029F7 C3
4589
                                  <1> vga_wtty_2:
                                  <1> cmp bh, 07h; bell (beep)
<1> je short vga_beeper; to
4590 000029F8 80FF07
4591 000029FB 74D2

<1> je short vga_beeper ; uii
<1> cmp bh, 08h ; backspace
<1> jne short vga_wtty_3
<1> ; if(xcurs>0)xcurs--;
<1> or dl, dl ; xcurs (column)
<1> jz short vga_wtty_1
<1> dec dl ; xcurs--;
<1> jmp short vga_wtty_12
<1> vga_wtty_3;
                                             jе
                                                    short vga_beeper ; u11
4592 000029FD 80FF08
4593 00002A00 7508
4594
4595 00002A02 08D2
4596 00002A04 74F1
4597 00002A06 FECA
4598 00002A08 EB59
4599
                                  <1> vga_wtty_3:
                                  <1> cmp bh, 0Dh ; carriage return (\r)
<1> jne short vga_wtty_4
4600 00002A0A 80FF0D
                          4601 00002A0D 7504
4603 00002A0F 28D2
4604 00002A11 EB50
4606 00002A13 80FF0A
                            jne short vga_wtty_5
<1> ; ycurs++;
<1> inc dh; next row
<1> jmp short vga_wtty_
<1> vga_wtty_5:
<1> cmp bh, 09h; tab stop
<1> jne short vga w++v o
4607 00002A16 7504
4608
4609 00002A18 FEC6
4610 00002A1A EB62
                                                      short vga_wtty_11
4611
4612 00002A1C 80FF09
4613 00002A1F 7527
                             4614 00002A21 88D0
                                                    al, dl
                                             mov
4615
                                             ;cbw
                                        xor
mov
4616 00002A23 30E4
                                                    ah, ah ; 09/12/2017
                                  <1>
<1>
4617 00002A25 B108
                                                    cl, 8
4618 00002A27 F6F1
                                             div
                                                    cl
4619 00002A29 28E1
                                  <1>
                                                    cl, ah
4620
                                  <1>
                                             ;
4621 00002A2B B720
                                  <1>
                                             mov
                                                    bh, 20h ; space
4622
                                  <1> vga_wtty_6: ; tab stop loop
                                         push cx
                                  <1>
4623 00002A2D 6651
                                         push bx
4624 00002A2F 6653
                                  <1>
                           <1>
<1>
4625 00002A31 E812000000
                                             call vga_wtty_8
                                         pop bx; bh = character, bl = color
4626 00002A36 665B
                                  <1>
                                         pop
dec
jz
4627 00002A38 6659
                                   <1>
                                                    CX
4628 00002A3A FEC9
                                  <1>
                                                    cl
4629 00002A3C 7409
                                  <1>
                                                    short vga_wtty_7
                                                    dx, [CURSOR_POSN] ; new cursor position (pg 0)
4630 00002A3E 668B15[DE580100] <1>
                                   <1> mov
<1> jmp
4631 00002A45 EBE6
                                                    short vga_wtty_6
4632
                                   <1> vga wtty 7:
4633 00002A47 C3
                                   <1>
                                           retn
4634
                                   <1>
4635
                                   <1> vga_wtty_8:
                                         add esi, vga_g_memmodel - (vga_g_modes + 1)
4636 00002A48 83C64F
                                   <1>
                                    <1>
                                              ; [ESI] = VGA memory model number (LINEAR8, PLANAR4, PLANAR1)
4638 00002A4B BF00000A00
                                   <1>
                                              mov
                                                     edi, 0A0000h
4639
                                   <1>
4640 00002A50 88F8
                                   <1>
                                                    al, bh ; character
                                              mov
4641
                                   <1>
4642 00002A52 803E04
                                   <1>
                                                    byte [esi], PLANAR4
                                              cmp
4643 00002A55 7414
                                   <1>
                                              jе
                                                     short vga_wtty_planar
4644 00002A57 803E03
                                   <1>
                                                     byte [esi], PLANAR1
                                              cmp
4645 00002A5A 740F
                                                    short vga_wtty_planar
                                   <1>
                                              iе
4646
                                   <1> vga_wtty_linear8:
4647
                                   <1>
                                             ; write_gfx_char_lin(car,attr,xcurs,ycurs,nbcols);
                                              ; AL = car, BL = attr (color), DL = xcurs, DH = ycurs,
4648
                                   <1>
4649
                                   <1>
                                              ; [CRT_COLS] = nbcols
                                              call write_gfx_char_lin
4650 00002A5C E8F3FCFFFF
                                   <1>
                                                   short vga_wtty_9
4651 00002A61 EB0D
                                   <1>
                                              jmp
                                   <1>
4653
                                   <1> vga_wtty_12:
                                             ; 09/07/2016
4654
                                   <1>
4655
                                   <1>
                                              ; set cursor position
4656
                                   <1>
                                             ; NOTE: Hardware cursor position will not be set
4657
                                    <1>
                                                  in any VGA modes (>7)
                                                 But, cursor position will be saved into
4658
                                   <1>
                                                 [CURSOR POSN].
4659
                                    <1>
                                   <1>
                                              ; TRDOS 386 (TRDOS v2.0) uses only one page
4660
```

```
4661
                                          ; (page 0) for all graphics modes.
4662
                                 <1>
4663 00002A63 668915[DE580100]
                                         mov [CURSOR_POSN], dx ; save cursor pos for pg 0
                                 <1>
4664
                                 <1>
                                          ; 04/08/2016
                                          ;mov bh, [ACTIVE_PAGE] ; = 0
4665
                                 <1>
4666
                                 <1>
                                           ;call _set_cpos
4667 00002A6A C3
                                 <1>
                                           retn
4668
                                 <1>
4669
                                 <1> vga_wtty_planar:
4670
                                 <1>
                                          ; write_gfx_char_pl4(car,attr,xcurs,ycurs,nbcols,cheight);
4671
                                 <1>
                                          ; AL = car, BL = attr (color), DL = xcurs, DH = ycurs,
4672
                                 <1>
                                          ; [CRT COLS] = nbcols, [CHAR HEIGHT] = cheight
4673 00002A6B E87BFDFFFF
                                 <1>
                                          call write_gfx_char_pl4
4674
                                 <1> vga_wtty_9:
                                 <1> inc dl; xcurs++;
4675 00002A70 FEC2
4676
                                 <1> vga_wtty_10:
4677
                                 <1> ; Do we need to wrap ?
4678
                                 <1>
                                          ; if(xcurs==nbcols)
                                          cmp dl, [CRT COLS]; [VGA COLS]
4679 00002A72 3A15[245F0000]
                                 <1>
4680 00002A78 7204
                                <1>
                                          jb
                                                 short vga_wtty_11 ; no
4681 00002A7A 28D2
                                 <1>
                                           sub dl, dl; xcurs=0;
4682 00002A7C FEC6
                                 <1>
                                          inc
                                                dh; ycurs++;
                                 <1> vga_wtty_11:
4683
4684
                                 <1> ; Do we need to scroll ?
4685
                                 <1>
                                          ; if(ycurs==nbrows)
4686 00002A7E 3A35[2A5F0000]
                                           cmp dh, [VGA_ROWS]
                                 <1>
                                         jb
4687 00002A84 72DD
                                 <1>
                                                 short vga_wtty_12 ; no
                                         ;
4688
                                 <1>
                                         ; biosfn_scroll (nblines,attr,rul,cul,rlr,clr,page,dir)
4689
                                 <1>
                                          ; al = n\overline{b}lines = 1, bl = attr (color) = 0
4690
                                 <1>
                                          ; ch = rul, cl = cul, dh = rlr, dl = clr, page = 0
4691
                                 <1>
                                          ; dir = SCROLL UP
4692
                                 <1>
4693
                                 <1>
4694 00002A86 B001
                                 <1>
                                                 al, 1
                                          mov
4695 00002A88 28DB
                                 <1>
                                           sub
                                                 bl, bl; 0; blank/black line (attr=0) will be used
4696 00002A8A 6629C9
                                 <1>
                                           sub
                                                cx, cx; 0,0
                                 <1>
                                          ; 06/08/2016
4698
                                 <1>
4699 00002A8D 8A35[2A5F0000]
                                 <1>
                                           mov dh, [VGA_ROWS]
4700 00002A93 FECE
                                 <1>
                                                dh ; nbrows -1
                                           dec
4701
                                 <1>
4702 00002A95 6652
                                 <1>
                                          push dx ; 04/08/2016
4703 00002A97 8A15[245F0000]
                                          mov dl, [CRT_COLS]
                                 <1>
4704 00002A9D FECA
                                           dec dl; nbcols -1
                                 <1>
4705
                                 <1>
4706 00002A9F 8A25[225F0000]
                                          mov ah, [CRT_MODE]
                                 <1>
                                 <1>
4708
                                          ; biosfn_scroll(0x01,0x00,0,0,nbrows-1,nbcols-1,page,SCROLL_UP);
                                 <1>
4709 00002AA5 E808F5FFFF
                                 <1>
                                           call vga graphics up
                                          ; 04/08/2016
4710
                                 <1>
                                          pop dx
4711 00002AAA 665A
                                 <1>
                                          ;dec dh ; ycurs-=1
jmp short vga_wtty_12
4712
                                 <1>
4713 00002AAC EBB5
                                 <1>
4714
                                 <1>
4715
                                 <1> font_setup:
4716
                                 <1>
                                          ; 09/07/2016
4717
                                 <1>
                                           ; character generator (font loading) functions
4718
                                 <1>
4719
                                 <1>
                                          ; derived from 'Plex86/Bochs VGABios' source code
4720
                                 <1>
                                          ; vgabios-0.7a (2011)
4721
                                 <1>
                                          ; by the LGPL VGABios developers Team (2001-2008)
                                           ; 'vgabios.c', 'int10 func'
4722
                                 <1>
4723
                                 <1>
4724
                                 <1>
                                           ; AX = 1100H ; Load User-Defined Font (EGA/VGA)
4725
                                 <1>
                                          ; BH
4726
                                 <1>
                                                    height of each character (bytes per character definition)
                                             ; (BL font block to load (EGA: 0-3; VGA: 0-7))
4727
                                 <1>
4728
                                 <1>
                                           ; CX number of characters to redefine (<=256)
                                          ; DX ASCII code of the first character defined at ES:BP ; EBP address of font-definition information
4729
                                 <1>
4730
                                 <1>
4731
                                 <1>
                                           ; (in user's memory space)
4732
                                 <1>
4733
                                 <1>
                                          ; case 0x11:
4734
                                 <1>
                                          ; switch(GET_AL())
4735
                                 <1>
                                                ; {
4736
                                 <1>
                                           ; case 0x00:
4737
                                 <1>
                                           ; case 0x10:
                                           ; biosfn load text user pat(GET AL(), ES, BP, CX, DX, GET BL(), GET BH());
4738
                                 <1>
4739
                                 <1>
4740
                                 <1>
                                           ; AX = 1110H ; Load and Activate User-Defined Font (EGA/VGA)
4741
                                 <1>
4742 00002AAE 08C0
                                 <1>
                                           or al, al; 0
4743 00002AB0 7404
                                 <1>
                                           jz
                                                 short font_setup_0
4744 00002AB2 3C10
                                 <1>
                                                al, 10h
4745 00002AB4 7511
                                 <1>
                                           ine
                                                 short font_setup_1
4746
                                <1> font_setup_0:
4747 00002AB6 E8B7000000
                                <1>
                                          call transfer user fonts
4748 00002ABB 721C
                                <1>
                                                 short font_setup_error
                                           jс
4749 00002ABD E8C2000000
                                           call load text user pat
                                <1>
4750 00002AC2 E996EAFFFF
                                                     VIDEO_RETURN
                                <1>
                                           jmp
4751
                                 <1> font_setup_1:
4752
                                        ; AX = 1101H ; Load ROM 8x14 Character Set (EGA/VGA)
                                 <1>
4753
                                 <1>
                                           ; case 0x01:
                                           ; case 0x11:
4754
                                 <1>
                                           ; biosfn_load_text_8_14_pat(GET_AL(),GET_BL());
4755
                                 <1>
4756
                                 <1>
                                            ; break;
4757 00002AC7 3C01
                                 <1>
                                          cmp al, 1
4758 00002AC9 7404
                                <1>
                                           jе
                                                 short font_setup_2
4759 00002ACB 3C11
                                 <1>
                                           cmp
                                                al, 11h
                                           ine short font_setup_3
4760 00002ACD 7511
                                <1>
4761
                                 <1> font_setup_2:
                                        ; AX = 1111H ; Load and Activate ROM 8x14 Character Set (EGA/VGA)
4762
                                 <1>
                                           ; (BL = font block to load (EGA: 0-3; VGA: 0-7))
4763
                                <1>
4764 00002ACF E8EE010000
                                 <1>
                                           call load_text_8_14_pat
4765 00002AD4 E984EAFFFF
                                 <1>
                                          jmp VIDEO RETURN
```

<1>

```
4766
                                 <1> font_setup_error:
                                           sub eax, eax; 0 -> fonts could not be loaded
4767 00002AD9 29C0
                                 <1>
4768 00002ADB E982EAFFFF
                                 <1>
                                           jmp
                                                  _video_return
4769
                                 <1> font_setup_3:
                                       ; AX = 1102H; Load ROM 8x8 Character Set (EGA/VGA)
4770
                                 <1>
4771
                                 <1>
                                           ; case 0x02:
                                          ; case 0x12:
; biosfn_load_text_8_8_pat(GET_AL(),GET_BL());
; break;
4772
                                 <1>
4773
                                 <1>
4774
                                 <1>
                                          cmp al, 2
4775 00002AE0 3C02
                                 <1>
                                        je
cmp
4776 00002AE2 7404
                                 <1>
                                                 short font_setup_4
                                          cmp al, 12h jne short font_setup_5
4777 00002AE4 3C12
                                 <1>
4778 00002AE6 750A
                                 <1>
4779
                                 <1> font_setup_4:
                                       ; AX = 1112H; Load and Activate ROM 8x8 Character Set (EGA/VGA)
4780
                                 <1>
4781
                                 <1>
                                           ; (BL = font block to load (EGA: 0-3; VGA: 0-7))
4782 00002AE8 E805020000
                                           call load text 8 8 pat
                                 <1>
4783 00002AED E96BEAFFFF
                                 <1>
                                         jmp
                                                     VIDEO RETURN
                                 <1> font setup 5:
4784
                                 <1> ; AX = 1104H; Load ROM 8x16 Character Set (EGA/VGA)
4785
                                           ; case 0x04:
4786
                                 <1>
                                          ; case 0x14:
; biosfn_load_text_8_16_pat(GET_AL(),GET_BL());
4787
                                  <1>
4788
                                 <1>
4789
                                 <1>
4790 00002AF2 3C04
                                 <1>
                                          cmp al, 4
4791 00002AF4 7404
                                 <1>
                                           jе
                                                  short font_setup_6
4792 00002AF6 3C14
                                 <1>
                                          cmp al, 14h
4793 00002AF8 750A
                                 <1>
                                          jne short font_setup_7
                                 <1> font_setup_6:
4794
4795
                                          ; AX = 1114H ; Load and Activate ROM 8x16 Character Set (EGA/VGA)
                                 <1>
4796
                                 <1>
                                           ; (BL = font block to load (EGA: 0-3; VGA: 0-7))
4797 00002AFA E823020000
                                 <1>
                                           call load text 8 16 pat
                                                     VIDEO RETURN
4798 00002AFF E959EAFFFF
                                 <1>
                                           jmp
4799
                                 <1> font setup 7:
                                         ; Note: AX=1120h (Setup INT 1Fh, EXT_PTR) is not needed
4800
                                 <1>
4801
                                 <1>
                                           ; for TRDOS 386 (TRDIOS v2.0) video functionality;
4802
                                           ; because, originally EXT PTR (font address) was used for
                                         ; chars 80h to 0FFh (after the first 128 ASCII char fonts), for
4803
                                  <1>
                                           ; CGA graphics mode; currenty, 'vgafont8' address has 256 chars!
4804
                                  <1>
4805
                                  <1>
                                          ; case 0x20:
4806
                                  <1>
                                          ; biosfn_load_gfx_8_8_chars(ES,BP);
; break;
4807
                                  <1>
4808
                                  <1>
4809
                                  <1>
                                           ; case 0x21:
                                           ; biosfn_load_gfx_user_chars(ES,BP,CX,GET_BL(),GET_DL());
4810
                                  <1>
4811
                                  <1>
                                             ; break;
4812
                                           ; AX = 1121H ; Setup User-Defined Font for Graphics Mode (VGA)
                                           ; BL screen rows code: 00H = user-specified (in DL)
                                  <1>
4813
4814
                                  <1>
                                                                      01H = 14 \text{ rows}
                                                                      02H = 25 \text{ rows}
4815
                                  <1>
4816
                                  <1>
                                                                      03H = 43 \text{ rows}
4817
                                  <1>
                                             ; CX bytes per character definition
4818
                                 <1>
                                             ; DL
                                                    (when BL=0) custom number of character rows on screen
4819
                                 <1>
                                            ; EBP address of font-definition information (user's mem space)
4820
                                 <1>
4821 00002B04 3C21
                                 <1>
                                           cmp al, 21h
                                           jne short font_setup_9
4822 00002B06 751A
                                 <1>
4823
                                 <1>
4824
                                 <1>
                                           ; TRDOS 386 modification !
4825
                                 <1>
                                           ; dh = 0 \rightarrow 256 characters
                                           ; dh = 80h \rightarrow 128 characters
4826
                                 <1>
                                           ; (If DH <> 0 and DH <> 80h -> invalid)
                                 <1>
4828 00002B08 20F6
                                          and dh, dh
                                 <1>
                                           jz
4829 00002B0A 7405
                                 <1>
                                                  short font_setup_8 ; 256 characters
                                           cmp dh, 80h; 128 characters
4830 00002B0C 80FE80
                                 <1>
4831 00002B0F 75C8
                                 <1>
                                           jne
                                                 short font_setup_error ; invalid !
                                 <1> font_setup_8:
4833 00002B11 E85C000000
                                 <1>
                                       call transfer_user_fonts
4834 00002B16 72C1
                                 <1>
                                                 short font_setup_error
                                 <1>
                                           ; ebp = user's font data address in system's memory space
4836 00002B18 E836020000
                                 <1>
                                           call load_gfx_user_chars
4837 00002B1D E93BEAFFFF
                                 <1>
                                           jmp
                                                     VIDEO RETURN
4838
                                 <1> font_setup_9:
                                       ; case 0x22:
4839
                                 <1>
                                           ; biosfn_load_gfx_8_14_chars(GET_BL());
; break;
4840
                                 <1>
4841
                                 <1>
4842 00002B22 3C22
                                 <1>
                                           cmp al, 22h
4843 00002B24 750A
                                 <1>
                                           jne
                                                 short font_setup_10
4844 00002B26 E866020000
                                 <1>
                                           call load_gfx_8_14_chars
4845 00002B2B E92DEAFFFF
                                           jmp
                                                  VIDEO RETURN
                                 <1>
4846
                                  <1> font_setup_10:
                                         ; case 0x23:
4847
                                  <1>
                                             ; biosfn_load_gfx_8_8_dd_chars(GET_BL());
4848
                                  <1>
4849
                                  <1>
4850 00002B30 3C23
                                  <1>
                                           cmp al, 23h
                                                 short font setup 11
4851 00002B32 750A
                                 <1>
                                           jne
                                           call load gfx 8 8 chars
4852 00002B34 E899020000
                                 <1>
4853 00002B39 E91FEAFFFF
                                 <1>
                                            jmp
                                                     VIDEO_RETURN
                                 <1> font setup 11:
4854
4855
                                 <1>
                                           ; case 0x24:
4856
                                 <1>
                                            ; biosfn_load_gfx_8_16_chars(GET_BL());
4857
                                  <1>
                                             ; break;
4858 00002B3E 3C24
                                 <1>
                                           cmp al, 24h
                                                 short font setup 12
4859 00002B40 750A
                                 <1>
4860 00002B42 E8CC020000
                                 <1>
                                           call load gfx 8 16 chars
4861 00002B47 E911EAFFFF
                                 <1>
                                            jmp
                                                     VIDEO_RETURN
                                 <1> font_setup_12:
4863
                                 <1>
                                           ; case 0x30:
                                            ; biosfn_get_font_info(GET_BH(), &ES, &BP, &CX, &DX);
4864
                                 <1>
4865
                                 <1>
                                             ; break;
4866 00002B4C 3C30
                                 <1>
                                           cmp al, 30h
                                           jne short font_setup_13
call get_font_info
4867 00002B4E 750A
                                 <1>
4868 00002B50 E8FF020000
                                 <1>
                                 <1>
                                           ; eax = return value (info: 4 bytes for 4 parms)
4870
                                  <1>
                                           ; eax = 0 -> invalid function (input)
```

```
_video_return
4871 00002B55 E908EAFFFF
                                   <1>
                                                jmp
                                    <1> font_setup_13:
                                   <1> cmp al, 03h; AX = 1103h
<1> jne short font_setup_14
4873 00002B5A 3C03
4874 00002B5C 750D
                                           ; biosfn_set_text_block_specifier:
4875
                                    <1>
4876
                                    <1>
                                              ; BL = font block selector code
                                             ; NOTE: TRDOS 386 only uses and sets font block 0
4877
                                    <1>
                                            ; (It is as BL = 0 for TRDOS 386)
mov dx, 3C4h; VGAREG_SEQU_ADDF
; mov ah, bl
4878
                                    <1>
4879 00002B5E 66BAC403
                                   <1>
                                              mov dx, 3C4h; VGAREG_SEQU_ADDRESS
4880
                                   <1>
4881 00002B62 28E4
                                           sub ah, ah; 0
                                   <1>
4882
                                    <1>
                                              ;mov al, 03h
                                               out dx, ax
4883 00002B64 66EF
                                   <1>
                                                     VIDEO RETURN
4884 00002B66 E9F2E9FFFF
                                   <1>
                                             jmp
4885
                                    <1>
                                    <1> font_setup_14:
4886
4887 00002B6B 29C0
                                   <1> sub eax, eax ; 0 = invalid function
4888 00002B6D E9F0E9FFFF
                                    <1>
                                               jmp _video_return
4889
                                    <1>
4890
                                    <1> transfer user fonts:
4891
                                    <1>
                                            ; 09/07/2016
4892
                                    <1>
                                              ;and ecx, OFFFFh
                                              ; ECX = byte count
4893
                                    <1>
4894
                                    <1>
                                            ;push ecx
                                           mov esi, ebp; user buffer
4895 00002B72 89EE
                                    <1>
4896 00002B74 BF00000700
                                    <1>
                                              mov
                                                      edi, Cluster_Buffer ; system buffer
                                             call transfer_from_user_buffer
4897 00002B79 E8C9BC0000
                                    <1>
4898
                                            ;pop ecx
                                    <1>
4899
                                    <1>
                                              ; ecx = transfer (byte) count = character count
4900 00002B7E BD00000700
                                    <1>
                                              mov ebp, Cluster_Buffer
4901
                                    <1>
                                               ; jc VIDEO_RETURN -> failed
4902 00002B83 C3
                                    <1>
                                              retn
4903
                                    <1>
4904
                                    <1> load text user pat:
                                           ; 26/07/2016
4905
                                    <1>
4906
                                    <1>
                                               ; 09/07/2016
4907
                                    <1>
                                               ; load user defined (EGA/VGA) text fonts
4908
                                    <1>
4909
                                    <1>
                                               ; derived from 'Plex86/Bochs VGABios' source code
                                               ; vgabios-0.7a (2011)
4910
                                    <1>
4911
                                    <1>
                                              ; by the LGPL VGABios developers Team (2001-2008)
                                               ; 'vgabios.c', 'biosfn load text user pat'
4912
                                     <1>
                                    <1>
4913
4914
                                    <1>
                                               ; biosfn_load_text_user_pat (AL,ES,BP,CX,DX,BL,BH)
4915
                                    <1>
                                               ; get_font_access();
4916
                                    <1>
                                               ; blockaddr = ((BL & 0x03) << 14) + ((BL & 0x04) << 11);
4917
                                    <1>
4918
                                    <1>
                                              ; for(i=0;i<CX;i++)
4919
                                    <1>
                                              ; {
                                               ; src = BP + i * BH;
4920
                                    <1>
4921
                                    <1>
                                               ; dest = blockaddr + (DX + i) * 32;
4922
                                    <1>
                                               ; memcpyb(0xA000, dest, ES, src, BH);
4923
                                    <1>
                                               ; }
4924
                                    <1>
                                              ; release_font_access();
4925
                                    <1>
                                              ; if (AL \ge 0 \times 10)
4926
                                    <1>
                                               ; {
4927
                                    <1>
                                               ; set_scan_lines(BH);
4928
                                    <1>
                                              ; }
4929
                                    <1>
4930 00002B84 50
                                               push eax
                                    <1>
4931 00002B85 E83C000000
                                   <1>
                                               call get_font_access
4932 00002B8A 28DB
                                    <1>
                                               sub
                                                     bl, bl; i = 0
                                   <1> ltup_1:
4933
                                  mov al, bl
mul bh
movzx esi, ax
novzx esi, ax
novzx esi, ebp
novzx esi, esi
                                  <1> mov
4934 00002B8C 88D8
4935 00002B8E F6E7
4936 00002B90 0FB7F0
                                 <1>
4937 00002B93 01EE
                                 <1>
4938 00002B95 88D8
4939 00002B97 20E4

4940 00002B99 6601D0

4941 00002B9C 66C1E005

4942 00002BA0 0FB7F8

4943 00002BA3 81C700000A00
4939 00002B97 28E4
                                              add edi, 0A0000h
                                             push ecx
                                   <1>
4945 00002BAA 0FB6CF
                                    <1>
                                              movzx ecx, bh
4946 00002BAD F3A4
                                    <1>
                                               rep
                                                     movsb
4947 00002BAF 59
                                    <1>
                                                      ecx
                                               pop
4948 00002BB0 FEC3
                                    <1>
                                               inc
                                                     bl
4949 00002BB2 38CB
                                    <1>
                                                      bl, cl
                                               cmp
4950 00002BB4 75D6
                                    <1>
                                               jne
                                                     short ltup_1
4951
                                    <1>
4952 00002BB6 E840000000
                                               call
                                    <1>
                                                     release_font_access
4953
                                    <1>
4954 00002BBB 58
                                    <1>
                                              pop
                                                      eax
                                              ; if (AL >= 0 \times 10)
4955
                                    <1>
                                              cmp al, 10h
jb short ltup_2
4956 00002BBC 3C10
                                    <1>
4957 00002BBE 7205
                                   <1>
                                               ; set_scan_lines(BH);
4958
                                   <1>
4959 00002BC0 E875000000
                                   <1>
                                               call set_scan_lines
                                    <1> ltup_2:
4960
4961 00002BC5 C3
                                    <1>
                                               retn
4962
                                    <1>
4963
                                    <1> get_font_access:
                                             ; \overline{0}9/07/2016
4964
                                    <1>
4965
                                    <1>
                                               ; derived from 'Plex86/Bochs VGABios' source code
4966
                                    <1>
                                              ; vgabios-0.7a (2011)
4967
                                    <1>
4968
                                    <1>
                                             ; by the LGPL VGABios developers Team (2001-2008)
                                              ; 'vgabios.c', 'get_font_access'
4969
                                    <1>
4970
                                    <1>
4971
                                              ; get_font_access()
                                    <1>
                                              push edx
mov dx, 3C4h; VGAREG_SEQU_ADDRESS
4972 00002BC6 52
                                    <1>
4973 00002BC7 66BAC403
                                   <1>
4974 00002BCB 66B80001
                                   <1>
                                                     ax, 0100h
4975 00002BCF 66EF
                                    <1>
                                               out dx, ax
```

```
4976 00002BD1 66B80204
4977 00002BD5 66EF
4978 00002BD7 66B80407
4979 00002BDB 66EF
4980 00002BDD 66B80003
4981 00002BE1 66EF
                                              dx, 3CEh ; VGAREG GRDC ADDRESS
4982 00002BE3 66BACE03
4983 00002BE7 66B80402
4984 00002BEB 66EF
4985 00002BED 66B80500
4986 00002BF1 66EF
4987 00002BF3 66B80604
4988 00002BF7 66EF
4989 00002BF9 5A
4990 00002BFA C3
4991
                               <1>
4992
                               <1> release font access:
                               <1>; 29/07/2016
4993
4994
                               <1>
                                         ; 09/07/2016
4995
                               <1>
4996
                                       ; derived from 'Plex86/Bochs VGABios' source code
                               <1>
4997
                                        ; vgabios-0.7a (2011)
                               <1>
4998
                               <1>
                                        ; by the LGPL VGABios developers Team (2001-2008)
4999
                               <1>
                                        ; 'vgabios.c', 'release_font_access'
5000
                               <1>
5001 00002BFB 66BAC403
                               <1>
                                        mov dx, 3C4h; VGAREG_SEQU_ADDRESS
                     mov ax, 0100h
5002 00002BFF 66B80001
                              <1>
5003 00002C03 66EF
5004 00002C05 66B80203
                                              ax, 0302h
5005 00002C09 66EF
                                              ax, 0304h
5006 00002C0B 66B80403
5007 00002C0F 66EF
5008 00002C11 66B80003
                                              ax, 0300h
5009 00002C15 66EF
5010 00002C17 66BACC03
                                              dx, 3CCh; VGAREG_READ_MISC_OUTPUT
5011 00002C1B EC
5012 00002C1C 2401
5013 00002C1E C0E002
5014 00002C21 0C0A
5015 00002C23 88C4
5016 00002C25 B006
                                              dx, 3CEh; VGAREG GRDC ADDRESS
5017 00002C27 66BACE03
5018 00002C2B 66EF
5019 00002C2D 66B80400
                                              ax, 0004h
5020 00002C31 66EF
5021 00002C33 66B80510
                                              ax, 1005h
5022 00002C37 66EF
5023 00002C39 C3
                               <1>
                                        retn
5024
                               <1>
                               <1> set_scan_lines:
5025
5026
                               <1> ; 09/07/2016
5027
                               <1>
                                        ; derived from 'Plex86/Bochs VGABios' source code
5028
                               <1>
5029
                               <1>
                                       ; vgabios-0.7a (2011)
                                      ; by the LGPL VGABios developers Team (2001-2008)
5030
                               <1>
                                       ; 'vgabios.c', 'set_scan_lines'
5031
                               <1>
5032
                               <1>
5033
                               <1>
                                      ; set_scan_lines(lines)
5034
                               <1>
                                        ; BH = lines
                                      ; outb(crtc_addr, 0x09);
mov dx, 3D4h : CPTC
                        5035
                               <1>
5036
                                        mov dx, 3D4h; CRTC_ADDRESS = 3D4h (always) mov al, 09h
5037 00002C3A 66BAD403
5038 00002C3E B009
5039 00002C40 EE
                                        out dx, al
                                      ; crtc_r9 = inb(crtc_addr+1);
inc dx; 3D5h
in al, dx
5040
                        5041 00002C41 6642
5042 00002C43 EC
5043
5044 00002C44 24E0
5045 00002C46 FECF
5046 00002C48 08F8
                               <1>
                                        ; outb(crtc_addr+1, crtc_r9);
5047
5048 00002C4A EE
                               <1>
                                        out dx, al
5049
                               <1>
                                    ; if(lines==8); cmp bh, 8 cmp bh, 7 jne short ssl_1; biosfn set [mail]
                                        ;inc bh
5050
                               <1>
                               <1>
5051
5052 00002C4B 80FF07
                            <1>
5053 00002C4E 7506
                               <1>
5054
                               <1>
                                        ; biosfn_set_cursor_shape(0x06,0x07);
5055 00002C50 66B90706
                               <1>
                                        mov cx, 0607h
5056 00002C54 EB06
                               <1>
                                       jmp short ssl 2
                               <1> ssl_1:
5057
                                         ; biosfn_set_cursor_shape(lines-4,lines-3);
                               <1>
5059 00002C56 88F9
                               <1>
                                         mov cl, bh; lines - 1
                                         mov ch, cl; lines - 1 (16 -> 15)
dec ch; lines - 2 (16 -> 14)
5060 00002C58 88CD
                               <1>
5061 00002C5A FECD
                               <1>
                               <1> ssl 2:
                                        ; CH = start line, CL = stop line
5063
                               <1>
                                         mov ah, 10; 6845 register for cursor set
5064 00002C5C B40A
                               <1>
5065 00002C5E 66890D[3B5F0000]
                                         mov [CURSOR MODE], cx; save in data area
                               <1>
                                         call m16 ; output cx register
5066 00002C65 E812F1FFFF
                               <1>
                                        ; write word (BIOSMEM SEG, BIOSMEM CHAR HEIGHT, lines);
                               <1>
                                        inc bh; lines
mov [CHAR_HEIGHT], bh
5068 00002C6A FEC7
                               <1>
5069 00002C6C 883D[265F0000]
                               <1>
                                        ; outb(crtc addr, 0x12);
                               <1>
5071 00002C72 66BAD403
                                        mov dx, 3D4h; CRTC_ADDRESS
                               <1>
5072 00002C76 B012
                                        mov al, 12h
                               <1>
5073 00002C78 EE
                                         out dx, al
                               <1>
5074
                               <1>
                                         ; vde = inb(crtc_addr+1);
5075 00002C79 6642
                               <1>
                                        inc dx
5076 00002C7B EC
                                        in
                                              al, dx
                               <1>
5077 00002C7C 88C4
                               <1>
                                        mov
                                              ah, al
                              <1>
                                        ; outb(crtc_addr, 0x07);
                                         dec dx
5079 00002C7E 664A
                               <1>
5080 00002C80 B007
                               <1>
                                        mov al, 07h
```

```
5081 00002C82 EE
                                <1>
                                          out dx, al
5082
                                 <1>
                                          ; ovl = inb(crtc addr+1);
5083 00002C83 6642
                                <1>
                                          inc dx
5084 00002C85 EC
                                <1>
                                          in
                                                al, dx
5085
                                <1>
                                          ; vde += (((ovl & 0x02) << 7) + ((ovl & 0x40) << 3) + 1);
5086 00002C86 88E2
                                <1>
                                          mov dl, ah; vde
5087 00002C88 88C6
                                <1>
                                          mov
                                                dh, al ; ovl
5088 00002C8A 6683E002
                                          and ax, 02h
                                <1>
5089 00002C8E 66C1E007
                                <1>
                                          shl
                                                ax, 7
                                          mov
5090 00002C92 6689C1
                                <1>
                                                cx, ax; (ovl & 0x02) << 7)
5091 00002C95 88F0
                                <1>
                                          mov al, dh; ovl
5092 00002C97 6683E040
                                <1>
                                          and
                                                ax, 40h
                                          shl ax, 3 ; (ovl & 0x40) << 3)
5093 00002C9B 66C1E003
                                <1>
5094 00002C9F 6640
                                <1>
                                        inc ax; + 1
                                        add
5095 00002CA1 6601C8
                                <1>
                                                ax, cx
5096 00002CA4 30F6
                                <1>
                                          xor
                                                dh, dh
                                        add ax, dx; + vde
5097 00002CA6 6601D0
                                <1>
5098
                                 <1>
                                          ; rows = vde / lines;
5099 00002CA9 F6F7
                                 <1>
                                          div bh
                                          ;dec al ; rows -1
5100
                                 <1>
                                          ; write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, rows-1);
5101
                                 <1>
5102 00002CAB A2[2A5F0000]
                                          mov [VGA ROWS], al; rows (not 'rows-1'!)
                                 <1>
                                          ; write_word(BIOSMEM_SEG,BIOSMEM_PAGE_SIZE, rows * cols * 2);
5103
                                 <1>
5104 00002CB0 8A25[245F0000]
                                 <1>
                                          mov ah, [CRT_COLS]
5105 00002CB6 F6E4
                                 <1>
                                          mul
                                                 ah
5106 00002CB8 66D1E0
                                 <1>
                                          shl
                                                 ax, 1
5107 00002CBB 66A3[58650100]
                                 <1>
                                          mov
                                                [CRT_LEN], ax
5108 00002CC1 C3
                                 <1>
                                          retn
5109
                                 <1>
5110
                                 <1> load_text_8_14_pat:
5111
                                 <1>
                                         ; 26/07/2016
                                          ; 25/07/2016
5112
                                 <1>
                                          ; 23/07/2016
5113
                                 <1>
5114
                                 <1>
                                         ; 09/07/2016
5115
                                 <1>
                                          ; load user defined (EGA/VGA) text fonts
5116
                                 <1>
                                          ; derived from 'Plex86/Bochs VGABios' source code
5117
5118
                                 <1>
                                          ; vgabios-0.7a (2011)
5119
                                 <1>
                                          ; by the LGPL VGABios developers Team (2001-2008)
                                          ; 'vgabios.c', 'biosfn_load_text_8_14_pat'
5120
                                 <1>
5121
                                 <1>
                                          ; biosfn_load_text_8_14_pat (AL,BL)
5122
                                 <1>
                                 <1>
5123
5124
                                 <1>
                                          ; get_font_access();
                                          ; blockaddr = ((BL & 0x03) << 14) + ((BL & 0x04) << 11);
5125
                                 <1>
5126
                                 <1>
                                          ; for (i=0; i<0x100; i++)
5127
                                 <1>
                                          ; {
                                 <1>
                                          ; src = i * 14;
5128
                                          ; dest = blockaddr + i * 32;
5129
                                 <1>
5130
                                 <1>
                                          ; memcpyb(0xA000, dest, 0xC000, vgafont14+src, 14);
5131
                                 <1>
                                          ; }
5132
                                 <1>
                                          ; release font access();
                                          ; if (AL>=\overline{0}x10)
                                 <1>
5133
5134
                                 <1>
5135
                                 <1>
                                          ; set_scan_lines(14);
5136
                                 <1>
                                          ; }
                                 <1>
5138 00002CC2 50
                                 <1>
                                          push eax
5139 00002CC3 E8FEFEFFF
                                 <1>
                                          call get_font_access
5140
                                 <1>
5141
                                 <1>
                                          ; blockaddr = ((BL & 0x03) << 14) + ((BL & 0x04) << 11);
5142
                                 <1>
                                          ;mov dl, bl
                                          ; and dl, 3
5143
                                 <1>
5144
                                 <1>
                                          ;shl dx, 14
5145
                                 <1>
                                          ;xchg dx, bx
5146
                                 <1>
                                          ;and dl, 4
5147
                                 <1>
                                          ;shl dx, 11
5148
                                 <1>
                                          ;add dx, bx
5149
                                 <1>
5150
                                 <1>
                                          ; xor dx, dx; blockaddr = 0
                                          ; Always block 0 for TRDOS 386 ! (blockaddr=0(
5151
                                 <1>
5152
                                 <1>
5153 00002CC8 28DB
                                                 bl, bl ; i = 0
                                 <1>
                                          sub
5154 00002CCA B70E
                                 <1>
                                                bh, 14
                                                esi, vgafont14
5155 00002CCC BE[3C350100]
                                 <1>
                                          mov
5156 00002CD1 BF00000A00
                                 <1>
                                          mov
                                                 edi, 0A0000h
                                 <1> 1t8 14 1:
5157
5158
                                 <1>
                                          ;mov al, bl
                                          ; mul bh
5159
                                 <1>
5160
                                 <1>
                                          ;movzx esi, ax
5161
                                 <1>
                                          ;add esi, vgafont14
5162
                                 <1>
                                           ;mov
                                                 al, bl
5163
                                 <1>
                                          ;sub ah, ah
                                 <1>
                                          ;shl ax, 5; * 32
                                          ;;add ax, dx; blockaddr + i * 32;
5165
                                 <1>
5166
                                 <1>
                                           ;movzx edi, ax ; dest
5167
                                 <1>
                                          ;add edi, 0A0000h
5168 00002CD6 0FB6CF
                                <1>
                                          movzx ecx, bh
5169 00002CD9 F3A4
                                <1>
                                                movsb
                                          rep
                                                edi, 18 ; 32 - 14
5170 00002CDB 83C712
                                <1>
                                          add
5171 00002CDE FEC3
                                <1>
                                          inc
                                                bl
                                 <1>
5172 00002CE0 75F4
                                          jnz
                                                short lt8_14_1
5173
                                <1>
                                          call release font access
5174 00002CE2 E814FFFFF
                                <1>
5175
                                 <1>
5176 00002CE7 58
                                 <1>
                                          pop
                                 <1>
                                          ; if (AL \ge 0 \times 10)
5178 00002CE8 3C10
                                 <1>
                                          cmp al, 10h
5179 00002CEA 7205
                                 <1>
                                          jb
                                                short lt8_14_4
5180
                                 <1>
                                          ; BH = 14
                                          ; set_scan_lines(14);
5181
                                <1>
5182 00002CEC E849FFFFFF
                                <1>
                                          call set_scan_lines
                                <1> lt8_14_4:
5183
5184 00002CF1 C3
                                 <1>
5185
                                 <1>
```

```
5186
                                   <1> load_text_8_8_pat:
                                          ; 26/07/2016
5187
                                    <1>
                                             ; 25/07/2016
5188
                                   <1>
5189
                                   <1>
                                             ; 23/07/2016
                                            ; 09/07/2016
5190
                                   <1>
5191
                                   <1>
                                              ; load user defined (EGA/VGA) text fonts
5192
                                   <1>
                                            ; derived from 'Plex86/Bochs VGABios' source code
5193
                                   <1>
5194
                                   <1>
                                             ; vgabios-0.7a (2011)
                                   <1>
                                             ; by the LGPL VGABios developers Team (2001-2008)
5195
5196
                                   <1>
                                              ; 'vgabios.c', 'biosfn_load_text_8_8_pat'
                                    <1>
5197
5198
                                   <1>
                                              ; biosfn_load_text_8_8_pat (AL,BL)
5199
                                   <1>
5200
                                   <1>
                                             ; get_font_access();
                                              ; blockaddr = ((BL & 0x03) << 14) + ((BL & 0x04) << 11);
5201
                                   <1>
5202
                                   <1>
                                              ; for (i=0; i<0x100; i++)
5203
                                   <1>
                                             ; {
                                             ; src = i * 8;
5204
                                   <1>
5205
                                   <1>
                                              ; dest = blockaddr + i * 32;
5206
                                   <1>
                                              ; memcpyb(0xA000, dest, 0xC000, vgafont8+src, 8);
5207
                                   <1>
                                             ; }
5208
                                   <1>
                                              ; release_font_access();
5209
                                   <1>
                                              ; if (AL >= 0 \times 10)
5210
                                   <1>
                                              ; {
5211
                                   <1>
                                              ; set_scan_lines(8);
5212
                                   <1>
                                              ; }
5213
                                   <1>
5214 00002CF2 50
                                   <1>
                                              push eax
5215 00002CF3 E8CEFEFFFF
                                   <1>
                                              call get_font_access
5216
                                   <1>
                                              ; blockaddr = ((BL & 0x03) << 14) + ((BL & 0x04) << 11);
5217
                                   <1>
5218
                                   <1>
                                              ;mov dl, bl
5219
                                   <1>
                                              ;and d1, 3
5220
                                   <1>
                                              ;shl dx, 14
5221
                                   <1>
                                              ;xchg dx, bx
5222
                                   <1>
                                              ;and dl, 4
5223
                                   <1>
                                              ;shl dx, 11
5224
                                   <1>
                                              ;add dx, bx
5225
                                   <1>
                                              ; xor dx, dx; blockaddr = 0
5226
                                   <1>
                                              ; Always block 0 for TRDOS 386 ! (blockaddr=0(
5227
                                   <1>
5228
                                   <1>
                                                    bl, bl; i = 0
5229 00002CF8 28DB
                                   <1>
                                              sub
5230 00002CFA B708
                                   <1>
                                              mov
                                                    bh, 8
5231 00002CFC BE[3C2D0100]
                                   <1>
                                              mov
                                                     esi, vgafont8
5232 00002D01 BF00000A00
                                   <1>
                                                    edi, 0A0000h
                                              mov
                                   <1> lt8_8_1:
5233
5234
                                   <1>
                                              ; mov
                                                    al, bl
                                              ;mul bh
5235
                                   <1>
5236
                                   <1>
                                              ;movzx esi, ax
5237
                                   <1>
                                             ;add esi, vgafont8
5238
                                   <1>
                                             ;mov al, bl
5239
                                   <1>
                                             ;sub ah, ah
5240
                                   <1>
                                             ;shl ax, 5 ; * 32
5241
                                   <1>
                                              ;;add ax, dx; blockaddr + i * 32;
5242
                                   <1>
                                             ;movzx edi, ax ; dest
5243
                                              ;add edi, 0A0000h
                                   <1>
5244 00002D06 0FB6CF
                                   <1>
                                              movzx ecx, bh
5245 00002D09 F3A4
                                   <1>
                                             rep movsb
5246 00002D0B 83C718
                                   <1>
                                              add
                                                     edi, 24 ; 32 - 8
5247 00002D0E FEC3
                                   <1>
                                              inc
                                                    bl
5248 00002D10 75F4
                                   <1>
                                                    short lt8_8_1
                                              jnz
5249
                                   <1>
                                              call release_font_access
5250 00002D12 E8E4FEFFFF
                                   <1>
5251
                                   <1>
5252 00002D17 58
                                   <1>
                                              pop
                                                    eax
                                              ; if (AL \ge 0 \times 10)
5253
                                   <1>
5254 00002D18 3C10
                                   <1>
                                                    al, 10h
                                              cmp
5255 00002D1A 7205
                                   <1>
                                              jb
                                                    short lt8_8_2
5256
                                   <1>
                                              ; BH = 8
                                              ; set_scan_lines(8);
call set_scan_lines
5257
                                   <1>
5258 00002D1C E819FFFFF
                                   <1>
5259
                                   <1> lt8_8_2:
5260 00002D21 C3
                                   <1>
                                              retn
5261
                                   <1>
5262
                                   <1> load_text_8_16_pat:
                                            ; 2<del>\overline{0}</del>; 2<del>\overline{0}</del>/<del>0</del>7/<del>2</del>016
5263
                                   <1>
                                             ; 25/07/2016
5264
                                    <1>
5265
                                             ; 23/07/2016
                                   <1>
5266
                                   <1>
                                             ; 09/07/2016
                                              ; load user defined (EGA/VGA) text fonts
5267
                                    <1>
5268
                                   <1>
5269
                                    <1>
                                              ; derived from 'Plex86/Bochs VGABios' source code
5270
                                    <1>
                                              ; vgabios-0.7a (2011)
                                              ; by the LGPL VGABios developers Team (2001-2008)
5271
                                    <1>
5272
                                              ; 'vgabios.c', 'biosfn_load_text_8_16_pat'
                                    <1>
5273
                                   <1>
5274
                                   <1>
                                              ; biosfn_load_text_8_16_pat (AL,BL)
5275
                                   <1>
5276
                                   <1>
                                              ; get_font_access();
                                              ; blockaddr = ((BL & 0x03) << 14) + ((BL & 0x04) << 11);
5277
                                    <1>
5278
                                   <1>
                                              ; for (i=0; i<0x100; i++)
5279
                                   <1>
                                              ; src = i * 16;
5280
                                   <1>
                                              ; dest = blockaddr + i * 32;
5281
                                   <1>
                                              ; memcpyb(0xA000, dest, 0xC000, vgafont16+src, 16);
5282
                                   <1>
5283
                                   <1>
5284
                                   <1>
                                              ; release_font_access();
                                              ; if (AL \ge 0 \times 10)
5285
                                   <1>
5286
                                   <1>
                                              ; set_scan_lines(16);
5287
                                    <1>
5288
                                   <1>
                                              ; }
5289
                                   <1>
5290 00002D22 50
                                   <1>
                                              push eax
```

```
5291 00002D23 E89EFEFFFF
                                  <1>
                                            call get_font_access
5292
                                  <1>
5293
                                            ; blockaddr = ((BL & 0x03) << 14) + ((BL & 0x04) << 11);
                                  <1>
5294
                                  <1>
                                            ;mov dl, bl
5295
                                  <1>
                                            ;and d1, 3
5296
                                  <1>
                                            ;shl dx, 14
5297
                                  <1>
                                            ;xchg dx, bx
                                            ;and dl, 4
5298
                                  <1>
5299
                                  <1>
                                            ;shl
                                                  dx, 11
                                  <1>
5300
                                            ; add dx, bx
5301
                                  <1>
                                            ; xor dx, dx; blockaddr = 0
5302
                                  <1>
5303
                                  <1>
                                            ; Always block 0 for TRDOS 386 ! (blockaddr=0(
5304
                                  <1>
5305 00002D28 28DB
                                  <1>
                                                  bl, bl; i = 0
                                            sub
5306 00002D2A B710
                                  <1>
                                            mov
                                                  bh, 16
5307 00002D2C BE[3C430100]
                                  <1>
                                            mov esi, vgafont16
5308 00002D31 BF00000A00
                                  <1>
                                            mov edi, 0A0000h
5309 00002D36 0FB6C7
                                  <1>
                                            movzx eax, bh
                                  <1> lt8_16_1:
5310
5311
                                  <1>
                                          ;mov al, bl
5312
                                            ;mul bh
                                  <1>
5313
                                  <1>
                                            ;movzx esi, ax
5314
                                  <1>
                                           ;add esi, vgafont16
5315
                                  <1>
                                           ;mov al, bl ; i
5316
                                  <1>
                                            ;sub ah, ah
                                           ;shl ax, 5 ; * 32
5317
                                  <1>
                                           ;;add ax, dx; blockaddr + i * 32;
5318
                                  <1>
5319
                                  <1>
                                            ;movzx edi, ax ; dest
                                           ;add edi, 0A0000h
5320
                                  <1>
5321
                                  <1>
                                            ; movzx ecx, bh
5322 00002D39 89C1
                                  <1>
                                           mov ecx, eax; 16
5323 00002D3B F3A4
                                  <1>
                                           rep
                                                  movsb
5324 00002D3D 01C7
                                 <1>
                                            add edi, eax; add edi, 16
5325 00002D3F FEC3
                                  <1>
                                            inc
                                                  bl
5326 00002D41 75F6
                                  <1>
                                            jnz
                                                  short lt8_16_1
                                  <1>
                                           call release_font_access
5328 00002D43 E8B3FEFFFF
                                  <1>
5329
                                  <1>
5330 00002D48 58
                                  <1>
                                           pop
                                                  eax
                                           ; if (AL > = 0 \times 10)
5331
                                  <1>
5332 00002D49 3C10
                                  <1>
                                            cmp al, 10h
5333 00002D4B 7205
                                  <1>
                                                  short lt8_16_2
                                            jb
5334
                                  <1>
                                            ; BH = 16
                                            ; set_scan_lines(16);
5335
                                  <1>
5336 00002D4D E8E8FEFFFF
                                  <1>
                                            call set_scan_lines
                                  <1> lt8_16_2:
5337
5338 00002D52 C3
                                  <1>
                                            retn
5339
                                  <1>
5340
                                  <1> load_gfx_user_chars:
                                          ; 08/08/2016
5341
                                  <1>
5342
                                  <1>
                                            ; 10/07/2016
5343
                                  <1>
                                            ; Setup User-Defined Font for Graphics Mode (VGA)
5344
                                  <1>
                                          ; derived from 'Plex86/Bochs VGABios' source code
5345
                                  <1>
5346
                                  <1>
                                            ; vgabios-0.7a (2011)
5347
                                  <1>
                                           ; by the LGPL VGABios developers Team (2001-2008)
5348
                                  <1>
                                           ; 'vgabios.c', 'biosfn_load_gfx_user_chars'
5349
                                  <1>
                                            ; biosfn_load_gfx_user_chars (ES,BP,CX,BL,DL)
5350
                                  <1>
5351
                                  <1>
                                            ; /* set 0x43 INT pointer */
5352
                                  <1>
                                            ; write word(0x0, 0x43*4, BP);
5353
                                  <1>
                                            ; write_word(0x0, 0x43*4+2, ES);
                                            xor eax, eax
5354 00002D53 31C0
                                  <1>
                                                  eax ; OFFFFFFFFh (user defined fonts)
5355 00002D55 48
                                  <1>
                                            dec
5356 00002D56 A3[6A650100]
                                  <1>
                                            mov
                                                   [VGA_INT43H], eax
5357
                                  <1>
                                                 screen rows code: 00H = user-specified (in DL)
5358
                                  <1>
                                            ; BL
                                            ;
5359
                                  <1>
                                                                       01H = 14 \text{ rows}
5360
                                  <1>
                                                                       02H = 25 \text{ rows}
                                              ;
5361
                                  <1>
                                                                       03H = 43 \text{ rows}
                                            ; CX bytes per character definition ; DL (when BL=0) custom number of character rows on screen
5362
                                  <1>
5363
                                  <1>
5364
                                  <1>
                                           ; dh = 0 \rightarrow 256 characters
5365
                                  <1>
                                           ; dh = 80h \rightarrow 128 characters
                                            ; (If DH <> 0 and DH <> 80h -> invalid)
5366
                                  <1>
5367
                                  <1>
                                             ; EBP address of font-definition information (user's mem space)
5368
                                  <1>
5369
                                  <1>
                                            ; switch (BL)
5370
                                            ; case 0:
                                  <1>
5371
                                  <1>
                                                 write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, DL-1);
5372
                                  <1>
                                                 break;
5373 00002D5B 20DB
                                 <1>
                                            and
                                                  bl, bl
5374 00002D5D 7508
                                  <1>
                                                   short l_gfx_uc_1
                                            jnz
                                                  [VGA ROWS], dl ; not DL-1 !
5375 00002D5F 8815[2A5F0000]
                                  <1>
                                            mov
5376 00002D65 EB23
                                  <1>
                                            jmp
                                                  short l_gfx_uc_4
                                  <1> l_gfx_uc_1:
5378
                                  <1>
                                           ; case 1:
5379
                                  <1>
                                                 write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, 13);
5380
                                  <1>
                                                break;
5381 00002D67 FECB
                                  <1>
                                            dec bl
                                                  short l_gfx_uc_2
5382 00002D69 7509
                                  <1>
                                            jnz
                                            ; bl = 1
                                  <1>
5383
5384 00002D6B C605[2A5F0000]0E
                                 <1>
                                            mov byte [VGA_ROWS], 14; not 13!
5385 00002D72 EB16
                                  <1>
                                            jmp
                                                  short l_gfx_uc_4
5386
                                  <1> l_gfx_uc_2:
5387 00002D74 FECB
                                 <1>
                                            dec
5388 00002D76 740B
                                 <1>
                                            jz
                                                   short l_gfx_uc_3; bl = 2
5389 00002D78 FECB
                                  <1>
                                            dec
                                                  bl
5390 00002D7A 750E
                                            jnz short l_gfx_uc_4; bl > 3
                                  <1>
5391
                                  <1>
                                           ; b1 = 3
5392
                                  <1>
                                           ; case 3:
                                                write byte (BIOSMEM SEG, BIOSMEM NB ROWS, 42);
5393
                                  <1>
5394
                                  <1>
5395 00002D7C C605[2A5F0000]2B
                                  <1>
                                           mov byte [VGA_ROWS], 43; not 42!
```

```
5396
                                  <1> l_gfx_uc_3:
                                       ; case 2:
5397
                                  <1>
5398
                                  <1>
                                           ; default:
5399
                                  <1>
                                          ; write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, 24);
5400
                                  <1>
                                                break;
                                         ;
                                           ; bl = 2 \text{ or } bl > 3
5401
                                  <1>
5402 00002D83 C605[2A5F0000]19
                                         mov byte [VGA_ROWS], 25; not 24!
                                  <1>
5403
                                  <1>
                                          ; }
5404
                                  <1> l_gfx_uc_4:
5405
                                           ; write byte (BIOSMEM SEG, BIOSMEM CHAR HEIGHT, CX);
                                  <1>
5406 00002D8A 880D[265F0000]
                                  <1>
                                            mov
                                                  [CHAR_HEIGHT], cl
5407
                                  <1>
                                           ; }
5408 00002D90 C3
                                  <1>
                                            retn
5409
                                  <1>
                                  <1> load_gfx_8_14_chars:
5410
5411
                                  <1>
                                         ; 08/08/2016
5412
                                           ; 10/07/2016
                                  <1>
5413
                                           ; Setup ROM 8x14 Font for Graphics Mode (VGA)
5414
                                  <1>
5415
                                  <1>
                                           ; derived from 'Plex86/Bochs VGABios' source code
                                           ; vgabios-0.7a (2011)
5416
                                  <1>
5417
                                  <1>
                                           ; by the LGPL VGABios developers Team (2001-2008)
                                  <1>
                                           ; 'vgabios.c', 'biosfn_load_gfx_8_14_chars'
5418
5419
                                  <1>
                                  <1>
5420
                                           ; biosfn_load_gfx_8_14_chars (BL)
                                           ; /* set 0x43 INT pointer */
5421
                                  <1>
5422
                                  <1>
                                           ; write word(0x0, 0x43*4, &vgafont14);
                                           ; write_word(0x0, 0x43*4+2, 0xC000);
5423
                                  <1>
5424 00002D91 C705[6A650100]-
                                  <1>
                                           mov dword [VGA_INT43H], vgafont14
5424 00002D97 [3C350100]
                                  <1>
5425
                                  <1>
                                                    screen rows code: 00H = user-specified (in DL)
5426
                                  <1>
                                           ; BL
                                                                        01H = 14 \text{ rows}
5427
                                  <1>
                                             ;
5428
                                  <1>
                                                                        02H = 25 \text{ rows}
5429
                                  <1>
                                                                        03H = 43 \text{ rows}
                                             ;
5430
                                  <1>
                                             ; DL
                                                      (when BL=0) custom number of char rows on screen
5431
                                  <1>
5432
                                  <1>
                                           ; switch (BL) {
5433
                                  <1>
                                           ; case 0:
5434
                                  <1>
                                           ; write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, DL-1);
5435
                                  <1>
                                           ; break;
                                           and bl, bl
jnz short l_gfx_8_14c_1
5436 00002D9B 20DB
                                  <1>
5437 00002D9D 7508
                                  <1>
                                            mov [VGA ROWS], dl ; not DL-1 !
5438 00002D9F 8815[2A5F0000]
                                 <1>
5439 00002DA5 EB23
                                  <1>
                                            qmj
                                                 short l gfx 8 14c 4
5440
                                  <1> l_gfx_8_14c_1:
                                        ; case 1:
5441
                                  <1>
                                           ; write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, 13);
; break;
5442
                                 <1>
5443
                                  <1>
                                           dec bl
5444 00002DA7 FECB
                                 <1>
5445 00002DA9 7509
                                 <1>
                                           jnz short l_gfx_8_14c_2
5446
                                  <1>
                                           ; bl = 1
                                            mov byte [VGA_ROWS], 14 ; not 13 !
5447 00002DAB C605[2A5F0000]0E
                                 <1>
5448 00002DB2 EB16
                                 <1>
                                            jmp short l_gfx_8_14c_4
                                 <1> l_gfx_8_14c_2:
5449
5450 00002DB4 FECB
                                 <1>
                                           dec bl
5451 00002DB6 740B
                                            jz
                                 <1>
                                                  short l_gfx_8_14c_3; bl = 2
5452 00002DB8 FECB
                                  <1>
                                            dec bl
                                           jnz
5453 00002DBA 750E
                                  <1>
                                                  short l_gfx_8_14c_4; bl > 3
5454
                                 <1>
                                           ; b1 = 3
5455
                                  <1>
                                           ; case 3:
                                           ;
5456
                                  <1>
                                                write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, 42);
5457
                                  <1>
                                                break;
5458 00002DBC C605[2A5F0000]2B
                                  <1>
                                          mov byte [VGA_ROWS], 43; not 42!
5459
                                  <1> l_gfx_8_14c_3:
                                        ; case 2:
5460
                                  <1>
5461
                                  <1>
                                           ; default:
                                                write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, 24);
5462
                                  <1>
                                          ;
                                          ;
5463
                                  <1>
5464
                                  <1>
                                           ; bl = 2 \text{ or } bl > 3
5465 00002DC3 C605[2A5F0000]19
                                  <1>
                                          mov byte [VGA_ROWS], 25; not 24!
5466
                                  <1>
                                           ; }
5467
                                  <1> l_gfx_8_14c_4:
                                       ; write_byte(BIOSMEM_SEG, BIOSMEM_CHAR_HEIGHT, 14);
5468
                                  <1>
5469 00002DCA C605[265F0000]0E
                                  <1>
                                                     byte [CHAR_HEIGHT], 14
                                            mov
5470
                                  <1>
                                           ; }
5471 00002DD1 C3
                                  <1>
                                           retn
5472
                                  <1>
5473
                                  <1> load_gfx_8_8_chars:
                                       ; 08/08/2016
5474
                                  <1>
                                            ; 10/07/2016
5475
                                  <1>
                                            ; Setup ROM 8x14 Font for Graphics Mode (VGA)
5476
                                  <1>
                                  <1>
5478
                                  <1>
                                            ; derived from 'Plex86/Bochs VGABios' source code
5479
                                  <1>
                                            ; vgabios-0.7a (2011)
5480
                                            ; by the LGPL VGABios developers Team (2001-2008)
                                  <1>
5481
                                  <1>
                                            ; 'vgabios.c', 'biosfn_load_gfx_8_8_dd_chars'
5482
                                  <1>
5483
                                  <1>
                                            ; biosfn_load_gfx_8_8_dd_chars (BL)
5484
                                  <1>
                                            ; /* set 0x43 INT pointer */
5485
                                  <1>
                                            ; write_word(0x0, 0x43*4, &vgafont8);
5486
                                  <1>
                                            ; write_word(0x0, 0x43*4+2, 0xC000);
5487 00002DD2 C705[6A650100]-
                                  <1>
                                            mov dword [VGA_INT43H], vgafont8
5487 00002DD8 [3C2D0100]
                                  <1>
5488
                                  <1>
5489
                                                    screen rows code: 00H = user-specified (in DL)
                                  <1>
                                            ; BL
5490
                                  <1>
                                                                        01H = 14 \text{ rows}
                                            ;
5491
                                  <1>
                                                                        02H = 25 \text{ rows}
                                             ;
5492
                                  <1>
                                                                        03H = 43 \text{ rows}
5493
                                  <1>
                                              ; DL
                                                      (when BL=0) custom number of char rows on screen
5494
                                  <1>
5495
                                  <1>
                                            ; switch (BL) {
5496
                                  <1>
                                            ; case 0:
5497
                                  <1>
                                            ; write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, DL-1);
5498
                                  <1>
                                               break;
```

```
5499 00002DDC 20DB
                                 <1>
                                           and bl, bl
                                                 short l_gfx_8_8c_1
[VGA_ROWS], dl ; not DL-1 !
5500 00002DDE 7508
                                 <1>
                                           jnz
5501 00002DE0 8815[2A5F0000]
                                 <1>
                                           mov
                                       jmp short l_gfx_8_8c_4
5502 00002DE6 EB23
                                 <1>
5503
                                 <1> l_gfx_8_8c_1:
                                        ; case 1:
5504
                                 <1>
                                         ; write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, 13);
; break;
5505
                                 <1>
5506
                                 <1>
5507 00002DE8 FECB
                                 <1>
                                          dec bl
                                          jnz short l_gfx_8_8c_2
5508 00002DEA 7509
                                 <1>
5509
                                 <1>
                                           ; bl = 1
5510 00002DEC C605[2A5F0000]0E
                                           mov byte [VGA_ROWS], 14 ; not 13 !
jmp short l_gfx_8_8c_4
                                 <1>
5511 00002DF3 EB16
                                 <1>
5512
                                 <1> l_gfx_8_8c_2:
5513 00002DF5 FECB
                                 <1>
                                       dec bl
5514 00002DF7 740B
                                 <1>
                                           jг
                                                 short l_gfx_8_8c_3 ; bl = 2
5515 00002DF9 FECB
                                <1>
                                          dec bl
                                         jnz short l_gfx_8_8c_4; bl > 3
5516 00002DFB 750E
                                 <1>
                                           ; bl = 3
5517
                                 <1>
5518
                                 <1>
                                           ; case 3:
                                         ; write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, 42);
5519
                                 <1>
                                               break;
5520
                                 <1>
                                           ;
                                         mov byte [VGA_ROWS], 43; not 42!
5521 00002DFD C605[2A5F0000]2B
                                 <1>
                                  <1> 1_gfx_8_8c_3:
                                        ; case 2:
5523
                                 <1>
5524
                                 <1>
                                           ; default:
                                         ; write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, 24);
5525
                                 <1>
                                        ; break;
5526
                                 <1>
5527
                                  <1>
                                        ; b1 = 2 or b1 > 3
mov byte [VGA_ROWS], 25 ; not 24 !
                                           ; bl = 2 \text{ or } bl > 3
5528 00002E04 C605[2A5F0000]19
                                 <1>
                                         ; }
5529
                                  <1>
5530
                                  <1> l_gfx_8_8c_4:
                                        ; write_byte(BIOSMEM_SEG, BIOSMEM_CHAR_HEIGHT, 8);
5531
                                 <1>
5532 00002E0B C605[265F0000]08
                                 <1>
                                                    byte [CHAR_HEIGHT], 8
5533
                                  <1>
                                           ; }
5534 00002E12 C3
                                 <1>
                                           retn
5535
                                  <1>
5536
                                 <1> load_gfx_8_16_chars:
                                        ; 0870872016
5537
                                  <1>
                                           ; 10/07/2016
5538
                                 <1>
5539
                                 <1>
                                         ; Setup ROM 8x14 Font for Graphics Mode (VGA)
5540
                                  <1>
5541
                                 <1>
                                           ; derived from 'Plex86/Bochs VGABios' source code
5542
                                  <1>
                                          ; vgabios-0.7a (2011)
5543
                                  <1>
                                           ; by the LGPL VGABios developers Team (2001-2008)
5544
                                  <1>
                                           ; 'vgabios.c', 'biosfn_load_gfx_8_16_chars'
5545
                                  <1>
5546
                                  <1>
                                           ; biosfn_load_gfx_8_16_chars (BL)
5547
                                  <1>
                                           ; /* set 0x43 INT pointer */
                                  <1>
                                           ; write_word(0x0, 0x43*4, &vgafont16);
5548
5549
                                  <1>
                                           ; write_word(0x0, 0x43*4+2, 0xC000);
5550 00002E13 C705[6A650100]-
                                  <1>
                                           mov dword [VGA_INT43H], vgafont16
5550 00002E19 [3C430100]
                                 <1>
5551
                                  <1>
5552
                                  <1>
                                           ; BL
                                                   screen rows code: 00H = user-specified (in DL)
5553
                                  <1>
                                                                       01H = 14 \text{ rows}
                                                                       02H = 25 \text{ rows}
5554
                                  <1>
                                             ;
                                                                       03H = 43 \text{ rows}
5555
                                  <1>
5556
                                  <1>
                                             ; DL
                                                     (when BL=0) custom number of char rows on screen
5557
                                 <1>
5558
                                 <1>
                                           ; switch (BL) {
5559
                                  <1>
                                           ; case 0:
5560
                                 <1>
                                           ; write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, DL-1);
5561
                                 <1>
                                           ; break;
5562 00002E1D 20DB
                                 <1>
                                           and bl, bl
5563 00002E1F 7508
                                 <1>
                                           jnz
                                                 short l_gfx_8_16c_1
5564 00002E21 8815[2A5F0000]
                                                 [VGA ROWS], dl ; not DL-1 !
                                 <1>
                                           mov
                                           jmp short l_gfx_8_16c_4
5565 00002E27 EB23
                                 <1>
5566
                                 <1> l_gfx_8_16c_1:
5567
                                 <1>
                                         ; case 1:
                                           ; write byte (BIOSMEM SEG, BIOSMEM NB ROWS, 13);
5568
                                 <1>
5569
                                 <1>
                                                break;
5570 00002E29 FECB
                                           dec bl
                                 <1>
5571 00002E2B 7509
                                 <1>
                                           jnz short l_gfx_8_16c_2
5572
                                 <1>
                                           ; bl = 1
5573 00002E2D C605[2A5F0000]0E
                                                 byte [VGA_ROWS], 14 ; not 13 !
                                <1>
                                           mov
5574 00002E34 EB16
                                 <1>
                                           jmp
                                                 short l_gfx_8_16c_4
5575
                                 <1> 1_gfx_8_16c_2:
                                                 bl
5576 00002E36 FECB
                                 <1>
                                           dec
5577 00002E38 740B
                                                 short l_gfx_8_16c_3; bl = 2
                                 <1>
                                           jΖ
5578 00002E3A FECB
                                 <1>
                                           dec bl
                                           jnz
5579 00002E3C 750E
                                  <1>
                                               short l_gfx_8_16c_4; bl > 3
                                           ; b1 = 3
5580
                                  <1>
5581
                                  <1>
                                           ; case 3:
5582
                                  <1>
                                                write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, 42);
                                           ;
5583
                                  <1>
                                                break;
5584 00002E3E C605[2A5F0000]2B
                                  <1>
                                           mov byte [VGA ROWS], 43; not 42!
5585
                                  <1> 1 gfx 8 16c 3:
5586
                                  <1>
                                           ; case 2:
5587
                                  <1>
                                           ; default:
5588
                                  <1>
                                                write_byte(BIOSMEM_SEG,BIOSMEM_NB_ROWS, 24);
                                                break;
5589
                                  <1>
                                           ;
                                           ; b1 = 2 \text{ or } b1 > 3
5590
                                  <1>
5591 00002E45 C605[2A5F0000]19
                                  <1>
                                           mov byte [VGA_ROWS], 25; not 24!
5592
                                  <1>
                                           ; }
5593
                                  <1> l_gfx_8_16c_4:
                                  <1>
                                           ; write_byte(BIOSMEM_SEG, BIOSMEM_CHAR_HEIGHT, 16);
5595 00002E4C C605[265F0000]10
                                 <1>
                                            mov
                                                     byte [CHAR_HEIGHT], 16
5596
                                  <1>
5597 00002E53 C3
                                  <1>
                                           retn
5598
                                 <1>
5599
                                  <1> get font info:
                                           ; 19/09/2016
5600
                                 <1>
5601
                                  <1>
                                           ; 08/08/2016
                                  <1>
5602
                                           ; 10/07/2016
```

```
5603
                                <1>
                                         ; Get Current Character Generator Info (VGA)
5604
                                <1>
5605
                                         ; derived from 'Plex86/Bochs VGABios' source code
                                <1>
5606
                                <1>
                                         ; vgabios-0.7a (2011)
5607
                                <1>
                                         ; by the LGPL VGABios developers Team (2001-2008)
                                         ; 'vgabios.c', 'biosfn_get_font_info'
5608
                                <1>
5609
                                <1>
                                        ; Modified for TRDOS 386 !
5610
                                <1>
5611
                                <1>
                                        ; INPUT ->
                                <1>
5612
                                        ; AX = 1130h
5613
                                <1>
5614
                                <1>
                                             BL = 0 -> Get info for current VGA font
5615
                                <1>
                                                     (BH = unused)
                                        ; 19/09/2016
5616
                                <1>
                                        ; BL > 0 -> Get requested character font data
5617
                                <1>
                                             BL = 1 -> vgafont8
5618
                                <1>
                                                BL = 2 \rightarrow vgafont14
5619
                                <1>
                                <1>
                                              BL = 3 \rightarrow vgafont16
5620
5621
                                <1>
                                               BL > 3 -> Invalid function (for now!)
                                              BH = ASCII code of the first character
5622
                                <1>
                                              ECX = Number of characters from the 1st char
5623
                                <1>
                                               ECX >= 256 \rightarrow All (256-BH) characters
5624
                                <1>
                                        ;
                                <1>
                                               ECX = 0 -> All characters (BH = unused)
5625
                                         ;
5626
                                <1>
                                                EDX = User's Buffer Address
                                      ; OUTPUT ->
                                <1>
5627
5628
                                <1>
                                        ; AL = height (scanlines), bytes per character
5629
                                <1>
                                             AH = screen rows
                                <1>
                                            Byte 16-23 of EAX = number of columns
5630
5631
                                <1>
                                             Byte 24-31 of EAX =
                                             0 -> default font (not configured yet)
5632
                                <1>
5633
                                <1>
                                                OFFh -> user defined font
5634
                                <1>
                                                14 = vgafont14
                                                 8 = vgafont8
5635
                                <1>
5636
                               <1>
                                                16 = vgafont16
5637
                               <1>
                                        ; If BL input > 0 ->
                                                EAX = Actual transfer count
5638
                               <1>
5639
                               <1>
                                      and bl, bl
5640 00002E54 20DB
                               <1>
5641 00002E56 7408
                               <1>
                                               short gfi_0
5642
                               <1>
                                        ; invalid function (input)
5643 00002E58 80FB03
                                        cmp bl, 3
                              <1>
5644 00002E5B 7642
                               <1>
                                        jna
                                               short gfi 4
5645 00002E5D 31C0
                               <1>
                                              eax, eax ; 0
                                         xor
5646 00002E5F C3
                               <1>
5647
                               <1> gfi_0:
                               <1> mov <1> mov
5648 00002E60 A0[265F0000]
                                               al, [CHAR_HEIGHT]
5649 00002E65 8A25[2A5F0000]
                                              ah, [VGA_ROWS]
                               <1>
                                        mov
                            5650 00002E6B C1E010
5651 00002E6E A0[245F0000]
                                               al, [CRT COLS]
5652 00002E73 8B0D[6A650100]
                                              ecx, [VGA_INT43H]
5653 00002E79 21C9
5654 00002E7B 741E
                                               short gfi 2; 0 = default font
                                               ecx ; OFFFFFFFF -> 0 (user defined font)
5655 00002E7D 41
5656 00002E7E 7504
                                              short gfi_1
5657 00002E80 FECC
                               <1>
                                              ah ; OFFh
                                         dec
5658 00002E82 EB17
                               <1>
                                         jmp
                                              short gfi_2
                               <1> gfi_1:
5660 00002E84 49
                                        dec
                                              ecx ; 08/08/2016
                               <1>
5661 00002E85 B40E
                               <1>
                                               ah, 14
                                         mov
                              5662 00002E87 81F9[3C350100]
                                               ecx, vgafont14
5663 00002E8D 740C
                                               short gfi_2
5664 00002E8F B408
                                               ah, 8
5665 00002E91 81F9[3C2D0100]
                                               ecx, vgafont8
5666 00002E97 7402
                                               short gfi_2
                                        ; vgafont16
5667
                               <1>
5668 00002E99 D0E4
                               <1>
                                         shl
                                              ah, 1 ; ah = 16
                               <1> gfi_2:
5669
5670 00002E9B C1C010
                               <1>
                                        rol
                                               eax, 16
5671
                               <1> gfi_3:
5672 00002E9E C3
                               <1>
                                        retn
                               <1> gfi_4:
5673
5674 00002E9F 89D7
                               <1> mov
                                               edi, edx ; **
5675 00002EA1 80FB02
                               <1>
                                        cmp
                                              bl, 2
                              5676 00002EA4 720B
5677 00002EA6 772F
5678
5679 00002EA8 BE[3C350100]
5680 00002EAD B30E
5681 00002EAF EB07
                               <1>
                                         jmp short gfi_6
                                <1> gfi_5:
5682
                                     ;BL = 1 \rightarrow vgafont8
5683
                                <1>
5684 00002EB1 BE[3C2D0100]
                                         mov esi, vgafont8; *
                                <1>
5685 00002EB6 B308
                               <1>
                                         mov
                                               bl, 8
                                <1> gfi_6:
                                    or
5687 00002EB8 09C9
                               <1>
                                               ecx, ecx
5688 00002EBA 7424
                                               short gfi_8 ; all chars from the 00h
                               <1>
                                         jz
5689 00002EBC 88F8
                              <1>
                                               al, bh ; character index
                                        mov
                              <1>
                                       mul bl ; char index * char height/size
5690 00002EBE F6E3
                              <1>
                                        movzx edx, ax
5691 00002EC0 0FB7D0
                                       add esi, edx; *
                              <1>
5692 00002EC3 01D6
5693 00002EC5 66BAFF00
                                     mov
                                               dx, 255
                              <1>
                               <1>
5694 00002EC9 28FA
                                        sub
                                               dl, bh
5695 00002ECB 6642
                               <1>
                                        inc
                                              dx
                                              ecx, edx
5696 00002ECD 39D1
                              <1>
                                       cmp
                                      ja
                                               short qfi 8
5697 00002ECF 770F
                               <1>
5698 00002ED1 7412
                               <1>
                                         jе
                                               short gfi_9
5699 00002ED3 89D1
                               <1>
                                               ecx, edx
                                        mov
5700 00002ED5 EB0E
                               <1>
                                        jmp
                                               short gfi_9
5701
                               <1> gfi 7:
                               <1>
5702
                                        ;BL = 3 \rightarrow vgafont16
5703 00002ED7 BE[3C430100]
                               <1>
                                         mov esi, vgafont16; *
5704 00002EDC B310
                               <1>
                                         mov
                                               bl, 16
5705 00002EDE EBD8
                               <1>
                                         jmp
                                              short gfi 6
                               <1> qfi 8:
5706
5707 00002EE0 B900010000
                               <1>
                                        mov
                                              ecx, 256
```

```
5708
                                            <1> gfi_9:
5709 00002EE5 6689C8
                                            <1> mov
                                                                  ax, cx ; character count
5710 00002EE8 30FF
                                            <1>
                                                         xor
                                                                  bh, bh
5711 00002EEA 66F7E3
                                           <1>
                                                         mul bx ; char count * char height/size
5712 00002EED 6689C1
                                            <1>
                                                       mov cx, ax
5713
                                            <1>
5714
                                            <1>
                                                       ; ESI = source address in system space
                                                       ; EDI = user's buffer address
5715
                                            <1>
5716
                                            <1>
                                                         ; ECX = transfer (byte) count
                                                       call transfer_to_user_buffer
5717 00002EF0 E808B90000
                                            <1>
5718 00002EF5 89C8
                                            <1>
                                                       mov eax, ecx; actual transfer count
5719 00002EF7 C3
                                             <1>
                                                         retn
5720
                                             <1>
                                             <1> vga pal funcs:
5721
                                                     ; 10/08/2016
5722
                                             <1>
5723
                                             <1>
                                                          ; VGA Palette functions
5724
                                                       ; derived from 'Plex86/Bochs VGABios' source code
5725
                                             <1>
                                                         ; vgabios-0.7a (2011)
5726
                                             <1>
5727
                                             <1>
                                                         ; by the LGPL VGABios developers Team (2001-2008)
5728
                                             <1>
                                                        ; 'vgabios.c', 'vgarom.asm'
5729
                                             <1>
5730 00002EF8 3C00
                                            <1>
                                                          cmp al, 0
                                           <1> cmp a1, 0
<1> je set_single_palette_reg
5731 00002EFA 0F848F000000
5732
                                            <1> vga_palf_1001:
5733 00002F00 3C01
                                            <1> cmp al, 1
5734 00002F02 0F84B4000000
                                           <1>
                                                                    set_overscan_border_color
                                                         jе
5735
                                            <1> vga_palf_1002:
                                      <1> vga_palf_1002:
<1> cmp al, 2
<1> je se
<1> vga palf 1003:
5736 00002F08 3C02
5737 00002F0A 0F84B0000000
                                                                       set_all_palette_reg
                                            <1> vga_palf_1003:
5738
5739 00002F10 3C03
                                            <1> cmp al, 3
5740 00002F12 0F84E8000000
                                            <1>
                                                                  toggle_intensity
5741
                                            <1> vga_palf_1007:
                                            <1> cmp al, 7
5742 00002F18 3C07
                                      <1> cmp
<1> je
<1> jb
<1> yga palf 10
5743 00002F1A 0F840D010000
                                                                  get_single_palette_reg
5744 00002F20 7266
                                                                  short vga_palf_unknown
                                   <1> vga_palf_1008:
5745
5746 00002F22 3C08
5747 00002F24 0F8437010000
                                                                  read_overscan_border_color
5748
5749 00002F2A 3C09
5750 00002F2C 0F8433010000
                                                                  get_all_palette_reg
5751
                                     <1> cmp al, 10h
<1> je set_single_dac_reg
<1> jb short vga_palf_unknown
<1> vga_palf_1012:
<1> cmp al, 12h
<1> je set_all_dac_reg
<1> jb short vga_palf_unknown
<1> vga_palf_1013:
                                           5752 00002F32 3C10
5753 00002F34 0F8487010000
5754 00002F3A 724C
5755
5756 00002F3C 3C12
5757 00002F3E 0F8498010000

      5758
      00002F44
      7242
      <1> jb short vga

      5759
      <1> vga_palf_1013:
      <1> cmp al, 13h

      5760
      00002F46
      3C13
      <1> cmp al, 13h

      5761
      00002F48
      0F84CC010000
      <1> je select

      5762
      <1> vga_palf_1015:

      5763
      00002F4E
      3C15
      <1> cmp al, 15h

      5764
      00002F50
      0F8412020000
      <1> je read_sind

      5765
      00002F56
      7230
      <1> jb short vga

      5766
      <1> vga_palf_1017:
      <1</td>

      5767
      00002F58
      3C17
      <1> cmp al, 17h

      5768
      00002F50
      0F8428020000
      <1</td>
      je read_

      5770
      <1</td>
      vga_palf_1018:
      <1</td>
      vga_palf_1018:

      5771
      00002F62
      3C18
      <1</td>
      cmp al, 18h

      5772
      00002F64
      0F8462020000
      <1</td>
      je read_

      5776
      <1</td>
      vga_palf_1018:
      <1</td>

      5777
      00002F62
      0F8462020000
      <1</td>
      je read_

      5777
      00002F72
      3C1A
      <1</td>
      cmp al, 18h

      5777
      00002F72

5758 00002F44 7242
                                                                     select_video_dac_color_page
                                                                  read single dac reg
                                                                  short vga_palf_unknown
                                                                  read_all_dac_reg
                                                                  short vga_palf_unknown
                                                                  set_pel_mask
                                                                      read_pel_mask
5776
                                            <1> vga_palf_101A:
                                            <1> cmp al, 1Ah <1> je read
5777 00002F72 3C1A
                                                                       read_video_dac_state
5778 00002F74 0F8468020000
                                            <1> vga_palf_101B:
5779
                                            <1> cmp al, 1Bh
5780 00002F7A 3C1B
5781
                                            <1>
                                                         ; jne short vga_palf_unknown
5782 00002F7C 770A
                                             <1>
                                                       ja short vga_palf_unknown
5783
                                             <1>
5784 00002F7E E80CF7FFFF
                                             <1> call gray_scale_summing
5785 00002F83 E9D5E5FFFF
                                             <1>
                                                         jmp
                                                                       VIDEO_RETURN
5786
                                             <1>
5787
                                             <1> vga_palf_unknown:
                                                     sub eax, eax; 0 = invalid function
5788 00002F88 29C0
                                             <1>
                                                                   _video_return
5789 00002F8A E9D3E5FFFF
                                             <1>
                                                          jmp
5790
                                             <1>
5791
                                             <1> set_single_palette_reg:
                                                      ; 10/08/2016
5792
                                             <1>
5793
                                             <1>
                                                          ; Set One Palette Register
5794
                                             <1>
                                                         ; BL = register number to set
                                                       ; (a 4-bit attribute nibble: 00h-0Fh)
5795
                                             <1>
5796
                                             <1>
                                                         ; BH = 6-bit RGB color to display
                                                       ; for that attribute
5797
                                             <1>
5798
                                             <1>
5799 00002F8F 80FB14
                                             <1>
                                                         cmp
                                                                  bl, 14h
                                            <1>
                                                                   short no_actl_reg1
5800
                                                          ;ja
5801 00002F92 0F87C5E5FFFF
                                            <1>
                                                                   VIDEO_RETURN
                                                          jа
                                                       push ax
5802 00002F98 6650
                                            <1>
5803 00002F9A 6652
                                            <1>
                                                         push dx
5804 00002F9C 66BADA03
                                                                  dx, 3DAh; VGAREG ACTL RESET
                                          <1>
                                                         mov
5805 00002FA0 EC
                                            <1>
                                                         in
                                                                  al, dx
5806 00002FA1 66BAC003
                                            <1>
                                                         mov
                                                                  dx, 3C0h; VGAREG_ACTL_ADDRESS
5807 00002FA5 88D8
                                           <1>
                                                                  al, bl
                                                         mov
5808 00002FA7 EE
                                           <1>
                                                       out
                                                                  dx, al
                                           <1>
<1>
5809 00002FA8 88F8
                                                         mov
                                                                  al, bh
5810 00002FAA EE
                                                         out
                                                                  dx, al
                                           <1>
                                                                  al, 20h
5811 00002FAB B020
5812 00002FAD EE
                                            <1>
                                                                  dx, al
                                                         out
```

```
; ifdef VBOX
mov dx, 3DAh ; VGAREG_ACTL_RESET
in al, dx
; endif ; VBOX
5813
                                                       <1>
5814 00002FAE 66BADA03
                                                      <1>
5815 00002FB2 EC
                                                      <1>
5816
                                                     <1>
                                               <1>
<1>
5817 00002FB3 665A
                                                                   pop dx
5818 00002FB5 6658
                                                                       pop
                                                                                 ax
                                                      <1> ;no actl reg1:
5820 00002FB7 E9A1E5FFFF
                                                      <1>
                                                                   jmp VIDEO_RETURN
5821
                                                       <1>
                                                       <1> set_overscan_border_color:
5822
                                                                ; 10/0<del>8</del>/2016
5823
                                                       <1>
5824
                                                                       ; Set Overscan/Border Color Register
                                                        <1>
                                                                       ; BH = 6-bit RGB color to display
5825
                                                       <1>
5826
                                                       <1>
                                                                                for that attribute
5827
                                                       <1>
5828 00002FBC B311
                                                                       mov bl, 11h
                                                       <1>
5829 00002FBE EBCF
                                                                    jmp short set single palette reg
                                                      <1>
5830
                                                       <1>
                                                       <1> set all palette reg:
5831
5832
                                                       <1> ; 10/08/2016
                                                                       ; Set All Palette Registers and Overscan
5833
                                                        <1>
                                                                    ; EDX = Address of 17 bytes;
; an rgbRGB value for each of 16 palette
5834
                                                        <1>
5835
                                                        <1>
5836
                                                        <1>
                                                                    ; registers plus one for the border.
5837
                                                       <1>
                                          5838 00002FC0 89D6
                                                                                  esi, edx ; user buffer
5839 00002FC2 B911000000
5840 00002FC7 89E7
5841 00002FC9 83EC14
                                                                       call transfer_from_user_buffer
5842 00002FCC E876B80000
5843
5844
                                            5845 00002FD1 66BADA03
5846 00002FD5 EC
5847 00002FD6 B100
5848 00002FD8 66BAC003
                                                                                 dx, 3C0h; VGAREG_ACTL_ADDRESS
                                         <1> set_palette_loop:
<1> mov al, cl
    out dx, al
    cl
    out dx, al
    cl
    inc edi
    cl
    inc cl
    inc cl
    inc short set_palette_loop
    cl> mov al, [edi]
    inc di
    inc cl
    inc cl
    inc al, 10h
    inc al, 11h
    out dx, al
    inc out dx, al

5849
5850 00002FDC 88C8
5851 00002FDE EE
5852 00002FDF 8A07
5853 00002FE1 EE
5854 00002FE2 47
5855 00002FE3 FEC1
5856 00002FE5 80F910
5857 00002FE8 75F2
5858 00002FEA B011
5859 00002FEC EE
5860 00002FED 8A07
5861 00002FEF EE
5862 00002FF0 B020
5863 00002FF2 EE
5864
5865 00002FF3 66BADA03
5866 00002FF7 EC
5868 00002FF8 83C414
5869 00002FFB E95DE5FFFF
5870
                                                       <1>
5871
                                                       <1> toggle_intensity:
5872
                                                       <1> ; 10/08/2016
5873
                                                        <1>
                                                                       ; Select Foreground Blink or Bold Background
                                                                       ; BL = 00h = enable bold backgrounds
5874
                                                        <1>
5875
                                                        <1>
                                                                      ;
                                                                                   (16 background colors)
5876
                                                        <1>
                                                                                    01h = enable blinking foreground
5877
                                                        <1>
                                                                    ;
                                                                                      (8 background colors)
5878
                                                       <1>

<1> mov dx, 3DAh ; VGAREG_ACTL_RESET

<1> in al, dx

<1> mov dx, 3C0h ; VGAREG_ACTL_ADDRESS

<1> mov al, 10h

<1> out dx, al

<1> in al, dx

<1> in al, dx

<1> in al, dx

<1> in al, dx

<1> and al, 0F7h

<1> and bl, 01h

<1> shl bl, 3

<1> or al, bl

<1> mov dx, 3C0h ; VGAREG_ACTL_ADDRESS

<1> out dx, al

5879 00003000 66BADA03
                                                      <1>
                                                                       mov
                                                                                 dx, 3DAh ; VGAREG_ACTL_RESET
5880 00003004 EC
5881 00003005 66BAC003
5882 00003009 B010
5883 0000300B EE
5884 0000300C 66BAC103
                                                                                 dx, 3C1h; VGAREG ACTL READ DATA
5885 00003010 EC
5886 00003011 24F7
5887 00003013 80E301
5888 00003016 C0E303
5889 00003019 08D8
5890 0000301B 66BAC003
5891 0000301F EE
                                                      <1>
                                                                       out
                                                                                 dx, al
5892 00003020 B020
                                                      <1>
                                                                       mov al, 20h
5893 00003022 EE
                                                       <1>
                                                                        out dx, al
                                                                        ; ifdef VBOX
5894
                                                        <1>
5895 00003023 66BADA03
                                                                                 dx, 3DAh ; VGAREG_ACTL_RESET
                                                      <1>
                                                                        mov
5896 00003027 EC
                                                        <1>
                                                                                al, dx
5897
                                                        <1>
                                                                        ; endif ; VBOX
5898 00003028 E930E5FFFF
                                                                        jmp VIDEO_RETURN
                                                       <1>
                                                       <1>
5900
                                                        <1> get_single_palette_reg:
5901
                                                        <1>
                                                                       ; 10/08/2016
                                                                        ; Read One Palette Register
5902
                                                        <1>
5903
                                                        <1>
                                                                        ; INPUT:
5904
                                                        <1>
                                                                       ; BL = Palette register to read (00h-0Fh)
5905
                                                        <1>
                                                                        ; OUTPUT:
5906
                                                        <1>
                                                                       ; BH = Current rgbRGB value of specified register
5907
                                                        <1>
                                                                                  for that attribute
                                                                       ;
5908
                                                        <1>
5909 0000302D 80FB14
                                                        <1>
                                                                        cmp bl, 14h
                                                                                   short no_actl_reg2
5910
                                                        <1>
                                                                        ;ja
                                                                                   VIDEO RETURN
5911 00003030 0F8727E5FFFF
                                                        <1>
                                                                        jа
5912
                                                       <1>
5913 00003036 66BADA03
                                                                                   dx, 3DAh ; VGAREG_ACTL_RESET
                                                      <1>
                                                                       mov
5914 0000303A EC
                                                       <1>
                                                                        in
                                                                                   al, dx
5915 0000303B 66BAC003
                                                      <1>
                                                                        mov
                                                                                  dx, 3C0h ; VGAREG_ACTL_ADDRESS
                                                                                  al, bl
5916 0000303F 88D8
                                                      <1>
5917 00003041 EE
                                                       <1>
                                                                        out
                                                                                  dx, al
```

```
5931
                               <1>
 5932
                                <1> read_overscan_border_color:
                                <1> ; 10/08/2016
 5933
                                         ; Read Overscan Register
 5934
                                <1>
                                       ; OUTPUT:
 5935
                                <1>
 5936
                                <1>
                                         ; BH = current rgbRGB value
                                       ; of the overscan/border register
 5937
                               <1>
                               <1> mov bl, 11h
<1> jmp short get_single_palette_reg
<1>
 5938
 5939 00003061 B311
 5940 00003063 EBC8
 5941
 5942
                                <1> get_all_palette_reg:
                                     ; 10/08/2016
                                <1>
 5943
 5944
                                <1>
                                         ; Read All Palette Registers
                                      ; EDX = Address of 17-byte buffer
 5945
                                <1>
                                       ; to receive data
 5946
                                <1>
 5947
                               <1>
                          <1>    mov    edi, edx
<1>    mov    ebx, esp
<1>    mov    esi, ebx
<1>    sub    esp, 20
 5948 00003065 89D7
 5949 00003067 89E3
 5950 00003069 89DE
 5951 0000306B 83EC14
 5952
                               <1>
                               <1> mov cl, 0
5953 0000306E B100
                            <1> mov
<1> get_palette_loop:
 5954
                               <1>
5985 000030AF B911000000
                                       mov ecx, 17; transfer (byte) count
                               <1>
                                       ; ESI = source address in system space
 5986
                               <1>
 5987
                               <1>
                                         ; EDI = user's buffer address
                         <1>
<1>
<1>
 5988 000030B4 E844B70000
                                     call transfer_to_user_buffer
 5989
 5990 000030B9 83C414
                               <1>
                                         add esp, 20
 5991 000030BC E99CE4FFFF
                               <1>
                                       jmp VIDEO_RETURN
 5992
                                <1>
 5993
                                <1> set_single_dac_reg:
                                <1> ; 10\(\bar{7}\)08\(\bar{2}\)016
 5994
                                       ; Set One DAC Color Register
 5995
                                <1>
                                       ; BX = color register to set (0-255)
; CH = green value (00h-3Fh)
; CL = blue value (00h-3Fh)
 5996
                                <1>
 5997
                                <1>
 5998
                                <1>
                                        ; DH = red value (00h-3Fh)
 5999
                                <1>
 6000
                                <1>
 6001 000030C1 6652
                                <1>
                                         push dx
 6002 000030C3 66BAC803
                                <1>
                                               dx, 3C8h; VGAREG DAC WRITE ADDRESS
                                         mov
 6003 000030C7 88D8
                               <1>
                                         mov
                                               al, bl
 6004 000030C9 EE
                               <1>
                                         out dx, al
 6005
                               <1>
                                         ;mov dx, 3C9h ; VGAREG_DAC_DATA
 6006 000030CA 6642
                               <1>
                                               dx
                                         inc
 6007 000030CC 6658
                               <1>
                                         pop
                                               ax
 6008 000030CE 88E0
                               <1>
                                         mov
                                               al, ah
 6009 000030D0 EE
                                <1>
                                         out
                                               dx, al
 6010 000030D1 88E8
                               <1>
                                         mov
                                               al, ch
 6011 000030D3 EE
                               <1>
                                               dx, al
 6012 000030D4 88C8
                               <1>
                                         mov
                                               al, cl
 6013 000030D6 EE
                                <1>
                                         out
                                               dx, al
 6014 000030D7 E981E4FFFF
                               <1>
                                               VIDEO RETURN
                                         jmp
 6015
                                <1>
                                <1> set all dac reg:
 6016
                                       ; 12/08/2016
 6017
                                <1>
                                         ; 11/08/2016
 6018
                                <1>
 6019
                                <1>
                                         ; 10/08/2016
                                         ; Set a Block of DAC Color Register
 6020
                                <1>
                                        ; BX = first DAC register to set (0-00FFh)
 6021
                                <1>
 6022
                                <1>
                                         ; ECX = number of registers to set (0-00FFh)
```

```
6023
                                                    <1>
                                                                   ; EDX = addr of a table of R,G,B values
6024
                                                    <1>
                                                                            (it will be CX*3 bytes long)
6025
                                                    <1>
6026 000030DC 89D6
                                                   <1>
                                                                   mov
                                                                            esi, edx ; user buffer
6027 000030DE 89CA
                                                   <1>
                                                                  mov
                                                                            edx, ecx
                                        shl add
.1> mov
<1> mov
<1> sub
<1> and c
.1> c
.1> c
.1> .1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> ...
.1> 
                                                                             cx, 1 ; *2
6028 000030E0 66D1E1
6029 000030E3 01D1
                                                                            ecx, edx; ecx = 3*ecx
6030 000030E5 89E5
                                                                            ebp, esp
6031 000030E7 89EF
                                                                            edi, ebp
6032 000030E9 29CF
                                                                            edi, ecx
6033 000030EB 6683E7FC
                                                                            di, OFFFCh ; (dword alignment)
6034 000030EF 89FC
                                                                             esp, edi
                                                                  call transfer_from_user_buffer
6035 000030F1 E851B70000
6036
                                                                            VIDEO_RETURN
6037
                                             <1> mov
<1> mov
<1> mov
<1> mov
6038 000030F6 89D1
                                                                             ecx, edx
6039 000030F8 66BAC803
                                                                             dx, 3C8h; VGAREG DAC WRITE ADDRESS
6040 000030FC 88D8
                                                                             al, bl
                                               <1>
                                                                             dx, al
6041 000030FE EE
                                                                   out
                                        6042 000030FF 66BAC903
                                                                  mov dx, 3C9h; VGAREG_DAC_DATA
6043
6044 00003103 8A07
                                                  6045 00003105 EE
6046 00003106 47
6047 00003107 8A07
                                                                            al, [edi]
6048 00003109 EE
                                                                            dx, al
6049 0000310A 47
6050 0000310B 8A07
                                                                            al, [edi]
6051 0000310D EE
                                                                            dx, al
6052 0000310E 47
6053 0000310F 6649
6054 00003111 75F0
                                                                            short set_dac_loop
6055 00003113 89EC
                                                  <1>
                                                                            esp, ebp
                                                                   mov
6056 00003115 E943E4FFFF
                                                  <1>
                                                                 jmp VIDEO RETURN
6057
                                                   <1>
                                                   <1> select_video_dac_color_page:
6058
                                                            ; 10/08/2016
6059
                                                    <1>
                                                                   ; DAC Color Paging Functions
6060
                                                    <1>
6061
                                                    <1>
                                                                   ; BL = 00H = select color paging mode
                                                                              BH = paging mode
6062
                                                    <1>
                                                                                          00h = 4 blocks of 64 registers
6063
                                                    <1>
                                                                                          01h = 16 blocks of 16 registers
6064
                                                    <1>
                                                                   ; BL = 01H = activate color page
                                                    <1>
6065
6066
                                                    <1>
                                                                               BH = DAC color page number
6067
                                                    <1>
                                                                                          00h-03h (4-page/64-reg mode)
                                                                                           00h-0Fh (16-page/16-reg mode)
6068
                                                    <1>
6069
                                                   <1>
6070 0000311A 66BADA03
                                                                  mov dx, 3DAh; VGAREG_ACTL_RESET
                                                   <1>
6071 0000311E EC
                                                                            al, dx
                                                   <1>
                                                                  in
                                         <1>
<1>
                                      mov
6072 0000311F 66BAC003
                                                                            dx, 3C0h ; VGAREG_ACTL_ADDRESS
6073 00003123 B010
                                                                            al, 10h
6074 00003125 EE
                                                                            dx, al
6075 00003126 66BAC103
                                                                            dx, 3C1h ; VGAREG_ACTL_READ_DATA
                                                                             al, dx
6076 0000312A EC
6077 0000312B 80E301
                                                                            bl, 01h
6078 0000312E 750E
                                                                             short set_dac_page
6079 00003130 247F
                                                                            al, 07Fh
6080 00003132 C0E707
                                                                            bh, 7
6081 00003135 08F8
                                                                             al, bh
6082 00003137 66BAC003
                                                                            dx, 3C0h; VGAREG_ACTL_ADDRESS
                                           mov dx,
<1> out dx,
<1> jmp sho
<1> set_dac_page:
<1> push ax
<1> mov dx,

6083 0000313B EE
                                                                            dx, al
6084 0000313C EB1D
                                                                            short set_actl_normal
6085
6086 0000313E 6650
                                             <1> mov dx, 3DAh
<1> in al, dx
<1> mov dx, 3C0h
<1> mov dx, 3C0h
<1> mov al, 14h
<1> out dx, al
<1> pop ax
<1> and al, 80h
<1> jnz short se
<1> shl bh, 2
<1> set dag 16 page:
                                                  <1>
                                                                  mov dx, 3DAh ; VGAREG_ACTL_RESET
6087 00003140 66BADA03
6088 00003144 EC
                                                                            dx, 3C0h; VGAREG_ACTL ADDRESS
6089 00003145 66BAC003
6090 00003149 B014
6091 0000314B EE
6092 0000314C 6658
6093 0000314E 2480
6094 00003150 7503
                                                                             short set_dac_16_page
6095 00003152 C0E702
                                                  <1> set_dac_16_page:
6096
                                                  <1>
<1>
                                                            and bh, OFh
6097 00003155 80E70F
6098 00003158 88F8
                                                                   mov
                                                                            al, bh
6099 0000315A EE
                                               <1>
                                                            out dx, al
6100
                                                  <1> set_actl_normal:
                                                  <1> mov al, 20h <1> out dx, al
6101 0000315B B020
6102 0000315D EE
                                                  <1>
                                                                  ; ifdef VBOX
6103
                                                   <1>
6104 0000315E 66BADA03
                                                    <1>
                                                                   mov dx, 3DAh; VGAREG_ACTL_RESET
6105 00003162 EC
                                                    <1>
                                                                   in
                                                                          al, dx
                                                    <1>
                                                                   ; endif ; VBOX
6107 00003163 E9F5E3FFFF
                                                    <1>
                                                                   jmp VIDEO RETURN
6108
                                                    <1>
6109
                                                    <1> read single dac reg:
                                                                ; 10/08/2<del>0</del>16
6110
                                                    <1>
6111
                                                    <1>
                                                                   ; Read One DAC Color Register
6112
                                                    <1>
                                                                  ; INPUT:
6113
                                                    <1>
                                                                ; BX = color register to read (0-255)
6114
                                                    <1>
                                                                  ; OUTPUT:
                                                                  ; CH = green value (00h-3Fh)
6115
                                                    <1>
                                                                  ; CL = blue value (00h-3Fh)
6116
                                                    <1>
                                                                   ; DH = red value (00h-3Fh)
6117
                                                    <1>
6118
                                                    <1>
6119 00003168 66BAC703
                                                    <1>
                                                                            dx, 3C7h ; VGAREG_DAC_READ_ADDRESS
                                                                  mov
6120 0000316C 88D8
                                                   <1>
                                                                  mov
                                                                             al, bl
6121 0000316E EE
                                                   <1>
                                                                   out
                                                                             dx, al
6122 0000316F 66BAC903
                                                  <1>
                                                                             dx, 3C9h; VGAREG_DAC_DATA
                                                                   mov
6123 00003173 EC
                                                  <1>
                                                                  in
                                                                             al, dx
6124 00003174 88442415
                                                  <1>
                                                                  mov
                                                                             [esp+21], al ; dh
6125 00003178 EC
                                                  <1>
                                                                  in
                                                                             al, dx
6126 00003179 88C5
                                                   <1>
                                                                   mov
                                                                            ch, al
6127 0000317B EC
                                                    <1>
                                                                   in
                                                                            al, dx
```

```
mov cl, al
 6128 0000317C 88C1
                                 <1>
 6129 0000317E 66894C2410
                                 <1>
                                           mov
                                                  [esp+16], cx ; cx
                                                 VIDEO RETURN
 6130 00003183 E9D5E3FFFF
                                 <1>
                                           jmp
 6131
                                  <1>
 6132
                                  <1> read_all_dac_reg:
                                        ; <u>1</u>2/0<u>8</u>/2016
 6133
                                  <1>
                                            ; 11/08/2016
 6134
                                  <1>
                                         ; 10/08/2016
 6135
                                  <1>
                                          ; Read a Block of DAC Color Registers
 6136
                                  <1>
                                           ; BX = first DAC register to read (0-00FFh); ECX = number of registers to read (0-00FFh)
 6137
                                  <1>
 6138
                                  <1>
                                              ; EDX = addr of a buffer to hold R,G,B values
 6139
                                  <1>
 6140
                                  <1>
                                                  (CX*3 bytes long)
 6141
                                  <1>
 6142 00003188 89D7
                                  <1>
                                                  edi, edx ; user buffer
                                           mov
6143 0000318A 89CA
                                  <1>
                                           mov
                                                  edx, ecx
                              <1>
<1>
<1>
 6144 0000318C 66D1E2
                                                  dx, 1 ; *2
                                           shl
 6175 000031C3 E995E3FFFF
                                 <1>
                                          jmp VIDEO_RETURN
 6176
                                  <1>
 6177
                                 <1> set_pel_mask:
                                 <1> ; 10/08/2016
 6178
                             <1>
<1>
 6179
                                           ; BL = mask value

<1> ; BL = mask value
<1> mov dx, 3C6h; V6
<1> mov al, bl
<1> out dx, al
<1> jmp VIDEO_RETURN
<1>
                                           mov dx, 3C6h; VGAREG_PEL_MASK
 6180 000031C8 66BAC603
 6181 000031CC 88D8
 6182 000031CE EE
 6183 000031CF E989E3FFFF
 6184
                                  <1>
 6185
                                  <1> read_pel_mask:
                                       ; 10/08/2016
 6186
                                  <1>
                                            ; Output: BL = mask value
 6187
                                 <1>
                                 <1>
 6188 000031D4 66BAC603
 6189 000031D8 EC
 6190 000031D9 8844240C
 6191 000031DD E97BE3FFFF
 6192
                                  <1>
 6193
                                  <1> read_video_dac_state:
                                  <1> ; 10/08/2016
 6194
 6195
                                  <1>
                                           ; Query DAC Color Paging State
                                         ; Output:
; BH = current active DAC color page
 6196
                                  <1>
 6197
                                  <1>
                                           ; BL = current active DAC paging mode
 6198
                                  <1>
 6199
                                  <1>
 6200 000031E2 66BADA03
                                           mov dx, 3DAh ; VGAREG_ACTL_RESET
                                  <1>
                                        in
                                                  al, dx
 6201 000031E6 EC
                                 <1>
                                                  dx, 3C0h ; VGAREG_ACTL_ADDRESS
                                       mov dx, 3C0
mov al, 10h
out dx, al
 6202 000031E7 66BAC003
                                 <1>
                                 <1>
 6203 000031EB B010
                                                  al, 10h
 6204 000031ED EE
                                 <1>
 6205 000031EE 66BAC103
                               <1>
                                           mov dx, 3C1h; VGAREG_ACTL_READ_DATA
 6206 000031F2 EC
                                 <1>
                                            in
                                                  al, dx
                                           mov
                                                  bl, al
 6207 000031F3 88C3
                                 <1>
 6208 000031F5 C0EB07
                                 <1>
                                           shr
                                                  bl, 7
 6209 000031F8 66BADA03
                                  <1>
                                            mov
                                                   dx, 3DAh ; VGAREG_ACTL_RESET
                          <1>
 6210 000031FC EC
                                            in
                                                   al, dx
 6211 000031FD 66BAC003
                                <1>
                                                   dx, 3C0h ; VGAREG_ACTL_ADDRESS
                        <1> mov dx, 3C0h; VGAREG_ACTL_ADDRESS
<1> mov al, 14h
<1> out dx, al
<1> mov dx, 3C1h; VGAREG_ACTL_READ_DATA
<1> in al, dx
<1> mov bh, al
<1> and bh, 0Fh
<1> test bl, 01
<1> jnz short get_dac_16_page
<1> shr bh, 2
<1> get_dac_16_page:
 6212 00003201 B014
 6213 00003203 EE
 6214 00003204 66BAC103
 6215 00003208 EC
 6216 00003209 88C7
 6217 0000320B 80E70F
 6218 0000320E F6C301
 6219 00003211 7503
 6220 00003213 C0EF02
                         <1> get_dac_16_page:
<1> mov dx, 3DAh; VGAREG_ACTL_RESET
<1> in al, dx
 6221
 6222 00003216 66BADA03
6223 0000321A EC
                                                  dx, 3C0h; VGAREG ACTL ADDRESS
```

```
6233
                                 <1>
6234
                                 <1> ; % include 'vidata.s' ; VIDEO DATA
6235
                                 <1>
6236
                                 <1>; /// End Of VIDEO FUNCTIONS ///
1942
1943
                                     setup_rtc_int:
                                     ; source: http://wiki.osdev.org/RTC
1944
1945 00003231 FA
                                          cli
                                                      ; disable interrupts
1946
                                          ; default int frequency is 1024 Hz (Lower 4 bits of register A is 0110b or 6)
1947
                                          ; in order to change this ...
                                          ; frequency = 32768 >> (rate-1) --> 32768 >> 5 = 1024
1948
1949
                                          ; (rate must be above 2 and not over 15)
                                          ; new rate = 15 --> 32768 >> (15-1) = 2 Hz
1950
1951 00003232 B08A
                                          mov al, 8Ah
1952 00003234 E670
                                                 70h, al ; set index to register A, disable NMI
                                          out
1953 00003236 90
                                          nop
1954 00003237 E471
                                          in
                                                 al, 71h; get initial value of register A
1955 00003239 88C4
                                          mov
                                                 ah, al
1956 0000323B 80E4F0
                                          and
                                                 ah, 0F0h
1957 0000323E B08A
                                                 al, 8Ah
                                          mov
1958 00003240 E670
                                          out
                                                 70h, al ; reset index to register A
1959 00003242 88E0
                                                 al, ah
                                          mov
                                                             ; new rate (0Fh -> 15)
1960 00003244 0C0F
                                                 al, OFh
                                          or
1961 00003246 E671
                                                71h, al; write only our rate to A. Note, rate is the bottom 4 bits.
1962
                                          ; enable RTC interrupt
1963 00003248 B08B
                                          mov
                                                 al, 8Bh ;
1964 0000324A E670
                                          out
                                                 70h, al ; select register B and disable NMI
1965 0000324C 90
                                          nop
1966 0000324D E471
                                                 al, 71h; read the current value of register B
                                          in
1967 0000324F 88C4
                                                 ah, al ;
                                          mov
1968 00003251 B08B
                                           mov
                                                 al, 8Bh ;
1969 00003253 E670
                                                 70h, al ; set the index again (a read will reset the index to register B)
                                          out
1970 00003255 88E0
                                                 al, ah ;
                                          mov
1971 00003257 0C40
                                                 al, 40h;
                                                 71h, al ; write the previous value ORed with 0x40. This turns on bit 6 of register B
1972 00003259 E671
                                          out
1973 0000325B FB
                                           sti
1974 0000325C C3
                                          retn
1975
1976
                                     ; Write memory information
1977
                                     ; 29/01/2016
1978
                                    ; 06/11/2014
1979
                                     ; 14/08/2015
1980
                                    memory_info:
1981 0000325D A1[C4580100]
                                     mov eax, [memory_size] ; in pages
1982 00003262 50
                                          push eax
1983 00003263 C1E00C
                                          shl eax, 12
                                                                       ; in bytes
1984 00003266 BB0A000000
                                          mov ebx, 10
                                                            ; 10
1985 0000326B 89D9
                                          mov ecx, ebx
1986 0000326D BE[59190100]
                                          mov
                                                 esi, mem_total_b_str
1987 00003272 E8BD000000
                                          call bintdstr
1988 00003277 58
                                          pop
                                                eax
1989 00003278 B107
                                          mov
                                                cl, 7
1990 0000327A BE[7D190100]
                                          mov
                                                esi, mem_total_p_str
                                          call bintdstr
1991 0000327F E8B0000000
                                          ; 14/08/2015
1993 00003284 E8C8000000
                                          call calc_free_mem
                                          ; edx = calculated free pages
                                          ; ecx = 0
1995
1996 00003289 A1[C8580100]
                                          mov eax, [free_pages]
                                          cmp eax, edx; calculated free mem value
1997 0000328E 39D0
1998
                                                ; and initial free mem value are same or not?
1999 00003290 751D
                                          jne
                                                short pmim ; print mem info with '?' if not
                                          push edx ; free memory in pages
2000 00003292 52
                                          ;mov eax, edx
2001
                                                 eax, 12; convert page count
2002 00003293 C1E00C
                                          shl
2003
                                                       ; to byte count
                                                cl, 10
2004 00003296 B10A
                                          mov
2005 00003298 BE[9D190100]
                                          mov
                                                esi, free_mem_b_str
2006 0000329D E892000000
                                          call bintdstr
2007 000032A2 58
                                          pop
                                                eax
2008 000032A3 B107
                                          mov
                                                cl, 7
                                                 esi, free_mem_p_str
2009 000032A5 BE[C1190100]
                                          mov
2010 000032AA E885000000
                                          call bintdstr
2011
                                     pmim:
2012 000032AF BE[47190100]
                                                esi, msg_memory_info
                                          mov
2013
                                          ;
2014 000032B4 B407
                                                 ah, 07h ; Black background,
                                          mov
2015
                                                      ; light gray forecolor
2016
                                     print_kmsg: ; 29/01/2016
2017 000032B6 8825[EF580100]
                                       mov [ccolor], ah
                                     pkmsg_loop:
2018
2019 000032BC AC
                                           lodsb
2020 000032BD 08C0
                                                 al, al
                                           or
2021 000032BF 7410
                                           jz short pkmsg_ok
2022 000032C1 56
                                          push esi
2023
                                           ; 13/05/2016
2024 000032C2 0FB61D[EF580100]
                                          movzx ebx, byte [ccolor]
2025
                                                      ; Video page 0 (bh=0)
                                                 _write_tty
2026 000032C9 E8EDE9FFFF
                                          call
2027 000032CE 5E
                                                esi
                                          pop
2028 000032CF EBEB
                                          jmp
                                                short pkmsg_loop
                                     pkmsg_ok:
2029
2030 000032D1 C3
                                          retn
2031
2032
                                     ; Convert binary number to hexadecimal string
                                     : 10/05/2015
2033
2034
                                     ; dsectpm.s (28/02/2015)
                                     ; Retro UNIX 386 v1 - Kernel v0.2.0.6
2035
2036
                                     ; 01/12/2014
2037
                                     ; 25/11/2014
2038
2039
                                     bytetohex:
                                          ; INPUT ->
2040
2041
                                          ; AL = byte (binary number)
2042
                                           ; OUTPUT ->
```

```
2043
                                                AX = hexadecimal string
2044
                                         ;
                                         push ebx
2045 000032D2 53
2046 000032D3 31DB
                                          xor
                                               ebx, ebx
2047 000032D5 88C3
                                               bl, al
                                         mov
2048 000032D7 C0EB04
                                         shr
                                               bl, 4
                                       mov bl, [ebx+hexchrs]
2049 000032DA 8A9B[24330000]
                                        xchg bl, al
2050 000032E0 86D8
2051 000032E2 80E30F
                                         and
                                               bl, 0Fh
2052 000032E5 8AA3[24330000]
                                         mov
                                               ah, [ebx+hexchrs]
2053 000032EB 5B
                                         pop
                                               ebx
2054 000032EC C3
                                         retn
2055
2056
                                    wordtohex:
                                         ; INPUT ->
2057
                                          ; AX = word (binary number)
2058
                                          ; OUTPUT ->
2059
2060
                                               EAX = hexadecimal string
                                         ;
2061
2062 000032ED 53
                                         push ebx
2063 000032EE 31DB
                                         xor ebx, ebx
2064 000032F0 86E0
                                         xchg ah, al
2065 000032F2 6650
                                         push ax
2066 000032F4 88E3
                                         mov
                                               bl, ah
2067 000032F6 C0EB04
                                         shr
                                               bl, 4
2068 000032F9 8A83[24330000]
                                         mov
                                                al, [ebx+hexchrs]
2069 000032FF 88E3
                                         mov
                                               bl, ah
2070 00003301 80E30F
                                         and bl, 0Fh
2071 00003304 8AA3[24330000]
                                         mov
                                               ah, [ebx+hexchrs]
2072 0000330A C1E010
                                         shl
                                               eax, 16
2073 0000330D 6658
                                         pop
                                               ax
2074 0000330F 5B
                                         pop
                                               ebx
2075 00003310 EBC0
                                               short bytetohex
                                          jmp
2076
                                         ;mov bl, al
                                          ;shr bl, 4
2077
2078
                                          ;mov bl, [ebx+hexchrs]
2079
                                          ;xchg bl, al
2080
                                          ; and bl, OFh
                                          ;mov ah, [ebx+hexchrs]
2081
                                          ;pop ebx
2082
                                          ;retn
2083
2084
2085
                                    dwordtohex:
2086
                                         ; INPUT ->
2087
                                          ; EAX = dword (binary number)
                                          ; OUTPUT ->
2088
                                               EDX:EAX = hexadecimal string
2089
                                         ;
2090
2091 00003312 50
                                         push eax
2092 00003313 C1E810
                                         shr
                                               eax, 16
2093 00003316 E8D2FFFFFF
                                         call wordtohex
2094 0000331B 89C2
                                         mov
                                               edx, eax
2095 0000331D 58
                                         pop
                                               eax
2096 0000331E E8CAFFFFF
                                         call wordtohex
2097 00003323 C3
                                         retn
2098
2099
                                    ; 10/05/2015
2100
                                    hex_digits:
2101
                                    hexchrs:
2102 00003324 303132333435363738-
                                     db '0123456789ABCDEF'
2102 0000332D 39414243444546
2103
2104
                                    ; Convert binary number to decimal/numeric string
2105
                                    ; 06/11/2014
2106
                                    ; Temporary Code
2107
2108
                                    bintdstr:
2109
2110
                                         ; EAX = binary number
                                         ; ESI = decimal/numeric string address
2111
2112
                                         ; EBX = divisor (10)
                                         ; ECX = string length (<=10)
2113
2114 00003334 01CE
                                         add esi, ecx
                                    btdstr0:
2115
2116 00003336 4E
                                         dec
                                                esi
2117 00003337 31D2
                                          xor
                                                edx, edx
2118 00003339 F7F3
                                          div
                                                ebx
                                               dl, 30h
2119 0000333B 80C230
                                          add
2120 0000333E 8816
                                          mov
                                                [esi], dl
2121 00003340 FEC9
                                          dec
                                               cl
                                                short btdstr2 ; 08/09/2016
2122 00003342 740C
                                          jz
2123 00003344 09C0
                                                eax, eax
                                          or
2124 00003346 75EE
                                         jnz
                                               short btdstr0
2125
                                    btdstr1:
2126 00003348 4E
                                         dec
                                               esi
2127 00003349 C60620
                                          mov byte [esi], 20h; blank space
2128 0000334C FEC9
                                          dec cl
2129 0000334E 75F8
                                         jnz
                                               short btdstr1
2130
                                    btdstr2:
2131 00003350 C3
                                         retn
2132
2133
                                    ; Calculate free memory pages on M.A.T.
                                    ; 06/11/2014
2134
2135
                                    ; Temporary Code
2136
2137
2138
                                    calc_free_mem:
2139 00003351 31D2
                                         xor edx, edx
2140
                                          ;xor
                                               ecx, ecx
                                         mov cx, [mat_size]; in pages
2141 00003353 668B0D[D8580100]
2142 0000335A C1E10A
                                         shl ecx, 10 ; 1024 dwords per page
2143 0000335D BE00001000
                                         mov esi, MEM_ALLOC_TBL
2144
                                    cfm0:
2145 00003362 AD
                                         lodsd
                                         push ecx
2146 00003363 51
```

```
mov
2147 00003364 B92000000
                                          ecx, 32
2148
                               cfm1:
2149 00003369 D1E8
                                     shr
                                          eax, 1
2150 0000336B 7301
                                         short cfm2
                                     jnc
2151 0000336D 42
                                    inc
                                          edx
2152
                                cfm2:
2153 0000336E E2F9
                                    loop
                                         cfm1
2154 00003370 59
                                     pop
                                          ecx
2155 00003371 E2EF
                                          cfm0
                                     loop
2156 00003373 C3
                                    retn
2157
2158
                               %include 'diskio.s' ; 07/03/2015
                            1
  2
                            <1>; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.2 - diskio.s
                            3
  4
                            <1> ; Last Update: 30/08/2020
                            <1> ; Beginning: 24/01/2016
  6
                            <1> ; ------
  7
  8
                            <1>; Assembler: NASM version 2.11 (trdos386.s)
                            9
 10
                            <1>; Turkish Rational DOS
                            <1>; Operating System Project v2.0 by ERDOGAN TAN (Beginning: 04/01/2016)
 11
 12
                            <1> ; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
 13
 14
                            <1>; diskio.inc (22/08/2015)
 15
                            <1>; Derived from 'IBM PC-XT-286' BIOS source code (1986)
 16
                            17
                            <1>
 18
 19
                            <1>; Retro UNIX 386 v1 Kernel - DISKIO.INC
 20
                            <1> ; Last Modification: 22/08/2015
                                  (Initialized Disk Parameters Data is in 'DISKDATA.INC')
 21
 22
                            <1>;
                                    (Uninitialized Disk Parameters Data is in 'DISKBSS.INC')
 23
                            <1>
                            <1>; DISK I/O SYSTEM - Erdogan Tan (Retro UNIX 386 v1 project)
 24
 25
                            <1> ; /////// DISK I/O SYSTEM //////////
 26
 27
                            <1>
                            <1>; 06/02/2015
 28
 29
                            <1> diskette_io:
                                 clc; 20/07/2020
 30 00003374 F8
                            <1>
 31 00003375 9C
                            <1>
                                    pushfd
                                push cs
 32 00003376 OE
                            <1>
                                    call DISKETTE_IO 1
 33 00003377 E809000000
                            <1>
 34 0000337C C3
                            <1>
                                    retn
                            <1>
                            36
                            37
 38
                            <1>; DISKETTE I/O - Erdogan Tan (Retro UNIX 386 v1 project)
 39
 40
                            <1>; 20/02/2015
                            <1>; 06/02/2015 (unix386.s)
 41
 42
                            <1>; 16/12/2014 - 02/01/2015 (dsectrm2.s)
 43
                            <1>;
 44
                            <1>; Code (DELAY) modifications - AWARD BIOS 1999 (ADISK.EQU, COMMON.MAC)
 45
                            <1>;
                            <1> ; ADISK.EQU
 46
 47
                            <1>
 48
                            <1> ;---- Wait control constants
 49
                            <1>
                            <1> ; amount of time to wait while RESET is active.
 50
 51
                            <1>
 52
                            <1> WAITCPU_RESET_ON EQU
                                                    21
                                                                ; Reset on must last at least 14us
 53
                            <1>
                                                                ;at 250 KBS xfer rate.
 54
                            <1>
                                                                ;see INTEL MCS, 1985, pg. 5-456
 55
                            <1>
                            <1> WAITCPU_FOR_STATUS EQU 100
 56
                                                                ;allow 30 microseconds for
 57
                            <1>
                                                                ; status register to become valid
 58
                            <1>
                                                                ; before re-reading.
 59
                            <1>
                            <1> ; After sending a byte to NEC, status register may remain
 60
 61
                            <1> ;incorrectly set for 24 us.
 62
                            <1>
                            <1> WAITCPU RQM LOW
 63
                                                   EQU 24
                                                                      ; number of loops to check for
                                                                ; RQM low.
 64
                            <1>
 65
                            <1>
                            <1>; COMMON.MAC
 66
 67
                            <1>;
 68
                            <1>;
                                    Timing macros
                            <1>;
 69
 70
                            <1>
 71
                            <1> %macro
                                           SIODELAY 0
                                                                     ; SHORT IODELAY
 72
                                          jmp short $+2
                            <1>
 73
                            <1> %endmacro
 74
                            <1>
 75
                                                                    ; NORMAL IODELAY
                            <1> %macro
                                               IODELAY 0
                                          jmp short $+2
 76
                            <1>
 77
                            <1>
                                          jmp short $+2
 78
                            <1> %endmacro
 79
                            <1>
 80
                            <1> %macro
                                               NEWIODELAY 0
 81
                            <1>
                                          out
                                               0ebh,al
 82
                            <1> %endmacro
 83
                            <1>
                            <1> ; (According to) AWARD BIOS 1999 - ATORGS.ASM (dw -> equ, db -> equ)
 84
 85
                            <1> ;;; WAIT FOR MEM
 86
                            <1> ; WAIT_FDU_INT_LO equ
                                                    017798 ; 2.5 secs in 30 micro units.
 87
                            <1>; WAIT_FDU_INT_HI equ
                                                     1
                            <1> WAIT FDU INT LH
                                                     equ 83334
                                                                    ; 27/02/2015 (2.5 seconds waiting)
 88
 89
                            <1> ;;; WAIT_FOR_PORT
                                                             ; .5 secons in 30 us units.
 90
                            <1>; WAIT FDU SEND LO equ
                                                     16667
                            <1>; WAIT FDU SEND HI equ
 91
                                                     0
                                                   16667 ; 27/02/2015
 92
                            <1> WAIT_FDU_SEND_LH equ
 93
                            <1> ; Time to wait while waiting for each byte of NEC results = .5
```

```
<1>; seconds. .5 seconds = 500,000 micros. 500,000/30 = 16,667.
                       95
96
                        <1> WAIT_FDU_RESULTS_LHequ 16667 ; 27/02/2015
97
98
                        <1> ;;; WAIT REFRESH
99
                        <1> ;amount of time to wait for head settle, per unit in parameter
                        <1> ; table = 1 ms.
100
                        <1> WAIT_FDU_HEAD_SETTLE
                                           equ 33 ; 1 ms in 30 micro units.
101
102
                       <1>
103
                       <1>
                        <1>; ///////// DISKETTE I/O ///////////
104
105
106
                       <1>; 11/12/2014 (copy from IBM PC-XT Model 286 BIOS - POSTEQU.INC)
107
                        <1>
108
                        <1> ;-----
                        <1>; EQUATES USED BY POST AND BIOS :
109
110
111
                        <1>
                        <1> ;----- 8042 KEYBOARD INTERFACE AND DIAGNOSTIC CONTROL REGISTERS ------
112
                       113
114
115
                        <1> ; REFRESH_BIT EQU 00010000B ; REFRESH TEST BIT
                       <1>
116
117
                        <1>; CMOS EQUATES FOR THIS SYSTEM :
118
119
                        <1> ;------
                       <1>;CMOS_PORT EQU 070H ; I/O ADDRESS OF CMOS ADDRESS PORT
<1>;CMOS_DATA EQU 071H ; I/O ADDRESS OF CMOS DATA PORT
<1>;NMI EQU 10000000B ; DISABLE NMI INTERRUPTS MASK -
120
121
122
                                                 ; HIGH BIT OF CMOS LOCATION ADDRESS
123
                       <1>
124
                        <1>
                        <1> ;----- CMOS TABLE LOCATION ADDRESS'S ## -----
125
                        <1> CMOS_DISKETTE EQU 010H ; DISKETTE DRIVE TYPE BYTE
126
127
                        <1>; EQU 011H ; - RESERVED ;C
                                  EQU 012H ; FIXED DISK TYPE BYTE
EQU 013H ; - RESERVED ; E
EQU 014H ; EQUIPMENT WORD LOW BYTE ; C
128
                        <1> CMOS_DISK
                                                                             ; H
                       <1> ;
129
                        <1> CMOS_EQUIP EQU
130
131
                       <1>
                       <1> ;----- DISKETTE EQUATES -------
132
133
134
135
136
137
138
139
140
141
                                               ; 320 K HEAD SETTLE TIME
; 2 SECONDS OF COUNTS FOR MOTOR TURN OFF
142
                        <1> MOTOR_WAIT EQU
143
144
                       <1>
145
                        <1> ;----- DISKETTE ERRORS ------
                       146
147
148
149
150
151
152
153
                       154
                       <1> WRITE_PROTECT EQU
<1> BAD_ADDR_MARK EQU
155
156
157
                        <1> BAD_CMD EQU 001H ; BAD COMMAND PASSED TO DISKETTE I/O
158
                        <1>
                        <1> ;----- DISK CHANGE LINE EQUATES -----
159
                        <1> NOCHGLN EQU 001H ; NO DISK CHANGE LINE AVAILABLE
160
                       <1> CHGLN EQU
                                       002H
                                                 ; DISK CHANGE LINE AVAILABLE
161
162
                        <1>
163
                        <1> ;----- MEDIA/DRIVE STATE INDICATORS -----
                        <1> TRK_CAPA EQU 00000001B ; 80 TRACK CAPABILITY
164
                                                ; MULTIPLE FORMAT CAPABILITY (1.2M) ; DRIVE DETERMINED
                        <1> FMT CAPA
165
                                   EQU
                                        00000010B
                                   EQU 00000100B
                        <1> DRV DET
166
                                   EQU 00010000B ; MEDIA DETERMINED BIT
167
                        <1> MED DET
                                                ; DOUBLE STEP BIT
; MASK FOR CLEARING ALL BUT RATE
                                        00100000B
                        <1> DBL STEP
                                   EQU
168
                                       11000000B
                        <1> RATE MSK
169
                                   EQU
                                       00000000B ; 500 KBS DATA RATE
170
                        <1> RATE 500
                                   EQU
                                       01000000B ; 300 KBS DATA RATE
                        <1> RATE_300
171
                                   EQU
                                                ; 250 KBS DATA RATE ; OPERATION START RATE MASK
                                        10000000B
172
                        <1> RATE 250
                                   EQU
                                   EQU 00001100B
173
                        <1> STRT MSK
                        <1> SEND_MSK
174
                                   EQU
                                       11000000B
                                                ; MASK FOR SEND RATE BITS
175
                        <1>
                        <1> ;----- MEDIA/DRIVE STATE INDICATORS COMPATIBILITY -----
176
                       <1> M3D3U EQU
<1> M3D1U EQU
<1> M1D1U EQU
177
                                        00000000B ; 360 MEDIA/DRIVE NOT ESTABLISHED
                       178
179
180
181
                       <1>
                        <1> ;----- INTERRUPT EQUATES -----
182
                       183
184
185
186
187
                        <1> INTB01
                                       EQU OA1H
188
                       <1>
                       <1> ;------
189
                       190
191
192
193
194
                       <1> ;------
                        <1> ;TIMER EQU 040H ; 8254 TIMER - BASE ADDRESS
195
196
                       <1>
197
                        <1> ;------
198
                        <1> DMA_PAGE EQU 081H ; START OF DMA PAGE REGISTERS
```

94

```
200
                             <1> ; 06/02/2015 (unix386.s, protected mode modifications)
201
                             <1>; (unix386.s <-- dsectrm2.s)
202
                             <1> ; 11/12/2014 (copy from IBM PC-XT Model 286 BIOS - DSEG.INC)
203
                             <1>
204
                             <1>; 27/05/2016 - TRDOS 386 (TRDOS v2.0)
                             <1> ; 10/12/2014
205
                             <1>;
206
207
                             <1> ;int40h:
                             <1>;
208
                                     pushf
                                     push cs
209
                             <1>;
210
                             <1>;
                                     ;cli
                             <1>;
211
                                      call DISKETTE_IO_1
212
                             <1>;
213
                             <1>
                             <1> ; DSKETTE ---- 04/21/86 DISKETTE BIOS
214
                             <1>; (IBM PC XT Model 286 System BIOS Source Code, 04-21-86)
215
216
                             <1>;
217
                             <1>
218
                             219
                             <1>; DISKETTE I/O
220
                                      THIS INTERFACE PROVIDES ACCESS TO THE 5 1/4 INCH 360 KB,
                                     1.2 MB, 720 KB AND 1.44 MB DISKETTE DRIVES.
221
                             <1>;
222
                             <1> ; INPUT
                             <1>; (AH) = 00H RESET DISKETTE SYSTEM
223
224
                             <1>;
                                          HARD RESET TO NEC, PREPARE COMMAND, RECALIBRATE REQUIRED
225
                             <1>;
                                           ON ALL DRIVES
226
                             (AH) = 01H READ THE STATUS OF THE SYSTEM INTO (AH)
@DISKETTE_STATUS FROM LAST OPERATION IS USED
227
                             <1>;
                             <1>;
228
229
                             <1> ;-----
230
                             <1>;
                                   REGISTERS FOR READ/WRITE/VERIFY/FORMAT
                                      (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
231
                             <1>;
232
                             <1>;
                                      (DH) - HEAD NUMBER (0-1 ALLOWED, NOT VALUE CHECKED)
233
                             <1>;
                                      (CH) - TRACK NUMBER (NOT VALUE CHECKED)
                                           MEDIA DRIVE TRACK NUMBER
234
                             <1>;
235
                             <1>;
                                            320/360 320/360
                                                     1.2M
236
                                           320/360
                                                                0 - 39
                             <1>;
                                           1.2M 1.2M
720K 720K
                                                       0-79
0-79
0-79
237
                             <1>;
238
                             <1>;
                                           1.44M 1.44M
239
                             <1>;
                                     (CL) - SECTOR NUMBER (NOT VALUE CHECKED, NOT USED FOR FORMAT)
240
                             <1>;
                             <1>;
                                           MEDIA DRIVE SECTOR NUMBER
241
242
                             <1>;
                                           320/360 320/360
                                                                       1-8/9
243
                             <1>;
                                           320/360
                                                       1.2M
                                                                  1-8/9
                                                        1-15
1-9
1-18
244
                             <1>;
                                            1.2M 1.2M
                                           720K 720K
245
                             <1>;
                                           1.44M 1.44M
246
                             <1>;
247
                             <1>;
                                    (AL) NUMBER OF SECTORS (NOT VALUE CHECKED)
248
                                           MEDIA DRIVE MAX NUMBER OF SECTORS
                             <1>;
                                            320/360 320/360
249
                             <1>;
250
                             <1>;
                                            320/360
                                                       1.2M
                             <1>;
251
                                           1.2M 1.2M
252
                             <1>;
                                           720K 720K
                                                            9
253
                             <1>;
                                            1.44M 1.44M
                                                             18
254
                             <1>;
255
                             <1>;
                                     (ES:BX) - ADDRESS OF BUFFER (NOT REQUIRED FOR VERIFY)
256
                             <1> ;
257
                             <1> ;------
258
                             <1>; (AH) = 02H READ THE DESIRED SECTORS INTO MEMORY
259
                             <1> ;------
                             <1>; (AH) = 03H WRITE THE DESIRED SECTORS FROM MEMORY
260
261
                             262
                             <1>; (AH)= 04H VERIFY THE DESIRED SECTORS
                             <1> ;-----
263
264
                             <1>; (AH)= 05H FORMAT THE DESIRED TRACK
265
                             <1>;
                                           (ES,BX) MUST POINT TO THE COLLECTION OF DESIRED ADDRESS FIELDS
                                           FOR THE TRACK. EACH FIELD IS COMPOSED OF 4 BYTES, (C,H,R,N),
266
                             <1>;
267
                                            WHERE C = TRACK NUMBER, H=HEAD NUMBER, R = SECTOR NUMBER,
                             <1>;
                                           N= NUMBER OF BYTES PER SECTOR (00=128,01=256,02=512,03=1024),
268
                             <1>;
269
                             <1>;
                                           THERE MUST BE ONE ENTRY FOR EVERY SECTOR ON THE TRACK.
270
                             <1>;
                                            THIS INFORMATION IS USED TO FIND THE REQUESTED SECTOR DURING
                             <1>;
271
                                           READ/WRITE ACCESS.
272
                             <1>;
                                           PRIOR TO FORMATTING A DISKETTE, IF THERE EXISTS MORE THAN
273
                                           ONE SUPPORTED MEDIA FORMAT TYPE WITHIN THE DRIVE IN QUESTION,
                             <1>;
                                           THEN "SET DASD TYPE" (INT 13H, AH = 17H) OR 'SET MEDIA TYPE'
274
                             <1>;
                                           (INT 13H, AH = 18H) MUST BE CALLED TO SET THE DISKETTE TYPE
275
                             <1>;
                                            THAT IS TO BE FORMATTED. IF "SET DASD TYPE" OR "SET MEDIA TYPE"
276
                             <1>;
277
                                            IS NOT CALLED, THE FORMAT ROUTINE WILL ASSUME THE
                             <1>;
278
                                            MEDIA FORMAT TO BE THE MAXIMUM CAPACITY OF THE DRIVE.
                             <1>;
                             <1>;
279
280
                             <1>;
                                            THESE PARAMETERS OF DISK BASE MUST BE CHANGED IN ORDER TO
281
                             <1>;
                                            FORMAT THE FOLLOWING MEDIAS:
282
283
                             <1>;
                                            : MEDIA : DRIVE : PARM 1 : PARM 2 :
284
                             <1>;
                                           : 320K : 320K/360K/1.2M : 50H : 8
285
                             <1>;
                                           : 360K : 320K/360K/1.2M : 50H : 9 : 1.2M : 1.2M : 54H : 15 : 720K : 720K/1.44M : 50H : 9 :
286
                             <1>;
                             <1>;
287
                                                                 : 50H : 9
288
                             <1>;
                                           : 1.44M : 1.44M : 6CH : 18
289
                             <1>;
290
                             <1>;
                                            NOTES: - PARM 1 = GAP LENGTH FOR FORMAT
                             <1>;
291
292
                             <1>;
                                                - PARM 2 = EOT (LAST SECTOR ON TRACK)
293
                             <1>;
                                                  - DISK BASE IS POINTED BY DISK POINTER LOCATED
294
                             <1>;
                                                  AT ABSOLUTE ADDRESS 0:78.
                             <1>;
                                                  - WHEN FORMAT OPERATIONS ARE COMPLETE, THE PARAMETERS
                                                  SHOULD BE RESTORED TO THEIR RESPECTIVE INITIAL VALUES.
296
                             <1>;
297
                             <1>;-
                                    (AH) = 08H READ DRIVE PARAMETERS
298
                             <1>;
299
                             <1>;
                                    REGISTERS
300
                             <1>;
                                       INPUT
                             <1>;
                                         (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
301
302
                             <1>;
                                          ** 27/05/2016 - TRDOS 386 (TRDOS v2.0) **
303
                                          ** EBX = Buffer address for floppy disk parameters table **
                             <1>;
```

199

<1>

```
304
                               <1>;
                                          OUTPUT
305
                                            (ES:DI) POINTS TO DRIVE PARAMETER TABLE
                                             *** TRDOS 386 note: floppy disk parameter table (16 bytes)
306
                               <1>;
307
                               <1>;
                                            will be returned to user in EBX, buffer address *** 27/05/2016 ***
308
                               <1>;
309
                               <1>;
                                             (CH) - LOW ORDER 8 OF 10 BITS MAXIMUM NUMBER OF TRACKS
                                             (CL) - BITS 7 & 6 - HIGH ORDER TWO BITS OF MAXIMUM TRACKS
310
                               <1>;
                                                   BITS 5 THRU 0 - MAXIMUM SECTORS PER TRACK
311
                               <1>;
312
                               <1>;
                                             (DH) - MAXIMUM HEAD NUMBER
                                            (DL) - NUMBER OF DISKETTE DRIVES INSTALLED
313
                               <1>;
                               <1> ;
314
                                             (BH) - 0
315
                                <1>;
                                            (BL) - BITS 7 THRU 4 - 0
                               <1>;
                                                   BITS 3 THRU 0 - VALID DRIVE TYPE VALUE IN CMOS
316
317
                               <1>;
                                            (AX) - 0
                                         UNDER THE FOLLOWING CIRCUMSTANCES:
318
                               <1>;
319
                               <1>;
                                            (1) THE DRIVE NUMBER IS INVALID,
320
                                            (2) THE DRIVE TYPE IS UNKNOWN AND CMOS IS NOT PRESENT,
                                            (3) THE DRIVE TYPE IS UNKNOWN AND CMOS IS BAD,
321
                               <1>;
322
                                <1>;
                                            (4) OR THE DRIVE TYPE IS UNKNOWN AND THE CMOS DRIVE TYPE IS INVALID
                                            THEN ES, AX, BX, CX, DH, DI=0 ; DL=NUMBER OF DRIVES.
323
                               <1>;
                               <1> ;
                                            IF NO DRIVES ARE PRESENT THEN: ES, AX, BX, CX, DX, DI=0.
324
325
                               <1>;
                                            @DISKETTE STATUS = 0 AND CY IS RESET.
326
                               <1>;-
                                      ______
327
                               <1>;
                                      (AH) = 15H READ DASD TYPE
                                      OUTPUT REGISTERS
328
                               <1>;
329
                               <1>;
                                        (AH) - ON RETURN IF CARRY FLAG NOT SET, OTHERWISE ERROR
330
                               <1>;
                                              00 - DRIVE NOT PRESENT
                                               01 - DISKETTE, NO CHANGE LINE AVAILABLE
331
                               <1>;
                                               02 - DISKETTE, CHANGE LINE AVAILABLE
332
                               <1>;
                                              03 - RESERVED (FIXED DISK)
333
                               <1>;
334
                               <1>;
                                        (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
335
                                <1> ;--
                               <1>;
                                        (AH) = 16H DISK CHANGE LINE STATUS
336
337
                               <1>;
                                        OUTPUT REGISTERS
338
                               <1>;
                                        (AH) - 00 - DISK CHANGE LINE NOT ACTIVE
                                               06 - DISK CHANGE LINE ACTIVE & CARRY BIT ON
339
                               <1>;
                                        (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
340
                               <1>;
341
                               342
                                         (AH) = 17H SET DASD TYPE FOR FORMAT
                               <1>;
                                        INPUT REGISTERS
343
                               <1>;
344
                               <1>;
                                        (AL) - 00 - NOT USED
345
                               <1>;
                                               01 - DISKETTE 320/360K IN 360K DRIVE
                                              02 - DISKETTE 360K IN 1.2M DRIVE
                               <1>;
346
347
                               <1>;
                                               03 - DISKETTE 1.2M IN 1.2M DRIVE
348
                               <1>;
                                              04 - DISKETTE 720K IN 720K DRIVE
349
                               <1>;
                                        (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED:
                                              (DO NOT USE WHEN DISKETTE ATTACH CARD USED)
350
                               <1>;
351
                               352
                               <1>;
                                        (AH) = 18H SET MEDIA TYPE FOR FORMAT
353
                                        INPUT REGISTERS
                               <1>;
354
                               <1>;
                                         (CH) - LOW ORDER 8 OF 10 BITS MAXIMUM TRACKS
355
                                <1>;
                                        (CL) - BITS 7 & 6 - HIGH ORDER TWO BITS OF MAXIMUM TRACKS
                                               BITS 5 THRU 0 - MAXIMUM SECTORS PER TRACK
356
                               <1>;
357
                               <1>;
                                        (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHACKED)
358
                               <1>;
                                        OUTPUT REGISTERS:
359
                               <1>;
                                         (ES:DI) - POINTER TO DRIVE PARAMETERS TABLE FOR THIS MEDIA TYPE,
360
                               <1>;
                                                UNCHANGED IF (AH) IS NON-ZERO
                                         (AH) - 00H, CY = 0, TRACK AND SECTORS/TRACK COMBINATION IS SUPPORTED
361
                               <1>;
362
                                             - 01H, CY = 1, FUNCTION IS NOT AVAILABLE
                                <1>;
                                             - OCH, CY = 1, TRACK AND SECTORS/TRACK COMBINATION IS NOT SUPPORTED
363
                               <1>;
                               <1> ;
364
                                             - 80H, CY = 1, TIME OUT (DISKETTE NOT PRESENT)
                               <1> ;-
365
                                       DISK CHANGE STATUS IS ONLY CHECKED WHEN A MEDIA SPECIFIED IS OTHER
366
                               <1>;
                               <1>;
367
                                        THAN 360 KB DRIVE. IF THE DISK CHANGE LINE IS FOUND TO BE
                                        ACTIVE THE FOLLOWING ACTIONS TAKE PLACE:
368
                               <1>;
369
                               <1>;
                                              ATTEMPT TO RESET DISK CHANGE LINE TO INACTIVE STATE.
370
                               <1>;
                                               IF ATTEMPT SUCCEEDS SET DASD TYPE FOR FORMAT AND RETURN DISK
                                               CHANGE ERROR CODE
371
                               <1>;
372
                                               IF ATTEMPT FAILS RETURN TIMEOUT ERROR CODE AND SET DASD TYPE
                               <1>;
373
                                              TO A PREDETERMINED STATE INDICATING MEDIA TYPE UNKNOWN.
                               <1>;
374
                               <1>;
                                        IF THE DISK CHANGE LINE IN INACTIVE PERFORM SET DASD TYPE FOR FORMAT.
                               <1>;
375
                               <1> ; DATA VARIABLE -- @DISK_POINTER
376
377
                               <1>; DOUBLE WORD POINTER TO THE CURRENT SET OF DISKETTE PARAMETERS
378
                               <1>;------
379
                               <1>; OUTPUT FOR ALL FUNCTIONS
380
                               <1>; AH = STATUS OF OPERATION
                                              STATUS BITS ARE DEFINED IN THE EQUATES FOR @DISKETTE STATUS
381
                               <1>;
382
                                               VARIABLE IN THE DATA SEGMENT OF THIS MODULE
                                <1>;
                                       CY = 0 SUCCESSFUL OPERATION (AH=0 ON RETURN, EXCEPT FOR READ DASD
383
                               <1>;
384
                               <1>;
                                             TYPE AH=(15)).
385
                                <1>;
                                         CY = 1 FAILED OPERATION (AH HAS ERROR REASON)
386
                                <1>;
                                         FOR READ/WRITE/VERIFY
387
                                              DS, BX, DX, CX PRESERVED
                                <1>;
388
                                         NOTE: IF AN ERROR IS REPORTED BY THE DISKETTE CODE, THE APPROPRIATE
                               <1>;
                               <1> ;
                                               ACTION IS TO RESET THE DISKETTE, THEN RETRY THE OPERATION.
389
390
                               <1>;
                                               ON READ ACCESSES, NO MOTOR START DELAY IS TAKEN, SO THAT
391
                                               THREE RETRIES ARE REQUIRED ON READS TO ENSURE THAT THE
                               <1>;
392
                                               PROBLEM IS NOT DUE TO MOTOR START-UP.
                               <1>;
393
                               <1>;-
                               <1> ;
394
395
                                <1>; DISKETTE STATE MACHINE - ABSOLUTE ADDRESS 40:90 (DRIVE A) & 91 (DRIVE B)
396
                               <1>;
                               <1>;
397
398
                               <1>;
399
                               <1>;
                               <1>;
400
401
                               <1>;
402
                               <1>;
                                                                              403
                               <1>;
404
                               <1>;
405
                               <1>;
                                                                 RESERVED
                                                                          PRESENT STATE
406
                               <1>;
407
                                                                  000: 360K IN 360K DRIVE UNESTABLISHED
                               <1>;
                                                                  001: 360K IN 1.2M DRIVE UNESTABLISHED
408
                               <1>;
```

```
<1> ;
                                                                                                                010: 1.2M IN 1.2M DRIVE UNESTABLISHED
409
410
                                                     <1>;
                                                                                                                011: 360K IN 360K DRIVE ESTABLISHED
                                                                                                               100: 360K IN 1.2M DRIVE ESTABLISHED
                                                     <1>;
411
412
                                                     <1>;
                                                                                                               101: 1.2M IN 1.2M DRIVE ESTABLISHED
                                                     <1>;
                                                                                                               110: RESERVED
413
                                                                                                               111: NONE OF THE ABOVE
414
                                                     <1>;
415
                                                                                                                          MEDIA/DRIVE ESTABLISHED
416
                                                     <1> ;
417
                                                     <1>;
                                                                                           ---->
                                                                                                                          DOUBLE STEPPING REQUIRED (360K IN 1.2M
418
                                                     <1>;
419
                                                     <1>;
                                                                                                              DRIVE)
420
                                                      <1>;
                                                                                                -----> DATA TRANSFER RATE FOR THIS DRIVE:
421
                                                     <1> ;
                                                     <1>;
422
                                                                                                                          00: 500 KBS
423
                                                     <1>;
                                                     <1>;
                                                                                                                          01: 300 KBS
424
                                                                                                                          10: 250 KBS
                                                                                                                          11: RESERVED
426
                                                     <1> ;
427
                                                     <1>;
428
429
430
                                                      <1>; STATE OPERATION STARTED - ABSOLUTE ADDRESS 40:92 (DRIVE A) & 93 (DRIVE B)
431
432
                                                      <1>; PRESENT CYLINDER NUMBER - ABSOLUTE ADDRESS 40:94 (DRIVE A) & 95 (DRIVE B)
433
                                                     434
                                                     <1>
                                                     <1> struc MD
                                                               .SPEC1 resb 1 ; SRT=D, HD UNLOAD=OF - 1ST SPECIFY BYTE .SPEC2 resb 1 ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
436 000000000 <res 00000001>
                                                     <1>
                                                                                                               ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
437 00000001 <res 00000001>
                                                     <1>
                                                                  .OFF_TIM resb 1
                                                             OFF_TIM resb 1 ; WAIT TIME AFTER O
BYT_SEC resb 1 ; 512 BYTES/SECTOR
SEC_TRK resb 1 ; EOT (LAST SECTOR
GAP resb 1 ; GAP LENGTH
DTL resb 1 ; DTL
GAP3 resb 1 ; GAP LENGTH FOR FO
FIL_BYT resb 1 ; FILL BYTE FOR FOR
HD_TIM resb 1 ; HEAD SETTLI
STR_TIM resb 1 ; MOTOR START TIME
MAX_TRK resb 1 ; MAX. TRACK NUMBER
438 00000002 <res 00000001>
                                                                                                              ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
                                                     <1>
439 00000003 <res 00000001>
                                                     <1>
                                                                                                         ; EOT (LAST SECTOR ON TRACK)
440 00000004 <res 00000001>
                                                     <1>
441 00000005 <res 00000001>
                                                     <1>
                                                                                        resb 1 ; DTL
resb 1 ; GAP LENGTH FOR FORMAT
resb 1 ; FILL RYTE TOT
442 00000006 <res 00000001>
                                                     <1>
443 00000007 <res 00000001>
                                                     <1>
444 00000008 <res 00000001>
                                                     <1>
445 00000009 <res 00000001>
                                                                                          resb 1 ; HEAD SETTLE TIME (MILLISECONDS)
                                                     <1>
                                                                                     resb 1 ; MOTOR START TIME (1/8 SECONDS)
resb 1 ; MAX. TRACK NUMBER
446 0000000A <res 00000001>
                                                     <1>
447 0000000B <res 00000001>
                                                     <1>
                                                                     .MAX_TRK
                                                                                                               ; MAX. TRACK NUMBER
448 0000000C <res 00000001>
                                                                                          resb 1
                                                     <1>
                                                                    .RATE
                                                                                                               ; DATA TRANSFER RATE
449
                                                     <1> endstruc
450
                                                     <1>
                                                     <1> BIT70FF
                                                                               EQU
                                                                                          7FH
451
452
                                                     <1> BIT7ON
                                                                               EQU
                                                                                          80H
453
                                                     <1>
454
                                                     <1>; 30/08/2020 - TRDOS 386 v2
                                                     <1>
                                                     <1>;;int13h: ; 16/02/2015
456
457
                                                     <1> ;; 16/02/2015 - 21/02/2015
                                                     <1> int40h:
459 0000337D 9C
                                                     <1>
                                                                  pushfd
460 0000337E 0E
                                                     <1>
                                                                    push cs
461 0000337F E801000000
                                                                     call DISKETTE_IO_1
                                                     <1>
462 00003384 C3
                                                     <1>
463
                                                     <1>
                                                     <1> DISKETTE_IO_1:
464
465
                                                     <1>
466 00003385 FB
                                                                                                              ; INTERRUPTS BACK ON
                                                                     STI
                                                     <1>
467 00003386 55
                                                     <1>
                                                                     PUSH
                                                                                                               ; USER REGISTER
                                                                              eBP
468 00003387 57
                                                                                                               ; USER REGISTER
                                                    <1>
                                                                     PUSH eDI
                                                                                                              ; HEAD #, DRIVE # OR USER REGISTER
469 00003388 52
                                                    <1>
                                                                     PUSH eDX
470 00003389 53
                                                     <1>
                                                                     PUSH
                                                                              eBX
                                                                                                              ; BUFFER OFFSET PARAMETER OR REGISTER
                                                                                                               ; TRACK #-SECTOR # OR USER REGISTER
471 0000338A 51
                                                     <1>
                                                                     PUSH
                                                                               eCX
                                                                                                                         ; BP
                                                                               eBP,eSP
472 0000338B 89E5
                                                     <1>
                                                                     VOM
                                                                                                                                     => PARAMETER LIST DEP. ON AH
                                                                                                               ; [BP] = SECTOR #
473
                                                     <1>
474
                                                     <1>
                                                                                                                ; [BP+1] = TRACK #
                                                                                                                ; [BP+2] = BUFFER OFFSET
475
                                                     <1>
                                                                                                                ; FOR RETURN OF DRIVE PARAMETERS:
476
                                                     <1>
477
                                                     <1>
                                                                                                                ; CL/[BP] = BITS 7&6 HI BITS OF MAX CYL
478
                                                     <1>
                                                                                                                               BITS 0-5 MAX SECTORS/TRACK
                                                                                                                ; CH/[BP+1] = LOW 8 BITS OF MAX CYL.
479
                                                     <1>
                                                                                                                ; BL/[BP+2] = BITS 7-4 = 0
480
                                                     <1>
                                                                                                                                   BITS 3-0 = VALID CMOS TYPE
                                                     <1>
481
482
                                                     <1>
                                                                                                                ; BH/[BP+3] = 0
483
                                                                                                                ; DL/[BP+4] = \# DRIVES INSTALLED
                                                     <1>
                                                                                                                ; DH/[BP+5] = MAX HEAD #
484
                                                     <1>
485
                                                     <1>
                                                                                                                ; DI/[BP+6] = OFFSET TO DISK BASE
486 0000338D 06
                                                                     push es ; 06/02/2015
                                                     <1>
487 0000338E 1E
                                                     <1>
                                                                     PUSH DS
                                                                                                               ; BUFFER SEGMENT PARM OR USER REGISTER
488 0000338F 56
                                                                     PUSH eSI
                                                                                                               ; USER REGISTERS
                                                     <1>
489
                                                     <1>
                                                                     ; CALL DDS
                                                                                                                ; SEGMENT OF BIOS DATA AREA TO DS
490
                                                     <1>
                                                                     ;mov cx, cs
                                             <1>
                                     es, cx

<1>
<1>
<1>
<in purple color="block">
</in purple color="block">
<in purple color="block">
<in purple color="block">
<in purple color="block">
<in purple color="block
                                                                    ;mov ds, cx
492 00003390 66B91000
493 00003394 8ED9
494 00003396 8EC1
496
497 00003398 80FC19
498 0000339B 7202
                                                                               short OK_FUNC ; FUNCTION OK
AH,14H ; REPLACE WITH KNOWN INVALID FUNCTION
499 0000339D B414
                                                   <1> OK_FUNC: 
<1> CMP
500
501 0000339F 80FC01
                                                                                                              ; RESET OR STATUS ?
                                                                   CMP
                                                                               AH,1
                                  <1> CMP
<1> JBE
<1> CMP
<1> CMP
<1> JZ
<1> CMP
<1> MP
<1 JBE
<1> MOV
                                                                               short OK_DRV ; IF RESET OR STATUS DRIVE ALWAYS OK
                                                                              AH,8 ; READ DRIVE PARMS?
short OK_DRV ; IF SO DRIVE CHECKED LATER
DL,1 ; DRIVES 0 AND 1 OK
short OK_DRV ; IF 0 OR 1 THEN JUMP
AH,14H ; REPLACE WITH VICEN
502 000033A2 760C
503 000033A4 80FC08
504 000033A7 7407
505 000033A9 80FA01
506 000033AC 7602
507 000033AE B414
                                                                                                            ; REPLACE WITH KNOWN INVALID FUNCTION
                                                   <1> OK_DRV:
                                           <1> OK_DRV:
<1> xor
509 000033B0 31C9
                                                                               ecx, ecx

<1>
<1>
    ;mov esi, ecx; occ...
<1>
    mov edi, ecx; 08/02/2015
<1>
    MOV CL, AH ; CL = FUNCTION
<1>
    ;XOR CH, CH ; CX = FUNCTION

510
511 000033B2 89CF
512 000033B4 88E1
513
```

```
;SHL CL, 1
SHL CL, 2;
MOV
 514
                                                               ; FUNCTION TIMES 2
                                 <1>
                                          SHL CL, 2; 20/02/2015; FUNCTION TIMES 4 (for 32 bit offset)

MOV eBX,FNC_TAB; LOAD START OF FUNCTION TABLE

ADD eBX,eCX; ADD OFFSET INTO TABLE => ROUTINE
 515 000033B6 C0E102
                                 <1>
515 000033B6 C0E102
516 000033B9 BB[F1330000]
517 000033BE 01CB
                                <1>
 517 000033BE 01CB
                                <1>
                                          ADD eba,eca

MOV AH,DH ; AX = HEAD T, T

XOR DH,DH ; DX = DRIVE #

MOV SI,AX ; SI = HEAD #,# OF SECTORS OR DASD TYPE

MOV DI,DX ; DI = DRIVE #
 518 000033C0 88F4
                                <1>
 519 000033C2 30F6
                                <1>
520 000033C4 6689C6
                                <1>
 521 000033C7 6689D7
                                         MOV
                                 <1>
522
                                 <1>
                                          ; 11/12/2014
 523
                                 <1>
                                          mov [cfd], dl
 524 000033CA 8815[455D0000]
                                 <1>
                                                                           ; current floppy drive (for 'GET_PARM')
                                 <1>
                                                 AH, [DSKETTE STATUS]
 526 000033D0 8A25[48590100]
                                           MOV
                                 <1>
                                                                          ; LOAD STATUS TO AH FOR STATUS FUNCTION
                                                 byte [DSKETTE_STATUS], 0 ; INITIALIZE FOR ALL OTHERS
 527 000033D6 C605[48590100]00
                                 <1>
                                 <1>
528
                                           THROUGHOUT THE DISKETTE BIOS, THE FOLLOWING INFORMATION IS CONTAINED IN
529
                                 <1>;
                                           THE FOLLOWING MEMORY LOCATIONS AND REGISTERS. NOT ALL DISKETTE BIOS
 530
                                 <1>;
                                           FUNCTIONS REQUIRE ALL OF THESE PARAMETERS.
 531
                                 <1>;
 532
                                 <1>;
533
                                 <1>;
                                                 DI : DRIVE #
                                                 SI-HI : HEAD #
 534
                                 <1>;
                                                 SI-LOW: # OF SECTORS OR DASD TYPE FOR FORMAT
 535
                                 <1>;
                                                 ES : BUFFER SEGMENT [BP] : SECTOR #
536
                                 <1>;
 537
                                 <1>;
                                                  [BP+1] : TRACK #
 538
                                                  [BP+2] : BUFFER OFFSET
 539
                                 <1>;
 540
                                 <1>;
                                           ACROSS CALLS TO SUBROUTINES THE CARRY FLAG (CY=1), WHERE INDICATED IN
 541
                                 <1>;
                                 <1>;
                                           SUBROUTINE PROLOGUES, REPRESENTS AN EXCEPTION RETURN (NORMALLY AN ERROR
 542
                                 <1>;
                                           CONDITION). IN MOST CASES, WHEN CY = 1, @DSKETTE_STATUS CONTAINS THE
 543
 544
                                 <1>;
                                           SPECIFIC ERROR CODE.
 545
                                 <1>;
 546
                                 <1>
                                                                    ; (AH) = @DSKETTE_STATUS
                                                                 ; CALL THE REQUESTED FUNCTION ; RESTORE ALL REGISTERS
 547 000033DD FF13
                                 <1>
                                           CALL dWORD [eBX]
                                                 eSI
 548 000033DF 5E
                                 <1>
                                           POP
549 000033E0 1F
                                 <1>
                                           POP
                                                 DS
 550 000033E1 07
                                <1>
                                                 es ; 06/02/2015
                                           pop
 551 000033E2 59
                                <1>
                                           POP
                                                 eCX
 552 000033E3 5B
                                 <1>
                                           POP
                                                 eBX
553 000033E4 5A
                                <1>
                                           POP
                                                 eDX
554 000033E5 5F
                                <1>
                                          POP
                                                 eDI
 555 000033E6 89E5
                                <1>
                                          VOM
                                                 eBP, eSP
                                          PUSH eAX
556 000033E8 50
                                <1>
                          557 000033E9 9C
                                          PUSHFd
                                       POP eAX
 558 000033EA 58
559
                                           ; MOV
                                                 [BP+6], AX
 560 000033EB 89450C
                                                 [ebp+12], eax ; 18/02/2015, flags
                                          mov
                                      POP
POP
 561 000033EE 58
                                                 eAX
 562 000033EF 5D
                                           POP
                                                 eBP
 563 000033F0 CF
                                          IRETd
 564
                                 <1>
565
                                 <1> ;-----
                                                                    ; AH = 06H; INVALID
; AH = 07H; INVALID
                                                 ; AH = 07H; INVALID

DSK_PARMS ; AH = 08H; READ DRIVE PARAMETERS

FNC_ERR ; AH = 09H · TANGATER
                                                                    ; AH = 09H; INVALID
; AH = 0AH; INVALID
                                                                         ; AH = 0BH; INVALID
; AH = 0CH; INVALID
; AH = 0DH; INVALID
                                                                      ; AH = ODH; INVALID
; AH = OEH; INVALID
; AH = OFH; INVALID
; AH = 10H; INVALID
; AH = 11H; INVALID
; AH = 12H
                               587 00003441 [97350000]
 588 00003445 [94360000]
589 00003449 [C4360000]
 590 0000344D [FE360000]
 591 00003451 [81370000]
 592
 593
                                 <1>
                                 594
 595
                                  <1>; DISK_RESET (AH = 00H)
                                                  RESET THE DISKETTE SYSTEM.
 596
                                 <1>;
 597
                                 <1>;
                                 <1>; ON EXIT: @DSKETTE STATUS, CY REFLECT STATUS OF OPERATION
 598
 599
                                 <1> ;-----
 600
                                 <1> DSK RESET:
                                                 DX,03F2H
 601 00003455 66BAF203
                                <1>
                                          MOV
                                                                  ; ADAPTER CONTROL PORT
 602 00003459 FA
                                 <1>
                                                                     ; NO INTERRUPTS
                                           CLI
 603 0000345A A0[46590100]
                                                 AL, [MOTOR STATUS] ; GET DIGITAL OUTPUT REGISTER REFLECTION
                                <1>
                                           MOV
 604 0000345F 243F
                                <1>
                                           AND
                                                 AL,00111111B ; KEEP SELECTED AND MOTOR ON BITS
                                                                   ; MOTOR VALUE TO HIGH NIBBLE
 605 00003461 C0C004
                                 <1>
                                         ROL
                                                 AL,4
                                                                    ; DRIVE SELECT TO LOW NIBBLE
                                 <1>
 606
                                                 AL,00001000B ; TURN ON INTERRUPT ENABLE DX,AL ; RESET THE ADAPTER
 607 00003464 0C08
                                <1>
                                         OR
                                      OUT
 608 00003466 EE
                                 <1>
                                                 byte [SEEK STATUS], 0 ; SET RECALIBRATE REQUIRED ON ALL DRIVES
 609 00003467 C605[45590100]00
                                <1>
                                          VOM
                                        ;JMP $+2
                                                                   ; WAIT FOR I/O
                                 <1>
                                         ;JMP $+2
                                                                    ; WAIT FOR I/O (TO INSURE MINIMUM
 611
                                 <1>
                                                                    ; PULSE WIDTH)
 612
                                 <1>
                                           ; 19/12/2014
613
                                 <1>
 614
                                 <1>
                                          NEWIODELAY
 614 0000346E E6EB
                                 <2> out 0ebh,al
615
                                 <1>
                                 <1>
                                           ; 17/12/2014
 616
 617
                                 <1>
                                           ; AWARD BIOS 1999 - RESETDRIVES (ADISK.ASM)
```

```
<1>
                                      mov ecx, WAITCPU_RESET ON
                                                                   ; cx = 21 -- Min. 14 micro seconds !?
 618 00003470 B915000000
                              <1> wdw1:
                              <1> NEWIODELAY ; 27/02/2015
 620
                          <2> out 0ebh,al
 620 00003475 E6EB
 621 00003477 E2FC
                              <1>
                                    loop wdw1
                             <1> OR AL,00000100B ; TURN OFF RESET BIT
<1> OUT DX,AL ; RESET THE ADAPTER
<1> ; 16/12/2014
<1> IODELAY
<2> imp classification
 622
                         <1> OR <1> OUT
 623 00003479 0C04
 624 0000347B EE
 625
 626
                         <2> jmp short $+2
<2> jmp short $+2
 626 0000347C EB00
 626 0000347E EB00
 627
                              <1>
                                                              ; ENABLE THE INTERRUPTS
 628
                             <1>
                                       ;STI
 WAIT_INT ; WAIT FOR THE INTERRUPT short DR_ERR ; IF ERROR, RETURN IT CX,11000000B ; CL = EXPECTED @NEC_STATUS
648 000034B3 E886030000
                                                             ; SEND SPECIFY COMMAND TO NEC
                              <1> RESBAC:
 649
                                                        ; VARIOUS CLEANUPS
; GET SAVED AL TO BL
; PUT BACK FOR RETURN
 650 000034B8 E83A090000
                             <1> CALL SETUP_END 
<1> MOV BX,SI
 651 000034BD 6689F3
                                             BX,SI
                              <1>
<1>
                                                               ; PUT BACK FOR RETURN
 652 000034C0 88D8
                              <1>
                                       MOV
                                             AL,BL
 653 000034C2 C3
                                       RETn
                              <1> DR_POP_ERR:
 654
 655 000034C3 6659
                              <1> POP
                                             CX
                                                               ; CLEAR STACK
                              <1> DR ERR:
                                             byte [DSKETTE_STATUS], BAD_NEC ; SET ERROR CODE
                             <1> OR <1> JMP
 657 000034C5 800D[48590100]20
                                       JMP SHORT RESBAC ; RETURN FROM RESET
 658 000034CC EBEA
                              <1>
 659
                              <1>
                               660
                               <1>; DISK STATUS (AH = 01H)
 661
                                      DISKETTE STATUS.
                               <1>;
 662
                               <1>;
 663
                               <1> ; ON ENTRY: AH : STATUS OF PREVIOUS OPERATION
 664
 665
                               <1>;
 666
                               <1>; ON EXIT: AH, @DSKETTE STATUS, CY REFLECT STATUS OF PREVIOUS OPERATION.
 667
                               <1> DSK_STATUS:
                              <1> MOV [DSKETTE_STATUS],AH; PUT BACK FOR SETUP END
<1> CALL SETUP_END ; VARIOUS CLEANUPS
<1> MOV BX,SI ; GET SAVED AL TO BL
 669 000034CE 8825[48590100]
 670 000034D4 E81E090000
                              <1> CALL SETUP_END
<1> MOV BX,SI
<1> MOV AL,BL
<1> RETn
 671 000034D9 6689F3
 672 000034DC 88D8
                                                              ; PUT BACK FOR RETURN
 673 000034DE C3
 674
                              <1>
 675
                               <1> ;------
                               <1>; DISK READ (AH = 02H)
 676
                               <1>;
 677
                                     DISKETTE READ.
 678
                               <1>;
                               <1>; ON ENTRY: DI : DRIVE # <1>; SI-HI : HEAD #
 679
 680
                                             SI-LOW: # OF SECTORS
 681
                               <1>;
                              682
 683
 684
 685
                               <1>;
                                             [BP+2] : BUFFER OFFSET
 686
                               <1> ; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
 687
                               <1> ;------
 688
 689
                               <1>
                               <1>; 06/02/2015, ES:BX -> EBX (unix386.s)
 690
                              <1> DSK_READ:
 693 000034DF 8025[46590100]7F
                               <1> AND
                                             byte [MOTOR STATUS], 011111111B; INDICATE A READ OPERATION
 694 000034E6 66B846E6
                                            AX,0E646H ; AX = NEC COMMAND, DMA COMMAND RD_WR_VF ; COMMON READ/WRITE/VERIFY
                                       VOM
                              <1>
                                    CALL RD_WR_VF
 695 000034EA E859040000
                              <1>
 696 000034EF C3
                               <1>
                                       RETn
                               <1>
 698
 699
                               <1> ; DISK WRITE (AH = 03H)
 700
                               <1>; DISKETTE WRITE.
 701
                               <1>;
                               <1> ; ON ENTRY: DI : DRIVE \#
 702
                                             SI-HI : HEAD #
 703
                               <1>;
                                             SI-LOW: # OF SECTORS
 704
                               <1>;
                                             ES : BUFFER SEGMENT [BP] : SECTOR #
 705
                               <1>;
 706
                               <1>;
                                              [BP+1] : TRACK #
 707
                               <1>;
                              <1>;
 708
                                             [BP+2] : BUFFER OFFSET
 709
                               <1>;
                               <1>; ON EXIT: @DSKETTE STATUS, CY REFLECT STATUS OF OPERATION
 710
 711
                               <1>
 712
 713
                               <1>; 06/02/2015, ES:BX -> EBX (unix386.s)
 714
                               <1>
 715
                               <1> DSK WRITE:
                                   MOV AX,0C54AH ; AX = NEC COMMAND, DMA COMMAND
OR byte [MOTOR_STATUS],10000000B ; INDICATE WRITE OPERATION
 716 000034F0 66B84AC5
                              <1>
 717 000034F4 800D[46590100]80
                              <1>
 718 000034FB E848040000
                              <1>
                                       CALL RD_WR_VF ; COMMON READ/WRITE/VERIFY
 719 00003500 C3
                               <1>
                                       RETn
```

```
<1>
720
721
                                         <1>; DISK VERF (AH = 04H)
722
723
                                         <1>;
                                                   DISKETTE VERIFY.
724
                                         <1>;
725
                                         <1> ; ON ENTRY: DI
                                                                     : DRIVE #
                                         <1> ; SI-HI : HEAD #
726
                                                SI-LOW: # OF SECTORS
ES: BUFFER SEGMENT
[BP]: SECTOR #
[BP+1]: TRACK #
[BP+2]: BUFFER OFFSET
727
                                         <1>;
728
                                         <1>;
                                         <1>;
729
730
                                         <1>;
731
                                         <1>;
732
                                         <1> ;
733
                                         <1> ; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
734
                                         <1> ;-----
                                         <1> DSK_VERF:
735
                                        <1> AND <1> MOV <1> CALL
736 00003501 8025[46590100]7F
                                                            byte [MOTOR_STATUS],011111111B; INDICATE A READ OPERATION
737 00003508 66B842E6
                                                    MOV AX,0E642H ; AX = NEC COMMAND, DMA COMMAND
738 0000350C E837040000
                                                                                   ; COMMON READ/WRITE/VERIFY
                                        <1>
                                                     CALL RD_WR_VF
739 00003511 C3
                                                  RETn
                                        <1>
740
                                        <1>
                                         <1> ;-----
741
                                         <1>; DISK_FORMAT (AH = 05H)
742
743
                                         <1> ; DISKETTE FORMAT.
744
                                         <1>;
                                         <1> ; ON ENTRY: DI : DRIVE #
745
                                         <1> ; SI-HI : HEAD \#
746
                                                            SI-LOW: # OF SECTORS
747
                                         <1>;
                                                 ES : BUFFER SEGMENT
[BP] : SECTOR #
[BP+1] : TRACK #
[BP+2] : BUFFER OFFSET
@DISK_POINTER POINTS TO
748
                                         <1>;
                                         <1>;
749
750
                                         <1>;
751
                                         <1>;
                                                            @DISK_POINTER POINTS TO THE PARAMETER TABLE OF THIS DRIVE
752
                                         <1>;
753
                                         <1>;
754
                                         <1>; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
755
                                         <1> ;------
                                         <1> DSK_FORMAT:
                                       <1> DSK_FORMAT:
<1> CALL XLAT_NEW ; TRANSLATE STATE TO PRESENT ARCH.
<1> CALL FMT_INIT ; ESTABLISH STATE IF UNESTABLISHED
<1> OR byte [MOTOR_STATUS], 10000000B; INDICATE WRITE OPERATION
<1> CALL MED_CHANGE ; CHECK MEDIA CHANGE AND RESET IF SO
<1> JC short FM_DON ; MEDIA CHANGED, SKIP
<1> CALL SEND_SPEC ; SEND SPECIFY COMMAND TO NEC
<1> CALL CHK_LASTRATE ; ZF=1 ATTEMPT RATE IS SAME AS LAST RATE
<1> JZ short FM_WR ; YES, SKIP SPECIFY COMMAND
<1> CALL SEND_RATE ; SEND DATA RATE TO CONTROLLER
<1> FM_WR:
757 00003512 E870030000
758 00003517 E86C050000
759 0000351C 800D[46590100]80
760 00003523 E8B4050000
761 00003528 725D
762 0000352A E80F030000
763 0000352F E81A060000
764 00003534 7405
                                      ; SEND DATA RATE TO CONTROLLER

(1) FM_WR:

(1) CALL FMTDMA_SET ; SET UP THE DMA FOR FORMAT

(1) MOV AH,04DH ; RETURN WITH ERROR

(1) CALL NEC_INIT ; INITIALIZE THE NEC

(1) JC short FM_DON ; ERROR - EXIT

(1) MOV eAX, FM_DON ; LOAD ERROR ADDRESS

(1) PUSH eAX ; PUSH NEC_OUT ERROR RETURN

(1) MOV DL,3 ; BYTES/SECTOR VALUE TO NEC

(1) CALL GET_PARM

(1) CALL NEC_OUTPUT

(1) MOV DL,4 ; SECTORS/TRACK VALUE TO NEC

(1) CALL GET_PARM

(1) CALL NEC_OUTPUT

(1) MOV DL,7 ; GAP_LENGTH_VALUE_TO_NEC
765 00003536 E8F1050000
767 0000353B E8A7060000
768 00003540 7245
769 00003542 B44D
770 00003544 E804070000
771 00003549 723C
772 0000354B B8[87350000]
773 00003550 50
774 00003551 B203
775 00003553 E873090000
776 00003558 E8740A0000
777 0000355D B204
778 0000355F E867090000
779 00003564 E8680A0000
                                              MOV DL,7
CALL GET_PARM
CALL NEC_OUTPUT
MOV DL,8
CALL GET_PARM
780 00003569 B207
                                        <1>
                                                                                     ; GAP LENGTH VALUE TO NEC
781 0000356B E85B090000
                                        <1>
782 00003570 E85C0A0000
                                        <1>
783 00003575 B208
                                        <1>
                                                                                     ; FILLER BYTE TO NEC
784 00003577 E84F090000
                                        <1>
                                                    CALL NEC_OUTPUT
785 0000357C E8500A0000
                                        <1>
                                                 POP ear
CALL NEC_TERM
                                                    POP eAX
786 00003581 58
                                        <1>
                                                                                     ; THROW AWAY ERROR
787 00003582 E844070000
                                                                                     ; TERMINATE, RECEIVE STATUS, ETC,
                                        <1>
788
                                        <1> FM_DON:
                                        <1> CALL XLAT_OLD <1> CALL SETUP_END
                                                                                    ; TRANSLATE STATE TO COMPATIBLE MODE
789 00003587 E82C030000
790 0000358C E866080000
                                                                                  ; VARIOUS CLEANUPS
                                                                                  ; GET SAVED AL TO BL
791 00003591 6689F3
                                                            BX,SI
                                        <1>
                                                    VOM
792 00003594 88D8
                                                                                     ; PUT BACK FOR RETURN
                                        <1>
                                                    MOV
                                                            AL,BL
793 00003596 C3
                                        <1>
                                                    RETn
794
                                         <1>
795
                                         <1> ;-----
                                         <1> ; FNC ERR
796
797
                                         <1>;
                                                    INVALID FUNCTION REQUESTED OR INVALID DRIVE:
798
                                                     SET BAD COMMAND IN STATUS.
                                         <1>;
799
                                         <1>;
                                         <1>; ON EXIT: @DSKETTE STATUS, CY REFLECT STATUS OF OPERATION
800
                                        <1> FNC ERR:
                                                                                     ; INVALID FUNCTION REQUEST
803 00003597 6689F0
                                                            AX,SI
                                        <1>
                                                     MOV
                                                                                    ; RESTORE AL
804 0000359A B401
                                        <1>
                                                     VOM
                                                            AH, BAD CMD
                                                                                     ; SET BAD COMMAND ERROR
805 0000359C 8825[48590100]
                                                            [DSKETTE_STATUS], AH; STORE IN DATA AREA
                                        <1>
                                                     MOV
806 000035A2 F9
                                        <1>
                                                                                    ; SET CARRY INDICATING ERROR
807 000035A3 C3
                                        <1>
                                                  RETn
808
                                        <1>
809
                                         <1>; 30/08/2020
810
                                         <1>; 29/08/2020
811
                                         <1> ; 01/06/2016
                                         <1>; 28/05/2016
812
813
                                         <1>; 27/05/2016 - TRDOS 386 (TRDOS v.2.0)
814
                                         <1> ;------
815
                                         <1>; DISK_PARMS (AH = 08H)
816
                                         <1>;
                                                 READ DRIVE PARAMETERS.
                                         <1>;
817
818
                                         <1> ; ON ENTRY: DI : DRIVE #
                                                            ; 27/05/2016
819
                                         <1>;
                                                             EBX = Buffer Address for floppy disk parameters table (16 bytes)
820
                                         <1>;
821
                                         <1>;
822
                                         <1> ; ON EXIT: CL/[BP] = BITS 7 & 6 HI 2 BITS OF MAX CYLINDER
823
                                         <1>;
                                                                            BITS 0-5 MAX SECTORS/TRACK
824
                                         <1>;
                                                             CH/[BP+1] = LOW 8 BITS OF MAX CYLINDER
```

```
<1> ;
825
                                                  BL/[BP+2] = BITS 7-4 = 0
826
                                                             BITS 3-0 = VALID CMOS DRIVE TYPE
                                                  BH/[BP+3] = 0
                                  <1>;
827
828
                                  <1>;
                                                DL/[BP+4] = # DRIVES INSTALLED (VALUE CHECKED)
829
                                  <1>;
                                                  DH/[BP+5] = MAX HEAD #
                                           ** 27/05/2016 - TRDOS 386 (TRDOS v2.0) **

** EBX = Buffer address for floppy disk parameters table **

;DI/[BP+6] = OFFSET TO DISK BASE
830
                                  <1>;
831
                                  <1>;
832
833
                                  <1>;
                                                 ;ES = SEGMENT OF DISK_BASE
834
                                  <1>;
835
                                  <1>;
                                                  AX
                                                          = 0
836
                                  <1>;
837
                                  <1> ;
                                                  NOTE: THE ABOVE INFORMATION IS STORED IN THE USERS STACK AT
                                                  THE LOCATIONS WHERE THE MAIN ROUTINE WILL POP THEM
838
                                  <1>;
839
                                  <1>;
                                                         INTO THE APPROPRIATE REGISTERS BEFORE RETURNING TO THE
840
                                 <1>;
                                                         CALLER.
                                 <1> ;------
                                 <1> DSK_PARMS:
842
                                       ___CALL XLAT_NEW
                                                                      ; TRANSLATE STATE TO PRESENT ARCH,
843 000035A4 E8DE020000
                                 <1>
                                          ; MOV WORD [BP+2],0 ; DRIVE TYPE = 0
                                 <1>
844
                                                                     ; LOAD EQUIPMENT FLAG FOR # DISKETTES
; KEEP DISKETTE DRIVE BITS
                                                   AX, [EQUIP_FLAG]
845
                                 <1>
                                       ; MOV
                                      ; AND
; MOV
                                                      AL,11000001B
846
                                 <1>
                                                      DL,2
                             ; DISKETTE DRIVES = 2
847
                                 <1>
                                                                     ; 2 DRIVES INSTALLED ?
                                                      AL,01000001B
                                                                             ; IF YES JUMP
                                                      short STO_DL
849
850
                                                                              ; DISKETTE DRIVES = 1
                                                  AL,00000001B
                                                                            ; 1 DRIVE INSTALLED ?
                                                   short NON_DRV
852
                                                                            ; IF NO JUMP
853 000035A9 29D2
854 000035AB 66A1[565D0000]
                                                  ax, [fd0_type]
855 000035B1 6621C0
856 000035B4 0F849B000000
                                                    NON DRV
857 000035BA FEC2
858 000035BC 20E4
859 000035BE 7402
                                                    short STO_DL
                            860 000035C0 FEC2
                                <1> STO_DL:
861
862
863 000035C2 6639FA
864 000035C5 0F868A000000
863 000035C2 6639FA
865
866
867 000035CB 895508
868 000035CE 6683FF01
869
870 000035D2 0F8780000000
872 000035D8 C6450901
873 000035DC E8E1080000
874
875
876
877 000035E1 740F
878 000035E3 E82B020000
879 000035E8 7208
880
                      <1>
<1>
882 000035EA 8A4B04
883 000035ED 8A6B0B
884 000035F0 EB36
885
                                <1> CHK_EST:
                                                  AH, [DSK STATE+eDI]; LOAD STATE FOR THIS DRIVE
886 000035F2 8AA7[55590100]
                                <1> MOV
                                                 AH, MED_DET ; CHECK FOR ESTABLISHED STATE
887 000035F8 F6C410
                                 <1>
                                           TEST
                                 <1> JZ
888 000035FB 745B
                                                  short NON_DRV1
                                                                            ; CMOS BAD/INVALID OR UNESTABLISHED
889
                                 <1> USE EST:
                                                 AH, RATE_MSK ; ISULALE ; AH, RATE_250 ; RATE 250 ? ; NO, GO CHECK OTHER RATE
                                 <1> AND
890 000035FD 80E4C0
891 00003600 80FC80
                                <1>
                                           CMP
                                       JNE
892 00003603 757B
                                 <1>
893
                                 <1>
894
                                 <1> ;----
                                                  DATA RATE IS 250 KBS, TRY 360 KB TABLE FIRST
895
                                 <1>

<1>
<1>
MOV AL,01 ; DRIVE TYPE 1 (360KB)
<1>
CALL DR_TYPE_CHECK ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
<1>
MOV CL, [eBX+MD.SEC_TRK] ; GET SECTOR/TRACK
<1>
MOV CH, [eBX+MD.MAX_TRK] ; GET MAX. TRACK NUMBER
<1>
TEST byte [DSK_STATE+eDI], TRK_CAPA ; 80 TRACK ?
<1>
JZ short STO_CX ; MUST BE 360KB DRIVE

896 00003605 B001
896 00003605 B001
897 00003607 E807020000
898 0000360C 8A4B04
899 0000360F 8A6B0B
899 0000360F 8A6B0B
900 00003612 F687[55590100]01
901 00003619 740D
902
                                 <1>
                                                  IT IS 1.44 MB DRIVE
903
                                  <1> ;----
904
                                  <1>
905
                                  <1> PARM144:
                                            MOV AL,04 ; DRIVE TYPE 4 (1.44MB)
CALL DR_TYPE_CHECK ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
906 0000361B B004
                                  <1>
907 0000361D E8F1010000
                                 <1>
908 00003622 8A4B04
                                 <1>
                                              MOV CL, [eBX+MD.SEC_TRK] ; GET SECTOR/TRACK
                                                                            ; GET MAX. TRACK NUMBER
909 00003625 8A6B0B
                                 <1>
                                              MOV
                                                      CH, [eBX+MD.MAX_TRK]
                                 <1> STO CX:
910
                                                 [eBP],eCX
                                                                      ; SAVE POINTER IN STACK FOR RETURN
911 00003628 894D00
                                 <1>
                                           MOV
912
                                 <1> ES_DI:
913
                                 <1>
                                           ;MOV [BP+6],BX
                                                              ; ADDRESS OF MEDIA/DRIVE PARM TABLE
914
                                            ;mov [ebp+12], ebx; 06/02/2015
                                 <1>
                                                          ; SEGMENT MEDIA/DRIVE PARAMETER TABLE
915
                                  <1>
                                            ; MOV AX, CS
916
                                  <1>
                                           ; MOV ES, AX
                                                                      ; ES IS SEGMENT OF TABLE
                                 <1>
917
                                           ; 28/05/2016
                                 <1>
918
919
                                  <1>
                                           ; 27/05/2016
920
                                 <1>
                                            ; return floppy disk parameters table to user
921
                                 <1>
                                           ; in user's buffer, which is pointed by EBX
922
                                 <1>
923 0000362B 57
                                 <1>
                                           push edi
924 0000362C 8B7D04
                                           mov edi, [ebp+4]
                                                                           ; ebx (input), user's buffer address
                                 <1>
                                           ; 29/08/2020
925
                                 <1>
926 0000362F 09FF
                                 <1>
                                                  edi, edi
                                           or
927 00003631 7417
                                 <1>
                                            jz
                                                  short no copy fdpt
                                 <1>
928
929 00003633 0FB6C0
                                 <1>
                                           movzx eax, al
```

```
mov [ebp+4], eax ; ebx ; drive type (for floppy drives)
; 01/06/2016 (INT 33h, disk type return for floppy disks, in BL)
mov [user_buffer], eax ; 01/06/2016 (overwrite ebx return value)
; (INT 33h, Function 08h will replace user's buffer addr with disk type!)
;
mov esi, ebx
 930 00003636 894504
 932 00003639 A3[4C650100]
 933
 934
 935 0000363E 89DE
 936 00003640 B91000000
937 00003645 E8B3B10000
                                     <1> mov ecx, 16; 16 bytes
<1> call transfer_to_user_buffer; trdosk6.s (16/05/2016)
                                     <1> no_copy_fdpt:
<1> pop e
 938
 939 0000364A 5F
                                                        edi
                                      <1> DP_OUT:
 940
                                      941 0000364B E868020000
                                                 XOR AX, AX
 942 00003650 6631C0
                                      <1>
                                                                                ; CLEAR
 943 00003653 F8
                                             CLC
                                      <1>
                                                RETn
 944 00003654 C3
                                      <1>
 945
                                      <1>
                                                         NO DRIYE PRESENT HANDLER
                                      <1> ;----
 946
 947
                                      <1>
                                       <1> NON DRV:
 948
                                      <1> ;MOV BYTE [BP+4],0
 949
                                                                               ; CLEAR NUMBER OF DRIVES
                                                         [ebp+8], edx; 0; 20/02/2015
 950 00003655 895508
                                      <1>
                                                  mov
                                      <1> NON DRV1:
                                                                                ; CHECK FOR FIXED MEDIA TYPE REQUEST
 952 00003658 6681FF8000
                                      <1> CMP
                                                         DI,80H
                                                         short NON DRV2
 953 0000365D 720C
                                      <1>
                                                                                      ; CONTINUE IF NOT REQUEST FALL THROUGH
 954
                                      <1>
                                                         FIXED DISK REQUEST FALL THROUGH ERROR
 955
                                      <1> ;----
                                     ; ELSE TRANSLATE TO COMPATIBLE MODE
 957 0000365F E854020000
                                                        AX,SI ; RESTORE AL
AH,BAD_CMD ; SET BAD COMMAND ERROR
 958 00003664 6689F0
 959 00003667 B401
 960 00003669 F9
                                     <1>
                                                  STC
 961 0000366A C3
                                      <1>
                                                 RETn
                                      <1>
 962
 963
                                      <1> NON_DRV2:
                                      <1>
                                                 ;XOR AX,AX
                                                                                ; CLEAR PARMS IF NO DRIVES OR CMOS BAD
 964

 965 0000366B 31C0
 966 0000366D 66894500
                                                                               ; TRACKS, SECTORS/TRACK = 0
 967
 968 00003671 886509
 969
                                             mov [ebp+12], eax; MOV ES, AX
 970 00003674 89450C
                                      <1>
                                                                              ; ES IS SEGMENT OF TABLE
 971
                                      <1>
                                                 ;JMP SHORT DP_OUT
 972
                                      <1>
 973
                                      <1>
                                            ; 30/08/2020
call XLAT_OLD
;mov ah, NOT_RDY; drive not ready
 974
                                      <1>
 975 00003677 E83C020000
                                      <1>
                                      <1>
                                             mov ah, INIT_FAIL; DRIVE PARAMETER ACTIVITY FAILED
 977 0000367C B407
                                      <1>
 978 0000367E F9
                                      <1>
                                                         ; cf -> 1, ah = 'drive not ready' error code
 979 0000367F C3
                                      <1>
 980
                                      <1>
 981
                                      <1> ;----
                                                         DATA RATE IS EITHER 300 KBS OR 500 KBS, TRY 1.2 MB TABLE FIRST
 982
                                      <1>
 983
                                      <1> USE_EST2:
                             <1> MOV AL,UZ
<1> CALL DR_TYPE_CHECK ; RTN CS:BX = MEDIA/DRIVE FARGE
<1> MOV CL, [eBX+MD.SEC_TRK] ; GET SECTOR/TRACK
<1> MOV CH, [eBX+MD.MAX_TRK] ; GET MAX. TRACK NUMBER
<1> CMP AH,RATE_300 ; RATE 300 ?
<1> JZ short STO_CX ; MUST BE 1.2MB DRIVE
<1> JMP SHORT PARM144 ; ELSE, IT IS 1.44MB DRIVE
<1>
                                                  MOV AL,02 ; DRIVE TYPE 2 (1.2MB)
CALL DR_TYPE_CHECK ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
 984 00003680 B002
                                      <1> MOV
 985 00003682 E88C010000
 986 00003687 8A4B04
 987 0000368A 8A6B0B
 988 0000368D 80FC40
 989 00003690 7496
 990 00003692 EB87
 991
 992
                                      <1>; 30/08/2020
 993
                                       <1>
 994
                                       <1> ;------
 995
                                       <1>; DISK_TYPE (AH = 15H)
                                       <1>; THIS ROUTINE RETURNS THE TYPE OF MEDIA INSTALLED.
 996
                                       <1>;
 997
 998
                                       <1>; ON ENTRY: DI = DRIVE #
999
                                       <1>;
                                       <1>; ON EXIT: AH = DRIVE TYPE, CY=0
1000
1001
                                       <1> ;-----
                                       <1> DSK_TYPE:
1002
                                      <1> CALL XLAT_NEW ; TRANSLATE STATE TO PRESENT ARCH.
1003 00003694 E8EE010000

<pre
1004 00003699 8A87[55590100]
1005 0000369F 08C0
1006 000036A1 7418
1007 000036A3 B401
1008 000036A5 A801
1009 000036A7 7402
1010 000036A9 B402
                                       <1> DT BACK:
1011
                                <1>
1012 000036AB 6650
                                                  PUSH AX
                                                                              ; SAVE RETURN VALUE
1013 000036AD E806020000
                                      <1>
                                                  CALL XLAT_OLD
                                                                              ; TRANSLATE STATE TO COMPATIBLE MODE
                                                                              ; RESTORE RETURN VALUE
                                                         AX
1014 000036B2 6658
                                      <1>
                                                  POP
                                                                             ; NO ERROR
; GET SAVED AL TO BL
1015 000036B4 F8
                                      <1>
                                                  CLC
1016 000036B5 6689F3
                                     <1>
                                                  MOV BX,SI
                                                                          ; GET SAVES ....
; PUT BACK FOR RETURN
1017 000036B8 88D8
                                     <1>
                                                  MOV AL, BL
1018 000036BA C3
                                      <1>
                                                  RETn
1019
                                      <1> NO DRV:
                                                                           ; NO DRIVE PRESENT OR UNKNOWN
1020
                                                 ;XOR AH,AH
                                      <1>
1021
                                       <1>
                                                 ; JMP SHORT DT BACK
1022
                                      <1>
                                               ; 30/08/2020
1023
                                      <1>
                                               call XLAT_OLD
1024 000036BB E8F8010000
                                      <1>
1025 000036C0 29C0
                                      <1>
                                                 sub
                                                        eax, eax
1026 000036C2 F9
                                                  stc ; cf = 1 \rightarrow drive not ready, ah = 0 (disk type = 0)
                                      <1>
1027 000036C3 C3
                                      <1>
1028
                                      <1>
                                       <1>;------
1029
                                       <1>; DISK CHANGE (AH = 16H)
1030
1031
                                       <1>;
                                                  THIS ROUTINE RETURNS THE STATE OF THE DISK CHANGE LINE.
1032
                                       <1>:
1033
                                       <1> ; ON ENTRY: DI = DRIVE #
1034
                                       <1>;
```

```
1035
                              <1> ; ON EXIT: AH = @DSKETTE_STATUS
                              <1>; 00 - DISK CHANGE LINE INACTIVE, CY = 0 <1>; 06 - DISK CHANGE LINE ACTIVE, CY = 1
1036
1037
<1> ;------
                                                             ; TRANSLATE STATE TO PRESENT ARCH.
                                                                    ; GO CHECK STATE OF DISK CHANGE LINE
1050 000036DE C605[48590100]06 <1> SETIT:
                                            MOV byte [DSKETTE_STATUS], MEDIA_CHANGE; INDICATE MEDIA REMOVED
                              <1>
1052 000036E5 E8CE010000
                             <1> FINIS:
                                            CALL XLAT OLD
                                                                   ; TRANSLATE STATE TO COMPATIBLE MODE
                                   CALL SETUP_END
1053 000036EA E808070000
                              <1>
                                                             ; VARIOUS CLEANUPS
                                                             ; GET SAVED AL TO BL
1054 000036EF 6689F3
                             <1>
                                      MOV BX,SI
                             <1> MOV AL, BL <1> RETn
1055 000036F2 88D8
                                                            ; PUT BACK FOR RETURN
1056 000036F4 C3
                             <1> DC_NON:
1058 000036F5 800D[48590100]80 <1> OR
                                            byte [DSKETTE_STATUS], TIME_OUT; SET TIMEOUT, NO DRIVE
1059 000036FC EBE7
                              <1>
                                            SHORT FINIS
                                      JMP
1060
                              <1>
                              <1> ;------
1061
1062
                              <1>; FORMAT_SET (AH = 17H)
1063
                                      THIS ROUTINE IS USED TO ESTABLISH THE TYPE OF MEDIA TO BE USED
                              <1>;
1064
                              <1>;
                                      FOR THE FOLLOWING FORMAT OPERATION.
                              <1> ;
1065
1066
                              <1> ; ON ENTRY: SI LOW = DASD TYPE FOR FORMAT
                              <1> ; DI = DRIVE #
1067
1068
                              <1>;
1069
                              <1> ; ON EXIT: @DSKETTE_STATUS REFLECTS STATUS
                              <1>;
1070
                                            AH = @DSKETTE_STATUS
                                            CY = 1 IF ERROR
1071
1072
                              <1> ;------
1073
                              <1> FORMAT SET:
                                     JMP
1083 0000371F EB48
                              <1>
                                            SHORT SO
                              <1>
                             <1> NOT_320:
1086 00003721 E8B6030000
                             <1> CALL MED_CHANGE
<1> CMP byte [DSKETT
<1> JZ short S0
                                                             ; CHECK FOR TIME OUT
                                            byte [DSKETTE_STATUS], TIME_OUT
1087 00003726 803D[48590100]80 <1>
1088 0000372D 743A
                                            short SO ; IF TIME OUT TELL CALLER
                             <1> S3:
1089
                             <1> DEC <1> JNZ <1> OR <1> JMP <1>
                                                             ; CHECK FOR 320/360K IN 1.2M DRIVE
1090 0000372F 664E
                                            short NOT 320 12 ; BYPASS IF NOT
1091 00003731 7509
1092 00003733 808F[55590100]70
                                      OR
                                            byte [DSK_STATE+eDI], MED_DET+DBL_STEP+RATE_300; SET STATE
1093 0000373A EB2D
                                            SHORT SO
1094
                             <1> NOT_320_12:
1095
                             <1> DEC <1> JNZ
                                            SI ; CHECK FOR 1.2M MEDIA IN 1.2M DRIVE short NOT_12 ; BYPASS IF NOT
1096 0000373C 664E
1097 0000373E 7509
                             <1>
1098 00003740 808F[55590100]10 <1>
                                            byte [DSK_STATE+eDI], MED_DET+RATE_500; SET STATE VARIABLE
1099 00003747 EB20
                              <1>
                                     JMP
                                            SHORT SO ; RETURN TO CALLER
1100
                              <1>
                             <1> NOT_12:
                                                         ; CHECK FOR SET DASD TYPE 04
1102 00003749 664E
                             <1> DEC
                                            SI
                                                             ; BAD COMMAND EXIT IF NOT VALID TYPE
1103 0000374B 752B
                             <1>
                                            short FS_ERR
                                      JNZ
1104
                             <1>
1105 0000374D F687[55590100]04 <1> TEST
1106 00003754 740B <1> JZ
1107 00003756 B050 <1> MOV
1108 00003758 F687[55590100]02 <1> TES
                                           byte [DSK_STATE+eDI], DRV_DET ; DRIVE DETERMINED ?
                                            short ASSUME ; IF STILL NOT DETERMINED ASSUME
                                            AL, MED_DET+RATE_300
                                      TEST byte [DSK_STATE+eDI], FMT_CAPA; MULTIPLE FORMAT CAPABILITY?
1109 0000375F 7502
                              <1>
                                      JNZ
                                           short OR_IT_IN
                                                                   ; IF 1.2 M THEN DATA RATE 300
1110
                              <1>
1111
                              <1> ASSUME:
1112 00003761 B090
                              <1>
                                             AL, MED_DET+RATE_250; SET UP
1113
                              <1>
                              <1> OR_IT_IN:
1114
1115 00003763 0887[55590100]
                              <1>
                                             [DSK_STATE+eDI], AL; OR IN THE CORRECT STATE
                              <1> S0:
                                                        ; TRANSLATE STATE TO COMPATIBLE MODE ; VARIOUS CLEANUPS
1117 00003769 E84A010000
                                       CALL XLAT OLD
                             <1>
1118 0000376E E884060000
                              <1>
                                                            ; GET SAVED AL TO BL
1119 00003773 665B
                              <1>
                                      POP
                                            BX
                                      MOV
1120 00003775 88D8
                              <1>
                                            AL,BL
                                                             ; PUT BACK FOR RETURN
1121 00003777 C3
                              <1>
                                      RETn
1122
                              <1>
1123
                              <1> FS_ERR:
                              <1> MOV
1124 00003778 C605[48590100]01
                                            byte [DSKETTE STATUS], BAD CMD; UNKNOWN STATE, BAD COMMAND
1125 0000377F EBE8
                              <1>
                                       JMP
                                            SHORT SO
1126
                              <1>
1127
                              1128
                              <1> ; SET MEDIA (AH = 18H)
1129
                              <1>;
                                       THIS ROUTINE SETS THE TYPE OF MEDIA AND DATA RATE
                                       TO BE USED FOR THE FOLLOWING FORMAT OPERATION.
1130
                              <1>;
1131
                              <1>;
                              <1> ; ON ENTRY:
1132
                              <1>; [BP] = SECTOR PER TRACK
1133
                                       [BP+1] = TRACK #
1134
                              <1>;
1135
                              <1>;
                                     DI = DRIVE #
1136
                              <1>;
1137
                              <1> ; ON EXIT:
                                       @DSKETTE STATUS REFLECTS STATUS
1138
                              <1>;
1139
                              <1>;
                                      IF NO ERROR:
```

```
AH = 0
CY = 0
ES = SEGMENT OF MEDIA/DRIVE PARAMETER TABLE
DI/[BP+6] = OFFSET OF MEDIA/DRIVE PARAMETER TABLE
1140
                             <1>;
1141
                             <1>;
1142
1143
                             <1>;
                                   IF ERROR:
                             <1>;
1144
                                   AH = @DSKETTE_STATUS
1145
                             <1>;
                                           CY = 1
1146
                             <1> ;
                             <1> ;-----
1147
1148
                             <1> SET_MEDIA:
<1> SM CMOS:
1156
1157 000037A4 E819070000
                             <1> CALL CMOS_TYPE ; RETURN DRIVE TYPE IN (AL)
<1> DR_SEARCH:
1167
                            1168 000037BA 8AA3[D05C0000]
                                           AH, [DR_TYPE+eBX] ; GET DRIVE TYPE
1169 000037C0 80E47F
                                     AND AH, BIT7OFF ; MASK OUT MSB
CMP AL, AH ; DRIVE TYPE MATCH ?
                            <1>
                            <1>
<1>
1170 000037C3 38E0
                                           short NXT_MD ; NO, CHECK NEXT DRIVE TYPE
1171 000037C5 7516
                                      JNE
                             <1> DR_FND:
1172
1173 000037C7 8BBB[D15C0000] <1> MOV eDI, [DR_TYPE+eBX+1] ; DI = MEDIA/DRIVE PARAM TABLE
                     1174
                             <1> MD_SEARCH:
1175 000037CD 8A6704
1176 000037D0 386500
1177 000037D3 7508
1178 000037D5 8A670B
1179 000037D8 386501
1180 000037DB 740F
                             <1> NXT_MD:
1181
                            1182
                                                            ; CHECK NEXT DRIVE TYPE
1183 000037DD 83C305
                            <1>
1184 000037E0 E2D8
                                                            ; RESTORE REG.
1185 000037E2 5F
                            <1> MD NOT FND:
1187 000037E3 C605[48590100]0C <1> MOV byte [DSKETTE_STATUS], MED_NOT_FND; ERROR, MEDIA TYPE NOT FOUND

1188 000037EA EB1C
1190 000037EC 8A470C
                                     CMP AL, RATE 300 ; DOUBLE STEP REQUIRED FOR RATE 300
JNE short MD_SET
1191 000037EF 3C40
1192 000037F1 7502
1193 000037F3 0C20
1194
                            <1> MD_SET:
<1> ;MOV AX, CS ; SEGMENT OF MEDIA/DRIVE PARAMETER TABLE
<1> ;MOV ES, AX ; ES IS SEGMENT OF TABLE
1201
1202
1203
                             <1> SM_RTN:
                             <1> CALL XLAT_OLD ; TRANSLATE STATE TO COMPATIBLE MODE 
<1> CALL SETUP_END ; VARIOUS CLEANUPS
1204 00003808 E8AB000000
1205 0000380D E8E5050000
1206 00003812 C3
                             <1>
                                     RETn
1207
                             <1>
1208
                             <1> ;-----
                             <1> ; DR_TYPE_CHECK
1209
                                      CHECK IF THE GIVEN DRIVE TYPE IN REGISTER (AL)
1210
                              <1>;
                                      IS SUPPORTED IN BIOS DRIVE TYPE TABLE
1211
                              <1>;
1212
                             <1> ; ON ENTRY:
1213
                             <1>; AL = DRIVE TYPE
                              <1> ; ON EXIT:
1214
                             1215
                             1217
1218
1219
                              <1> ; REGISTERS ALTERED: eBX
                              <1> ;-----
1220
1221
                              <1> DR TYPE CHECK:
1222 00003813 6650
                                      PUSH AX
                            <1>
1223 00003815 51
                             <1>
                                      PUSH eCX
1224 00003816 31DB
                             <1>
                                      XOR
                                           eBX,eBX
                                                                  ; BX = INDEX TO DR TYPE TABLE
1225 00003818 B906000000
                                           eCX, DR CNT ; CX = LOOP COUNT
                            <1>
                                      VOM
                            <1> TYPE CHK:
1227 0000381D 8AA3[D05C0000] <1>
                                     MOV AH, [DR_TYPE+eBX] ; GET DRIVE TYPE
1228 00003823 38E0
                             <1>
                                      CMP
                                                            ; DRIVE TYPE MATCH?
1229 00003825 740D
                                      JE
                            <1>
                                           short DR_TYPE_VALID; YES, RETURN WITH CARRY RESET
                                     ; ADD BX, 3 ; CHECK NEXT DRIVE TYPE
1230
                            <1>
                                       add ebx, 5; 16/02/2015 (32 bit address modification)
1231 00003827 83C305
                             <1>
1232 0000382A E2F1
                            <1>
                                     LOOP
                                           TYPE CHK
1233
                             <1>
                                                          ; 1.44MB fd parameter table
                                    mov
1234 0000382C BB[2F5D0000]
                             <1>
                                           ebx, MD TBL6
1235
                             <1>
                                                             ; Default for GET_PARM (11/12/2014)
1236
                             <1>
1237 00003831 F9
                                      STC
                             <1>
                                                            ; DRIVE TYPE NOT FOUND IN TABLE
1238 00003832 EB06
                             <1>
                                      JMP
                                            SHORT TYPE_RTN
                             <1> DR_TYPE_VALID:
1240 00003834 8B9B[D15C0000] <1>
                                           eBX, [DR TYPE+eBX+1] ; BX = MEDIA TABLE
                                     MOV
1241
                             <1> TYPE RTN:
1242 0000383A 59
                            <1>
                                      POP
                                            eCX
1243 0000383B 6658
                             <1>
                                      POP
                                            ΑX
1244 0000383D C3
                             <1>
                                      RETn
```

```
1245
                                   <1>
1246
1247
                                   <1> ; SEND SPEC
1248
                                            SEND THE SPECIFY COMMAND TO CONTROLLER USING DATA FROM
1249
                                            THE DRIVE PARAMETER TABLE POINTED BY @DISK POINTER :
                                   <1>;
                                   <1>; ON ENTRY: @DISK POINTER = DRIVE PARAMETER TABLE
1250
                                   <1>; ON EXIT: NONE
1251
                                   <1> ; REGISTERS ALTERED: CX, DX
1252
1253
                                   <1>;------
1254
                                  <1> SEND_SPEC:
1267
                                  <1> SPECBAC:
1268 00003865 58
                                  <1> POP
                                                                     ; RESTORE ORIGINAL AX VALUE
1269 00003866 C3
                                  <1>
                                            RETn
1270
                                  <1>
                                   <1> ;------
1271
                                   <1> ; SEND SPEC MD
1272
1273
                                   <1>;
                                            SEND THE SPECIFY COMMAND TO CONTROLLER USING DATA FROM
                                   <1>;
1274
                                            THE MEDIA/DRIVE PARAMETER TABLE POINTED BY (CS:BX) :
1275
                                   <1>; ON ENTRY: CS:BX = MEDIA/DRIVE PARAMETER TABLE
                                   <1>; ON EXIT: NONE
1276
                                   <1> ; REGISTERS ALTERED: AX
1277
77
78
279
280 00003867 50
.281 00003868 B8[85380000]
1282 0000386D 50
1283 0000386E B403
1284 00003870 E85C070000
1285 00003875 8A23
100003877 E855070000
1270 8A6301
140070000
                                   <1> ;-----
                            <1> SEND_SPEC_MD:
<1> PUSH eAX ; SAVE RATE DATA
<1> MOV eAX, SPEC_ESBAC ; LOAD ERROR ADDRESS
<1> PUSH eAX ; PUSH NEC_OUT ERROR RETURN
<1> MOV AH,03H ; SPECIFY COMMAND
<1> CALL NEC_OUTPUT ; OUTPUT THE COMMAND
<1> MOV AH,[eBX+MD.SPEC1] ; GET 1ST SPECIFY BYTE
<1> CALL NEC_OUTPUT ; OUTPUT THE COMMAND
<1> MOV AH,[eBX+MD.SPEC2] ; GET SECOND SPECIFY BYTE
<1> CALL NEC_OUTPUT ; OUTPUT THE COMMAND
<1> MOV AH,[eBX+MD.SPEC2] ; GET SECOND SPECIFY BYTE
<1> CALL NEC_OUTPUT ; OUTPUT THE COMMAND
<1> POP eAX ; POP ERROR RETURN
<1> SPEC ESBAC:
                                  <1> SEND_SPEC_MD:
                                  <1> SPEC_ESBAC:
1290
1291 00003885 58
                                                                    ; RESTORE ORIGINAL AX VALUE
                                  <1> POP eAX
1292 00003886 C3
                                  <1>
                                            RETn
1293
                                  <1>
1294
                                   <1> ;-----
                                   <1> ; XLAT NEW
1295
                                   <1>; \overline{\text{TRANSLATES}} DISKETTE STATE LOCATIONS FROM COMPATIBLE <1>; MODE TO NEW ARCHITECTURE.
1296
1297
1298
                                   <1>;
1299
                                   <1> ; ON ENTRY: DI = DRIVE #
1300
                                   <1> ;------
                                  <1> XLAT NEW:
                                                  eDI, 1 ; VALID DRIVE short XN_OUT : TF TMX/ATTR
1302 00003887 83FF01
                                  <1> CMP eDI,1 <1> JA short X
                                                  byte [DSK_STATE+eDI], 0 ; NO DPICE ; NO DPICE
1313
                                  <1> XN_OUT:
1314 000038B1 C3
                                  <1> RETn
                                  <1> DO_DET:
1315
1316 000038B2 E8BF080000
                                                                            ; TRY TO DETERMINE
                                  <1> CALL DRIVE_DET
1317 000038B7 C3
                                  <1>
                                            RETn
1318
                                   <1>
1319
                                   <1> ;-----
                                   <1> ; XLAT OLD
1320
1321
                                   <1>; TRANSLATES DISKETTE STATE LOCATIONS FROM NEW
                                            ARCHITECTURE TO COMPATIBLE MODE.
1322
                                   <1>;
1323
                                   <1>;
1324
                                   <1>; ON ENTRY: DI = DRIVE
1325
                                   <1> ;-----
1326
                                   <1> XLAT OLD:
1327 000038B8 83FF01
                                            CMP eDI,1
                                                                      ; VALID DRIVE ?
                               <1>
                                 <1>
                                            ;JA short XO_OUT
                                                                      ; IF INVALID BACK
                                            ja XO_OUT

CMP byte [DSK_STATE+eDI],0 ; NO DRIVE ?

TF NO DRIVE TRANS
1329 000038BB 0F8786000000
                                  <1>
1330 000038C1 80BF[55590100]00 <1>
                                          JZ short XO_OUT ; IF NO DRIVE TRANSLATE DONE
1331 000038C8 747D
                                 <1>
1332
                                  <1>
                                                   TEST FOR SAVED DRIVE INFORMATION ALREADY SET
1333
                                  <1> ;----
                                 <1>
1334
                                                  , CX = DRIVE NUMBER

; CL = SHIFT COUNT, A=0, B=4

AH, FMT_CAPA

; LOAD MULTIPLE DATA DATA

AH, CL

; PORTATION
1335 000038CA 6689F9
                                            MOV
                                 <1>
                                       ShL
MOV
ROR
TEST
JNZ
                                 <1>
1336 000038CD C0E102
                                 <1>
1337 000038D0 B402
                                                                      ; LOAD MULTIPLE DATA RATE BIT MASK
                                                                   ; ROTATE BY MASK
; MULTIPLE-DATA RATE DETERMINED ?
; IF SO, NO NEED TO RE-SAVE
1338 000038D2 D2CC
                                <1>
1339 000038D4 8425[54590100]
                                 <1>
<1>
                                                  [HF_CNTRL], AH
1340 000038DA 751C
                                                  short SAVE_SET
                                  <1>
                                                   ERASE DRIVE BITS IN @HF_CNTRL FOR THIS DRIVE
1342
                                  <1> ;----
1343
                                  <1>
                                                   AH, DRV_DET+FMT_CAPA+TRK_CAPA ; MASK TO KEEP
1344 000038DC B407
                                 <1>
                                 <1>
                                                   AH, CL ; FIX MASK TO KEEP
1345 000038DE D2CC
                                        ROR
                                                                      ; TRANSLATE MASK
1346 000038E0 F6D4
                                  <1>
                                            NOT
                                                   AH
                                 <1> NOT <1> AND
                                                                      ; KEEP BITS FROM OTHER DRIVE INTACT
1347 000038E2 2025[54590100]
                                                  [HF_CNTRL], AH
1348
                                  <1>
1349
                                  <1> ;----
                                                   ACCESS CURRENT DRIVE BITS AND STORE IN @HF CNTRL
```

```
1350
                                                 <1>
1351 000038E8 8A87[55590100]
                                                                       AL, [DSK STATE+eDI]; ACCESS STATE
1352 000038EE 2407
                                                                       AL,DRV_DET+FMT_CAPA+TRK_CAPA ; KEEP DRIVE BITS
                                                <1>
                                                              AND
1353 000038F0 D2C8
                                                <1>
                                                               ROR
                                                                       AL,CL ; FIX FOR THIS DRIVE
                                                                                                ; UPDATE SAVED DRIVE STATE
1354 000038F2 0805[54590100]
                                                <1>
                                                                       [HF_CNTRL], AL
                                    1355
                                                <1>
1356
1357
1358
1359 000038F8 8AA7[55590100]
1360 000038FE 88E7
1361 00003900 80E4C0
1362 00003903 80FC00
1363 00003906 7410
1364 00003908 B001
1365 0000390A 80FC40
1366 0000390D 7518
1367 0000390F F6C720
1368 00003912 751F
1369
1370 00003914 B007
1371 00003916 EB22
                                       CALL CMOS_TYPE ; RETURN DRIVE 1

1> ;;20/02/2015

1> ;;JC short UNKNO ; ERROR, SET 'NC

1> jz short UNKNO ;; 20/02/2015

1> CMP AL,2 ; 1.2MB DRIVE ?

1> JNE short UNKNO ; NO, GO SET 'NO

1> MOV AL,M1D1U ; AL = 1.2 IN 1.1

1> JMP SHORT TST_DET

1> CHK_250:
                                               <1> CHK_144:
1372
; RETURN DRIVE TYPE IN (AL)
1374
1375
                                                                                                   ; ERROR, SET 'NONE OF ABOVE'
1376 0000391D 74F5
1377 0000391F 3C02
1378 00003921 75F1
                                                                                                   ; NO, GO SET 'NONE OF ABOVE'
1379 00003923 B002
                                                                                                    ; AL = 1.2 IN 1.2 UNESTABLISHED
1380 00003925 EB0C
1381

<!>CHK_250:
<!>CHK_250:
<!>AL = 360 IN 360
<!>AL = 360 IN 360
<!>AL = 360 IN 360
<!>RATE 250 ?
<!>AL = 360 IN 360
<!>RATE 250 ?
<!>SATE 250 ?
<!>AL = 360 IN 360
<!>RATE 250 ?
<!>RATE 250 ?
<!>RATE 250 ?
<!>AL = 360 IN 360
<!>RATE 250 ?
<!>RATE 250 ?
<!>RATE 250 ?
<!>SATE 250 ?
<!>AL = 360 IN 360
<!>RATE 250 ?
<!->CHK 250 ?
<!>RATE 250 ?
<!>RATE 250 ?
<!>AL = 360 IN 360
<!>RATE 250 ?
<!
SATE 250 ?
<!->CHK 250 ?
<!
CHK 250 ?
<!->CHK 250 ?
<!-
1382 00003927 B000
                                                                                                   ; AL = 360 IN 360 UNESTABLISHED
1383 00003929 80FC80
1384 0000392C 75E6
1385 0000392E F6C701
                                                                                                   ; 80 TRACK CAPABILITY ?
                                                                                                   ; IF SO JUMP, FALL THRU TEST DET
1386 00003931 75E1
                                             <1> TST_DET:
1387
                                               <1> TEST BH, MED_DET
1388 00003933 F6C710
                                                                                                   ; DETERMINED ?
                                                                       short AL_SET
1389 00003936 7402
                                                              JZ
                                               <1>
                                                                                                   ; IF NOT THEN SET
                                               <1> ADD
1390 00003938 0403
                                                                                                   ; MAKE DETERMINED/ESTABLISHED
1391
                                                <1> AL SET:
1392 0000393A 80A7[55590100]F8 <1> AND 1393 00003941 0887[55590100] <1> OR
                                                                        byte [DSK_STATE+eDI], ~(DRV_DET+FMT_CAPA+TRK_CAPA) ; CLEAR DRIVE
                                                                        [DSK_STATE+eDI], AL; REPLACE WITH COMPATIBLE MODE
                                                 <1> XO OUT:
1394
1395 00003947 C3
                                                 <1>
                                                            RETn
1396
                                                 <1>
1397
                                                 1398
                                                 <1> ; RD_WR_VF
1399
                                                 <1>; COMMON READ, WRITE AND VERIFY:
1400
                                                 <1>;
                                                              MAIN LOOP FOR STATE RETRIES.
1401
                                                 <1>;
                                                 <1> ; ON ENTRY: AH = READ/WRITE/VERIFY NEC PARAMETER
1402
1403
                                                 <1> ; AL = READ/WRITE/VERIFY DMA PARAMETER
1404
                                                 <1>;
1405
                                                 <1> ; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
                                                 <1> ;-----
1406
                                              ; SAVE DMA, NEC PARAMETERS

<1> CALL XLAT_NEW ; TRANSLATE STATE TO PRESENT ARCH.

<1> CALL SETUP_STATE ; INITIALIZE START AND END RATE

<1> POP AX ; RESTORE READ/WRITE/VERTEY

<1> DO_AGAIN:
1407
1408 00003948 6650
1409 0000394A E838FFFFFF
1410 0000394F E8F3000000
1411 00003954 6658
                                               <1> DO_AGAIN:
1412
                                             <1> PUSH AX
<1> CALL MED_CHANGE
<1> POP AX
<1> JC RWV_END
                                                              PUSH AX ; SAVE READ/WRITE/VERIFY PARAMETER
CALL MED_CHANGE ; MEDIA CHANGE AND RESET IF CHANGED
POP AX ; RESTORE READ/WRITE/VERIFY
1413 00003956 6650
1414 00003958 E87F010000
1415 0000395D 6658
1416 0000395F 0F82C9000000
                                                                                                              ; MEDIA CHANGE ERROR OR TIME-OUT
                                               <1> RWV:
1417
1428 00003982 7413
                                                <1>
                                                              JZ
                                                                        short RWV_2 ; YES, CMOS IS CORRECT
1429 00003984 B002
                                                <1>
                                                              MOV
                                                                       AL,2
                                                                                                    ; CHANGE TO 1.2M
                                                                       SHORT RWV_2
1430 00003986 EB0F
                                                 <1>
                                                              JMP
                                                 <1> RWV 1:
1431
                                                          JB
1432 00003988 720D
                                                                         short RWV_2 ; NO DRIVE SPECIFIED, CONTINUE
                                              <1>
1433 0000398A F687[55590100]01 <1>
                                                                       byte [DSK_STATE+eDI], TRK_CAPA ; IS IT REALLY 40 TRACK?
                                                              TEST
                                                                        short RWV_2 ; NO, 80 TRACK
1434 00003991 7504
                                               <1>
                                                              JNZ
                                              <1> MOV <1> jmp
                                                                                                   ; IT IS 40 TRACK, FIX CMOS VALUE
1435 00003993 B001
                                                                        AL,1
                                                                       short rwv 3
1436 00003995 EB04
                                               <1> RWV_2:
1437
                                               <1> OR <1> .T7
                                                                                                    ; TEST FOR NO DRIVE
1438 00003997 08C0
                                                                        AL, AL
                                                                        short RWV_ASSUME ; ASSUME TYPE, USE MAX TRACK
1439 00003999 742D
                                               <1>
                                                              JΖ
                                               <1> rwv 3:
1440
                                                                                                    ; RTN CS:BX = MEDIA/DRIVE PARAM TBL.
1441 0000399B E873FEFFFF
                                               <1> CALL DR_TYPE_CHECK
                                                                       short RWV_ASSUME ; TYPE NOT IN TABLE (BAD CMOS)
                                               <1>
1442 000039A0 7226
                                                               JC
1443
                                               <1>
1444
                                                <1> ;----
                                                                       SEARCH FOR MEDIA/DRIVE PARAMETER TABLE
1445
                                                <1>
                                                        PUSH eDI
XOR eBX,eBX
                                                                                                   ; SAVE DRIVE #
1446 000039A2 57
                                               <1>
                                                                       eDI ; SAVE DRIVE #
eBX,eBX ; BX = INDE
eCX,DR_CNT ; CX = LOOP COUNT
1448 000039A5 B906000000 <1>
1449
                                                                                                     ; BX = INDEX TO DR_TYPE TABLE
                                               <1>
                                                              MOV
                                              <1> RWV DR SEARCH:
1451 000039B0 80E47F
                                                                       AH,BIT7OFF ; MASK OUT MSB
AL,AH ; DRIVE TYPE M
                                               <1>
                                                               AND
                                               <1> CMP AL,AH ; DRIVE TYPE MATCH?
<1> JNE short RWV_NXT_MD ; NO, CHECK NEXT DRIVE TYPE
1452 000039B3 38E0
1453 000039B5 750B
                                                <1> RWV DR FND:
```

```
1455 000039B7 8BBB[D15C0000] <1> MOV eDI, [DR_TYPE+eBX+1] ; DI = MEDIA/DRIVE PARAMETER TABLE
                    <1> RWV MD SEARH:
  1457 000039BD 3A770C
  1458 000039C0 741B
  1459
                      <1> ;ADD BX,3
<1> add eBX,5
<1> LOOP RWV_DR_SEARCH
<1> POP eDI
<1>
  1460
                                                       ; CHECK NEXT DRIVE TYPE
  1461 000039C2 83C305
  1462 000039C5 E2E3
  1463 000039C7 5F
                                                       ; RESTORE DRIVE #
  1464
                           <1>
                                        ASSUME PRIMARY DRIVE IS INSTALLED AS SHIPPED
  1465
                            <1> ;----
  1466
                            <1>
  1467
                            <1> RWV_ASSUME:
  1468 000039C8 BB[EE5C0000]
                           ; POINT TO 40 TRACK 250 KBS
 1460 000039CD F687[55590100]01 <1> TEST byte [DSK_STATE+eDI], TRK_CAPA; TEST FOR 80 TR
1470 000039D4 740A <1> JZ short RWV_MD_FND1; MUST BE 40 TRACK
1471 000039D6 BB[085D0000] <1> MOV eBX, MD_TBL3; POINT TO 80 TRACK 500 KBS
1472 000039DB EB03 <1> JMP short RWV_MD_FND1; GO SPECIFY PARAMTERS
1473 <1>
                                   TEST byte [DSK_STATE+eDI], TRK_CAPA; TEST FOR 80 TRACK
                                        CS:BX POINTS TO MEDIA/DRIVE PARAMETER TABLE
  1474
                           <1> ;----
  1475
                            <1>
                            <1> RWV MD FND:
  1476
                           <1> MOV eBX,eDI <1> POP eDI
  1477 000039DD 89FB
                                                            ; BX = MEDIA/DRIVE PARAMETER TABLE
  1478 000039DF 5F
                           <1>
                                                       ; RESTORE DRIVE #
  1479
                            <1>
1480
                            <1> ;----
                                        SEND THE SPECIFY COMMAND TO THE CONTROLLER
  1481
                           <1>
                           <1>
<1>
  1520 00003A46 C3
                            <1>
                                   RETn
  1521
                            <1>
  1522
                            <1> ;------
                            <1>; SETUP_STATE: INITIALIZES START AND END RATES.
  1523
                            <1> ;-----
  1524
                            <1> SETUP STATE:
  1525
 <1> AX_SET:
  1536 00003A71 08A7[55590100]
                                   OR [DSK_STATE+eDI], AH; RATE FIRST TO TRY
                            <1>
  1537 00003A77 8025[50590100]F3
                                   AND byte [LASTRATE], ~STRT_MSK; ERASE LAST TO TRY RATE BITS
                         <1>
                                   ROR AL,4 ; TO OPERATION LAST RATE LOCATION OR [LASTRATE], AL ; LAST RATE
  1538 00003A7E C0C804
                           <1>
  1539 00003A81 0805[50590100]
                            <1>
                                   OR
                            <1> J1C:
  1540
  1541 00003A87 C3
                            <1>
  1542
                            <1>
  1543
                            <1> ;-----
  1544
                            <1> ; FMT_INIT: ESTABLISH STATE IF UNESTABLISHED AT FORMAT TIME.
  1545
                            <1> ;-----
  1546
                            <1> FMT INIT:
 1547 00003A88 F687[55590100]10
                           <1> TEST byte [DSK_STATE+eDI], MED_DET; IS MEDIA ESTABLISHED
                           <1>
<1>
<1>
  1559 00003AA7 80CC90
                                   OR AH, MED_DET+RATE_250; ESTABLISH MEDIA
```

```
; SKIP OTHER STATE PROCESSING
1560 00003AAA EB25
                                  <1>
                                                   SHORT SKP_STATE
                                           JMP
                                  <1> N 360:
                                                   AL ; 1.2 M DRIVE short N_12 ; JUMP IF NOT
1562 00003AAC FEC8
                                             DEC
                                  <1>
1563 00003AAE 7505
                                  <1>
                                             JNZ
1564
                                  <1> F1 RATE:
                                 <1>
<1>
                                                    AH, MED_DET+RATE_500; SET FORMAT RATE
1565 00003AB0 80CC10
                                            OR
                                                   SHORT SKP_STATE ; SKIP OTHER STATE PROCESSING
1566 00003AB3 EB1C
                                 <1> N_12:
1567

<1> N_12:
<1> DEC AL ; CHECK FOR TYPE 3
<1> JNZ short N_720 ; JUMP IF NOT
<1> TEST AH, DRV_DET ; IS DRIVE DETERMINED
<1> JZ short ISNT_12 ; TREAT AS NON 1.2 DRIVE
<1> TEST AH, FMT_CAPA ; IS 1.2M
<1> JZ short ISNT_12 ; JUMP IF NOT
<1> OR AH, MED_DET+RATE_300 ; RATE 300
<1> JMP SHORT SKP_STATE ; CONTINUE
<1> N_720:

1568 00003AB5 FEC8
1569 00003AB7 750F
1570 00003AB9 F6C404
1571 00003ABC 7410
1572 00003ABE F6C402
1573 00003AC1 740B
1574 00003AC3 80CC50
1575 00003AC6 EB09
                                                   ; CHECK FOR TYPE 4
short CL_DRV ; NO DRIVE CMC5
                                 <1> N_720:
1576
                                 <1> DEC <1> JNZ <1> JMP
1577 00003AC8 FEC8
                                                   AL
1578 00003ACA 750C
                                                                       ; NO DRIVE, CMOS BAD
1579 00003ACC EBE2
                                                   SHORT F1_RATE
                                  <1> ISNT_12:
1580
                                  <1>
1581 00003ACE 80CC90
                                                   AH, MED DET+RATE 250; MUST BE RATE 250
1582
                                  <1>
1583
                                  <1> SKP_STATE:
1584 00003AD1 88A7[55590100]
                                  <1> MOV
                                                   [DSK STATE+eDI], AH; STORE AWAY
                                  <1> F1_OUT:
1585
1586 00003AD7 C3
                                  <1> RETn
                                  <1> CL_DRV:
1587
                                                   AH,AH ; CLEAR STATE SHORT SKP_STATE ; SAVE IT
                                  <1> XOR
1588 00003AD8 30E4
1589 00003ADA EBF5
                                  <1>
1590
                                  <1>
                                   <1> ;-----
1591
                                   <1> ; MED_CHANGE
1592
1593
                                   <1>; CHECKS FOR MEDIA CHANGE, RESETS MEDIA CHANGE,
1594
                                   <1>;
                                            CHECKS MEDIA CHANGE AGAIN.
1595
                                   <1>;
                                   <1>; ON EXIT: CY = 1 MEANS MEDIA CHANGE OR TIMEOUT
                                   <1>; @DSKETTE_STATUS = ERROR CODE
1597
1598
                                   <1> ;-----
                                   <1> MED_CHANGE:
1599
1600 00003ADC E888060000
                                  <1> CALL READ_DSKCHNG ; READ DISK CHANCE LINE STATE
<1> JZ short MC_OUT ; BYPASS HANDLING DISK CHANGE LINE
1601 00003AE1 7447
                                                  byte [DSK_STATE+eDI], ~MED_DET; CLEAR STATE FOR THIS DRIVE
                                             AND
1602 00003AE3 80A7[55590100]EF
                                  <1>
1603
                                   <1>
                                             THIS SEQUENCE ENSURES WHENEVER A DISKETTE IS CHANGED THAT
1604
                                   <1>;
                                             ON THE NEXT OPERATION THE REQUIRED MOTOR START UP TIME WILL
1605
                                   <1>;
                                             BE WAITED. (DRIVE MOTOR MAY GO OFF UPON DOOR OPENING).
1606
                                   <1>;
1607
                                  <1>
                                                                    ; CL = DRIVE 0
; MOTOR ON BIT MASK
1608 00003AEA 6689F9
                                  <1>
                                             VOM
                                                   CX,DI
                                                   AL,1 ; MOTOR ON BIT MASK
AL,CL ; TO APPROPRIATE POSITION
AL ; KEEP ALL BUT MOTOR ON
; NO INTERRUPTS
1609 00003AED B001
                                  <1>
                                             MOV AL, 1
1610 00003AEF D2E0
                                             SHL
                                 <1>
1611 00003AF1 F6D0
                                  <1>
                                             NOT
1612 00003AF3 FA
                                  <1>
                                                                       ; NO INTERRUPTS
                                             CLI
                                 <1><1>
                                             AND [MOTOR_STATUS], AL ; TURN MOTOR OFF INDICATOR
1613 00003AF4 2005[46590100]
                                                                 ; INTERRUPTS ENABLED
1614 00003AFA FB
                                  <1>
                                            STI
                                  <1>
1615 00003AFB E810040000
                                             CALL MOTOR ON
                                                                       ; TURN MOTOR ON
                                THIS SEQUENCE

<1>
CALL DSK_RESET

NOV CH,01H

CALL SEEK

XOR CH,CH

CALL SEEK

MOV

AND CALL SEEK

CALL SEEK

CALL SEEK

CALL SEEK

CALL SEEK
1616
                                  <1>
                                                   THIS SEQUENCE OF SEEKS IS USED TO RESET DISKETTE CHANGE SIGNAL
1617
1618
                                                                      ; RESET NEC
1619 00003B00 E850F9FFFF
                                                                    ; MOVE TO CYLINDER 1
1620 00003B05 B501
                                                                      ; ISSUE SEEK
1621 00003B07 E8FF040000
                                            XOR CH,CH ; MOVE TO CYLINDER 0
CALL SEEK ; ISSUE SEEK
1622 00003B0C 30ED
1623 00003B0E E8F8040000
1624 00003B13 C605[48590100]06
                                            MOV byte [DSKETTE_STATUS], MEDIA_CHANGE; STORE IN STATUS
                                  <1> OK1:
1625
                                  <1>
                                             CALL READ_DSKCHNG ; CHECK MEDIA CHANGED AGAIN
JZ short OK2 ; IF ACTIVE, NO DISKETTE, TIMEOUT
1626 00003B1A E84A060000
1627 00003B1F 7407
                                  <1>
1628
                                  <1> OK4:
1629 00003B21 C605[48590100]80
                                  <1>
                                             MOV byte [DSKETTE_STATUS], TIME_OUT; TIMEOUT IF DRIVE EMPTY
1630
                                   <1> OK2:
                                                                       ; MEDIA CHANGED, SET CY
1631 00003B28 F9
                                   <1>
1632 00003B29 C3
                                  <1>
                                             RETn
1633
                                   <1> MC_OUT:
1634 00003B2A F8
                                   <1>
                                                                       ; NO MEDIA CHANGED, CLEAR CY
                                             CLC
1635 00003B2B C3
                                   <1>
1636
                                   <1>
1637
                                   1638
                                   <1> ; SEND RATE
1639
                                   <1>; SENDS DATA RATE COMMAND TO NEC
                                   <1> ; ON ENTRY: DI = DRIVE #
1640
1641
                                   <1> ; ON EXIT:
                                                   NONE
1642
                                   <1> ; REGISTERS ALTERED: DX
1643
                                   <1> ;-----
                                   <1> SEND RATE:
1644
                                         PUSH AX
1645 00003B2C 6650
                                   <1>
                                                                       ; SAVE REG.
1646 00003B2E 8025[50590100]3F
                                             AND byte [LASTRATE], ~SEND MSK; ELSE CLEAR LAST RATE ATTEMPTED
                                  <1>
                                 <1> MOV <1> AND <1> OR
1647 00003B35 8A87[55590100]
                                                   AL, [DSK_STATE+eDI]; GET RATE STATE OF THIS DRIVE
                                                                   ; KEEP ONLY RATE BITS
1648 00003B3B 24C0
                                                   AL, SEND MSK
                                                                       ; SAVE NEW RATE FOR NEXT CHECK
1649 00003B3D 0805[50590100]
                                                   [LASTRATE], AL
                                        ROL
                                                   AL,2 ; MOVE TO BIT OUTPUT POSITIONS DX,03F7H ; OUTPUT NEW DATA RATE
1650 00003B43 C0C002
                                  <1>
1651 00003B46 66BAF703
                                  <1>
                                            VOM
1652 00003B4A EE
                                  <1>
                                             OUT
                                                   DX,AL
                                             POP
1653 00003B4B 6658
                                  <1>
                                                                       ; RESTORE REG.
                                                   AX
1654 00003B4D C3
                                   <1>
                                            RETn
1655
                                   <1>
                                   <1> ;------
1656
                                   <1> ; CHK_LASTRATE
1657
1658
                                   <1>;
                                            CHECK PREVIOUS DATE RATE SNT TO THE CONTROLLER.
                                   <1> : ON ENTRY:
1659
                                   <1>; DI = DRIVE #
1660
1661
                                   <1> ; ON EXIT:
                                   <1> ;         ZF = 1 DATA RATE IS THE SAME AS THE LAST RATE SENT TO NEC
<1> ;         ZF = 0 DATA RATE IS DIFFERENT FROM LAST RATE
1662
1663
                                   <1> ; REGISTERS ALTERED: DX
1664
```

```
1665
                                   <1>;------
   1666
                                   <1> CHK LASTRATE:
                                   - PUSH AX ; SAVE REG
<1> AND AH, [LASTRATE] ; GET LAST DATA RATE SELECTED
   1667 00003B4E 6650
   1668 00003B50 2225[50590100] <1>
1669 00003B56 8A87[55590100] <1>
                                  <1> MOV AL, [DSK_STATE+eDI]; GET RATE STATE OF THIS DRIVE
<1> AND AX, SEND_MSK*257; KEEP ONLY RATE BITS C
<1> CMP AL, AH; COMPARE TO PREVIOUSLY TRIED
   1670 00003B5C 6625C0C0
                                            AND AX, SEND_MSK*257 ; KEEP ONLY RATE BITS OF BOTH
   1671 00003B60 38E0
                                                                     ; ZF = 1 RATE IS THE SAME
   1672
                                   <1>
   1673 00003B62 6658
                                   <1>
                                            POP
                                                                     ; RESTORE REG.
                                           RETn
   1674 00003B64 C3
                                   <1>
   1675
                                   <1>
   1676
                                   <1> ;-----
   1677
                                   <1> ; DMA_SETUP
   1678
                                   <1>;
                                            THIS ROUTINE SETS UP THE DMA FOR READ/WRITE/VERIFY OPERATIONS.
   1679
                                   <1>;
                                   <1>; ON ENTRY: AL = DMA COMMAND
   1680
   1681
   1682
                                   <1>; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
                                   <1> ;------
   1683
   1684
                                   <1>
   1685
                                   <1> ; SI = Head #, # of Sectors or DASD Type
   1686
   1687
                                   <1>; 22/08/2015
   1688
                                   <1> ; 08/02/2015 - Protected Mode Modification
   1689
                                   <1> ; 06/02/2015 - 07/02/2015
   1690
                                    <1>; NOTE: Buffer address must be in 1st 16MB of Physical Memory (24 bit limit).
   1691
                                   <1> ; (DMA Addres = Physical Address)
                                   <1> ; (Retro UNIX 386 v1 Kernel/System Mode Virtual Address = Physical Address)
   1692
   1693
                                   <1>;
                                   <1>
   1694
   1695
                                   <1>
   1696
                                   <1>; 04/02/2016 (clc)
                                   <1> ; 20/02/2015 modification (source: AWARD BIOS 1999, DMA_SETUP)
   1697
   1698
                                   <1>; 16/12/2014 (IODELAY)
   1699
                                   <1>
                                   <1> DMA_SETUP:
   1700
   1701
                                   <1>
                                   <1> ;; 20/02/2015
   1702
<1> mov edx, [ebp+4] ; Buffer address
<1> test edx, 0FF000000h ; 16 MB limit (22/08/2015, bugfix)
   1703 00003B65 8B5504
   jnz short dma_bnd_err_stc
                                                                     ; DMA command
                                                                   ; *
                                                                   ; GET BYTES/SECTOR PARAMETER
                                                                   ;
                                                                           ; SHIFT COUNT (0=128, 1=256, 2=512 ETC)
                                                                   ; Sector count
                                                                  ; AH = # OF SECTORS
; AL = 0, AX = # SECTORS * 256
                                                                    ; AX = # SECTORS * 128
                                                                     ; SHIFT BY PARAMETER VALUE
                                                                     ; -1 FOR DMA VALUE
                                                                     ; check for overflow
   1727 00003B9F 723E
                                                   short dma_bnd_err
                                             jс
   1728
                                  <1>
   1729 00003BA1 6629CA
                                   <1>
                                                                   ; Restore start address
                                            sub
                                                  dx, cx
   1730
                                   <1> J33:
                                  <1>
   1731 00003BA4 FA
                                                                     ; DISABLE INTERRUPTS DURING DMA SET-UP
                                  <1>
                                                                   ; SET THE FIRST/LA5T F/F
   1732 00003BA5 E60C
                                                 DMA+12,AL
                                            OUT
                               IODELAY

<2> jmp short $+2

<2> jmp short $+2

<1>
   1733
                                                                           ; WAIT FOR I/O
   1733 00003BA7 EB00
   1733 00003BA9 EB00
                                                              ; OUTPUT THE MODE BYTE ; Buffer address
   1734 00003BAB E60B
                                  <1>
<1>
                                        OUT DMA+11,AL mov eax, edx
   1735 00003BAD 89D0
                                            OUT DMA+4,AL
                                                                   ; OUTPUT LOW ADDRESS
   1736 00003BAF E604
                                  <1>
                                  <1> IODELAY <2> jmp short $+2
   1737
                                                                           ; WAIT FOR I/O
                               <2> jmp short $+2
<2> jmp short $+2
   1737 00003BB1 EB00
   1737 00003BB3 EB00
                                        MOV AL, AH
                                  <1>
   1738 00003BB5 88E0
   1739 00003BB7 E604
                                   <1>
                                            OUT
                                                  DMA+4, AL
                                                                     ; OUTPUT HIGH ADDRESS
                                        shr eax, 16
   1740 00003BB9 C1E810
                                  <1>
   1741
                                   <1>
                                                                            ; I/O WAIT STATE
                                   <2> jmp short $+2
   1741 00003BBC EB00
   1741 00003BBE EB00
                                        jmp short $+2
   1742 00003BC0 E681
                                   <1>
                                             OUT 081H, AL
                                                                            ; OUTPUT highest BITS TO PAGE REGISTER
   1743
                                   <1>
                                            IODELAY
   1743 00003BC2 EB00
                                  <2> jmp short $+2
                                  <2> jmp short $+2
   1743 00003BC4 EB00
                                                            ; Byte count - 1
                                  <1>
   1744 00003BC6 6689C8
                                        mov ax, cx
   1745 00003BC9 E605
                                  <1>
                                             OUT
                                                  DMA+5,AL
                                                                     ; LOW BYTE OF COUNT
                                                                          ; WAIT FOR I/O
   1746
                                  <1>
                                            IODELAY
   1746 00003BCB EB00
                                  <2> jmp short $+2
   1746 00003BCD EB00
                                  <2> jmp short $+2
   1747 00003BCF 88E0
                                  <1>
                                           MOV AL, AH
   1748 00003BD1 E605
                                  <1>
                                             OUT
                                                 DMA+5,AL
                                                                   ; HIGH BYTE OF COUNT
                                            IODELAY
   1749
                                   <1>
                               <2> jmp short $+2
<2> jmp short $+2
<1> STI
   1749 00003BD3 EB00
   1749 00003BD5 EB00
                                                                   ; RE-ENABLE INTERRUPTS
   1750 00003BD7 FB
                                                   AL, 2
                                                                     ; MODE FOR 8237
   1751 00003BD8 B002
                                  <1>
                                             MOV
                                                  DMA+10, AL
                                                                     ; INITIALIZE THE DISKETTE CHANNEL
   1752 00003BDA E60A
                                  <1>
                                             OUT
   1753
                                  <1>
   1754 00003BDC F8
                                   <1>
                                            clc
                                                  ; 04/02/2016
   1755 00003BDD C3
                                  <1>
                                            retn
   1756
                                   <1>
   1757
                                   <1> dma_bnd_err_stc:
```

```
1758 00003BDE F9
                                  <1>
                                          stc
                                  <1> dma bnd err:
                                 1760 00003BDF C605[48590100]09
1761 00003BE6 C3
                                  <1>
                                                                      ; CY SET BY ABOVE IF ERROR
1762
                                  <1>
1763
                                  <1> ;; 16/12/2014
                                  <1> ;; CLI
1764
                                                                     ; DISABLE INTERRUPTS DURING DMA SET-UP
                                  <1>;; OUT DMA+12,AL
<1>;; ;JMP $+2
<1>;; IODELAY
                                                                  ; SET THE FIRST/LAST F/F
1765
1766
                                                                      ; WAIT FOR I/O
1767
                                  , COTPUT THE MODE BYTE

, COMPAL, 42H

(1>;; ; ; JNE short NOT_VERF

(1>;; ; ; XOR AX, AX

(1>;; ; JMP SHORT J33

(1>;;; NOT VERF:
1768
1769
1770
1771
1772
1773
1774
                                  <!> ;;; NOT_VERF:
<!> ;; ; MOV AX, ES ; GET THE ES VALUE
<!> ;; ; ROL AX, 4 ; ROTATE LEFT
<!> ;; ; MOV CH, AL ; GET HIGHEST NIBBLE OF ES TO CH
<!> ;; ; AND AL, 11110000B ; ZERO THE LOW NIBBLE FROM SEGMENT
<!> ;; ; ADD AX, [BP+2] ; TEST FOR CARRY FROM ADDITION
<!> ;; mov eax, [ebp+4] ; 06/02/2015
1775
1776
1777
1778
1779
1780
                                  <1> ;; ;JNC short J33
1781
                                                                     ; CARRY MEANS HIGH 4 BITS MUST BE INC
1782
                                  <1> ;;
                                            ;INC CH
                                  <1> ;;;J33:
1783
1784
1785
                                  <1> ;; ;JMP $+2 <1> ;; IODELAY
1786
1787
1788
                                  <1> ;; MOV AL, AH
                                  <1>;; OUT DMA+4,AL ; OUTPUT HIGH ADDR
<1>;; shr eax, 16 ; 07/02/2015
<1>;; ;MOV AL,CH ; GET HIGH 4 BITS
<1>;; ;JMP $+2 ; I/O WAIT STATE
<1>;; IODELAY
1789
                                                                     ; OUTPUT HIGH ADDRESS
1790
1791
1792
1793
                                  <1>;; 10DELA1
<1>;; ;AND AL,00001111B
1794
                                  <1> ;; OUT 081H,AL
1795
                                                                            ; OUTPUT HIGH 4 BITS TO PAGE REGISTER
1796
                                  <1> ;;
                                            ;SIODELAY
1797
                                  <1> ;;
                                 1798
                                  <1> ;;;---- DETERMINE COUNT
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
                                            PUSH eAX ; 08/02/2015 ; SAVE COUNT VALUE
1811
                                  <1> ;;
                                  <1> ;; OUT DMA+5,AL ; LOW BYTE OF COUNT <1> ;; JMP $+2 ; WAIT FOR I/O
1812
1813
                                  <1>;; IODELAY
1814
                                  <1> ;; MOV AL, AH
1815
                                                  DMA+5,AL
1816
                                  <1> ;;
                                            OUT
                                                                    ; HIGH BYTE OF COUNT
                                  <1> ;;
1817
                                            ; IODELAY
                                  1818
1819
1820
1821
                                  <1>;; add ecx, eax; 08/02/2015 <1>;; MOV AL, 2; N
1822
                                                  AL, 2 ; MODE FOR 8237
$+2 ; WAIT FOR I/O
1823
                                  <1>;; ;JMP $+2
1824
                                  <1>;; OUT DMA+10, AL <1>;; ;JNC shows ;
1825
                                           OUT DMA+10, AL ; INITIALIZE THE DISKETTE CHANNEL ; JNC short NO_BAD ; CHECK FOR ERROR
1826
1827
                                                  short dma bnd err ; 08/02/2015
1828
                                  <1> ;; jc
                                  <1>;; and <1>:: iz
                                                  ecx, OFFF00000h; 16 MB limit
1829
                                            jz
                                  <1> ;;
1830
                                                  short NO_BAD
1831
                                  <1> ;;dma_bnd_err:
                                  <1> ;; MOV byte [DSKETTE_STATUS], DMA_BOUNDARY ; SET ERROR
1832
                                  <1> ;; NO BAD:
1833
1834
                                  <1> ;; RETn
                                                                      ; CY SET BY ABOVE IF ERROR
1835
                                  <1>
1836
                                  <1> ;-----
1837
                                  <1> ; FMTDMA SET
                                  <1>;
1838
                                            THIS ROUTINE SETS UP THE DMA CONTROLLER FOR A FORMAT OPERATION.
                                   <1>;
1839
                                  <1> ; ON ENTRY: NOTHING REQUIRED
1840
1841
1842
                                  <1>; ON EXIT: @DSKETTE STATUS, CY REFLECT STATUS OF OPERATION
1843
                                  <1> ;------
1844
                                  <1> FMTDMA SET:
1845
                                  <1>;; 20/\overline{02}/2015 modification
1846
                                 <1>
                                          mov edx, [ebp+4]
                                            mov edx, [ebp+4] ; Buffer address test edx, 0FFF00000h ; 16 MB limit
1847 00003BE7 8B5504
1848 00003BEA F7C20000F0FF
1849 00003BF0 75EC
                                 <1>
1849 00003BF0 75EC
                                 <1>
                                            jnz short dma bnd err stc
                                 <1>
1850
; SECTORS/TRACK VALUE IN PARM TABLE
                                            call GET PARM
                                                                  : AX = SECTORS/TRACK VALUE

: AX = SECTORS/TRACK VALUE

: AX = SEC/TRY
                                       sub
shl
                                 <1>
1855 00003BFD 28E4
                                                  ah, ah
                           <1> dec <1> mov <1> pop <1> add <1> jc ;
                                                                     ; AX = SEC/TRK * 4 (OFFSET C,H,R,N)
; -1 FOR DMA VALUE
1856 00003BFF 66C1E002
                                                  ax, 2
                                         dec ax
1857 00003C03 6648
1858 00003C05 6689C1
                                                  cx, ax
                                                                      ; *
1859 00003C08 665A
                                                  dx
                                                  dx, cx
1860 00003C0A 6601CA
                                                                       ; check for overflow
1861 00003C0D 72D0
                                                   short dma_bnd_err
                                 <1>
                                            ;
```

```
; DISABLE INTERRUPTS DURING DMA SET-UP
                                                                                 ; OUTPUT highest BITS TO PAGE REGISTER
                                                CLI ; DISABLE INTERRUPTS DURING DMA SET-UP
OUT DMA+12,AL ; SET THE FIRST/LAST F/F
; JMP $+2 ; WAIT FOR I/O
                                         <1> ;;
                                         <1>;; IODELAY
        1895
                                         <1>;; OUT DMA+11,AL ; OUTPUT THE MODE BYTE <1>;; ;MOV AX,ES ; GET THE ES VALUE
        1896
        1897
                                         1898
        1899
        1900
        1901
        1902
                                                                            ; CARRY MEANS HIGH 4 BITS MUST BE INC
        1903
                                         <1> ;; mov
        1904
                                                         eax, [ebp+4]; 08/02/2015
        1905
                                         <1> ;;; J33A:
                                         1906
        1907
                                         <1> ;; ;JMP $+2
        1908
        1909
                                         <1> ;;
                                                  IODELAY
                                         <1>;; MOV AL, AH
        1910
                                         <1>; OUT DMA+4, AL ; OUTPUT HIGH ADDRESS
        1911
                                                shr
                                                        eax, 16 ; 08/02/2015
        1912
                                         <1> ;;
                                                        AL,CH ; GET HIGH 4 BITS
$+2 ; I/O WAIT STATE
        1913
                                         <1> ;;
                                                  ; MOV
        1914
                                         <1> ;;
                                                 ;JMP $+2
        1915
                                         <1> ;;
                                                  IODELAY
                                                  ;AND AL,00001111B
        1916
                                         <1> ;;
        1917
                                         <1> ;;
                                                        081H,AL
                                                                                 ; OUTPUT HIGH 4 BITS TO PAGE REGISTER
        1918
                                         <1> ;;
        1919
                                         <1> ;;;---- DETERMINE COUNT
                                         <1>;; sub eax, eax; 08/02/2015
        1920
                                         <1> ; Sub eax, eax, 00/02/2015
<1> ;; MOV DL, 4 ; SECTORS/TRACK VALUE IN PARM TABLE
<1> ;; CALL GET_PARM ; "
<1> ;; XCHG AL, AH ; AL = SECTORS/TRACK VALUE
<1> ;; SUB AH, AH ; AX = SECTORS/TRACK VALUE
<1> ;; SHL AX, 2 ; AX = SEC/TRK * 4 (OFFSET C,H,R,N)
<1> ;; DEC AX ; -1 FOR DMA VALUE
<1> ;; PUSH eAX ; 08/02/2015 ; SAVE # OF BYTES TO BE TRANSFERED
        1921
        1922
        1923
        1924
        1925
        1926
        1927
                                         <1>;; OUT DMA+5,AL ; LOW BYTE OF COUNT
        1928
                                                  ;JMP $+2
        1929
                                         <1> ;;
                                                                           ; WAIT FOR I/O
        1930
                                         <1> ;;
                                                  IODELAY
                                                  MOV AL, AH
        1931
                                         <1> ;;
                                                                   ; HIGH BYTE OF COUNT
                                         <1> ;;
                                                   OUT
                                                         DMA+5,AL
        1933
                                         <1> ;;
                                                                           ; RE-ENABLE INTERRUPTS
        1934
                                         <1> ;;
                                                         eCX ; 08/02/2015 ; RECOVER COUNT VALUE
                                         <1> ;;
                                                        eAX ; 08/02/2015 ; RECOVER ADDRESS VALUE
        1935
                                                   POP
                                         <1> ;;
                                                   ; ADD AX, CX ; ADD, TEST FOR 64K OVERFLOW
        1936
                                                   add ecx, eax; 08/02/2015
        1937
                                         <1> ;;
                                                  MOV
        1938
                                         <1> ;;
                                                        AL, 2 ; MODE FOR 8237
                                                                           ; WAIT FOR I/O
        1939
                                         <1> ;;
                                                  ;JMP $+2
                                                SIODELAY
                                                  ; INITIALIZE THE DISKETTE CHANNEL

; JNC short FMTDMA_OK

; CHECK FOR THE CHANNEL
        1940
                                         <1> ;;
                                         <1> ;; OUT DMA+10, AL
        1941
                                         <1> ;;
        1942
                                         <1> ;;
                                                         short fmtdma_bnd_err; 08/02/2015
                                                   jс
        1943
        1944
                                         <1> ;; and
                                                        ecx, OFFF00000h ; 16 MB limit
                                         <1>;; jz short FMTDMA_OK <1>;; stc ; 20/02/2015
        1945
        1946
                                         <1> ;;
        1947
                                         <1> ;;fmtdma bnd err:
                                         <1> ;; MOV byte [DSKETTE_STATUS], DMA_BOUNDARY ; SET ERROR
        1948
        1949
                                         <1> ;; FMTDMA OK:
                                         <1>;; RETn
                                                                            ; CY SET BY ABOVE IF ERROR
        1950
        1951
                                         <1>
        1952
                                         <1> ;----
                                         <1> ; NEC_INIT
        1953
        1954
                                         <1> ;
                                                  THIS ROUTINE SEEKS TO THE REQUESTED TRACK AND INITIALIZES
        1955
                                         <1>;
                                                  THE NEC FOR THE READ/WRITE/VERIFY/FORMAT OPERATION.
```

```
1956
                            <1>;
1957
                            <1> ; ON ENTRY: AH = NEC COMMAND TO BE PERFORMED
1958
1959
                            <1> ; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
1960
                            1961
                            <1> NEC INIT:
                           <1> PUSH AX
                                                  ; SAVE NEC COMMAND
; TURN MOTOR ON FOR SPECIFIC DRIVE
1962 00003C4D 6650
1963 00003C4F E8BC020000
                           <1>
                                   CALL MOTOR_ON
1964
                            <1>
1965
                                         DO THE SEEK OPERATION
                           <1> ;----
1966
                           <1>
                                                        ; CH = TRACK #
                                                         ; MOVE TO CORRECT TRACK
                                                        ; RECOVER COMMAND
                                         short ER_1 ; ERROR ON SEEK
eBX, ER_1 ; LOAD ERROR ADDRESS
eBX ; PUSH NEC_OUT ERROR RETURN
1974
                            <1> ;----
                                          SEND OUT THE PARAMETERS TO THE CONTROLLER
                                                        ; OUTPUT THE OPERATION COMMAND
1984
                            <1> ER_1:
1985 00003C7E C3
                            <1>
                                   RETn
1986
                            <1>
1987
                            <1> ;-----
1988
                            <1> ; RWV_COM
1989
                            <1>;
                                    THIS ROUTINE SENDS PARAMETERS TO THE NEC SPECIFIC TO THE
                                    READ/WRITE/VERIFY OPERATIONS.
1990
                            <1>;
1991
                            <1>;
                            <1> ; ON ENTRY: CS:BX = ADDRESS OF MEDIA/DRIVE PARAMETER TABLE
                            <1> ; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
1993
1994
                            <1> ;-------
<1> RWV_COM:
1995
                                                         ; BYTES/SECTOR PARAMETER FROM BLOCK
                            <1> ER_2:
2017
2018 00003CCA C3
                            <1>
                                    RETn
2019
                            <1>
2020
                            2021
                            <1> ; NEC_TERM
2022
                            <1>;
                                    THIS ROUTINE WAITS FOR THE OPERATION THEN ACCEPTS THE STATUS
                            <1>;
                                    FROM THE NEC FOR THE READ/WRITE/VERIFY/FORWAT OPERATION.
2023
2024
2025
                            <1> ; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
2026
                            <1> ;-----
2027
                            <1> NEC_TERM:
2028
                            <1>
2029
                            <1> ;----
                                         LET THE OPERATION HAPPEN
2030
                            <1>
                           2031 00003CCB 56
                                                         ; SAVE HEAD #, # OF SECTORS
2032 00003CCC E80D040000
2033 00003CD1 9C
2034 00003CD2 E837040000
                                                              ; GET THE NEC STATUS
2035 00003CD7 724B
                                POPF
2036 00003CD9 9D
                            <1>
2037 00003CDA 723E
                            <1>
                                    JC
                                          short SET_END
                                                      ; LOOK FOR ERROR
                            <1>
2038
2039
                                          CHECK THE RESULTS RETURNED BY THE CONTROLLER
                            <1> ;----
2040
                            <1>
                                                         ; SET THE CORRECT DIRECTION
2041 00003CDC FC
                           <1>
                                    CLD
2041 00003CDC FC
2042 00003CDD BE[49590100]
2043 00003CE2 AC
2044 00003CE3 24C0
2045 00003CE5 7433
                                  MOV eSI, NEC_STATUS
                           <1>
                                                              ; POINT TO STATUS FIELD
                          <1> lodsb <1> AND <1> JZ <1> CMP
                                                         ; GET STO
                                                        ; TEST FOR NORMAL TERMINATION
                                   AND AL,11000000B
                                         short SET END
2046 00003CE7 3C40
                                         AL,01000000B ; TEST FOR ABNORMAL TERMINATION
                           <1>
2047 00003CE9 7527
                                    JNZ
                                         short J18
                                                         ; NOT ABNORMAL, BAD NEC
                           <1>
2048
2049
                           <1> ;----
                                          ABNORMAL TERMINATION, FIND OUT WHY
2050
                           <1>
                      <1><1><1><1></1></1>
2051 00003CEB AC
                                    lodsb
                                                          ; GET ST1
2052 00003CEC D0E0
                                    SAL AL,1
                                                          ; TEST FOR EDT FOUND
2053 00003CEE B404
                                    MOV
                                         AH, RECORD_NOT_FND
2054 00003CF0 7222
                           <1>
                                    JC
                                          short J19
                           <1>
2055 00003CF2 C0E002
                                    SAL
                                         AL,2
                                  MOV
                                         AH, BAD CRC
2056 00003CF5 B410
                           <1>
                           <1>
<1>
2057 00003CF7 721B
                                    JC
                                         short J19
2058 00003CF9 D0E0
                                    SAL
                                         AL,1
                                                          ; TEST FOR DMA OVERRUN
2059 00003CFB B408
                           <1>
                                    MOV
                                         AH, BAD DMA
2060 00003CFD 7215
                           <1>
                                    JC
                                         short J19
```

```
<1>
                                                                     ; TEST FOR RECORD NOT FOUND
2061 00003CFF C0E002
                                           SAL
                                                 AL, 2
2062 00003D02 B404
                                 <1>
                                           VOM
                                                 AH, RECORD NOT FND
2063 00003D04 720E
                                           JC
                                <1>
                                                 short J19
                                <1>
<1>
<1>
<1>
2064 00003D06 D0E0
                                           SAL
                                                 AL,1
2065 00003D08 B403
                                          VOM
                                                 AH, WRITE_PROTECT ; TEST FOR WRITE_PROTECT
2066 00003D0A 7208
                                           JC
                                                  short J19
                                <1>
2067 00003D0C D0E0
                                                                     ; TEST MISSING ADDRESS MARK
                                           SAL
                                                 AL,1
                                                 AH, BAD_ADDR MARK
2068 00003D0E B402
                                <1>
                                           MOV
2069 00003D10 7202
                                 <1>
                                           JC
                                                 short J19
2070
                                 <1>
2071
                                 <1> ;----
                                                 NEC MUST HAVE FAILED
2072
                                 <1> J18:
2073 00003D12 B420
                                 <1>
                                           MOV
                                                 AH, BAD_NEC
2074
                                 <1> J19:
2075 00003D14 0825[48590100]
                                 <1>
                                          OR
                                                 [DSKETTE_STATUS], AH
                                 <1> SET END:
2076
2077 00003D1A 803D[48590100]01 <1> CMP
                                                 byte [DSKETTE_STATUS], 1 ; SET ERROR CONDITION
2078 00003D21 F5
                                           CMC
                                 <1>
2079 00003D22 5E
                                 <1>
                                           POP
2080 00003D23 C3
                                 <1>
                                           RETn
                                                                     ; RESTORE HEAD #, # OF SECTORS
2081
                                 <1>
2082
                                 <1> SET END POP:
2083 00003D24 9D
                                 <1> POPF
                                           JMP SHORT SET_END
2084 00003D25 EBF3
                                 <1>
2085
                                 <1>
2086
                                 <1> ;------
2087
                                 <1> ; DSTATE: ESTABLISH STATE UPON SUCCESSFUL OPERATION.
2088
                                 <1> ;-----
2089
                                 <1> DSTATE:
2090 00003D27 803D[48590100]00
                                 <1> CMP
                                                 byte [DSKETTE_STATUS], 0 ; CHECK FOR ERROR

2091 00003D2E 753E
                                                 byte [DSK_STATE+eDI], MED_DET; NO ERROR, MARK MEDIA AS DETERMINED
2092 00003D30 808F[55590100]10
2093 00003D37 F687[55590100]04
2094 00003D3E 752E
2095 00003D40 8A87[55590100]
2096 00003D46 24C0
2097 00003D48 3C80
2098 00003D4A 751B
2099
                                 <1>
                                                 CHECK IF IT IS 1.44M
2100
                                 <1> ;----
                                 <1>
2101
                               2102 00003D4C E871010000
2103
2104
2105 00003D51 7414
2106 00003D53 3C04
2107 00003D55 7410
                                 <1> M_720:
2108
                                      AND
2109 00003D57 80A7[55590100]FD
                                <1>
                                                 byte [DSK_STATE+eDI], ~FMT_CAPA ; TURN OFF FORMAT CAPABILITY
                                                 byte [DSK_STATE+eDI], DRV_DET ; MARK DRIVE DETERMINED
2110 00003D5E 808F[55590100]04
                               <1>
                                           OR
                                 <1>
2111 00003D65 EB07
                                           JMP
                                                 SHORT SETBAC
                                                                  ; BACK
2112
                                 <1> M 12:
                                                 byte [DSK_STATE+eDI],DRV_DET+FMT_CAPA
2113 00003D67 808F[55590100]06
                                 <1> OR
                                                                   ; TURN ON DETERMINED & FMT CAPA
2114
                                 <1>
2115
                                 <1> SETBAC:
2116 00003D6E C3
                                 <1>
                                           RETn
2117
                                 <1>
2118
                                 2119
                                  <1> ; RETRY
2120
                                           DETERMINES WHETHER A RETRY IS NECESSARY.
                                 <1>;
2121
                                 <1>;
                                           IF RETRY IS REQUIRED THEN STATE INFORMATION IS UPDATED FOR RETRY.
2122
                                  <1>;
2123
                                 <1>; ON EXIT: CY = 1 FOR RETRY, CY = 0 FOR NO RETRY
                                  <1> ;-----
2124
2125
                                 <1> RETRY:
                                                 byte [DSKETTE_STATUS],0 ; GET STATUS OF OPERATION short NO_RETRY ; SUCCESSFUL OPERATION
2126 00003D6F 803D[48590100]00
                                 <1> CMP
2127 00003D76 7445
                                 <1>
                                          JZ
                                byte [DSKETTE_STATUS], TIME_OUT; IF TIME OUT NO RETRY
2128 00003D78 803D[48590100]80
2129 00003D7F 743C
2130 00003D81 8AA7[55590100]
                                           MOV AH, [DSK_STATE+eDI] ; GET MEDIA STATE OF DRIVE
                                                                 ; ESTABLISHED/DETERMINED ?
2131 00003D87 F6C410
                                                 short NO_RETRY ; IF ESTABLISHED STATE THEN TRUE ERROR
AH,RATE_MSK ; ISOLATE RATE
CH,[LASTRATE] ; GET START OPERATION STATE
2132 00003D8A 7531
2133 00003D8C 80E4C0
                                                 CH,4 ; TO CORRESPONDING BITS
CH,RATE_MSK ; ISOLATE RATE BITS
CH,AH ; ALL RATES TRIED
short NO_RETRY ; IF YES THEN
2134 00003D8F 8A2D[50590100] <1>
                                           MOV CH, [LASTRATE]
2135 00003D95 C0C504
                                 <1>
                                           ROL
2136 00003D98 80E5C0
                                 <1>
                                           AND
2137 00003D9B 38E5
                                 <1>
                                           CMP
2138 00003D9D 741E
                                                                          ; IF YES, THEN TRUE ERROR
                                 <1>
                                           JE
2139
                                 <1>
2140
                                           SETUP STATE INDICATOR FOR RETRY ATTEMPT TO NEXT RATE
                                 <1>;
                                 <1> ;
                                           00000000B (500) -> 10000000B (250)
2141
                                            10000000B (250) -> 01000000B 01000000B (300) -> 00000000B
2142
                                  <1>;
                                                                            (300)
2143
                                 <1>;
                                                                            (500)
2144
                                 <1>
                                                 AH, RATE_500+1
                                                                   ; SET CY FOR RATE 500
2145 00003D9F 80FC01
                                 <1>
                                           CMP
                                                 AH,1 ; TO NEXT STATE
AH,RATE_MSK ; KEEP ONLY RATE BITS
2146 00003DA2 D0DC
                                 <1>
                                           RCR
2147 00003DA4 80E4C0
                                 <1>
                                                 byte [DSK_STATE+eDI], ~(RATE_MSK+DBL_STEP)
2148 00003DA7 80A7[55590100]1F
                                <1>
                                           AND
2149
                                 <1>
                                                                ; RATE, DBL STEP OFF
                                                  [DSK STATE+eDI], AH ; TURN ON NEW RATE
2150 00003DAE 08A7[55590100]
                                 <1>
2151 00003DB4 C605[48590100]00
                                           VOM
                                                 byte [DSKETTE_STATUS],0 ; RESET STATUS FOR RETRY
                                 <1>
2152 00003DBB F9
                                  <1>
                                           STC
                                                                    ; SET CARRY FOR RETRY
2153 00003DBC C3
                                 <1>
                                           RETn
                                                                     ; RETRY RETURN
2154
                                 <1>
                                 <1> NO RETRY:
2155
                                                                     ; CLEAR CARRY NO RETRY
2156 00003DBD F8
                                 <1>
                                           CLC
2157 00003DBE C3
                                                                     ; NO RETRY RETURN
                                 <1>
2158
                                 <1>
2159
                                  <1> ;-----
2160
                                 <1> ; NUM TRANS
                                           THIS ROUTINE CALCULATES THE NUMBER OF SECTORS THAT WERE
2161
                                 <1>;
2162
                                  <1>;
                                           ACTUALLY TRANSFERRED TO/FROM THE DISKETTE.
2163
                                 <1>;
2164
                                  <1>; ON ENTRY: [BP+1] = TRACK
2165
                                 <1>; SI-HI = HEAD
```

```
<1>;
2166
                                          [BP] = START SECTOR
2167
2168
                             <1> ; ON EXIT: AL = NUMBER ACTUALLY TRANSFERRED
2169
                             <1> ;------
2170
                             <1> NUM TRANS:
2171 00003DBF 30C0
                             <1> XOR
                                                           ; CLEAR FOR ERROR
2172 00003DC1 803D[48590100]00
                                         byte [DSKETTE_STATUS], 0 ; CHECK FOR ERROR
                            <1>
                                          DIF_HD ; IF ON SAME HEAD, THEN NO ADJUST
                                         CH, [NEC_STATUS+3] , GET TRACK ENDED OF ON

CH, [eBP+1] ; IS IT ASKED FOR TRACK

short SAME_TRK ; IF SAME TRACK NO INCREASE

BL,AH ; ADD SECTORS/TRACK
2184
                            <1> DIF_HD:
2185 00003DEF 00E3
                            <1> ADD
                                                          ; ADD SECTORS/TRACK
                                          BL,AH
                            <1> SAME_TRK:
2186
2187 00003DF1 2A5D00
                            <1>
<1>
                                          BL,[eBP]
                                     SUB
                                                          ; SUBTRACT START FROM END
2188 00003DF4 88D8
                                     VOM
                                                           ; TO AL
                                          AL,BL
2189
                            <1> NT_OUT:
2190 00003DF6 C3
                            <1>
                                   RETn
2191
                            <1>
                            <1> ;------
2192
                             <1> ; SETUP_END
2193
2194
                                     RESTORES @MOTOR_COUNT TO PARAMETER PROVIDED IN TABLE
                             <1>;
                                     AND LOADS @DSKETTE STATUS TO AH, AND SETS CY.
2195
                             <1>;
2196
                             <1>;
2197
                             <1> ; ON EXIT:
                             <1> ; AH, @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
2198
2199
                             2200
                            <1> SETUP_END:
                                                      ; GET THE MOTOR WAIT PARAMETER
                            <1> MOV DL,2
<1> PUSH AX
<1> CALL GET_PARM
2201 00003DF7 B202
2202 00003DF9 6650
                                                          ; SAVE NUMBER TRANSFERRED
2203 00003DFB E8CB000000
                          2204 00003E00 8825[47590100]
2205 00003E06 6658
2205 00003E06 6658
2206 00003E08 8A25[48590100]
2207 00003E0E 08E4
2208 00003E10 7402
2209 00003E12 30C0
2210
                            <1> NUN ERR:
2211 00003E14 80FC01
                            <1> CMP
                                          AH,1
                                                          ; SET THE CARRY FLAG TO INDICATE
2212 00003E17 F5
                            <1>
                                                           ; SUCCESS OR FAILURE
2213 00003E18 C3
                            <1>
                                    RETn
2214
                            <1>
2215
                            <1> ;-----
                             <1> ; SETUP_DBL
2216
2217
                             <1>; CHECK DOUBLE STEP.
2218
                             <1>;
2219
                             <1> ; ON ENTRY : DI = DRIVE
2220
                             <1>;
2221
                             <1>; ON EXIT : CY = 1 MEANS ERROR
                             <1> ;-----
                            <1> SETUP DBL:
2223
2224 00003E19 8AA7[55590100]
                            <1> MOV
                                          AH, [DSK_STATE+eDI]; ACCESS STATE
                                    TEST AH, MED DET ; ESTABLISHED STATE ?
2225 00003E1F F6C410
                            <1>
2226 00003E22 757E
                            <1> JNZ short NO_DBL
                                                           ; IF ESTABLISHED THEN DOUBLE DONE
2227
                            <1>
                                          CHECK FOR TRACK 0 TO SPEED UP ACKNOWLEDGE OF UNFORMATTED DISKETTE
                            <1> ;----
2228
                            <1>
2229
                           ; SET RECALIBRATE REQUIRED ON ALL DRIVES
2230 00003E24 C605[45590100]00
2231 00003E2B E8E0000000
                                    CALL MOTOR_ON ; ENSURE MOTOR STAY ON MOV CH,0 ; LOAD TRACK 0
2232 00003E30 B500
2233 00003E32 E8D4010000
2234 00003E37 E868000000
                                                                ; READ ID FUNCTION
2235 00003E3C 7249
2236
                            <1>
                                          INITIALIZE START AND MAX TRACKS (TIMES 2 FOR BOTH HEADS)
2237
                            <1> ;----
2238
                            <1>
                            <1> MOV CX,0450H
2239 00003E3E 66B95004
                                                          ; START, MAX TRACKS
2240 00003E42 F687[55590100]01
                                    TEST byte [DSK_STATE+eDI], TRK_CAPA; TEST FOR 80 TRACK CAPABILITY
                            <1>
2241 00003E49 7402
                                    JZ short CNT_OK ; IF NOT COUNT IS SETUP
                            <1>
2242 00003E4B B1A0
                            <1>
                                   MOV CL, 0A0H
                                                               ; MAXIMUM TRACK 1.2 MB
2243
                            <1>
                                     ATTEMPT READ ID OF ALL TRACKS, ALL HEADS UNTIL SUCCESS; UPON SUCCESS,
2244
                             <1>;
                                     MUST SEE IF ASKED FOR TRACK IN SINGLE STEP MODE = TRACK ID READ; IF NOT
2245
                             <1>;
                             <1> ;
2246
                                     THEN SET DOUBLE STEP ON.
2247
                             <1>
                            <1> CNT_OK:
2249 00003E4D C605[47590100]FF
                                      MOV
                                             byte [MOTOR_COUNT], OFFH ; ENSURE MOTOR STAYS ON FOR OPERATION
                            <1>
2250 00003E54 6651
                            <1>
                                     PUSH CX
                                                         ; SAVE TRACK, COUNT
2251 00003E56 C605[48590100]00
                                          byte [DSKETTE_STATUS], 0 ; CLEAR STATUS, EXPECT ERRORS
                            <1>
                                    VOM
                                                      ; RESTORE COUNT
; IF OK, ASKED = RETURNED TRACK ?
; INC FOR NEXT TRACK
                       CH CT.
                                                      ; REACHED MAXIMUM YET ; CONTINUE TILL ALL TRIED
                                   CMP
2266 00003E83 38CD
                           <1>
                                          CH,CL
2267 00003E85 75C6
                            <1>
                                    JNZ
                                          short CNT_OK
                            <1>
2268
2269
                            <1> ;----
                                          FALL THRU, READ ID FAILED FOR ALL TRACKS
2270
                            <1>
```

```
2271
                             <1> SD_ERR:
                             <1>
2272 00003E87 F9
                                                           ; SET CARRY FOR ERROR
                                                            ; SETUP_DBL ERROR EXIT
2273 00003E88 C3
                             <1>
                                     RETn
2274
                             <1>
2275
                             <1> DO CHK:
byte [DSK_STATE+eDI], DBL_STEP; TURN ON DOUBLE STEP REQUIRED
                            <1> NO DBL:
                            <1> _ CLC
2283 00003EA2 F8
                                                           ; CLEAR ERROR FLAG
2284 00003EA3 C3
                             <1>
                                     RETn
2285
                             <1>
                             <1> ;------
2286
                             <1> ; READ ID
2287
                                     READ ID FUNCTION.
2288
                             <1>;
2289
                             <1>;
                             <1>; ON ENTRY: DI : BIT 2 = HEAD; BITS 1,0 = DRIVE
2290
2291
                             <1>;
2292
                             <1>; ON EXIT: DI : BIT 2 IS RESET, BITS 1,0 = DRIVE
2293
                             <1>; @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
2294
                             <1> ;------
2295
                             <1> READ ID:
                            <1> MOV 
<1> PUSH
2296 00003EA4 B8[C13E0000]
                                          eAX, ER_3
                                                           ; MOVE NEC OUTPUT ERROR ADDRESS
; WAIT FOR OPERATION, GET STATUS
2305
                             <1> ER_3:
2306 00003EC1 C3
                             <1>
                                    RETn
2307
                             <1>
2308
                             <1> ;-----
                             <1>; CMOS TYPE
2309
2310
                             <1>; RETURNS DISKETTE TYPE FROM CMOS
2311
                             <1>;
                             <1> ; ON ENTRY: DI = DRIVE #
2312
                             <1>;
2313
2314
                             <1> ; ON EXIT: AL = TYPE; CY REFLECTS STATUS
2315
                             <1> ;------
2316
                             <1>
                             <1> CMOS TYPE: ; 11/12/2014
                             2318 00003EC2 8A87[565D0000]
2319 00003EC8 20C0
2320 00003ECA C3
                             <1> retn
2321
                             <1>
2322
                             <1> ; CMOS TYPE:
                             2323
2324
                                     TEST AL, BAD_BAT+BAD_CKSUM ; BATTERY GOOD AND CHECKSUM VALID STC ; SET CY = 1 INDICATING ERROR FOR RETURN JNZ short BAD_CM ; ERROR IF EITHER BIT ON
2325
                             <1>;
2326
                             <1>;
2327
                             <1>;
                                     MOV AL, CMOS_DISKETTE ; ADDRESS OF DISKETTE BYTE IN CMOS
2328
                             <1>;
                                     CALL CMOS_READ ; GET DISKETTE BYTE
OR DI,DI ; SEE WHICH DRIVE IN QUESTION
2329
                             <1>;
                                     OR DI,DĪ
2330
                             <1>;
                                     JNZ short TB ; IF DRIVE 1, DATA IN LOW NIBBLE ROR AL,4 ; EXCHANGE NIBBLES IF SECOND DRIVE.
2331
                             <1>;
                                                          ; EXCHANGE NIBBLES IF SECOND DRIVE
2332
                             <1>;
                                    ROR AL,4
2333
                             <1> ;TB:
                                                         ; KEEP ONLY DRIVE DATA, RESET CY, 0
2334
                             <1> ; AND
                                          AL,OFH
2335
                             <1> ;BAD CM:
2336
                             <1>;
                                    RETn
                                                           ; CY, STATUS OF READ
2337
                             <1>
                             2338
2339
                             <1> ; GET PARM
2340
                                     THIS ROUTINE FETCHES THE INDEXED POINTER FROM THE DISK BASE
                             <1>;
2341
                             <1>;
                                     BLOCK POINTED TO BY THE DATA VARIABLE @DISK POINTER. A BYTE FROM
                                     THAT TABLE IS THEN MOVED INTO AH, THE INDEX OF THAT BYTE BEING
2342
                             <1>;
2343
                             <1>;
                                     THE PARAMETER IN DL.
                             <1>;
2344
2345
                             <1> ; ON ENTRY: DL = INDEX OF BYTE TO BE FETCHED
2346
                             <1>;
2347
                             <1>; ON EXIT: AH = THAT BYTE FROM BLOCK
2348
                             <1>; AL, DH DESTROYED
2349
                             <1> ;------
2350
                             <1> GET PARM:
2351
                             <1> ; PUSH DS
2352 00003ECB 56
                             <1>
                                     PUSH
                                           eSI
                                                           ; DS = 0, BIOS DATA AREA
2353
                             <1>
                                     ; SUB AX, AX
2354
                             <1>
                                     ; MOV DS, AX
2355
                             <1>
                                     ;; mov ax, cs
2356
                             <1>
                                     ;;mov ds, ax
2357
                             <1>
                                     ; 08/02/2015 (protected mode modifications, bx -> ebx)
                                                   ; BL = INDEX
2358 00003ECC 87D3
                            <1>
                                     XCHG eDX, eBX
                                     ; SUB BH, BH
2359
                             <1>
                                                           ; BX = INDEX
                                     and ebx, 0FFh
2360 00003ECE 81E3FF000000
                            <1>
                                     ;LDS SI, [DISK_POINTER] ; POINT TO BLOCK
2361
                             <1>
2362
                             <1>
                                     ;
                                     ; 17/12/2014
2363
                            <1>
2364 00003ED4 66A1[455D0000]
                            <1>
                                     mov ax, [cfd]; current (AL) and previous fd (AH)
2365 00003EDA 38E0
                             <1>
                                     cmp
                                          al, ah
2366 00003EDC 7425
                            <1>
                                     jе
                                           short gpndc
                                     mov
                                          [pfd], al ; current drive -> previous drive
2367 00003EDE A2[465D0000]
                            <1>
2368 00003EE3 53
                            <1>
                                     push ebx; 08/02/2015
2369 00003EE4 88C3
                            <1>
                                     mov
                                           bl, al
2370
                            <1>
                                     : 11/12/2014
                                     mov al, [eBX+fd0_type] ; Drive type (0,1,2,3,4)
2371 00003EE6 8A83[565D0000]
                            <1>
2372
                            <1>
                                     ; 18/12/2014
2373 00003EEC 20C0
                            <1>
                                     and al, al
                            <1>
                                          short gpdtc
2374 00003EEE 7507
                                     jnz
2375 00003EF0 BB[2F5D0000]
                            <1>
                                     mov ebx, MD_TBL6
                                                         ; 1.44 MB param. tbl. (default)
```

```
2376 00003EF5 EB05
                                 <1>
                                           jmp
                                                  short gpdpu
                                 <1> gpdtc:
                                <1> call DR_TYPE_CHECK
2378 00003EF7 E817F9FFFF
2379
                                <1>
                                          ; cf = 1 \rightarrow \text{eBX} points to 1.44\text{MB} fd parameter table (default)
2380
                                 <1> gpdpu:
                                                [DISK POINTER], ebx
2381 00003EFC 891D[CC5C0000]
                                <1> mov
2382 00003F02 5B
                                <1>
                                          pop ebx
                                <1> gpndc:
2383
                                <1> mov
<1> MOV
<1> XCHG
<1> POP
<1> ;POP
2384 00003F03 8B35[CC5C0000]
                                                esi, [DISK_POINTER] ; 08/02/2015, si -> esi
2385 00003F09 8A241E
                                          MOV AH, [eSI+eBX] ; GET THE WORD XCHG eDX,eBX ; RESTORE BX
                                          XCHG eDX, eBX
2386 00003F0C 87D3
2387 00003F0E 5E
                                                eSI
                                          ; POP DS
2388
2389 00003F0F C3
                                <1>
                                        RETn
2390
                                 <1>
2391
                                 <1> ;-----
                                 <1> ; MOTOR ON
2392
2393
                                 <1>;
                                          TURN MOTOR ON AND WAIT FOR MOTOR START UP TIME. THE @MOTOR COUNT
2394
                                           IS REPLACED WITH A SUFFICIENTLY HIGH NUMBER (0FFH) TO ENSURE
                                 <1>;
2395
                                 <1>;
                                          THAT THE MOTOR DOES NOT GO OFF DURING THE OPERATION. IF THE
                                        MOTOR NEEDED TO BE TURNED ON, THE MULTI-TASKING HOOK FUNCTION
2396
                                 <1>;
                                          (AX=90FDH, INT 15) IS CALLED TELLING THE OPERATING SYSTEM
2397
                                 <1>;
                                 <1>;
                                          THAT THE BIOS IS ABOUT TO WAIT FOR MOTOR START UP. IF THIS
2398
2399
                                 <1>;
                                          FUNCTION RETURNS WITH CY = 1, IT MEANS THAT THE MINIMUM WAIT
2400
                                          HAS BEEN COMPLETED. AT THIS POINT A CHECK IS MADE TO ENSURE
                                 <1>;
2401
                                 <1>;
                                          THAT THE MOTOR WASN'T TURNED OFF BY THE TIMER. IF THE HOOK DID
2402
                                 <1>;
                                          NOT WAIT, THE WAIT FUNCTION (AH=086H) IS CALLED TO WAIT THE
                                        PRESCRIBED AMOUNT OF TIME. IF THE CARRY FLAG IS SET ON RETURN,
2403
                                 <1>;
2404
                                 <1>;
                                          IT MEANS THAT THE FUNCTION IS IN USE AND DID NOT PERFORM THE
                                 <1>;
                                          WAIT. A TIMER 1 WAIT LOOP WILL THEN DO THE WAIT.
2405
2406
                                 <1>;
2407
                                 <1> ; ON ENTRY: DI = DRIVE #
                                 <1>; ON EXIT: AX,CX,DX DESTROYED
2408
2409
                                 <1> ;------
2410
                                 <1> MOTOR_ON:
                                2411 00003F10 53
2412 00003F11 E82A000000
2413 00003F16 7226
2414 00003F18 E89BF9FFFF
2415 00003F1D E865F9FFFF
2416
2417
                                <1> M_WAIT:
2418
                                <1> M_WAIT:
<1> MOV DL,10
<1> CALL GET_PARM
<1> ;MOV AL,AH
<1> ;XOR AH,AH
<1> ;CMP AL,8
<1> cmp ah, 8
<1> ;JAE short GP2
<1> ja short J13
<1> ;MOV AL,8
<1> mov ah, 8
2419 00003F22 B20A
                                                                   ; GET THE MOTOR WAIT PARAMETER
2420 00003F24 E8A2FFFFF
2421
                                                                    ; AL = MOTOR WAIT PARAMETER
                                                                    ; AX = MOTOR WAIT PARAMETER
2422
                                                                    ; SEE IF AT LEAST A SECOND IS SPECIFIED
2423
2424 00003F29 80FC08
                                                                    ; IF YES, CONTINUE
2425
2426 00003F2C 7702
2427
                                                                    ; ONE SECOND WAIT FOR MOTOR START UP
2428 00003F2E B408
                                 <1>
                                          mov
                                                ah, 8
2429
                                 <1>
2430
                                 <1> ;----
                                                 AS CONTAINS NUMBER OF 1/8 SECONDS (125000 MICROSECONDS) TO WAIT
2431
                                 <1> GP2:
                                 <1> ;----
                                                 FOLLOWING LOOPS REQUIRED WHEN RTC WAIT FUNCTION IS ALREADY IN USE
                                                             ; WAIT FOR 1/8 SECOND PER (AL)
                                 <1> J13:
2433
                                <1> MOV eCX,8286
<1> CALL WAITF
<1> ;DEC AL
<1> dec ah
<1> JNZ short J13
2434 00003F30 B95E200000
                                                                    ; COUNT FOR 1/8 SECOND AT 15.085737 US
2435 00003F35 E8BCDEFFFF
                                                                    ; GO TO FIXED WAIT ROUTINE
2436
                                                                   ; DECREMENT TIME VALUE
2437 00003F3A FECC
                                                short J13 ; ARE WE DONE YET
2438 00003F3C 75F2
2439
                                <1> MOT_IS_ON:
                                 <1>
2440 00003F3E 5B
                                          POP
                                                                    ; RESTORE REG.
                                                eBX
2441 00003F3F C3
                                 <1>
                                           RETn
2442
                                 <1>
2443
                                 <1> ;------
2444
                                 <1> ; TURN ON
2445
                                 <1>;
                                          TURN MOTOR ON AND RETURN WAIT STATE.
2446
                                 <1>;
                                 <1>; ON ENTRY: DI = DRIVE #
2447
2448
                                 <1>;
                                 <1>; ON EXIT: CY = 0 MEANS WAIT REQUIRED
2449
                                 2450
2451
                                 <1>;
                                                AX, BX, CX, DX DESTROYED
                                 <1> ;-----
2452
                                 <1> TURN ON:
2453
2454 00003F40 89FB
                                 <1> MOV
                                                eBX,eDI
                                                                           ; BX = DRIVE #
                                          MOV CL,BL ; CL = DRIVE #
ROL BL,4 ; BL = DRIVE SELECT
. NO INTERRUPTS WHIL
                                                                 ; CL = DRIVE #
2455 00003F42 88D9
                                 <1>
                                 <1> ROL <1> CLI
2456 00003F44 C0C304
                                                 ; NO INTERRUPTS WHILE DETERMINING STATUS byte [MOTOR_COUNT], OFFH ; ENSURE MOTOR STAYS ON FOR OPERATION
2457 00003F47 FA
2458 00003F48 C605[47590100]FF
                              <1>
                                          VOM
2459 00003F4F A0[46590100]
                                <1>
                                           VOM
                                                 AL, [MOTOR STATUS] ; GET DIGITAL OUTPUT REGISTER REFLECTION
2460 00003F54 2430
                                 <1>
                                          AND
                                                 AL,00110000B ; KEEP ONLY DRIVE SELECT BITS
2461 00003F56 B401
                                 <1>
                                                                    ; MASK FOR DETERMINING MOTOR BIT
                                          VOM
                                                 AH,1
2462 00003F58 D2E4
                                <1>
                                         SHL
                                                 AH,CL
                                                                    ; AH = MOTOR ON, A=00000001, B=00000010
2463
                                 <1>
                                 <1> ; AL = DRIVE SELECT FROM @MOTOR STATUS
2464
2465
                                 <1> ; BL = DRIVE SELECT DESIRED
                                 <1> ; AH = MOTOR ON MASK DESIRED
2466
2467
                                 <1>
                                          CMP AL,BL ; REQUESTED DRIVE ALREADY SELECTED ?
JNZ short TURN_IT_ON ; IF NOT SELECTED JUMP
2468 00003F5A 38D8
                                 <1>
2469 00003F5C 7508
                                 <1>
                                                AH, [MOTOR_STATUS] ; TEST MOTOR ON BIT short NO_MOT_WAIT ; JUMP IF MOTOR ON AND SELECTED
2470 00003F5E 8425[46590100]
                                 <1>
                                          TEST
2471 00003F64 7535
                                 <1>
                                          JNZ
                                 <1>
                                 <1> TURN_IT_ON:
2473
                                 <1>
                                                AH, BL
                                                                    ; AH = DRIVE SELECT AND MOTOR ON
2474 00003F66 08DC
                                          OR
2475 00003F68 8A3D[46590100]
                                                 BH, [MOTOR STATUS] ; SAVE COPY OF @MOTOR STATUS BEFORE
                                          VOM
                                <1>
                                <1> AND <1> AND <1> OR <1> MOV <1> MOV
                                          AND
2476 00003F6E 80E70F
                                                 BH,00001111B ; KEEP ONLY MOTOR BITS
2477 00003F71 8025[46590100]CF
                                                 byte [MOTOR STATUS], 11001111B; CLEAR OUT DRIVE SELECT
                                                 [MOTOR_STATUS],AH ; OR IN DRIVE SELECTED AND MOTOR ON
2478 00003F78 0825[46590100]
2479 00003F7E A0[46590100]
                                                 AL, [MOTOR_STATUS] ; GET DIGITAL OUTPUT REGISTER REFLECTION
2480 00003F83 88C3
                                                           ; BL=@MOTOR STATUS AFTER, BH=BEFORE
                                                 BL,AL
```

```
; KEEP ONLY MOTOR BITS
                                                               STI ; ENABLE INTERRUPTS AGAIN
AND AL,00111111B ; STRIP AWAY UNWANTED BITS
ROL AL,4 ; PUT BITS IN DESIRED POSITIONS
                                                                        AL,00001100B ; NO RESET, ENABLE DMA/INTERRUPT
DX.03F2H : SELECT DRIVE AND TURN ON MOTOR
                                                                                                     ; SELECT DRIVE AND TURN ON MOTOR
                                                                        BL,BH ; NEW MOTOR TURNED ON ? short NO_MOT_WAIT ; NO WAIT REQUIRED IF JUST SELECT
                                                                         short no_mot_w1 ; 27/02/2015
                                                                                                      ; (re) SET CARRY MEANING WAIT
                                                 <1>
 2493
                                                 <1> NO_MOT_WAIT: <1> sti
 2494
 2495 00003F9B FB
                                                 <1> no_mot_w1: ; 27/02/2015
 2496
 2497 00003F9C F9
                                                                                                      ; SET NO WAIT REQUIRED
                                                 <1> STC
                                                                                                      ; INTERRUPTS BACK ON
                                                               ;STI
 2498
                                                 <1>
 2499 00003F9D C3
                                                 <1>
                                                                RETn
 2500
                                                 <1>
 2501
                                                  <1> ;------
 2502
                                                  <1> ; HD WAIT
 2503
                                                  <1>; WAIT FOR HEAD SETTLE TIME.
 2504
                                                  <1>;
 2505
                                                  <1>; ON ENTRY: DI = DRIVE #
 2506
                                                  <1>;
 2507
                                                  <1>; ON EXIT: AX, BX, CX, DX DESTROYED
                                                  <1> ;-----
 2508
2509
                                                 <1> HD_WAIT:
 2524
                                                 <1> ;GP3:
                                                                         AH, HD320_SETTLE
                                                                                                            ; USE 320/360 HEAD SETTLE
 2525 00003FC0 B414
                                                 <1> MOV
                                                            JMP
 2526
                                                 <1>;
                                                                         SHORT DO WAT
 2527
                                                  <1>
                                                  <1> ;ISNT WRITE:
 2528
                                                               OR AH,AH ; CHECK FOR NO WAIT

JZ short HW_DONE ; IF NOT WRITE AND 0 ITS OK
                                                 <1> ;
 2529
 2530
                                                  <1>;
 2531
                                                  <1>
 2532
                                                  <1> ;----
                                                                         AH CONTAINS NUMBER OF MILLISECONDS TO WAIT
                                                  <1> DO_WAT:
 2533
                                                 <1>; MOV AL, AH ; AL = # MILLISECONDS 
<1>; ; XOR AH, AH ; AX = # MILLISECONDS 
<1> J29: ; 1 MILLISECOND LOOP
 2534
 2535
 2536

// call waitf
// call waitf
// call dec ah

/ mov cx, wait_fdu_Head_Settle; 33; 1 ms in 30 micro units.
// call waitf ; count at 15.085737 us per count
// call waitf ; delay for 1 millisecond
// dec ah
// dec ah
// call indicate the count
// dec ah
// call indicate the count
// call indicate the call indicate the count
// call indicate the call i
 2538 00003FC2 B942000000
 2539 00003FC7 E82ADEFFFF
 2540
                                                <1> dec ah <1> JNZ short
 2541 00003FCC FECC
                                                                         short J29 ; DO AL MILLISECOND # OF TIMES
 2542 00003FCE 75F2
 2543
                                                 <1> HW_DONE:
 2544 00003FD0 C3
                                                  <1>
 2545
                                                  <1>
 2546
                                                  <1> ;------
 2547
                                                  <1> ; NEC OUTPUT
                                                  <1> ; This routine sends a byte to the nec controller after testing
 2548
 2549
                                                  <1>;
                                                                FOR CORRECT DIRECTION AND CONTROLLER READY THIS ROUTINE WILL
 2550
                                                  <1>;
                                                             TIME OUT IF THE BYTE IS NOT ACCEPTED WITHIN A REASONABLE AMOUNT
 2551
                                                  <1>;
                                                               OF TIME, SETTING THE DISKETTE STATUS ON COMPLETION.
 2552
                                                  <1>;
                                                  <1> ; ON ENTRY: AH = BYTE TO BE OUTPUT
 2553
 2554
 2555
                                                  <1>; ON EXIT: CY = 0 SUCCESS
                                                                         CY = 1 FAILURE -- DISKETTE STATUS UPDATED
                                                  <1>;
 2556
 2557
                                                                                     IF A FAILURE HAS OCCURRED, THE RETURN IS MADE ONE LEVEL
                                                                                     HIGHER THAN THE CALLER OF NEC OUTPUT. THIS REMOVES THE
 2558
                                                  <1>;
 2559
                                                                                     REQUIREMENT OF TESTING AFTER EVERY CALL OF NEC_OUTPUT.
                                                  <1>;
 2560
                                                                        AX,CX,DX DESTROYED
                                                  <1>;
 2561
                                                  <1> ;------
 2562
                                                  <1>
                                                  <1>; 09/12/2014 [Erdogan Tan]
 2563
 2564
                                                                (from 'PS2 Hardware Interface Tech. Ref. May 88', Page 09-05.)
 2565
                                                  <1> ; Diskette Drive Controller Status Register (3F4h)
                                                  <1>;
 2566
                                                                This read only register facilitates the transfer of data between
                                                                the system microprocessor and the controller.
 2567
 2568
                                                  <1>; Bit 7 - When set to 1, the Data register is ready to transfer data
 2569
                                                                   with the system micrprocessor.
                                                  <1>;
                                                  <1>; Bit 6 - The direction of data transfer. If this bit is set to 0,
 2570
                                                                   the transfer is to the controller.
 2571
                                                  <1>;
 2572
                                                  <1>; Bit 5 - When this bit is set to 1, the controller is in the non-DMA mode.
                                                  <1> ; Bit 4 - When this bit is set to 1, a Read or Write command is being executed.
 2573
 2574
                                                  <1> ; Bit 3 - Reserved.
 2575
                                                  <1>; Bit 2 - Reserved.
                                                  <1> ; Bit 1 - When this bit is set to 1, dskette drive 1 is in the seek mode.
 2576
 2577
                                                  <1>; Bit 0 - When this bit is set to 1, dskette drive 1 is in the seek mode.
 2578
                                                  <1>
 2579
                                                  <1>; Data Register (3F5h)
                                                  <1> ; This read/write register passes data, commands and parameters, and provides
 2580
 2581
                                                  <1> ; diskette status information.
 2582
                                                  <1>
                                                  <1> NEC_OUTPUT:
 2583
 2584
                                                  <1>
                                                                ; PUSH BX
                                                                                                     ; SAVE REG.
 2585 00003FD1 66BAF403
                                                  <1>
                                                                MOV DX,03F4H
                                                                                                      ; STATUS PORT
```

```
2586
                                <1>
                                         ;MOV BL,2
                                                                  ; HIGH ORDER COUNTER
2587
                                <1>
                                         ;XOR CX,CX
                                                                  ; COUNT FOR TIME OUT
                                         ; 16/12/2014
2588
                                <1>
                                         ; waiting for (max.) 0.5 seconds
2589
                                <1>
2590
                                <1>
                                                  byte [wait_count], 0 ;; 27/02/2015
2591
                                <1>
                                         ; 17/12/2014
2592
                                <1>
                                        ; Modified from AWARD BIOS 1999 - ADISK.ASM - SEND COMMAND
2593
                                <1>
2594
                                <1>
2595
                                <1>
                                         ;WAIT_FOR_PORT: Waits for a bit at a port pointed to by DX to
2596
                                <1>
2597
                                <1>
                                         ; INPUT:
2598
                                <1>
                                         ; AH=Mask for isolation bits.
2599
                                <1>
                                               AL=pattern to look for.
2600
                                <1>
                                               DX=Port to test for
2601
                                <1>
                                               BH:CX=Number of memory refresh periods to delay.
2602
                                                    (normally 30 microseconds per period.)
2603
                                <1>
2604
                                <1>
                                         ;WFP_SHORT:
2605
                                <1>
                                              Wait for port if refresh cycle is short (15-80 Us range).
2606
                                <1>
2607
                                <1>
                                         mov bl, WAIT_FDU_SEND_HI+1 ; 0+1
2608
                                <1>;
                                <1>;
2609
                                         mov cx, WAIT_FDU_SEND_LO ; 16667
2610 00003FD5 B91B410000
                                         mov ecx, WAIT_FDU_SEND_LH ; 16667 (27/02/2015)
                                <1>
2611
                                <1>;
2612
                                <1> ; WFPS_OUTER_LP:
                                <1>; ;
2613
2614
                                <1> ; WFPS_CHECK_PORT:
2615
                                <1> J23:
                                         IN AL,DX ; GET STATUS
AND AL,11000000B ; KEEP STATUS AND DIRECTION
CMP AL,1000000B ; STATUS 1 AND DIRECTION 0 ?
JZ short J27 ; STATUS AND DIRECTION OK
2616 00003FDA EC
                               <1>
2617 00003FDB 24C0
                               <1>
                             <1> CMP <1> JZ
2618 00003FDD 3C80
2619 00003FDF 7418
                               <1> WFPS_HI:
<1> IN
2620
                                         IN AL, PORT_B ;061h ; SYS1; wait for hi to lo
TEST AL,010H ; transition on memory
JNZ SHORT WFPS_HI ; refresh.
2621 00003FE1 E461
2622 00003FE3 A810
                              <1>
                             <1> JNZ
2623 00003FE5 75FA
2624
                               <1> WFPS_LO:
                                                                  ; SYS1
2625 00003FE7 E461
                               <1> IN
                                               AL, PORT_B
                               2626 00003FE9 A810
                               <1>
                                       TEST AL,010H
2627 00003FEB 74FA
                                               SHORT WFPS LO
                                         ;LOOP SHORT WFPS_CHECK_PORT
2628
                                       loop J23 ; 27/02/2015
2629 00003FED E2EB
                               <1>
                                <1>;
2630
2631
                                <1>;
                                         dec
2632
                                <1>;
                                         jnz short WFPS OUTER LP
                                <1>;
                                               short WFPS_TIMEOUT ; fail
2633
                                         jmp
2634
                                <1> ;J23:
                                                                  ; GET STATUS
                                <1> ; IN
2635
                                               AL,DX
                                         AND AL,11000000B; KEEP STATUS AND DIRECTION CMP AL,10000000B; STATUS 1 AND DIRECTION 0?
2636
                                <1>;
                                         CMP AL,10000000B
JZ short J27
2637
                                <1>;
2638
                                <1>;
                                                                  ; STATUS AND DIRECTION OK
2639
                                <1>
                                         ;LOOP J23
                                                                 ; CONTINUE TILL CX EXHAUSTED
                                                                 ; DECREMENT COUNTER
2640
                                <1>
                                        ;DEC BL
                                         ;JNZ short J23
2641
                                <1>
                                                                  ; REPEAT TILL DELAY FINISHED, CX = 0
2642
                                <1>
                                         ;;27/02/2015
                                <1>
2643
2644
                                <1>
                                         ;16/12/2014
2645
                                <1>
                                          ;;cmp byte [wait_count], 10 ; (10/18.2 seconds)
2646
                                <1>
                                         ;;jb short J23
2647
                                <1>
                                <1> ; WFPS_TIMEOUT:
2648
2649
                                <1>
2650
                                <1> ;----
                                               FALL THRU TO ERROR RETURN
2651
                                <1>
2652 00003FEF 800D[48590100]80 <1>
                                               byte [DSKETTE_STATUS],TIME_OUT
                                         ; POP BX
                                                                  ; RESTORE REG.
                               <1>
2653
                                                                  ; DISCARD THE RETURN ADDRESS
2654 00003FF6 58
                                <1>
                                         POP
                                               eAX ; 08/02/2015
2655 00003FF7 F9
                                <1>
                                         STC
                                                                   ; INDICATE ERROR TO CALLER
2656 00003FF8 C3
                                <1>
2657
                                <1>
                                                DIRECTION AND STATUS OK; OUTPUT BYTE
2658
                                <1> ;----
2659
                                <1>
2660
                                <1> J27:
                               2661 00003FF9 88E0
                                                                  ; GET BYTE TO OUTPUT
2662 00003FFB 6642
                                                                  ; DATA PORT = STATUS PORT + 1
                                                                  ; OUTPUT THE BYTE
2663 00003FFD EE
2664
2665
2666 00003FFE 9C
                                                                  ; SAVE FLAGS
                                                                   ; 30 TO 45 MICROSECONDS WAIT FOR
2667 00003FFF B903000000
                                                eCX, 3
                                <1>
                                         MOV
2668 00004004 E8EDDDFFFF
                                                          ; NEC FLAGS UPDATE CYCLE ; RESTORE FLAGS FOR EXIT
                            <1>
                                        CALL
                                                WAITF
2669 00004009 9D
                                <1>
                                         POPF
                                                                 ; RESTORE REG
2670
                                <1>
                                         ; POP BX
2671 0000400A C3
                                <1>
                                         RETn
                                                                  ; CY = 0 FROM TEST INSTRUCTION
2672
                                <1>
2673
                                <1> ;------
                                <1> ; SEEK
2674
2675
                                <1>;
                                         THIS ROUTINE WILL MOVE THE HEAD ON THE NAMED DRIVE TO THE NAMED
                                         TRACK. IF THE DRIVE HAS NOT BEEN ACCESSED SINCE THE DRIVE
2676
                                <1>;
2677
                                <1>;
                                         RESET COMMAND WAS ISSUED, THE DRIVE WILL BE RECALIBRATED.
                                <1>;
2678
2679
                                <1>; ON ENTRY: DI = DRIVE #
2680
                                <1>; CH = TRACK #
2681
                                <1>;
2682
                                <1> ; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION.
                                <1>; AX,BX,CX DX DESTROYED
2683
2684
                                <1> ;--
2685
                                <1> SEEK:
                               <1> MOV eBX,eDI
2686 0000400B 89FB
                                                                        ; BX = DRIVE #
                              2687 0000400D B001
2688 0000400F 86CB
2689 00004011 D2C0
2690 00004013 86CB
```

```
2697
                                                                   <1>
                                                                 2698
 2699
 2700 0000402A C605[48590100]00 <1>
 2701 00004031 E854000000
 2702 00004036 7251
                                                                                                                                        ; IF RECALIBRATE FAILS TWICE THEN ERROR
 2703
 2704
                                                                   <1> AFT_RECAL:
                                                                                      MOV byte [DSK_TRK+eDI],0 ; SAVE NEW CYLINDER AS PRESENT POSITION OR CH,CH ; CHECK FOR SEEK TO TRACK 0

JZ short DO_WAIT ; HEAD SETTLE, CY = 0 IF JUMP
 2705 00004038 C687[59590100]00 <1> MOV
 2706 0000403F 08ED
                                                                   <1>
                                                                   <1> JZ
 2707 00004041 743F
 2708
                                                                   <1>
 2709
                                                                    <1> ;----
                                                                                                     DRIVE IS IN SYNCHRONIZATION WITH CONTROLLER, SEEK TO TRACK
 2710
                                                                   <1>
 short R7 ; SINGLE STEP REQUIRED BYPASS DOUBLE
                                                                              JZ
 2712 0000404A 7402
                                                                   <1>
                                                                                                                                          ; DOUBLE NUMBER OF STEP TO TAKE
 2713 0000404C D0E5
                                                                  <1>
                                                                                       SHL
                                                                                                   CH,1
 2714
                                                                  <1>
 CH, [DSK_TRK+eDI] ; SEE IF ALREADY AT THE DESIRED TRACK
 2716 00004054 7433
| 2/16 | 00004054 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 7433 | 743
                                                                               JE
                                                                                                    short RB
                                                                                                                                           ; IF YES, DO NOT NEED TO SEEK
                                                                                                                                          ; ON STACK FOR NEC OUTPUT ERROR
                                                                                                   [DSK_TRK+eDI], CH ; SAVE NEW CYLINDER AS PRESENT POSITION
                                                                                                                                          ; ENDING INTERRUPT AND SENSE STATUS
 2729
                                                                   <1>
 2730
                                                                   <1> ;----
                                                                                                     WAIT FOR HEAD SETTLE
 2731
                                                                   <1>
                                                                   <1> DO WAIT:
 2732
                                                                  <1> PUSHF <1> CALL
                                                                                      PUSHF ; SAVE STATUS

CALL HD_WAIT ; WAIT FOR HEAD SETTLE TIME

POPF . PESTORE STATUS
 2733 00004082 9C
 2734 00004083 E816FFFFF
                                                                  <1>
 2735 00004088 9D
                                                                                                                                          ; RESTORE STATUS
                                                                   <1>
                                                                                      POPF
                                                                   <1> RB:
 2736
 2737
                                                                   <1> NEC ERR:
                                                                   - ; 08/02/2015 (code trick here from original IBM PC/AT DISKETTE.ASM)
 2738
 2739
                                                                    <1>
                                                                                        ; (*) nec_err -> retn (push edx -> pop edx) -> nec_err -> retn
                                                                                                                                        ; RETURN TO CALLER
 2740 00004089 C3
                                                                    <1>
 2741
                                                                    <1>
 2742
                                                                    <1> ;-----
 2743
                                                                    <1> ; RECAL
 2744
                                                                    <1>;
                                                                                        RECALIBRATE DRIVE
 2745
                                                                    <1>;
 2746
                                                                    <1> ; ON ENTRY: DI = DRIVE #
 2747
 2748
                                                                    <1> ; ON EXIT: CY REFLECTS STATUS OF OPERATION.
                                                                    <1> ;-----
 2749
 2750
                                                                   <1> RECAL:

PUSH CX

NOV eAX, RC_BACK

PUSH eAX

NOV AH,07H

CALL NEC_OUTPUT

NOV eBX,eDI

NOV AH,BL

CALL NEC_OUTPUT

CALL CHK_STAT_2

CALL CHK_STAT_2

THROW AWAY ERROR

CALC

THROW

TH
 2753 00004091 50

2754 00004092 B407

2755 00004094 E838FFFFFF

2756 00004099 89FB

2757 0000409B 88DC

2758 0000409D E82FFFFFF

2759 000040A2 E804000000

2760 000040A7 58
 2760 000040A7 58
 2761
 2762 000040A8 6659
 2763 000040AA C3
                                                                   <1>
                                                                                        RETn
 2764
                                                                    <1>
 2765
                                                                    <1> ;------
 2766
                                                                    <1>; CHK STAT 2
 2767
                                                                    <1>; THIS ROUTINE HANDLES THE INTERRUPT RECEIVED AFTER RECALIBRATE,
                                                                                      OR SEEK TO THE ADAPTER. THE INTERRUPT IS WAITED FOR, THE
 2768
                                                                    <1>;
                                                                                      INTERRUPT STATUS SENSED, AND THE RESULT RETURNED TO THE CALLER.
 2769
                                                                     <1>;
 2770
                                                                    <1>;
                                                                     <1>; ON EXIT: @DSKETTE STATUS, CY REFLECT STATUS OF OPERATION.
 2771
 2772
                                                                   <1> CHK_STAT_2:
 2774 000040AB B8[D3400000]
                                                                                                           eAX, CS_BACK
                                                                                                                                                        ; LOAD NEC OUTPUT ERROR ADDRESS
                                                                   <1>
                                                                                          MOV
                                                                                        PUSH eAX
 2775 000040B0 50
                                                                   <1>
                                                                                                                                          ; WAIT FOR THE INTERRUPT
 2776 000040B1 E828000000
                                                                  <1>
                                                                                       CALL WAIT_INT
                                                                             CALL WAIT_INT
JC short J34
MOV AH,08H
CALL NEC_OUTPUT
CALL RESULTS
JC short J34
MOV AL,[NEC_ST
AND AL,0110000
CMP AL,0110000
 2777 000040B6 721A
                                                                  <1>
                                                                                                   short J34
                                                                                                                                          ; IF ERROR, RETURN IT
; SENSE INTERRUPT STATUS COMMAND
                                                                                     CALL NEC OUTPUT
                                                                                                                                                        ; READ IN THE RESULTS
                                                                                                   Short U34
AL,[NEC_STATUS] ; GET THE BITS
FOR CORRECT
                                                                                                    short J34
                                                                                                                                                        ; GET THE FIRST STATUS BYTE
                                                                                                   AL,01100000B ; ISOLATE INE DITC

AT. 01100000B ; TEST FOR CORRECT VALUE
 2783 000040CB 2460
                                                                  <1>
 2784 000040CD 3C60
                                                                  <1>
                                                                                                    short J35
                                                                                                                                       ; IF ERROR, GO MARK IT
 2785 000040CF 7403
                                                                   <1>
                                                                                       JZ
 2786 000040D1 F8
                                                                   <1>
                                                                                       CLC
                                                                                                                                           ; GOOD RETURN
                                                                   <1> J34:
 2788 000040D2 58
                                                                                       POP
                                                                    <1>
                                                                                                                                           ; THROW AWAY ERROR RETURN
                                                                                                     eAX
 2789
                                                                    <1> CS_BACK:
 2790 000040D3 C3
                                                                    <1>
                                                                                       RETn
                                                                    <1> J35:
 2791
 2792 000040D4 800D[48590100]40
                                                                   <1>
                                                                                        OR
                                                                                                    byte [DSKETTE_STATUS], BAD_SEEK
                                                                                                                                          ; ERROR RETURN CODE
 2793 000040DB F9
                                                                    <1>
                                                                                        STC
 2794 000040DC EBF4
                                                                    <1>
                                                                                                     SHORT J34
                                                                                        JMP
 2795
                                                                    <1>
```

```
2796
                               <1> ;-----
2797
                               <1> ; WAIT INT
2798
                               <1>; THIS ROUTINE WAITS FOR AN INTERRUPT TO OCCUR A TIME OUT ROUTINE
2799
                               <1>;
                                      TAKES PLACE DURING THE WAIT, SO THAT AN ERROR MAY BE RETURNED
2800
                               <1>;
                                      IF THE DRIVE IS NOT READY.
2801
                               <1>;
                               <1>; ON EXIT: @DSKETTE STATUS, CY REFLECT STATUS OF OPERATION.
2802
2803
                               <1> ;-----
2804
                               <1>
2805
                               <1> ; 17/12/2014
2806
                               <1> ; 2.5 seconds waiting !
2807
                               <1> ; (AWARD BIOS - 1999, WAIT FDU INT LOW, WAIT FDU INT HI)
2808
                               <1>; amount of time to wait for completion interrupt from NEC.
2809
                               <1>
2810
                               <1>
                               <1> WAIT_INT:
2811
2812 000040DE FB
                                                                ; TURN ON INTERRUPTS, JUST IN CASE
                               <1> STI
2813 000040DF F8
                                      CLC
                                                                ; CLEAR TIMEOUT INDICATOR
                               <1>
                                      ;MOV BL,10
;XOR CX,CX
                                                                ; CLEAR THE COUNTERS
2814
                               <1>
2815
                               <1>
                                                                 ; FOR 2 SECOND WAIT
2816
                               <1>
                                      ; Modification from AWARD BIOS - 1999 (ATORGS.ASM, WAIT
2817
                               <1>
2818
                               <1>
                                    ; WAIT_FOR_MEM: ; Waits for a bit at a specified memory location pointed
2819
                               <1>
2820
                               <1>
2821
                               <1>
                                              to by ES:[DI] to become set.
                                      ; INPUT:
2822
                               <1>
                                      ; AH=Mask to test with.
                               <1>
2823
                               ;
<1> ;
2824
                                              ES:[DI] = memory location to watch.
2825
                                              BH:CX=Number of memory refresh periods to delay.
2826
                               <1>
                                                   (normally 30 microseconds per period.)
2827
                               <1>
2828
                               <1>
                                        ; waiting for (max.) 2.5 secs in 30 micro units.
2829
                               <1>; mov cx, WAIT_FDU_INT_LO ; 017798
                               <1> ;; mov bl, WAIT_FDU_INT_HI
<1> ; mov bl, WAIT_FDU_INT_HI + 1
2830
2831
                               <1>
                                        ; 27/02/2015
2832
                               <1> mov ecx, WAIT_FDU_INT_LH ; 83334 (2.5 seconds)
2833 000040E0 B986450100
                               <1> WFMS CHECK MEM:
2834
2835 000040E5 F605[45590100]80
                              <1> test byte [SEEK_STATUS], INT_FLAG; TEST FOR INTERRUPT OCCURRING
2836 000040EC 7516
                              <1>
                                        jnz
                                               short J37
                              <1> WFMS_HI:
<1> IN AL,PORT_B ; 061h ; SYS1, wait for lo to hi
<1> TEST AL,010H ; transition on mem
2837

<1> WFMS_HI:
<1> IN AL,PORT_B
<1> TEST AL,010H
<1> JNZ SHORT WFMS_
<1> WFMS_LO:
<1> IN AL,PORT_B
<1> TEST AL,010H

2838 000040EE E461
                                       TEST AL,010H ; transition on memory JNZ SHORT WFMS_HI ; refresh.
2839 000040F0 A810
2840 000040F2 75FA
2841
2842 000040F4 E461
                                                               ;SYS1
                              <1>
2843 000040F6 A810
                                        TEST AL,010H
                              2844 000040F8 74FA
2845 000040FA E2E9
                               <1> ; WFMS_OUTER_LP:
                               short J36A ; Check outer counter ; WFMS TIMPOUT
2846
2847
                               <1> ;;
2848
2849
                               <1>;
                                      dec bl
                                      jz short J36A
jmp short WFMS_CHECK_MEM
2850
                               <1>;
2851
                               <1>;
2852
                               <1>
                                      ;17/12/2014
2853
                               <1>
2854
                               <1>
                                        ;16/12/2014
                                         mov byte [wait_count], 0 ; Reset (INT 08H) counter
2855
                               <1>;
2856
                               <1> ;J36:
                                        TEST byte [SEEK_STATUS], INT_FLAG; TEST FOR INTERRUPT OCCURRING JNZ short J37
                               <1> ;
2857
2858
                               <1>;
                                        ;16/12/2014
2859
                               <1>
                                        ;LOOP J36
                                                              ; COUNT DOWN WHILE WAITING
2860
                               <1>
                                        ; DEC BL
2861
                               <1>
                                                                ; SECOND LEVEL COUNTER
                                        ;JNZ short J36
2862
                               <1>
                                         cmp byte [wait_count], 46 ; (46/18.2 seconds)
2863
                               <1>;
2864
                               <1>;
                                        jb short J36
2865
                               <1>
                               <1> ; WFMS TIMEOUT:
2866
2867
                               <1> ;J36A:
                               <1> OR 
<1> STC
2868 000040FC 800D[48590100]80
                                               byte [DSKETTE_STATUS], TIME_OUT; NOTHING HAPPENED
2869 00004103 F9
                                                               ; ERROR RETURN
2870
                               <1> J37:
                              <1> PUSHF
<1> AND
<1> POPF
2871 00004104 9C
                                                                ; SAVE CURRENT CARRY
2872 00004105 8025[45590100]7F
                                        AND byte [SEEK STATUS], ~INT FLAG; TURN OFF INTERRUPT FLAG
                                                               ; RECOVER CARRY
2873 0000410C 9D
2874 0000410D C3
                                                                 ; GOOD RETURN CODE
                               <1>
                                        RETn
2875
                               <1>
                               <1> ;------
2876
2877
                                <1> ; RESULTS
2878
                               <1>;
                                        THIS ROUTINE WILL READ ANYTHING THAT THE NEC CONTROLLER RETURNS
2879
                               <1>;
                                        FOLLOWING AN INTERRUPT.
2880
                               <1>;
                               <1>; ON EXIT: @DSKETTE STATUS, CY REFLECT STATUS OF OPERATION.
2881
                               <1>; AX, BX, CX, DX DESTROYED
2882
2883
                               <1> ;------
                               <1> RESULTS:
2884
2885 0000410E 57
                              <1> PUSH eDI
                                              eDI, NEC_STATUS ; POINTER TO DATA AREA
                                      MOV
2886 0000410F BF[49590100]
                              <1>
                                              BL,7 ; MAX STATUS BYTES
2887 00004114 B307
                               <1>
                                        VOM
2888 00004116 66BAF403
                               <1>
                                        MOV
                                              DX,03F4H
                                                                ; STATUS PORT
2889
                               <1>
2890
                               <1> ;---- WAIT FOR REQUEST FOR MASTER
2891
                               <1>
2892
                               <1> _R10:
                                       ; 16/12/2014
2893
                               <1>
                                        ; wait for (max) 0.5 seconds
2894
                               <1>
2895
                               <1>
                                        ;MOV BH,2 ; HIGH ORDER COUNTER
2896
                               <1>
                                        ;XOR CX,CX
                                                                ; COUNTER
2897
                               <1>
                                       ;Time to wait while waiting for each byte of NEC results = .5
2898
                               <1>
                                       ; seconds. .5 \text{ seconds} = 500,000 \text{ micros}. 500,000/30 = 16,667.
2899
                               <1>
2900
                               <1>
                                        ; 27/02/2015
```

```
mov ecx, WAIT_FDU_RESULTS_LH; 16667; mov cx, WAIT_FDU_RESULTS_LO; 16667
2901 0000411A B91B410000
                               <1>
2902
                                   <1>
                                             ;mov bh, WAIT_FDU_RESULTS_HI+1; 0+1
2903
                                   <1>
2904
                                   <1>
2905
                                   <1> WFPSR_OUTER_LP:
2906
                                   <1>
                                   <1> WFPSR_CHECK_PORT:
2907
                                             ; WAIT FOR MASTER

IN AL,DX ; GET STATUS

AND AL,11000000B ; KEEP ONLY STATUS AND DIRECTION

CMP AL,11000000B ; STATUS 1 AND DIRECTION 1 ?

JZ short J42 ; STATUS AND DIRECTION OK
                                   <1> J39:
2908
2909 0000411F EC
                                   <1>
2910 00004120 24C0
                                  <1>
                                  <1>
<1>
2911 00004122 3CC0
                                <1>
                            2912 00004124 7418
2913
2914 00004126 E461
2915 00004128 A810
2916 0000412A 75FA
2918 0000412C E461
2919 0000412E A810
                                        JZ SHORT WFPSR_LO

LOOP WFPSR_CHECK_PORT

;; 27/02/2015

;; dec bh

;; jnz short WFPSR_OUTER_LP
2920 00004130 74FA
2921 00004132 E2EB
                                   <1>
2922
2923
                                   <1>
2924
                                   <1>
2925
                                   <1>
                                             ;jmp short WFPSR_TIMEOUT; fail
2926
                                   <1>
                                             ; WAIT FOR MASTER

IN AL,DX ; GET STATUS

AND AL,11000000B ; KEEP ONLY STATUS AND DIRECTION

CMP AL,11000000B ; STATUS 1 AND DIRECTION 1 ?

JZ short J42 ; STATUS AND DIRECTION OF
2927
                                   <1>
                                   <1>; J39:
2928
2929
                                   <1>;
2930
2931
                                   <1>;
                                                    short J42
                                                                       ; STATUS AND DIRECTION OK
2932
                                   <1>;
                                           ; LOOP J39 ; LOOP TILL TIMEOUT
; DEC BH ; DECREMENT HIGH ORDER CO
; JNZ short J39 ; REPEAT TILL DELAY DONE
2933
                                   <1>
2934
                                   <1>
                                                                       ; DECREMENT HIGH ORDER COUNTER
2935
                                   <1>
2936
                                   <1>
                                          ;;cmp byte [wait;;jb short J39
                                             ;;cmp byte [wait count], 10 ; (10/18.2 seconds)
2938
                                   <1>
2939
                                   <1>
2940
                                   <1> ; WFPSR TIMEOUT:
2941 00004134 800D[48590100]80
                                  <1> OR byte [DSKETTE_STATUS],TIME_OUT
                                             STC ; SET ERROR RETURN

JMP SHORT POPRES ; POP REGISTERS AND RETURN
2942 0000413B F9
                                   <1>
2943 0000413C EB29
                                   <1>
2944
                                   <1>
                                                    READ IN THE STATUS
2945
                                   <1> ;----
2946
                                   <1>
                                   <1> J42:
2947
                                  2948 0000413E EB00
                                                                      ; POINT AT DATA PORT
2949 00004140 6642
2950 00004142 EC
                                                                        ; GET THE DATA
2951
2952
                                  <2> out 0ebh,al
<1> MOV [eDI],AL ; STORE THE BYTE
<1> INC eDI ; INCREMENT THE POINTER
2952 00004143 E6EB
2953 00004145 8807
2954 00004147 47
2955
                                   <1>
                                             ; 16/12/2014
2956
                                   <1>;
                                             push cx
                                   <1>;
2957
                                             mov cx, 30
2958
                                   <1> ; wdw2:
2959
                                   <1>; NEWIODELAY
2960
                                   <1>;
                                             loop wdw2
                                             pop cx
2961
                                   <1>;
2962
                                   <1>
                                                                     ; MINIMUM 24 MICROSECONDS FOR NEC
; WAIT 30 TO 45 MICROSECONDS
; POINT AT STRATUS
                                        MOV eCX,3
CALL WAITF
DEC DX
2963 00004148 B903000000
                                  <1>
2964 0000414D E8A4DCFFFF
                                   <1>
                                           DEC DX
IN AL,DX
2965 00004152 664A
                                  <1>
                                                                        ; POINT AT STATUS PORT
2966 00004154 EC
                                  <1>
                                                                        ; GET STATUS
2967
                                  <1>
                                             ; 16/12/2014
                                  <1> ; 16/12/201
<1> NEWIODELAY
2968
2968 00004155 E6EB
                                  <2> out Oebh,al
2969
                                  <1>
                                             TEST AL,00010000B ; TEST FOR NEC STILL BUSY JZ short POPRES ; RESULTS DONE ?
2970 00004157 A810
                                   <1>
2971 00004159 740C
                                  <1>
                                             JZ
2972
                                  <1>
                                                                     ; DECREMENT THE STATUS COUNTER ; GO BACK FOR MORE
2973 0000415B FECB
                                   <1>
                                            DEC BL
                                             JNZ short _R10
2974 0000415D 75BB
                                   <1>
                                              OR byte [DSKETTE_STATUS], BAD_NEC; TOO MANY STATUS BYTES
2975 0000415F 800D[48590100]20
                                  <1>
                                                                      ; SET ERROR FLAG
2976 00004166 F9
                                   <1>
2977
                                   <1>
2978
                                   <1>;----
                                                    RESULT OPERATION IS DONE
2979
                                   <1> POPRES:
2980 00004167 5F
                                    <1>
2981 00004168 C3
                                   <1>
                                              RETn
                                                                         ; RETURN WITH CARRY SET
                                   <1>
2983
                                   <1> ;------
2984
                                   <1> ; READ DSKCHNG
2985
                                   <1>;
                                            READS THE STATE OF THE DISK CHANGE LINE.
2986
                                   <1>;
2987
                                   <1>; ON ENTRY: DI = DRIVE #
                                   <1>;
2988
2989
                                   <1>; ON EXIT: DI = DRIVE #
2990
                                   <1>; ZF = 0 : DISK CHANGE LINE INACTIVE
                                   <1>;
                                                    ZF = 1 : DISK CHANGE LINE ACTIVE
2991
2992
                                                   AX,CX,DX DESTROYED
                                   <1>;
2993
                                   <1> ;-----
                                   <1> READ_DSKCHNG:
2994
                                           CALL MOTOR_ON ; TURN ON THE MOTOR IF OFF
MOV DX,03F7H ; ADDRESS DIGITAL INPUT REGISTER
IN AL,DX ; INPUT DIGITAL INPUT REGISTER
TEST AL,DSK_CHG ; CHECK FOR DISK CHANGE LINE ACTIVE
2995 00004169 E8A2FDFFFF
                                  <1> CALL MOTOR_ON
2996 0000416E 66BAF703
                                  <1>
2997 00004172 EC
                                   <1>
2998 00004173 A880
                                  <1>
2999 00004175 C3
                                   <1>
                                            RETn
                                                                       ; RETURN TO CALLER WITH ZERO FLAG SET
3000
                                   <1>
3001
                                   3002
                                   <1> ; DRIVE DET
3003
                                   <1>; DETERMINES WHETHER DRIVE IS 80 OR 40 TRACKS AND
```

```
3004
                                  <1>; UPDATES STATE INFORMATION ACCORDINGLY.
 3005
                                   <1>; ON ENTRY: DI = DRIVE #
3006
                                  <1> ;-----
                                  <1> DRIVE DET:
                                                                            ; SENSE DRIVE STATUS COMMAND BYTE
                                                  CH,CH ; IS HOME AT TRACK 0 short IS_80 ; MUST BE 80 TRACK DRIVE
 3033 000041C2 7408
                                  <1>
                                            JΖ
 3034
                                  <1>
 3035
                                           DRIVE IS A 360; SET DRIVE TO DETERMINED;
                                  <1>;
                                           SET MEDIA TO DETERMINED AT RATE 250.
 3036
                                  <1>;
 3037
                                  <1>
 3038 000041C4 808F[55590100]94
                                                  byte [DSK_STATE+eDI], DRV_DET+MED_DET+RATE_250
                                  <1>
                                           OR
                                  <1>
 3039 000041CB C3
                                           RETn
                                                                     ; ALL INFORMATION SET
                                  <1> IS_80:
 3041 000041CC 808F[55590100]01
                                  <1> OR
                                                  byte [DSK_STATE+eDI], TRK_CAPA; SETUP 80 TRACK CAPABILITY
 3042
                                  <1> DD_BAC:
 3043 000041D3 C3
                                  <1> RETn
 3044
                                  <1> POP_BAC:
 3045 000041D4 6659
                                                 CX ; THROW AWAY
                                  <1> POP
                                  <1>
 3046 000041D6 C3
                                            RETn
 3047
                                  <1>
                                  <1> fdc_int:
 3048
                                  <1> ; 30/07/2015
<1> ; 16/02/2015
 3049
 3050
 3051
                                  <1> ;int_0Eh: ; 11/12/2014
 3052
                                   <1>
                                   <1> ;--- HARDWARE INT OEH -- ( IRQ LEVEL 6 ) ------
 3053
 3054
                                   <1> ; DISK_INT
 3055
                                   <1>; THIS ROUTINE HANDLES THE DISKETTE INTERRUPT.
 3056
                                   <1> ;
 3057
                                   <1> ; ON EXIT: THE INTERRUPT FLAG IS SET IN @SEEK_STATUS.
 3058
                                  <1> ;-----
 3059
                                  <1> DISK_INT_1:
                                <1>
<1> PUSH AX
<1> push ds
<1> mov ax, KDATA
<1> mov ds, ax
<1> OR byte [SEEK_STATUS], INT_FLAG; TURN ON INTERRUPT OCCURRED
<1> MOV AL,EOI ; END OF INTERRUPT MARKER
<1> OUT INTA00,AL ; INTERRUPT CONTROL PORT
<1> pop ds
<1> POP AX ; RECOVER REGISTER
IRETD ; RETURN FROM INTERRUPT
 3060
                                  <1>
                                                        ; SAVE WORK REGISTER
 3061 000041D7 6650
 3062 000041D9 1E
 3063 000041DA 66B81000
3064 000041DE 8ED8
 3064 000041DE 8ED8
 3065 000041E0 800D[45590100]80
 3066 000041E7 B020
 3067 000041E9 E620
 3068 000041EB 1F
 3069 000041EC 6658
 3070 000041EE CF
 3071
 3072
 3073
                                  <1> ; DSKETTE SETUP
                                  <1> ;
 3074
                                           THIS ROUTINE DOES A PRELIMINARY CHECK TO SEE WHAT TYPE OF
                                           DISKETTE DRIVES ARE ATTACH TO THE SYSTEM.
 3075
                                   <1>;
 3076
                                  <1> ;-----
 3077
                                  <1>
                                   <1>; 29/05/2016 - TRDOS 386 (TRDOS v2.0)
 3078
 3079
                                   <1>
 3080
                                   <1> DSKETTE SETUP:
                                        ; PUSH AX
 3081
                                  <1>
                                                                     ; SAVE REGISTERS
                                           ; PUSH BX
 3082
                                   <1>
                                           ; PUSH CX
 3083
                                  <1>
                                       PUSH eDX
 3084 000041EF 52
                                  <1>
                                       ;PUSH DI
                                   <1>
 3086
                                   <1>
                                            ;; PUSH DS
                                   <1>
                                            ; 14/12/2014
 3087
                                            ; mov word [DISK POINTER], MD TBL6
 3088
                                   <1>
                                                 [DISK POINTER+2], cs
 3089
                                   <1>
                                            ;mov
 3090
                                   <1>
 3091
                                   <1>
                                            ;OR
                                                  byte [RTC_WAIT_FLAG], 1 ; NO RTC WAIT, FORCE USE OF LOOP
 3092 000041F0 31FF
                                   <1>
                                            XOR
                                                                            ; INITIALIZE DRIVE POINTER
 3093 000041F2 66C705[55590100]00- <1>
                                                  WORD [DSK_STATE],0 ; INITIALIZE STATES
                                            MOV
 3093 000041FA 00
                                   <1>
                                                  byte [LASTRATE],~(STRT MSK+SEND MSK) ; CLEAR START & SEND
 3094 000041FB 8025[50590100]33
                                   <1>
                                            AND
                                                  byte [LASTRATE], SEND_MSK ; INITIALIZE SENT TO IMPOSSIBLE
 3095 00004202 800D[50590100]C0
                                            OR
                                  <1>
 3096 00004209 C605[45590100]00
                                  <1>
                                            MOV
                                                  byte [SEEK_STATUS], 0 ; INDICATE RECALIBRATE NEEDED
                                                                         ; INITIALIZE MOTOR COUNT
 3097 00004210 C605[47590100]00
                                   <1>
                                            MOV
                                                  byte [MOTOR_COUNT], 0
 3098 00004217 C605[46590100]00
                                                                           ; INITIALIZE DRIVES TO OFF STATE
                                  <1>
                                            MOV
                                                  byte [MOTOR_STATUS],0
                                                  byte [DSKETTE STATUS], 0 ; NO ERRORS
 3099 0000421E C605[48590100]00
                                   <1>
                                            VOM
 3100
                                  <1>
 3101
                                  <1>
                                            ; 28/02/2015
 3102
                                            ;mov word [cfd], 100h
                                  <1>
 3103 00004225 E82BF2FFFF
                                  <1>
                                            call DSK_RESET
 3104 0000422A 5A
                                  <1>
                                            pop
                                                  edx
 3105 0000422B F8
                                                  ; 29/05/2016
                                  <1>
                                            clc
 3106 0000422C C3
                                  <1>
 3107
                                  <1>
```

```
3108
                            <1> ;SUP0:
                            3109
                                                         ; TRANSLATE STATE TO COMPATIBLE MODE
3110
                                   ; 02/01/2015
3111
                            <1>;
                                 ;INC DI
;CMP DI,MAX_DRV
                                                        ; POINT TO NEXT DRIVE
3112
                            <1>;
                                   ;CMP DI,MAX_DRV ; SEE IF DONE
;JNZ short SUPO ; REPEAT FOR EACH ORIVE
3113
                            <1>;
3114
                            <1>;
                           <1> ;
                                    cmp byte [fd1_type], 0
3115
                                   jna short sup1
or di, di
3116
                            <1>;
                            <1>;
3117
3118
                            <1>;
                                    jnz short sup1
                           3119
3120
3121
                            3122
3123
                                   CALL SETUP_END ; VARIOUS CLEANUPS ;; POP DS ; RESTORE CALLERS REGISTERS
3124
                            <1>;
                                  ;;POP DS
3125
                            <1>;
                                   ; POP DI
3126
                            <1>;
                            <1>;
3127
                                   POP
                                        eDX
3128
                            <1> ;
                                   ; POP CX
3129
                            <1>;
                                   ; POP
                                         BX
                            <1>;
3130
                                   ; POP AX
                            <1>;
3131
3132
                            <1>
                            3133
3134
                            3135
                            <1>;
3136
                            <1>
3137
                           <1> int13h: ; 21/02/2015
3138 0000422D F8
                           <1>
                                clc ; 20/07/2020
                                  pushfd
3139 0000422E 9C
                           <1>
3140 0000422F 0E
                           <1>
                                   push cs
                                  call DISK_IO
3141 00004230 E848010000
                           <1>
3142 00004235 C3
                           <1>
                                   retn
3143
                           <1>
                            3144
3145
                            3146
                            <1>
                            <1>; DISK I/O - Erdogan Tan (Retro UNIX 386 v1 project)
3147
3148
                            <1> ; 18/02/2016
                            <1> ; 17/02/2016
3149
                            <1>; 23/02/2015
3150
                            <1>; 21/02/2015 (unix386.s)
3151
3152
                            <1>; 22/12/2014 - 14/02/2015 (dsectrm2.s)
3153
                            <1>;
                            <1>; Original Source Code:
3154
                            <1>; DISK ---- 09/25/85 FIXED DISK BIOS
3155
3156
                            <1>; (IBM PC XT Model 286 System BIOS Source Code, 04-21-86)
3157
3158
                            <1>; Modifications: by reference of AWARD BIOS 1999 (D1A0622)
3159
                            <1>;
                                  Source Code - ATORGS.ASM, AHDSK.ASM
                            <1>;
3160
3161
                            <1>
3162
                            <1>
3163
                            <1> ; The wait for controller to be not busy is 10 seconds.
                            <1>;10,000,000 / 30 = 333,333. 333,333 decimal = 051615h
3164
                            <1> ;; WAIT_HDU_CTLR_BUSY_LO equ 1615h
3165
                            3166
3167
3168
                            <1>
3169
                            <1> ;The wait for controller to issue completion interrupt is 10 seconds.
                            <1>;10,000,000 / 30 = 333,333. 333,333 decimal = 051615h
3170
3171
                            <1>;;WAIT_HDU_INT_LO equ 1615h
                                                   05h
3172
                            <1>;;WAIT_HDU_INT_HI equ
3173
                            <1> WAIT_HDU_INT_LH
                                                   equ
                                                        51615h; 21/02/2015
3174
                            <1>
3175
                            <1> ; The wait for Data request on read and write longs is
                            <1> ;2000 us. (?)
3176
3177
                            <1> ;; WAIT_HDU_DRQ_LO equ 1000 ; 03E8h
                            <1>;;WAIT_HDU_DRQ_HI equ 0
3178
                            <1> WAIT_HDU_DRQ_LH
                                                   equ 1000 ; 21/02/2015
3179
3180
                            <1>
3181
                            <1>; Port 61h (PORT_B)
                            <1> SYS1 equ 61h ; PORT_B (diskette.inc)
3182
3183
                            <1>
                            <1>; 23/12/2014
3184
                            <1> %define CMD_BLOCK
3185
                                                  eBP-8 ; 21/02/2015
3186
                            <1>
3187
                            <1>; 30/08/2020 - TRDOS 386 v2
3188
                            <1>
3189
                            3190
                            <1> ; FIXED DISK I/O INTERFACE
3192
                            <1>;
                                    THIS INTERFACE PROVIDES ACCESS TO 5 1/4" FIXED DISKS THROUGH
3193
                            <1>;
3194
                            <1>;
                                   THE IBM FIXED DISK CONTROLLER.
                            <1>;
3195
                            <1> ;
                                   THE BIOS ROUTINES ARE MEANT TO BE ACCESSED THROUGH
3196
                                   SOFTWARE INTERRUPTS ONLY. ANY ADDRESSES PRESENT IN
3197
                            <1>;
                                   THESE LISTINGS ARE INCLUDED ONLY FOR COMPLETENESS,
3198
                            <1>;
                                   NOT FOR REFERENCE. APPLICATIONS WHICH REFERENCE ANY
3199
                            <1>;
                                   ABSOLUTE ADDRESSES WITHIN THE CODE SEGMENTS OF BIOS
                            <1>;
3200
3201
                            <1>;
                                    VIOLATE THE STRUCTURE AND DESIGN OF BIOS.
                            <1>;
3202
3203
                            <1> ;--
3204
                            <1>;
3205
                            <1> ; INPUT (AH) = HEX COMMAND VALUE
                            <1>;
3206
3207
                            <1>;
                                    (AH) = 00H RESET DISK (DL = 80H, 81H) / DISKETTE
                                    (AH) = 01H READ THE STATUS OF THE LAST DISK OPERATION INTO (AL)
3208
                            <1>;
                                   NOTE: DL < 80H - DISKETTE
DL > 80H - DISK
3209
                            <1>;
3210
                            <1>;
3211
                            <1>;
                                    (AH) = 02H READ THE DESIRED SECTORS INTO MEMORY
3212
                            <1>;
                                    (AH) = 03H WRITE THE DESIRED SECTORS FROM MEMORY
```

```
<1> ;
                                          (AH) = 04H VERIFY THE DESIRED SECTORS
3214
                                <1>;
                                          (AH) = 05H FORMAT THE DESIRED TRACK
                                <1>;
                                          (AH) = 06H UNUSED
3215
3216
                                <1>;
                                         (AH) = 07H UNUSED
3217
                                <1>;
                                         (AH) = 08H RETURN THE CURRENT DRIVE PARAMETERS
3218
                                <1>;
                                         (AH) = 09H INITIALIZE DRIVE PAIR CHARACTERISTICS
                                         INTERRUPT 41 POINTS TO DATA BLOCK FOR DRIVE 0
                                <1>;
                                <1>;
                                                   INTERRUPT 46 POINTS TO DATA BLOCK FOR DRIVE 1
3220
3221
                                <1>;
                                         (AH) = 0AH READ LONG
3222
                                        (AH) = 0BH WRITE LONG (READ & WRITE LONG ENCOMPASS 512 + 4 BYTES ECC) :
                                <1>;
3223
                                <1>;
                                       (AH) = OCH SEEK
                                         (AH) = 0DH ALTERNATE DISK RESET (SEE DL)
3224
                                <1>;
                                         (AH) = 0EH UNUSED
                                <1>;
3225
3226
                                <1>;
                                        (AH) = OFH UNUSED
                                         (AH) = 10H TEST DRIVE READY
3227
                                <1>;
                                         (AH) = 11H RECALIBRATE
3228
                                <1>;
                                       (AH) = 12H UNUSED
                                       (AH) = 13H UNUSED
3230
                                <1>;
3231
                                <1>;
                                         (AH) = 14H CONTROLLER INTERNAL DIAGNOSTIC
                                       (AH) = 15H READ DASD TYPE
3232
                                <1>;
3233
                                <1>;
3234
                                <1> ;------
                                <1>;
3235
3236
                                       REGISTERS USED FOR FIXED DISK OPERATIONS
3237
                                <1>;
                                              (DL) - DRIVE NUMBER (80H-81H FOR DISK. VALUE CHECKED) : (DH) - HEAD NUMBER (0-15 ALLOWED, NOT VALUE CHECKED) :
3238
                                <1>;
3239
                                               (CH) - CYLINDER NUMBER (0-1023, NOT VALUE CHECKED) (SEE CL):
                                <1>;
3240
                                               (CL) - SECTOR NUMBER (1-17, NOT VALUE CHECKED) :
                                <1>;
3241
3242
                                <1> ;
3243
                                                         NOTE: HIGH 2 BITS OF CYLINDER NUMBER ARE PLACED
                                                             IN THE HIGH 2 BITS OF THE CL REGISTER :
3244
3245
                                                              (10 BITS TOTAL)
                                <1>;
3246
                                <1>;
                                              (AL) - NUMBER OF SECTORS (MAXIMUM POSSIBLE RANGE 1-80H, :
3247
                                           FOR READ/WRITE LONG 1-79H

(ES:BX) - ADDRESS OF BUFFER FOR READS AND WRITES,
3248
                                <1>;
                                                           FOR READ/WRITE LONG 1-79H) :
3250
                                <1> ;
                                               (NOT REQUIRED FOR VERIFY)
                                <1>;
3251
3252
3253
                                               FORMAT (AH=5) ES:BX POINTS TO A 512 BYTE BUFFER. THE FIRST
                                <1>;
                                               2*(SECTORS/TRACK) BYTES CONTAIN F,N FOR EACH SECTOR.:

F = 00H FOR A GOOD SECTOR

80H FOR A BAD SECTOR

N = SECTOR NUMBER
:
3254
3255
                                <1>;
3256
3257
                                <1>;
                                                         FOR AN INTERLEAVE OF 2 AND 17 SECTORS/TRACK
3258
                                <1>;
                                                       THE TABLE SHOULD BE:
3259
3260
                                <1>;
                                               DB 00H,01H,00H,0AH,00H,02H,00H,0BH,00H,03H,00H,0CH : DB 00H,04H,00H,0DH,00H,05H,00H,0EH,00H,06H,00H,0FH : DB 00H,07H,00H,10H,00H,08H,00H,11H,00H,09H :
                                <1> ;
3261
3262
3263
                                <1>;
3264
3265
                                <1> ; --
3266
                                <1>
3267
                                <1> ;-----
                                <1> ; AH = STATUS OF CURRENT OPERATION
<1> ; STATUS DIEGO ADD DETECTION
3268
3269
                                         STATUS BITS ARE DEFINED IN THE EQUATES BELOW
CY = 0 SUCCESSFUL OPERATION (AH=0 ON RETURN)
3270
3271
                                <1>;
                                       CY = 1 FAILED OPERATION (AH HAS ERROR REASON)
3272
                                <1>;
3273
                                <1>;
                                         NOTE: ERROR 11H INDICATES THAT THE DATA READ HAD A RECOVERABLE
3274
                                <1>;
                                                ERROR WHICH WAS CORRECTED BY THE ECC ALGORITHM. THE DATA
3275
                                <1>;
3276
                                <1>;
                                                IS PROBABLY GOOD, HOWEVER THE BIOS ROUTINE INDICATES AN
3277
                                <1>;
                                                ERROR TO ALLOW THE CONTROLLING PROGRAM A CHANCE TO DECIDE
                                                FOR ITSELF. THE ERROR MAY NOT RECUR IF THE DATA IS
3278
                                <1>;
                                <1>;
                                <1>;
3280
                                       IF DRIVE PARAMETERS WERE REQUESTED (DL >= 80H),
3281
                                <1>;
                                       INPUT:
  (DL) = DRIVE NUMBER
3282
                                <1>;
3283
                                <1>;
                                              ; 27/05/2016 - TRDOS 386 (TRDOS v2.0)
3284
                                          EBX = Buffer address for fixed disk parameters table (32 bytes)
                                <1>;
3285
                                           OUTPUT:
3286
                                <1>;
                                       OUTPUT: :

(DL) = NUMBER OF CONSECUTIVE ACKNOWLEDGING DRIVES ATTACHED (1-2) :
3287
                                <1>;
                                              (CONTROLLER CARD ZERO TALLY ONLY)
                                           (DH) = MAXIMUM USEABLE VALUE FOR HEAD NUMBER
(CH) = MAXIMUM USEABLE VALUE FOR HEAD NUMBER
3289
                                <1>;
3290
                                               (CH) = MAXIMUM USEABLE VALUE FOR CYLINDER NUMBER
                                <1>;
3291
                                              (CL) = MAXIMUM USEABLE VALUE FOR SECTOR NUMBER
                                <1>;
                                                   AND CYLINDER NUMBER HIGH BITS
3292
3293
                                 <1>;
                                <1>;
                                          IF READ DASD TYPE WAS REQUESTED,
3295
                                <1>;
3296
                                <1>;
                                          AH = 0 - NOT PRESENT
                                           1 - DISKETTE - NO CHANGE LINE AVAILABLE
3297
                                <1>;
                                              2 - DISKETTE - CHANGE LINE AVAILABLE
3298
                                <1>;
                                <1>;
                                              3 - FIXED DISK
3299
                                <1>;
3300
3301
                                <1>;
                                          CX,DX = NUMBER OF 512 BYTE BLOCKS WHEN AH = 3
3302
                                <1> ;
                                          REGISTERS WILL BE PRESERVED EXCEPT WHEN THEY ARE USED TO RETURN
3303
                                <1>;
3304
                                <1>;
                                          INFORMATION.
3305
                                <1>;
3306
                                <1>;
                                          NOTE: IF AN ERROR IS REPORTED BY THE DISK CODE, THE APPROPRIATE
3307
                                <1>;
                                                ACTION IS TO RESET THE DISK, THEN RETRY THE OPERATION.
3308
                                <1> ;------
3309
3310
                                <1>
                                <1> SENSE_FAIL EQU OFFH ; NOT IMPLEMENTED
<1> NO_ERR EQU OEOH ; STATUS ERROR/ERROR REGISTER=0
3311
                                ; STATUS ERROR/ERROR REGI

<1> WRITE FAULT EQU OCCH ; WRITE FAULT ON SELECTED DRIVE

<1> UNDEF ERR EQU OBBH ; UNDEFINED ERROR OCCUPAND

<1> NOT DEV.
3312
3313
3314
                                3315
3316
```

3213

```
; CONTROLLER HAS FAILED
11H ; ECC CORRECTED DATA ERROR
3318
                               <1> DATA CORRECTED
3319
                                                     EQU
                                                         ; BAD ECC ON DISK READ
; NOT IMPLEMENTED
; BAD SECTOR FIAS DETER
3320
                               <1> BAD_ECC EQU 10H
                               <1> BAD_TRACK EQU 0BH <1> BAD_SECTOR EQU 0AH
3321
                               3322
                                                          , DAIA EXTENDS TOO FAR
; DRIVE PARAMETER ACTIVITY FAILED
3323
                               3324
3325
3326
3327
3328
                                <1>;BAD_CMD EQU 01H
                                                          ; BAD COMMAND PASSED TO DISK I/O
3329
                                <1>
                                <1> ;-----
3330
3331
                                <1>;
                               <1> ; FIXED DISK PARAMETER TABLE
3332
3333
                                <1> ; - THE TABLE IS COMPOSED OF A BLOCK DEFINED AS: :
3334
                               <1>;
                                <1>; +0 (1 WORD) - MAXIMUM NUMBER OF CYLINDERS
3335
                               <1>; +0 (1 WORD) - MAXIMUM NUMBER OF HEADS
3336
                                <1>; +3 (1 WORD) - NOT USED/SEE PC-XT
3337
                               <1> ; +5 (1 WORD) - STARTING WRITE PRECOMPENSATION CYL :
<1> ; +7 (1 BYTE) - MAXIMUM ECC DATA BURST LENGTH :
3338
3339
3340
                                <1> ; +8 (1 BYTE) - CONTROL BYTE
                               3341
3342
3343
                               <1> ; +9 (3 BYTES) - NOT USED/SEE PC-XT
<1> ; +12 (1 WORD) - LANDING ZONE
3344
                               3345
3346
                                     TO DYNAMICALLY DEFINE A SET OF PARAMETERS :
BUILD A TABLE FOR UP TO 15 TYPES AND PLACE :
THE CORRESPONDING VECTOR INTO INTERRIPT 41
FOR DRIVE 0 AND TYPES
3347
3348
                               <1>;
3349
3350
                               <1>;
                               <1> ;
3351
3352
                               <1>;
3353
                                <1> ;------
3354
3355
                                <1>
3356
                               <1> ;
3357
                                <1> ; HARDWARE SPECIFIC VALUES
3358
                               <1>;
3359
3360
                                <1>; - CONTROLLER I/O PORT
3361
                                <1>;
3362
                                <1>;
                                        > WHEN READ FROM:
                                        HF PORT+0 - READ DATA (FROM CONTROLLER TO CPU)
3363
                                <1>;
                                        HF_PORT+1 - GET ERROR REGISTER :
HF_PORT+2 - GET SECTOR COUNT :
3364
                               <1>;
3365
                               <1>;
                                        HF_PORT+3 - GET SECTOR NUMBER
HF_PORT+4 - GET CYLINDER LOW
3366
                               <1>;
3367
                               <1>;
                                        HF_PORT+5 - GET CYLINDER HIGH (2 BITS)
HF_PORT+6 - GET SIZE/DRIVE/HEAD :
3368
                                <1>;
                               <1>;
3369
                                         HF_PORT+7 - GET STATUS REGISTER
3370
                               <1>;
3371
                                <1>;
3372
                               <1>;
                                         > WHEN WRITTEN TO:
                                         HF PORT+0 - WRITE DATA (FROM CPU TO CONTROLLER) :
3373
                                <1>;
                                        HF_PORT+1 - SET PRECOMPENSATION CYLINDER : HF_PORT+2 - SET SECTOR COUNT :
3374
                               <1>;
                                        HF_PORT+3 - SET SECTOR NUMBER
HF_PORT+4 - SET CYLINDER LOW
HF_PORT+5 - SET_CYLINDER
3375
                                <1>;
3376
                               <1>;
3377
                               <1>;
                                        HF_PORT+5 - SET CYLINDER HIGH (2 BITS)
HF_PORT+6 - SET SIZE/DRIVE/HEAD :
3378
                                <1>;
                               <1>;
3379
3380
                                <1>;
                                        HF_PORT+7 - SET COMMAND REGISTER
3381
                                <1>;
3382
                                3383
                                <1>
                               <1>;HF_PORT EQU 01F0H; DISK PORT <1>;HF1_PORT equ 0170h
3384
3385
                                <1>; HF_REG_PORT EQU 03F6H
3386
                                <1>;HF1_REG_PORT equ 0376h
3387
3388
                                <1>
                               3389
3390
3391
                                <1>
                                <1> align 2
3392
3393
                                <1>
                                <1> ;----
                                                    STATUS REGISTER
3394
3395
                                <1>
3396
                                <1> ST ERROR
                                              EQU 0000001B
                                                     00000010B
3397
                                <1> ST_INDEX EQU
                                                                 ; ECC CORRECTION SUCCESSFUL
                                                     00000100B
3398
                                <1> ST_CORRCTD EQU
                                                     EQU 00001000B
                                <1> ST DRQ
3399
                                                                      ; SEEK COMPLETE
3400
                                <1> ST SEEK COMPL
                                                     EQU 00010000B
3401
                                <1> ST WRT FLT EQU
                                                     00100000B
                                                                ; WRITE FAULT
                                                     01000000B
                                <1> ST READY
3402
                                              EQU
                                                     10000000B
3403
                                <1> ST_BUSY
                                               EQU
3404
                               <1>
3405
                               <1> ;----
                                                     ERROR REGISTER
3406
                               <1>
                                <1> ERR DAM
                                               EQU
                                                     00000001B
                                                                 ; DATA ADDRESS MARK NOT FOUND
3407
                                                                ; TRACK 0 NOT FOUND ON RECAL
3408
                                <1> ERR TRK 0
                                                     00000010B
                                               EQU
                                                                ; ABORTED COMMAND
                                <1> ERR_ABORT
3409
                                                     00000100B
                                               EOU
                                                                ; NOT USED
3410
                                <1>;
                                                     00001000B
                                                     EOU 00010000B ; ID NOT FOUND
3411
                                <1> ERR ID
                                                     00100000B ; NOT USED
3412
                               <1>;
                                              EQU
                                <1> ERR DATA ECC EQU
3413
                                                     01000000B
                                <1> ERR_BAD_BLOCK
                                                     EQU 10000000B
3414
3415
                               <1>
3416
                               <1>
                                                               READ (20H); WRITE
                               <1> RECAL CMD
                                              EQU
                                                    00010000B
3417
3418
                                <1> READ CMD
                                               EQU
                                                     00100000B
                               <1> WRITE CMD
                                                     00110000B
3419
                                              EOU
                                                                ; VERIFY (40H)
3420
                               <1> VERIFY_CMD EQU
                                                     01000000B
3421
                               <1> FMTTRK CMD EQU
                                                     01010000B
                                                                ; FORMAT TRACK
                                                                                    (50H)
```

3317

```
EQU 01100000B ; INITIALIZE
EQU 01110000B ; SEEK (70H)
EQU 10010000B ; DIAGNOSTIC (90H)
3422
                                     <1> INIT CMD
                                                                                                   (60H)
                                     <1> SEEK_CMD EQU
<1> DIAG_CMD EQU
3423
3424
                                     <1> SET_PARM_CMD EQU 10010001B ; DRIVE PARMS (91H)
3425
                                                               00000001B ; CHD MODIFIER
3426
                                      <1> NO_RETRIES EQU
                                                                                                    (01H)
                                      <1> ECC_MODE EQU
3427
                                                               00000010B
                                                                             ; CMD MODIFIER
                                                                                                    (02H)
                                                               00001000B ; CMD MODIFIER
3428
                                     <1> BUFFER_MODE EQU
                                                                                                    (08H)
3429
                                     <1>
3430
                                     <1> ; MAX_FILE EQU
                                                               2
                                     <1>; S MAX FILE EQU
3431
                                                               2
                                                                         ; 22/12/2014
                                                               4
3432
                                      <1> MAX_FILE equ
                                      <1> S MAX_FILE equ
3433
                                                                             ; 22/12/2014
3434
                                     <1>
                                                                           ; DELAY FOR OPERATION COMPLETE
3435
                                     <1> DELAY 1
                                                        EQU
                                                              25H
                                     <1> DELAY_2
3436
                                                       EQU
                                                               0600H
                                                                          ; DELAY FOR READY
3437
                                      <1> DELAY_3
                                                       EQU
                                                              0100H
                                                                             ; DELAY FOR DATA REQUEST
3438
                                      <1>
                                                                             ; CMOS FLAG IN BYTE OEH
                                                       EQU 08H
                                     <1> HF_FAIL
3439
3440
                                      <1>
3441
                                     <1> ;----
                                                             COMMAND BLOCK REFERENCE
3442
                                     <1>
                                     <1> ; CMD_BLOCK EQU
3443
                                                                 BP-8
                                                                                      ; @CMD BLOCK REFERENCES BLOCK HEAD IN SS
                                                                             ; (BP) POINTS TO COMMAND BLOCK TAIL
3444
                                     <1>
                                                                             ; AS DEFINED BY THE "ENTER" PARMS
3445
                                      <1>
                                     <1> ; 19/12/2014
3446
                                     <1> ORG_VECTOR equ 4*13h ; INT 13h vector
<1> DISK_VECTOR equ 4*40h ; INT 40h vector (for floppy disks)
<1> ; HDISK_INT equ 4*76h ; Primary HDC - Hardware interrupt (IRQ14)
<1> ; HDISK_INT1 equ 4*76h ; Primary HDC - Hardware interrupt (IRQ14)
<1> ; HDISK_INT2 equ 4*77h ; Secondary HDC - Hardware interrupt (IRQ15)
<1> ; HF_TBL_VEC equ 4*41h ; Pointer to 1st fixed disk parameter table
<1> ; HF1_TBL_VEC equ 4*46h ; Pointer to 2nd fixed disk parameter table
                                      <1> ORG_VECTOR equ 4*13h
3447
3448
3449
3450
3451
3452
3453
3454
                                     <1>
3455
                                      <1> align 2
3456
                                      <1>
3457
                                      <1>; FIXED DISK I/O SETUP
3458
3459
                                      <1>;
                                      <1> ; - ESTABLISH TRANSFER VECTORS FOR THE FIXED DISK
3460
                                      <1>; - PERFORM POWER ON DIAGNOSTICS
3461
                                      <1>; SHOULD AN ERROR OCCUR A "1701" MESSAGE IS DISPLAYED
3462
3463
3464
                                     <1> ;------
3465
                                      <1>
3466
                                      <1>; 29/05/2016 - TRDOS 386 (TRDOS v2.0)
3467
                                      <1>
                                     <1> DISK SETUP:
3468
                                           ;CLI
3469
                                    ;xor ax,ax

;mov ds,ax

;mov ax, ax

;mov ax, [org_vector]

;mov [disk_vector], ax

;mov ax, [org_vector+2]

;mov [disk_vector+2]

;mov [disk_vector+2]

;mov [disk_vector+2]

;mov [disk_vector+2]

;mov [disk_vector+2]

;mov [org_vector+2]

;mov [org_vector+2]
                                     <1>
3470
                                                                                   ; GET ABSOLUTE SEGMENT
3471
3472
                                                                                   ; SET SEGMENT REGISTER
3473
                                                                              ; GET DISKETTE VECTOR
3474
                                                                                    ; INTO INT 40H
3475
3476
                                                ;MOV word [ORG_VECTOR], DISK_IO ; FIXED DISK HANDLER
3477
3478
                                            ; 1st controller (primary master, slave) - IRQ 14
3479
                                     <1>
                                               ;;MOV word [HDISK_INT],HD_INT ; FIXED DISK INTERRUPT ;mov word [HDISK_INT1],HD_INT ;
3480
                                     <1>
3481
                                     <1>
                                           ;mov [HDISK_INT+2],CS; 2nd controller (second;mov word)
3482
                                     <1>
3483
                                      <1>
                                     <1>
                                                ; 2nd controller (secondary master, slave) - IRQ 15
3484
3485
                                      <1>
                                                ;mov word [HDISK_INT2],HD1_INT ;
3486
                                     <1>
                                                ;mov [HDISK_INT2+2],CS
3487
                                     <1>
3488
                                     <1>
                                                ;; MOV word [HF TBL VEC], HDO DPT ; PARM TABLE DRIVE 80
3489
                                     <1>
                                                ;; MOV word [HF_TBL_VEC+2], DPT_SEGM
3490
                                     <1>
                                                ;; MOV word [HF1_TBL_VEC], HD1_DPT; PARM TABLE DRIVE 81
                                                ;;MOV word [HF1_TBL_VEC+2],DPT_SEGM
3491
                                     <1>
                                                ;push cs
3492
                                      <1>
3493
                                      <1>
                                                ;pop ds
                                                       word [HDPM_TBL_VEC], HDO_DPT
                                                                                            ; PARM TABLE DRIVE 80h
3494
                                     <1>
                                                ;mov
3495
                                     <1>
                                                       word [HDPM_TBL_VEC+2], DPT_SEGM
3496 00004236 C705[60590100]0000- <1>
                                                       dword [HDPM_TBL_VEC], (DPT_SEGM*16)+HD0_DPT
                                                mov
3496 0000423E 0900
                                     <1>
                                                                                            ; PARM TABLE DRIVE 81h
3497
                                     <1>
                                                      word [HDPS_TBL_VEC], HD1_DPT
                                                ; mov
3498
                                                ;mov word [HDPS_TBL_VEC+2],DPT_SEGM
                                     <1>
3499 00004240 C705[64590100]2000- <1>
                                                       dword [HDPS TBL VEC], (DPT SEGM*16)+HD1 DPT
                                                mov
3499 00004248 0900
                                    <1>
                                                ;mov word [HDSM_TBL_VEC],HD2_DPT
                                                                                            ; PARM TABLE DRIVE 82h
3500
                                     <1>
3501
                                      <1>
                                                ;mov
                                                       word [HDSM_TBL_VEC+2], DPT_SEGM
                                                        dword [HDSM_TBL_VEC], (DPT_SEGM*16)+HD2_DPT
3502 0000424A C705[68590100]4000- <1>
                                                mov
3502 00004252 0900
                                                ;mov word [HDSS TBL VEC], HD3 DPT
                                                                                            ; PARM TABLE DRIVE 83h
3503
                                     <1>
                                                       word [HDSS_TBL_VEC+2], DPT_SEGM
3504
                                     <1>
                                                ; mov
3505 00004254 C705[6C590100]6000- <1>
                                                       dword [HDSS_TBL_VEC], (DPT_SEGM*16)+HD3_DPT
                                                mov
3505 0000425C 0900
                                     <1>
3506
                                     <1>
3507
                                     <1>
                                                ;;IN AL,INTB01 ; TURN ON SECOND INTERRUPT CHIP
3508
                                                ;;;AND AL,OBFH
                                     <1>
3509
                                     <1>
                                                ;;and al, 3Fh
                                                                                   ; enable IRQ 14 and IRQ 15
                                                ;;;JMP $+2
3510
                                     <1>
3511
                                     <1>
                                                ;;IODELAY
                                                ;;OUT INTB01,AL
3512
                                     <1>
3513
                                     <1>
                                                ;;IODELAY
                                                                         ; LET INTERRUPTS PASS THRU TO
3514
                                     <1>
                                                ;;IN AL,INTA01
                                                                          ; SECOND CHIP
3515
                                     <1>
                                                ;;AND AL,OFBH
3516
                                     <1>
                                                ;;;JMP $+2
3517
                                     <1>
                                                ::IODELAY
                                                ;;OUT INTA01,AL
3518
                                     <1>
3519
                                     <1>
                                                ;STI
3520
                                     <1>
                                                                         ; MOVE ABSO POINTER TO ; EXTRA SEGMENT POINTER
3521
                                     <1>
                                                ;; PUSH DS
3522
                                     <1>
                                                ;;POP ES
```

```
; ESTABLISH DATA SEGMENT
; RESET THE STATUS INDICATOR
; ZERO NUMBER OF FIXED DISKS
                                        ;;;CALL
3523
                                 <1>
                                                     DDS
3524
                                 <1>
                                          ;;MOV byte [DISK STATUS1],0
                                          ;;MOV byte [HF_NUM],0
3525
                                 <1>
3526
                                 <1>
                                        ;;MOV byte [CONTROL_BYTE],0
3527
                                 <1>
                                         ;; MOV byte [PORT OFF], 0 ; ZERO CARD OFFSET
                                         ; 20/12/2014 - private code by Erdogan Tan
; (out of original PC-AT, PC-XT BIOS code)
3528
                                 <1>
3529
                                 <1>
                                          ;mov si, hd0_type
3530
                                <1>
                                         mov esi, hd0_type; mov cx, 4
3531 0000425E BE[585D0000]
                                <1>
3532
                                <1>
3533 00004263 B904000000
                                <1> mov ecx, 4
                                <1> hde l:
3534
                                <1> lodsb
3535 00004268 AC
                                          cmp al, 80h
3536 00004269 3C80
                                <1>
                                                                        ; 8?h = existing
3537 0000426B 7206
                                <1>
                                                short L4
                               <1> jb short L4
<1> inc byte [HF_NUM] ; + 1 hard (fixed) disk drives
                                          jb
3538 0000426D FE05[5C590100]
                                <1> _L4: ; 26/02/2015
<1> loop hde_1
3540 00004273 E2F3
                                                                   ; 0 <= [HF NUM] =< 4
3541
                                 <1> ;_L4:
                                 <1> ; L4:
3542
3543
                                 <1>
3544
                                 <1>
                                          ;; 31/12/2014 - cancel controller diagnostics here
                                          ;;;mov cx, 3 ; 26/12/2014 (Award BIOS 1999)
3545
                                 <1>
3546
                                 <1>
3547
                                 <1>
                                          ;;
                                          ;;MOV DL,80H
                                                                  ; CHECK THE CONTROLLER
3548
                                 <1>
3549
                                 <1> ;;hdc_dl:
                                      ;; MOV AH,14H ; USE CONTROLLER DIAGNOSTIC COMMAND ::INT 13H : CALL BIOS WITH DIAGNOSTIC COMMAND
3550
                                 <1>
                                         ;;INT 13H ; CALL BIOS WITH DIAGNOSTIC COMMAND ;;;JC short CTL_ERRX ; DISPLAY ERROR MESSAGE IF BAD RETURN
3551
                                 <1>
                                      ;;;JC short CTL_ERRX
;;;jc short POD_DONE ;22/12/2014
3552
                                 <1>
3553
                                 <1>
                                        ;;jnc short hdc_reset0
3554
                                 <1>
                                          ;;loop hdc_dl
3555
                                 <1>
3556
                                 <1>
                                        ;;; 27/12/2014
                                        ;;stc
3557
                                 <1>
                                        ;;retn
;
3558
                                 <1>
3559
                                 <1>
                                <1> ;;hdc_reset0:
3560
                                      ; 18/01/2015
3561
                                <1>
3562 00004275 8A0D[5C590100]
                                <1>
                                          mov cl, [HF_NUM]
3563 0000427B 20C9
                                <1>
                                          and cl, cl
3564 0000427D 740E
                                <1>
                                          jz short POD_DONE
3565
                                <1>
                                          ;
                                      mov
                                                dl, 7Fh
3566 0000427F B27F
                                <1>
                                <1> hdc_reset1:
3567
                                      inc dl
3568 00004281 FEC2
                                <1>
                                          ;; 31/12/2015
3569
                                <1>
3570
                                <1>
                                        ;;push dx
3571
                                <1>
                                          ;;push cx
3572
                                 <1>
                                          ;;push ds
3573
                                 <1>
                                          ;;sub ax, ax
3574
                                 <1>
                                          ;;mov ds, ax
                                                                   ; GET START TIMER COUNTS
                                          ;;MOV AX, [TIMER_LOW]
3575
                                 <1>
3576
                                 <1>
                                         ;;pop ds
                                        ;;MOV BX,AX
3577
                                 <1>
                                          ;;ADD AX,6*182 ; 60 SECONDS* 18.2
3578
                                 <1>
                                          ;; MOV CX, AX
3579
                                 <1>
                                 <1>
                                          ;;mov word [wait_count], 0 ; 22/12/2014 (reset wait counter)
3580
3581
                                 <1>
                                          ;;
                                          ;; 31/12/2014 - cancel HD_RESET 1
3582
                                 <1>
3583
                                 <1>
                                          ;;CALL HD_RESET_1 ; SET UP DRIVE 0, (1,2,3)
                                          ;;pop cx
3584
                                 <1>
3585
                                 <1>
                                          ;;pop dx
3586
                                 <1>
                                          ;;
                                          ; 18/01/2015
3587
                                <1>
3588 00004283 B40D
                                <1>
                                          mov ah, ODh ; ALTERNATE RESET
                                <1>
                                          ;int 13h
3590 00004285 E8A3FFFFF
                                <1>
                                          call int13h
3591 0000428A E2F5
                                <1>
                                          loop hdc_reset1
3592 0000428C F8
                                <1>
                                         clc ; 29/05/2016
3593
                                <1> POD DONE:
3594 0000428D C3
                                 <1>
3595
                                 <1>
3596
                                 <1> ;;----
                                                POD_ERROR
3597
                                 <1>
                                 <1> ;; CTL ERRX:
3598
                                 3599
                                 <1>;
                                        ;CALL SET_FAIL ; DO NOT IPL FROM DISK
3600
                                                                   ; DISPLAY ERROR AND SET (BP) ERROR FLAG
3601
                                 <1>;
                                          ;CALL E_MSG
                                 <1> ;
3602
                                          ;JMP short POD_DONE
3603
                                 <1>
3604
                                 <1> ;; HD_RESET_1:
                                 <1>;;
3605
                                          ; PUSH BX
                                                                   ; SAVE TIMER LIMITS
3606
                                        ; PUSH CX
                                 <1> ;;
                                                                  ; SET DRIVE PARAMETERS
3607
                                 <1> ;; RES_1: MOV AH, 09H
3608
                                 <1> ;; INT 13H
3609
                                 <1> ;;
                                        JC short RES 2
                                                                 ; RECALIBRATE DRIVE
3610
                                 <1>;; MOV AH,11H
3611
                                 <1> ;;
                                          INT
                                                13H
                                        JNC short RES_CK
                                 <1> ;; JNC short RES_CK ; DRIVE OK
<1> ;;RES_2: ;CALL POD_TCHK ; CHECK TIME OUT
3612
3613
                                 <1> ;;    cmp word [wait_count], 6*182 ; waiting time (in timer ticks)
3614
                                 <1> ;;
                                                                   ; (30 seconds)
3615
3616
                                 <1> ;;
                                 <1>;; ;JNC short RES_1 <1>;; jb short RES_1
3617
3618
                                 <1>;;;RES FL: ;MOV SI,OFFSET F1781 ; INDICATE DISK 1 FAILURE;
3619
                                 <1> ;;  ;TEST DL,1
3620
3621
                                 <1> ;;
                                          ;JNZ RES_E1
                                 <1>;; ;UNZ KES_E1
<1>;; ;MOV SI,OFFSET F1780
                                                                   ; INDICATE DISK 0 FAILURE
3622
                                 ; DO NOT TRY TO IPL DISK 0
3623
3624
                                 <1> ;;
                                          ;JMP SHORT RES E1
                                 <1>;; RES ER: ; 22/12/2014
3625
3626
                                 <1> ;; RES OK:
3627
                                 ; RESTORE TIMER LIMITS
```

```
3628
                                 <1> ;;
                                          ; POP BX
3629
                                 <1> ;;
                                          RETn
3630
                                 <1> ;;
3631
                                 <1> ;; RES_RS: MOV
                                                      AH,00H
                                                                          ; RESET THE DRIVE
                                 <1> ,, NEC_10. 
<1> ;; INT 13H
3632
                                                       AH,08H ; GEI IIIII ; SAVE DRIVE CODE
3633
                                 <1> ;; RES_CK: MOV
                                                                         ; GET MAX CYLINDER, HEAD, SECTOR
                                 <1> ;; MOV BL,DL
3634
                                                13H
3635
                                          INT
                                 <1> ;;
3636
                                 <1> ;;
                                          JC
                                                 short RES_ER
                                 <1> ;;
                                                [NEC_STATUS],CX ; SAVE MAX CYLINDER, SECTOR
3637
                                          MOV
                                 3638
3639
                                                                   ; VERIFY THE LAST SECTOR
                                 <1> ;; INT 13H
3640
                                                                ; VERIFY OK
; OK ALSO IF JUST ID READ
3641
                                 <1> ;;
                                                short RES_OK
                                 <1> ;; CMP
<1> ;; JE
3642
                                                AH, BAD_SECTOR
3643
                                                short RES_OK
                                 <1> ;;
                                          CMP AH, DATA CORRECTED
3644
                                 <1> ;;
                                          JE
                                                short RES_OK
3645
3646
                                 <1> ;;
                                          CMP
                                                AH,BAD_ECC
                                          JE short RES_OK
                                 <1> ;;
3647
                                          ;CALL POD_TCHK _
                                                                    ; CHECK FOR TIME OUT
3648
                                 <1> ;;
                                         cmp word [wait_count], 6*182; waiting time (in timer ticks)
                                 <1> ;;
3649
                                 <1> ;;
                                                                    ; (60 seconds)
3650
3651
                                 <1> ;;
                                                short RES_ER ; FAILED
CX,[NEC_STATUS] ; GET SECTOR ADDRESS, AND CYI
AL,CL ; SEPARATE OUT SECTOR NUMBER
3652
                                 <1> ;;
                                          JC
3653
                                 <1> ;;
                                          VOM
                                                                    ; GET SECTOR ADDRESS, AND CYLINDER
3654
                                 <1> ;;
                                          VOM
                                          JZ short RES_RS ; WE'VE TRIED ALL SECTORS ON TRACK
AND CL,0C0H ; KEEP CYLINDER BITS
OR CL,AL ; MERGE SECTOR MITTER
MOV [NEC CT.]
3655
                                 <1> ;;
3656
                                 <1> ;;
                                 <1> ;;
3657
3658
                                 <1> ;;
                                 <1> ;; OR CL,AL ; MERGE SECTOR WITH CYLINDER BITS
<1> ;; MOV [NEC_STATUS],CX ; SAVE CYLINDER, NEW SECTOR NUMBER
<1> ;; JMP short RES_3 ; TRY AGAIN
3659
3660
3661
3662
                                 <1> ;;;RES_ER: MOV SI,OFFSET F1791 ; INDICATE DISK 1 ERROR
                                 <1>;; ;TEST DL,1 <1>;; ;JNZ short
3663
                                 <1> ;;
                                          ;JNZ short RES E1
3664
                                 <1>;; ;MOV SI,OFFSET F1790 ; INDICATE DISK 0 ERROR
3665
3666
                                 <1> ;;;RES_E1:
3667
                                 <1> ;; ; ; CALL E_MSG
                                                                  ; DISPLAY ERROR AND SET (BP) ERROR FLAG
3668
                                 <1> ;;; RES_OK:
                                 <1>;; ;POP
3669
                                                CX
                                                                    ; RESTORE TIMER LIMITS
                                          ; POP BX
                                 <1> ;;
3670
                                         ;RETn
3671
                                 <1> ;;
3672
                                 <1>;
                                 <1> ;; SET_FAIL:
3673
                                 3674
                                        ;CALL CMOS_READ
;OR AL, HF_FAIL
;XCHG AL AT
3675
                                       ;OR AL,HF_FAIL ; SET DO NOT IPL FROM DISK FLAG
;XCHG AH,AL ; SAVE IT
;CALL CMOS_WRITE ; PUT IT OUT
;RETn
                                 <1>;
3676
                                 <1>;
                                 <1>;
3677
3678
                                 <1>;
3679
                                 <1>;
                                          ;RETn
3680
                                 <1>;
3681
                                 <1> ;; POD_TCHK:
                                                                  ; CHECK FOR 30 SECOND TIME OUT
                                 <1>; ; POP AX <1>; POP CX
3682
                                                                 ; SAVE RETURN
3683
                                                                    ; GET TIME OUT LIMITS
                                          ; POP BX
3684
                                 <1>;
                                        ; PUSH BX
                                                                  ; AND SAVE THEM AGAIN
3685
                                 <1>;
                                        ; PUSH CX ; PUSH AX
3686
                                 <1>;
3687
                                 <1>;
3688
                                 <1>;
                                        ;push ds
                                        ;xor ax, ax
3689
                                 <1>;
                                          ;mov ds, ax
3690
                                 <1>;
                                                                    ; RESTORE RETURN
                                          ;MOV AX, [TIMER_LOW]
3691
                                 <1>;
                                                                    ; AX = CURRENT TIME
3692
                                 <1>;
                                                                    ; BX = START TIME
                                          ;
3693
                                 <1>;
                                                                    ; CX = END TIME
3694
                                 <1>;
                                          ;pop ds
3695
                                 <1>;
                                        ; CMP BX, CX
3696
                                 <1>;
                                                short TCHK1
                                                                    ; START < END
                                          ;JB
3697
                                 <1>;
                                          ; CMP BX, AX
                                                               ; END < START < CURRENT
                                          ;JB short TCHKG
3698
                                 <1>;
                                          ; JMP SHORT TCHK2
3699
                                 <1>;
                                                                    ; END, CURRENT < START
                                 <1> ;; TCHK1: CMP AX, BX
3700
                                                                    ; CURRENT < START < END
3701
                                 <1> ;; JB short TCHKNG
3702
                                 <1> ;; TCHK2: CMP AX, CX
                                 <1> ;; JB short TCHKG
3703
                                                                   ; START < CURRENT < END
                                                                    ; OR CURRENT < END < START
3704
                                 <1> ;;
                                 <1> ;; TCHKNG: STC
3705
                                                                          ; CARRY SET INDICATES TIME OUT
                                 <1> ;; RETn
3706
3707
                                 <1> ;; TCHKG: CLC
                                                                    ; INDICATE STILL TIME
3708
                                 <1> ;;
                                         RETn
3709
                                 <1> ;;
3710
                                 <1> ;;int_13h:
3711
3712
                                 <1> ;------
                                 <1>; FIXED DISK BIOS ENTRY POINT
3713
3714
3715
                                 <1>
3716
                                 <1>; 30/08/2020
                                 <1>; 29/08/2020
3717
3718
                                 <1> ; 15/01/2017
3719
                                 <1> ; 14/01/2017
                                 <1> ; 07/01/2017
3720
3721
                                 <1>; 02/01/2017
3722
                                 <1>; 01/06/2016
                                 <1>; 16/05/2016, 27/05/2016, 28/05/2016, 29/05/2016
3723
                                 <1>; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
3724
3725
                                 <1> int33h: ; DISK I/O
3726
                                 <1>
                                          ; 29/05/2016
                                          and byte [esp+8], 111111110b ; clear carry bit of eflags register
3727 0000428E 80642408FE
                                <1>
                                        ; 16/05/2016
3728
                                <1>
                                        push ds
3729 00004293 1E
                                <1>
3730 00004294 53
                                <1>
                                          push ebx ; user's buffer address (virtual)
3731 00004295 66BB1000
                                <1>
                                          mov bx, KDATA; System (Kernel's) data segment
3732 00004299 8EDB
                                 <1>
                                          mov
                                                ds, bx
```

```
3733
                                 <1>
3734
                                 <1>
                                          ;;15/01/2017
3735
                                          ; 14/01/2017
                                 <1>
3736
                                 <1>
                                          ; 02/01/2017
3737
                                 <1>
                                          ;;mov byte [intflg], 33h ; disk io interrupt
3738
                                 <1>
                                          ;pop ebx
3739
                                          ;mov [user buffer], ebx
                                 <1>
3740
                                <1>
3741 0000429B 8F05[4C650100]
                                <1>
                                                dword [user_buffer] ; 01/06/2016
                                          pop
3742
                                <1>
3743 000042A1 C605[865E0100]00
                                <1>
                                          mov
                                                byte [scount], 0 ; sector count for transfer
3744 000042A8 80FC03
                                                ah, 03h; chs write
                                <1>
                                          cmp
                                                short int33h 2
3745 000042AB 7744
                                <1>
                                          jа
3746 000042AD 7407
                                <1>
                                                short int33h_0
                                          jе
3747 000042AF 80FC02
                                <1>
                                                ah, 02h ; chs read
                                          cmp
3748 000042B2 726F
                                <1>
                                          jb
                                                short int33h_5
3749 000042B4 EB68
                                <1>
                                          jmp
                                                short int33h 4
                                <1> int33h_0:
3750
3751
                                <1>
                                          ; transfer user's buffer content to sector buffer
3752 000042B6 51
                                          push ecx
                                <1>
3753 000042B7 0FB6C8
                               <1>
                                          movzx ecx, al
                                <1> int33h 1:
3755 000042BA 56
                                <1>
                                         push esi
3756 000042BB 8B35[4C650100] <1>
                                          mov
                                                esi, [user_buffer]
3757
                                <1>
                                          ; esi = user's buffer address (virtual, ebx)
3758 000042C1 57
                                <1>
                                          push edi
                                         push es
3759 000042C2 06
                                <1>
                                        push eax
3760 000042C3 50
                                <1>
                                                ax, KDATA
3761 000042C4 66B81000
                                <1>
                                          mov
3762 000042C8 8EC0
                                <1>
                                                es, ax
                                          mov
3763 000042CA BF00000700
                                <1>
                                          mov
                                                edi, Cluster_Buffer
3764 000042CF C1E109
                                <1>
                                          shl
                                                ecx, 9 ; * 512
3765 000042D2 E870A50000
                                          call transfer_from_user_buffer
                                <1>
3766 000042D7 58
                                <1>
                                          pop
3767 000042D8 07
                                <1>
                                          pop
                                                es
3768 000042D9 5F
                                <1>
                                                edi
                                          pop
3769 000042DA 5E
                                <1>
                                                esi
                                          pop
3770 000042DB 59
                                <1>
                                          pop
                                                ecx
                                                short int33h 5
3771 000042DC 7345
                                <1>
                                          jnc
3772 000042DE 8B1D[4C650100]
                                                ebx, [user_buffer] ; 01/06/2016
                                <1>
                                          mov
3773 000042E4 1F
                                <1>
                                         pop
3774
                                 <1>
                                          ;;15/01/2017
3775
                                <1>
                                          ; 02/01/2017
3776
                                 <1>
3777
                                 <1>
                                          ;cli
3778
                                          ;;mov byte [ss:intflg], 0 ; 07/01/2017
                                 <1>
3779
                                 <1>
3780
                                <1>
                                          ; (*) 29/05/2016
3781
                                 <1>
                                          ; (*) retf 4 ; skip eflags on stack
3782
                                <1>
                                          ; 29/05/2016 -set carry flag on stack-
3783
                                <1>
3784
                                 <1>
                                          ; [esp] = EIP
                                          ; [esp+4] = CS
3785
                                <1>
                                          ; [esp+8] = E-FLAGS
3786
                                <1>
3787 000042E5 804C240801
                                <1>
                                          or byte [esp+8], 1 ; set carry bit of eflags register
3788
                                <1>
                                          ; [esp+12] = ESP (user)
                                          ; [esp+16] = SS (User)
3789
                                <1>
3790 000042EA B8FF000000
                                <1>
                                                eax, OFFh ; Unknown error !?
                                          mov
3791
                                <1>
                                          ;iretd
3792 000042EF EB7E
                                <1>
                                               short int33h_7 ; 07/01/2017
                                          jmp
3793
                                <1>
3794
                                          ; (*) 29/05/2016 - 'ref 4' intruction causes to stack fault
                                 <1>
3795
                                 <1>
                                          ; (OUTER-PRIVILEGE-LEVEL)
3796
                                 <1>
                                          ; INTEL 80386 PROGRAMMER'S REFERENCE MANUAL 1986
3797
                                 <1>
                                          ; // RETF instruction:
3798
                                 <1>
3799
                                 <1>
                                          ; IF OperandMode=32 THEN
3800
                                <1>
                                         ; Load CS:EIP from stack;
3801
                                 <1>
                                               Set CS RPL to CPL;
3802
                                <1>
                                              Increment eSP by 8 plus the immediate offset if it exists;
3803
                                 <1>
                                             Load SS:eSP from stack;
                                          ; ELSE (* OperandMode=16 *)
3804
                                 <1>
                                          ; Load CS: IP from stack;
3805
                                <1>
                                               Set CS RPL to CPL;
3806
                                <1>
3807
                                 <1>
                                               Increment eSP by 4 plus the immediate offset if it exists;
                                          ;
3808
                                 <1>
                                          ;
                                               Load SS:eSP from stack;
3809
                                 <1>
                                          ; FI;
3810
                                 <1>
3811
                                 <1>
3812
                                 <1>
3813
                                 <1> int33h 2:
3814 000042F1 80FC05
                                                ah, 05h ; format track
                                 <1>
                                          cmp
3815 000042F4 770A
                                <1>
                                                 short int33h_3
                                          jа
3816 000042F6 722B
                                <1>
                                          jb
                                                 short int33h_5
3817 000042F8 51
                                <1>
                                          push ecx
3818 000042F9 B901000000
                                <1>
                                          mov
                                                 ecx, 1
                                                short int33h 1
3819 000042FE EBBA
                               <1>
                                          jmp
3820
                               <1> int33h_3:
3821 00004300 80FC1C
                               <1>
                                                ah, 1Ch ; LBA write
                                          cmp
                               <1>
                                                short int33h 5
3822 00004303 771E
                                          jа
3823 00004305 74AF
                               <1>
                                          jе
                                                short int33h_0
3824 00004307 80FC1B
                                <1>
                                          cmp
                                                ah, 1Bh ; LBA read
3825 0000430A 7412
                                                short int33h_4
                                <1>
                                          jе
                                          ; 29/08/2020
                                <1>
                                               ah, 08h ; get disk parameters
3827 0000430C 80FC08
                                <1>
                                          cmp
                                                short int33h 5
3828
                                <1>
                                          ;jne
3829 0000430F 7405
                                <1>
                                                short int33h 10
                                          jе
3830 00004311 80FC15
                                <1>
                                                ah, 15h ; read DASD type (get disk size)
                                          cmp
3831 00004314 750D
                                <1>
                                          jne
                                                 short int33h_5
                                <1> int33h 10:
3832
3833
                                <1>
                                          ; 01/06/2016
3834 00004316 8B1D[4C650100]
                                <1>
                                          mov
                                                ebx, [user buffer]; user's buffer address
3835 0000431C EB0A
                                <1>
                                          jmp
                                                short int33h_6
                                <1> int33h 4:
3837 0000431E A2[865E0100]
                                <1>
                                         mov
                                                byte [scount], al ; <= 128 sectors</pre>
```

```
3839 00004323 BB00000700
                                <1>
                                        mov
                                              ebx, Cluster_Buffer; max. 65536 bytes
3840
                               <1>
                                                             ; buf. addr: 70000h
3841
                               <1>
                                              byte [ClusterBuffer_Valid], 0
3842
                               <1> int.33h 6:
3843 00004328 1F
                               <1>
                                        pop
3844 00004329 9C
                               <1>
                                         pushfd
3845 0000432A 0E
                               <1>
                                        push cs
3846 0000432B E84D000000
                               <1>
                                               DISK IO
                                        call
3847 00004330 2E8B1D[4C650100] <1>
                                        mov
                                               ebx, [CS:user_buffer]; 01/06/2016
3848 00004337 723D
                               <1>
                                               short int33h_9
3849
                                <1>
                                        ;
3850 00004339 2E803D[865E0100]00 <1>
                                               byte [CS:scount], 0
                                        cmp
3851 00004341 762C
                                       jna
                               <1>
                                               short int33h_7
3852
                               <1>
3853
                               <1>
                                         ; transfer sector buffer content to user's buffer
3854 00004343 06
                                        push es
                               <1>
3855 00004344 1E
                               <1>
                                         push ds
3856 00004345 50
                               <1>
                                        push eax
3857 00004346 66B81000
                              <1>
                                        mov ax, KDATA
3858 0000434A 8ED8
                               <1>
                                        mov ds, ax
3859 0000434C 8EC0
                               <1>
                                        mov
                                              es, ax
3860 0000434E 51
                               <1>
                                        push ecx
                                     push esi
3861 0000434F 56
                               <1>
                                      push edi
3862 00004350 57
                               <1>
3863 00004351 0FB60D[865E0100] <1>
                                        movzx ecx, byte [scount]
3864 00004358 C1E109
                               <1>
                                        shl ecx, 9; * 512 bytes
3865 0000435B 89DF
                               <1>
                                        mov
                                              edi, ebx ; user's buffer address
3866 0000435D BE00000700
                               <1>
                                        mov
                                               esi, Cluster_Buffer
                                        call transfer_to_user_buffer
3867 00004362 E896A40000
                              <1>
3868 00004367 5F
                               <1>
                                              edi
                                        pop
3869 00004368 5E
                               <1>
                                               esi
                                        pop
3870 00004369 59
                               <1>
                                         pop
                                               ecx
3871 0000436A 58
                              <1>
                                        pop
                                               eax
3872 0000436B 1F
                               <1>
                                               ds
                                         pop
3873 0000436C 07
                               <1>
                                               es
                                         pop
                                               short int33h 8
3874 0000436D 7202
                              <1>
                                         jс
                               <1> int33h_7:
3875
3876 0000436F FA
                               <1>
                                        cli
                                         ;;15/01/2017
3877
                               <1>
3878
                               <1>
                                        ;;mov byte [ss:intflg], 0 ; 07/01/2017
                                        ; cf = 0 ; use eflags which is in stack
3879
                               <1>
3880 00004370 CF
                               <1>
                                        iretd
3881
                               <1> int33h 8:
3882 00004371 B8FF000000
                               <1>
                                        mov
                                               eax, OFFh; Unknown error!?
3883
                                <1> int33h_9:
3884
                                <1>
                                       ; cf = 1
3885
                                <1>
                                        ; (*) 29/05/2016
3886
                                <1>
                                        ; (*) retf 4 ; skip eflags on stack
3887
                                <1>
                                        ; Note: This 'retf 4' was wrong, -it was causing
3888
                                <1>
3889
                                <1>
                                       ;
                                                to stack errors in ring 3-
                                               POP sequence of 'retf 4' is as
3890
                                <1>
                                         ;
3891
                                <1>
                                               "eip, cs, eflags, esp, ss, +4 bytes"
3892
                                <1>
                                                 it is not as "eip, cs, +4 bytes, esp, ss" !
3893
                                <1>
3894
                                <1>
                                         ; 29/05/2016 -set carry flag on stack-
3895 00004376 804C240801
                                         or byte [esp+8], 1 ; set carry bit of eflags register
                                <1>
3896
                                <1>
                                         ;iretd
3897 0000437B EBF2
                                              short int33h_7 ; 07/01/2017
                                <1>
                                         jmp
3898
                                <1>
3899
                                <1>; 30/08/2020
3900
                                <1>; 09/12/2017
3901
                                <1> ; 29/05/2016
3902
                                <1> ; 27/05/2016 - TRDOS 386 (TRDOS v2.0)
3903
                                <1>
3904
                                <1> DISK IO:
3905 0000437D 80FA80
                               <1>
                                        CMP DL,80H
                                                                 ; TEST FOR FIXED DISK DRIVE
                                         ;JAE short A1
                                                                 ; YES, HANDLE HERE
3906
                                <1>
                                         ;;;INT 40H
3907
                                <1>
                                                                 ; DISKETTE HANDLER
3908
                                <1>
                                        ;;call int40h
3909 00004380 0F82FFEFFFF
                                <1>
                                             DISKETTE_IO_1
                                         jb
                                <1> ; RET_2:
3910
3911
                                <1>
                                        ;RETf 2
                                                                 ; BACK TO CALLER
3912
                                <1>;
                                        retf 4
                                <1> A1:
3913
3914 00004386 FB
                                <1>
                                                                  ; ENABLE INTERRUPTS
                                        ;; 04/01/2015
3915
                                <1>
3916
                                <1>
                                         ;;OR AH,AH
3917
                                         ;;JNZ short A2
                                <1>
                                                                  ; RESET NEC WHEN AH=0
3918
                                <1>
                                         ;;INT 40H
                                         ;;SUB AH,AH
3919
                                <1>
                                              DL, (80H + S_MAX_FILE - 1) ; 83h ; 30/08/2020
3920 00004387 80FA83
                              <1>
                                         CMP
3921
                               <1>
                                         ;JA short RET 2
                                               short _A0
                                         jna
3922 0000438A 7614
                               <1>
                                         ; 29/05/2016
3923
                               <1>
                                        push ds
3924 0000438C 1E
                              <1>
                              <1>
3925 0000438D 50
                                        push eax; 30/08/2020 (ax --> eax)
3926 0000438E 66B81000
                              <1>
                                               ax, KDATA
                                         mov
3927 00004392 8ED8
                              <1>
                                              ds, ax
                                        mov
3928 00004394 58
                              <1>
                                       pop eax ;
                                        mov ah, OAAh
3929 00004395 B4AA
                               <1>
                                                               ; Hard disk drive not ready !
                                                  ; (Programmer's guide to AMIBIOS, 1992)
3930
                               <1>
3931 00004397 8825[5B590100] <1>
                                               byte [DISK_STATUS1], ah
                                              ds
3932 0000439D 1F
                               <1>
                                         pop
                                              short RET 2
3933 0000439E EB38
                               <1>
                                        jmp
                               <1> _A0:
3935
                                        ; 18/01/2015
                               <1>
3936 000043A0 08E4
                               <1>
                                         or ah, ah
3937 000043A2 743A
                                               short A4
                               <1>
                                         jz
                              <1>
3938 000043A4 80FC0D
                                         cmp ah, ODh; Alternate reset
3939 000043A7 7504
                               <1>
                                         jne
                                              short A2
                               <1>
3940 000043A9 28E4
                                         sub
                                              ah,ah ; Reset
3941 000043AB EB31
                                         jmp short A4
                               <1>
                               <1> A2:
```

<1> int33h 5:

3838

```
CMP AH,08H
3943 000043AD 80FC08
                                <1>
                                                                    ; GET PARAMETERS IS A SPECIAL CASE
3944
                                 <1>
                                          ;JNZ short A3
                                           ;JMP GET PARM N
3945
                                <1>
                                <1>
3946 000043B0 0F8452030000
                                           je GET_PARM_N
                                          CMP AH,15H;JNZ short A4
3947 000043B6 80FC15
                                <1> A3: CMP
                                                                    ; READ DASD TYPE IS ALSO
                                <1>
3948
                                          ;JMP READ_DASD_TYPE
je READ_DASD_TYPE
3949
                                <1>
3950 000043B9 0F84DB020000
                                      je READ; 02/02/2015
cmp ah, 1Dh
; 12/01/2015
                                <1>
3951
                                <1>
3952 000043BF 80FC1D
                                <1>
                                                                           ; (Temporary for Retro UNIX 386 v1)
3953
                                <1>
3954 000043C2 F5
                                 <1>
                                          cmc
3955 000043C3 7319
                                <1>
                                          jnc
                                                 short A4
3956
                                <1> int33h_bad_cmd:
                                      ; 16/05/2016
3957
                                <1>
                                          ; 30/01/2015
3958
                                <1>
                                          ; 29/05/2016
3959
                                <1>
3960 000043C5 1E
                                        push ds
                                <1>
3961 000043C6 6650
                                <1>
                                          push ax
3962 000043C8 66B81000
                                <1>
                                                ax, KDATA
                                          mov
3963 000043CC 8ED8
                                <1>
                                          mov
                                                ds, ax
3964 000043CE 6658
                                 <1>
                                          pop
                                                ax
                                                ah, BAD CMD
3965 000043D0 B401
                                <1>
                                          mov
                                          mov [DISK_STATUS1], ah ; BAD_CMD ; COMMAND ERROR
3966 000043D2 8825[5B590100]
                                <1>
3967
                                 <1>
                                           ;jmp short RET_2
                                 <1> RET_2:
3968
                                      -
; (*) 29/05/2016
3969
                                 <1>
3970
                                 <1>
                                          ; (*) retf 4
3971 000043D8 804C240801
                                 <1>
                                                byte [esp+8], 1; set carry bit of eflags register
3972 000043DD CF
                                 <1>
                                          iretd
3973
                                 <1> A4:
                                                                    ; SAVE REGISTERS DURING OPERATION
3974 000043DE C8080000
                                                                    ; SAVE (BP) AND MAKE ROOM FOR @CMD BLOCK
                                 <1>
                                           ENTER 8,0
3975 000043E2 53
                                           PUSH eBX
                                                                    ; IN THE STACK, THE COMMAND BLOCK IS:
                                <1>
3976 000043E3 51
                                <1>
                                           PUSH eCX
                                                                    ; @CMD BLOCK == BYTE PTR [BP]-8
3977 000043E4 52
                                <1>
                                          PUSH eDX
3978 000043E5 1E
                                <1>
                                           PUSH DS
3979 000043E6 06
                                <1>
                                           PUSH ES
3980 000043E7 56
                                          PUSH eSI
                                <1>
3981 000043E8 57
                                <1>
                                           PUSH
                                                eDI
                                          ;;04/01/2015
3982
                                <1>
3983
                                <1>
                                          ;;OR AH,AH
                                                                    ; CHECK FOR RESET
3984
                                 <1>
                                          ;;JNZ short A5
3985
                                                                    ; FORCE DRIVE 80 FOR RESET
                                 <1>
                                          ;;MOV DL,80H
3986
                                 <1> ;; A5:
                                          ;push cs
3987
                                 <1>
3988
                                 <1>
                                          ;pop ds
                                          ; 21/02/2015
3989
                                <1>
3990 000043E9 6650
                                <1>
                                          push ax
3991 000043EB 66B81000
                                <1>
                                          mov
                                                ax, KDATA
3992 000043EF 8ED8
                                <1>
                                                ds, ax
                                          mov
3993 000043F1 8EC0
                                 <1>
                                          mov es, ax
3994 000043F3 6658
                                 <1>
                                          pop
                                                 ax
                                          CALL DISK_IO_CONT
                                                                    ; PERFORM THE OPERATION
3995 000043F5 E88D000000
                                 <1>
3996
                                 <1>
                                          ;; CALL DDS
                                                                    ; ESTABLISH SEGMENT
3997 000043FA 8A25[5B590100]
                                 <1>
                                          MOV AH, [DISK_STATUS1] ; GET STATUS FROM OPERATION
                                          ; (*) CMP AH,1 ; SET THE CARRY FLAG TO INDICATE ; (*) CMC ; SUCCESS OR FAILURE
3998
                                 <1>
3999
                                 <1>
4000 00004400 5F
                                          POP eDI
                                                                    ; RESTORE REGISTERS
                                 <1>
4001 00004401 5E
                                 <1>
                                           POP
                                                eSI
                                          POP ES
POP DS
4002 00004402 07
                                <1>
4003 00004403 1F
                                <1>
4004 00004404 5A
                                <1>
                                          POP
                                                eDX
                                          POP
4005 00004405 59
                                <1>
                                                eCX
                                          POP eBX
4006 00004406 5B
                                <1>
4007 00004407 C9
                                <1>
                                          LEAVE
                                                                    ; ADJUST (SP) AND RESTORE (BP)
4008
                                <1>
                                          ;RETf 2
                                                                     ; THROW AWAY SAVED FLAGS
                                          ; (*) 29/05/2016
4009
                                <1>
4010
                                          ; (*) retf 4
                                <1>
4011 00004408 80FC01
                                <1>
                                           cmp
4012 0000440B 7205
                                <1>
                                           jс
                                                 short A5
4013 0000440D 804C240801
                                 <1>
                                                 byte [esp+8], 1; set carry bit of eflags register
                                 <1> _A5:
4014
4015 00004412 CF
                                 <1>
                                           iretd
4016
                                 <1>
4017
                                 <1> ; 21/02/2015
                                             dw --> dd
4018
                                 <1>;
4019
                                 <1> D1:
                                                                    ; FUNCTION TRANSFER TABLE
4020 00004413 [D6450000]
                                 <1>
                                                                    ; 000H
                                           dd
                                                 DISK RESET
                                                                    ; 001H
4021 00004417 [4D460000]
                                 <1>
                                           dd
                                                 RETURN_STATUS
4022 0000441B [5A460000]
                                 <1>
                                                 DISK READ
                                                                    ; 002H
                                          dd
                                                 DISK_WRITE
4023 0000441F [63460000]
                                 <1>
                                          dd
                                                                    ; 003H
                                                                    ; 004H
4024 00004423 [6C460000]
4025 00004427 [84460000]
                                                 DISK_VERF
                                 <1>
                                           dd
                                 <1>
                                           dd
                                                 FMT TRK
                                                                    ; 005H
                                                                    ; 006H FORMAT BAD SECTORS
4026 0000442B [CC450000]
                                 <1>
                                                 BAD COMMAND
                                                                    ; 007H FORMAT DRIVE
4027 0000442F [CC450000]
                                 <1>
                                                 BAD COMMAND
                                           dd
                                                                    ; 008H RETURN PARAMETERS
4028 00004433 [CC450000]
                                                 BAD COMMAND
                                 <1>
                                           dd
4029 00004437 [A1470000]
                                                 INIT DRV
                                 <1>
                                           dd
                                                                    ; 009Н
4030 0000443B [00480000]
                                                                   ; 00AH
                                 <1>
                                           dd
                                                 RD LONG
                                                                    ; 00BH
4031 0000443F [09480000]
                                 <1>
                                           dd
                                                 WR LONG
4032 00004443 [12480000]
                                                 DISK SEEK
                                 <1>
                                                                    ; 00CH
                                           dd
                                                                   ; 00DH
4033 00004447 [D6450000]
                                                 DISK RESET
                                 <1>
                                           dd
                                                                   ; 00EH READ BUFFER
4034 0000444B [CC450000]
                                 <1>
                                                 BAD COMMAND
                                           dd
4035 0000444F [CC450000]
                                                 BAD COMMAND
                                 <1>
                                                                    ; 00FH WRITE BUFFER
                                           dd
                                                 TST RDY
4036 00004453 [3A480000]
                                 <1>
                                           dd
                                                                   ; 010H
                                                                   ; 011H
                                                 HDISK RECAL
4037 00004457 [5E480000]
                                 <1>
                                           dd
                                 <1>
4038 0000445B [CC450000]
                                                 BAD COMMAND
                                                                    ; 012H MEMORY DIAGNOSTIC
                                           dd
                                                                   ; 013H DRIVE DIAGNOSTIC
4039 0000445F [CC450000]
                                                 BAD COMMAND
                                 <1>
                                           dd
                                                 CTLR_DIAGNOSTIC ; 014H CONTROLLER DIAGNOSTIC
4040 00004463 [94480000]
                                 <1>
                                           dd
                                           ; 02/02/2015 (Temporary - Retro UNIX 386 v1 - DISK I/O test)
4041
                                 <1>
                                                 BAD COMMAND
4042 00004467 [CC450000]
                                 <1>
                                                                    ; 015h
                                           dd
                                                 BAD_COMMAND
4043 0000446B [CC450000]
                                 <1>
                                           dd
                                                                    ; 016h
                                                                   ; 017h
4044 0000446F [CC450000]
                                 <1>
                                           dd
                                                 BAD COMMAND
4045 00004473 [CC450000]
                                                 BAD COMMAND
                                                                    ; 018h
                                 <1>
                                           dd
4046 00004477 [CC450000]
                                 <1>
                                           dd
                                                 BAD COMMAND
                                                                    ; 019h
4047 0000447B [CC450000]
                                 <1>
                                           dd
                                                 BAD COMMAND
                                                                    ; 01Ah
```

```
dd DISK_READ
dd DISK_WRITE
4048 0000447F [5A460000]
                               <1>
                                <1>
                                                                ; 01Bh ; LBA read
4049 00004483 [63460000]
                                                                  ; 01Ch ; LBA write
                                <1> D1L
                                         EQU $ - D1
4051
                                <1>
4052
                                <1> DISK IO CONT:
4053
                                <1>
                                         ;; CALL DDS
                                                                  ; ESTABLISH SEGMENT
                                          CMP AH,01H
4054 00004487 80FC01
                                                                  ; RETURN STATUS
                                <1>
                                         ;;JNZ short SU0
4055
                                <1>
4056
                                <1>
                                          ;;JMP RETURN STATUS
4057 0000448A 0F84BD010000
                                          je RETURN_STATUS
                                <1>
                                <1> SU0:
4058
                                               byte [DISK_STATUS1],0 ; RESET THE STATUS INDICATOR
4059 00004490 C605[5B590100]00
                                <1>
                                         ;; PUSH BX
4060
                                <1>
                                                            ; SAVE DATA ADDRESS
                                         ;mov si, bx ;; 14/02/2015
4061
                                <1>
4062 00004497 89DE
                                <1>
                                               esi, ebx ; 21/02/2015
                                         mov
                                         MOV BL, [HF_NUM] ; GET NUMBER OF DRIVES
4063 00004499 8A1D[5C590100]
                                <1>
                                         ;; 04/01/2015
                                <1>
                                         ;; PUSH AX
4065
                                <1>
                                          AND DL, 7FH
4066 0000449F 80E27F
                                <1>
                                                                  ; GET DRIVE AS 0 OR 1
4067
                                <1>
                                                                  ; (get drive number as 0 to 3)
                                         CMP BL, DL
4068 000044A2 38D3
                                <1>
                                         ;; JBE BAD_COMMAND_POP ; INVALID DRIVE jbe BAD_COMMAND ;; 14/02/2015
                                <1>
4070 000044A4 0F8622010000
                                <1>
4071
                                <1>
                                         ;;03/01/2015
4072
                                <1>
4073 000044AA 29DB
                                <1>
                                          sub ebx, ebx
4074 000044AC 88D3
                                <1>
                                         mov bl, dl
4075
                                         ; sub bh, bh
                                <1>
4076 000044AE 883D[70590100]
                                <1>
                                         mov [LBAMode], bh ; 0
4077
                                         ;;test byte [bx+hd0_type], 1
                                                                         ; LBA ready ?
                                <1>
4078
                                <1>
                                         ;test byte [ebx+hd0_type], 1
4079
                                <1>
                                         jz short sul;
                                                                ; no
                                         ;inc byte [LBAMode]
4080
                                <1>
4081
                                <1> ;su1:
                                         ; 21/02/2015 (32 bit modification)
4082
                                <1>
4083
                                <1>
                                          ;04/01/2015
                                         push ax ; ***
4084 000044B4 6650
                                <1>
                                          ;PUSH ES ; **
4085
                                <1>
4086 000044B6 6652
                                <1>
                                          PUSH DX ; *
                                         push ax
4087 000044B8 6650
                                <1>
4088 000044BA E8BB060000
                               <1>
                                         CALL GET_VEC
                                                                  ; GET DISK PARAMETERS
                                         ; 02/02/2015
4089
                                <1>
                                         ;mov ax, [ES:BX+16] ; I/O port base address (1F0h, 170h)
4090
                                <1>
4091 000044BF 668B4310
                                <1>
                                         mov ax, [ebx+16]
                                               [HF_PORT], ax
4092 000044C3 66A3[485D0000]
                                <1>
                                         mov
4093
                                <1>
                                         ;mov dx, [ES:BX+18]; control port address (3F6h, 376h)
4094 000044C9 668B5312
                                <1>
                                         mov dx, [ebx+18]
4095 000044CD 668915[4A5D0000]
                                         mov
                                               [HF_REG_PORT], dx
                               <1>
                                         ;mov al, [ES:BX+20]; head register upper nibble (A0h, B0h, E0h, F0h)
4096
                                <1>
                                         mov al, [ebx+20]
4097 000044D4 8A4314
                                <1>
                                         ; 23/02/2015
4098
                                <1>
4099 000044D7 A840
                                <1>
                                          test al, 40h
                                                            ; LBA bit (bit 6)
4100 000044D9 7406
                                <1>
                                          jz
                                                short sul
4101 000044DB FE05[70590100]
                               <1>
                                         inc byte [LBAMode] ; 1
4102
                                <1> su1:
4103 000044E1 C0E804
                                <1>
                                          shr
                                               al, 4
4104 000044E4 2401
                                <1>
                                          and al, 1
4105 000044E6 A2[4C5D0000]
                                <1>
                                          mov [hf_m_s], al
4106
                                <1>
4107
                                <1>
                                          ; 03/01/2015
4108
                                <1>
                                          ; MOV AL, byte [ES:BX+8] ; GET CONTROL BYTE MODIFIER
4109 000044EB 8A4308
                                <1>
                                          mov
                                                al, [ebx+8]
                                <1>
                                               DX,[HF_REG_PORT]
4110
                                          ; MOV
                                                                  ; Device Control register
                                                                   ; SET EXTRA HEAD OPTION
4111 000044EE EE
                                <1>
4112
                                <1>
                                                                  ; Control Byte: (= 08h, here)
4113
                                <1>
                                                                   ; bit 0 - 0
                                                                   ; bit 1 - nIEN (1 = disable irq)
4114
                                <1>
                                                                   ; bit 2 - SRST (software RESET)
4115
                                <1>
4116
                                <1>
                                                                   ; bit 3 - use extra heads (8 to 15)
                                                                            -always set to 1-
4117
                                <1>
                                                                   ;
4118
                                <1>
                                                                   ; (bits 3 to 7 are reserved
4119
                                <1>
                                                                             for ATA devices)
4120 000044EF 8A25[5D590100]
                                               AH, [CONTROL BYTE] ; SET EXTRA HEAD OPTION IN
                                          MOV
                                <1>
4121 000044F5 80E4C0
                                                                   ; CONTROL BYTE
                                <1>
                                          AND
                                               AH, OCOH
4122 000044F8 08C4
                                <1>
                                          OR
                                                AH, AL
4123 000044FA 8825[5D590100]
                                <1>
                                         VOM
                                               [CONTROL_BYTE], AH
                                <1>
                                         ; 04/01/2015
4125 00004500 6658
                                <1>
                                          pop ax
4126 00004502 665A
                                               dx ; * ;; 14/02/2015
                                <1>
                                          pop
4127 00004504 20E4
                                <1>
                                          and ah, ah; Reset function?
4128 00004506 7507
                                <1>
                                          jnz short su2
                                         ;;pop dx ; * ;; 14/02/2015 ;pop es ; **
4129
                                <1>
4130
                                <1>
4131 00004508 6658
                                <1>
                                          pop ax; ***
                                <1>
4132
                                          ;;pop bx
4133 0000450A E9C7000000
                                          jmp DISK_RESET
                                <1>
                                <1> su2:
                                         cmp byte [LBAMode], 0
jna short su3
4135 0000450F 803D[70590100]00
                               <1>
4136 00004516 7662
                                <1>
4137
                                <1>
                                         ; 02/02/2015 (LBA read/write function calls)
4138
                               <1>
4139 00004518 80FC1B
                                          cmp ah, 1Bh
                                <1>
4140 0000451B 720B
                               <1>
                                               short lbarw1
                                          jb
4141 0000451D 80FC1C
                                          cmp ah, 1Ch
                               <1>
4142 00004520 775D
                                <1>
                                         ja short invldfnc
                                         ;;pop dx ; * ; 14/02/2015
4143
                               <1>
4144
                               <1>
                                         ;mov ax, cx; Lower word of LBA address (bits 0-15)
4145 00004522 89C8
                                         mov eax, ecx; LBA address (21/02/2015)
                               <1>
                                         ;; 14/02/2015
4146
                                <1>
4147 00004524 88D1
                               <1>
                                         mov cl, dl; 14/02/2015
4148
                               <1>
                                         ;;mov dx, bx
4149
                                <1>
                                         ;mov dx, si; higher word of LBA address (bits 16-23)
4150
                               <1>
                                         ;;mov bx, di
4151
                               <1>
                                         ;mov si, di ; Buffer offset
4152 00004526 EB32
                                <1>
                                         jmp short lbarw2
```

```
4153
                                \langle 1 \rangle lbarw1:
                                      ; convert CHS to LBA
4154
                                 <1>
4155
                                 <1>
4156
                                <1>
                                          ; LBA calculation - AWARD BIOS - 1999 - AHDSK.ASM
                                <1>
                                         ; LBA = "# of Heads" * Sectors/Track * Cylinder + Head * Sectors/Track
4157
4158
                                <1>
                                               + Sector - 1
4159 00004528 6652
                                          push dx ; * ;; 14/02/2015
                                <1>
                                          ;xor dh, dh
                                <1>
4160
4161 0000452A 31D2
                                <1>
                                          xor
                                                edx, edx
                                          ;mov dl, [ES:BX+14] ; sectors per track (logical)
4162
                                <1>
4163 0000452C 8A530E
                                <1>
                                          mov dl, [ebx+14]
4164
                                <1>
                                          ;xor ah, ah
4165 0000452F 31C0
                                <1>
                                          xor
                                                eax, eax
                                          ;mov al, [ES:BX+2]; heads (logical)
4166
                                <1>
4167 00004531 8A4302
                                <1>
                                          mov
                                                al, [ebx+2]
4168 00004534 FEC8
                                <1>
                                          dec
                                                al
4169 00004536 6640
                                <1>
                                          inc
                                                ax
                                                             ; 0 = 256
                                               dx
4170 00004538 66F7E2
                                <1>
                                          mul
4171
                                <1>
                                                ; AX = # of Heads" * Sectors/Track
4172 0000453B 6689CA
                                <1>
                                                dx, cx
                                          mov
4173
                                <1>
                                          ;and cx, 3Fh
                                                             ; sector (1 to 63)
4174 0000453E 83E13F
                                <1>
                                          and
                                                ecx, 3fh
4175 00004541 86D6
                                <1>
                                          xchg dl, dh
4176 00004543 C0EE06
                                <1>
                                                dh, 6
                                                ; DX = cylinder (0 \text{ to } 1023)
4177
                                <1>
4178
                                <1>
                                          ;mul
                                               dx
4179
                                <1>
                                                ; DX:AX = # of Heads" * Sectors/Track * Cylinder
4180 00004546 F7E2
                                          mul edx
                                <1>
4181 00004548 FEC9
                                <1>
                                          dec
                                                cl ; sector - 1
                                          ;add ax, cx
4182
                                <1>
4183
                                <1>
                                          ;adc dx, 0
4184
                                <1>
                                                ; DX:AX = # of Heads" * Sectors/Track * Cylinder + Sector -1
4185 0000454A 01C8
                                <1>
                                          add
                                                eax, ecx
4186 0000454C 6659
                                <1>
                                          pop cx; *; ch = head, cl = drive number (zero based)
4187
                                <1>
                                          ;push dx
4188
                                <1>
                                          ;push ax
4189 0000454E 50
                                <1>
                                          push eax
                                          ;mov al, [ES:BX+14] ; sectors per track (logical)
                                <1>
4190
4191 0000454F 8A430E
                                <1>
                                          mov
                                                al, [ebx+14]
4192 00004552 F6E5
                                <1>
                                          mul ch
                                          ; AX = Head * Sectors/Track movzx eax, ax; 09/12/2017
4193
                                <1>
4194 00004554 OFB7C0
                                <1>
4195
                                <1>
                                          ;pop dx
4196 00004557 5A
                                <1>
                                          pop
                                                edx
4197
                                <1>
                                          ;add ax, dx
4198
                                <1>
                                          ;pop
                                                dx
                                          ;adc dx, 0 ; add carry bit
4199
                                <1>
4200 00004558 01D0
                                <1>
                                          add
                                                eax, edx
4201
                                <1> lbarw2:
4202 0000455A 29D2
                                <1> sub
                                                edx, edx; 21/02/2015
4203 0000455C 88CA
                                <1>
                                          mov dl, cl ; 21/02/2015
4204 0000455E C645F800
                                <1>
                                          mov
                                                byte [CMD BLOCK], 0 ; Features Register
4205
                                <1>
                                                             ; NOTE: Features register (1F1h, 171h)
4206
                                <1>
                                                             ; is not used for ATA device R/W functions.
4207
                                 <1>
                                                             ; It is old/obsolete 'write precompensation'
4208
                                 <1>
                                                             ; register and error register
4209
                                 <1>
                                                             ; for old ATA/IDE devices.
                                          ; 18/01/2014
4210
                                 <1>
4211
                                 <1>
                                          ;mov ch, [hf_m_s] ; Drive 0 (master) or 1 (slave)
4212 00004562 8A0D[4C5D0000]
                                          mov cl, [hf_m_s]
                                <1>
                                                            ; bit 4 (drive bit)
4213
                                <1>
                                          ;shl ch, 4
4214
                                 <1>
                                          ;or
                                                ch, 0E0h
                                                             ; bit 5 = 1
                                                             ; bit 6 = 1 = LBA \mod e
4215
                                <1>
4216
                                <1>
                                                             ; bit 7 = 1
4217 00004568 80C90E
                                <1>
                                                cl, 0Eh ; 1110b
                                          or
4218
                                <1>
                                          ;and dh, 0Fh
                                                                   ; LBA byte 4 (bits 24 to 27)
4219 0000456B 25FFFFFF0F
                                <1>
                                          and
                                                eax, OFFFFFFh
                                          shl
                                                ecx, 28 ; 21/02/2015
4220 00004570 C1E11C
                                <1>
4221
                                <1>
                                          ;or
                                                dh, ch
4222 00004573 09C8
                                <1>
                                                eax, ecx
                                          or
                                          ;;mov [CMD_BLOCK+2], al ; LBA byte 1 (bits 0 to 7)
4223
                                <1>
4224
                                 <1>
                                                               ; (Sector Number Register)
4225
                                          ;;mov [CMD_BLOCK+3], ah ; LBA byte 2 (bits 8 to 15)
                                <1>
4226
                                 <1>
                                                              ; (Cylinder Low Register)
                                                [CMD_BLOCK+2], ax ; LBA byte 1, 2
[CMD_BLOCK+4], dl ; LBA byte 3 (bits 16 to 23)
4227
                                 <1>
                                          ; mov
4228
                                 <1>
                                          ;mov
4229
                                 <1>
                                                               ; (Cylinder High Register)
4230
                                 <1>
                                          ;;mov [CMD_BLOCK+5], dh ; LBA byte 4 (bits 24 to 27)
4231
                                 <1>
                                                               ; (Drive/Head Register)
4232
                                 <1>
4233
                                 <1>
                                          ;mov [CMD_BLOCK+4], dx ; LBA byte 4, LBA & DEV select bits
4234 00004575 8945FA
                                                 [CMD_BLOCK+2], eax ; 21/02/2015
                                 <1>
                                          mov
4235
                                <1>
                                          ;14/02/2015
4236
                                <1>
                                          ;mov dl, cl; Drive number (INIT_DRV)
4237 00004578 EB38
                                <1>
                                          jmp short su4
4238
                                <1> su3:
                                          ; 02/02/2015
                                <1>
4240
                                <1>
                                          ; (Temporary functions 1Bh & 1Ch are not valid for CHS mode)
4241 0000457A 80FC14
                                <1>
                                               ah, 14h
                                          cmp
4242 0000457D 7604
                               <1>
                                               short chsfnc
                                          ina
                                <1> invldfnc:
4243
                                          ; 14/02/2015
4244
                                <1>
                                          ;pop es ; **
4245
                                <1>
                                          pop ax ; ***
4246 0000457F 6658
                                <1>
4247
                                                    short BAD COMMAND POP
                                <1>
                                            amṛ;
4248 00004581 EB49
                                <1>
                                           jmp
                                                   short BAD_COMMAND
                                <1> chsfnc:
                                      ;MOV AX,[ES:BX+5]
4250
                                                                   ; GET WRITE PRE-COMPENSATION CYLINDER
                                <1>
4251 00004583 668B4305
                                <1>
                                          mov
                                                ax, [ebx+5]
                                      SHR
                               <1>
4252 00004587 66C1E802
                                               AX,2
                                        MOV [CMD_BLOCK],AL
4253 0000458B 8845F8
                               <1>
                                       ;; MOV AL, [ES:BX+8] ; GET CONTROL BYTE MODIFIER
4254
                                <1>
4255
                                <1>
                                          ;; PUSH DX
                                        ;;MOV DX,[HF_REG_PORT]
4256
                                <1>
4257
                                <1>
                                                                   ; SET EXTRA HEAD OPTION
                                        ;;OUT DX,AL
```

```
4258
                                                      <1>
                                                                  ;; POP DX ; *
                                                                    ;; POP ES ; **
 4259
                                                      <1>
                                                                    ;; MOV AH, [CONTROL_BYTE] ; SET EXTRA HEAD OPTION IN
 4260
                                                      <1>
                                                                    ;; AND AH, OCOH
 4261
                                                      <1>
                                                                                                             ; CONTROL BYTE
 4262
                                                     <1>
                                                                    ;;OR AH,AL
                                                                    ;;MOV [CONTROL_BYTE],AH
 4263
                                                     <1>
 4264
                                                    <1>
 4265 0000458E 88C8
                                                                                                              ; GET SECTOR NUMBER
                                                                    MOV
                                                    <1>
                                                                              AL,CL
 4266 00004590 243F
                                                    <1>
                                                                     AND
                                                                              AL,3FH
                                                                     MOV [CMD BLOCK+2], AL
 4267 00004592 8845FA
                                                    <1>
                                                                    VOM
 4268 00004595 886DFB
                                                 <1>
                                                                             [CMD_BLOCK+3],CH ; GET CYLINDER NUMBER
 4269 00004598 88C8
                                             MOV
                                                                              AL,CL
                                                                    SHR AL, 6
 4270 0000459A C0E806
MOV [CMD_BI

<1>;;05/01/2015

<1>;;05/01/2015

<1>;;MOV AL,DL

4274 000045A0 A0[4C5D0000] <1> mov al, [hf]

4275 000045A5 C0E004 <1> SHL AL,4

4276 000045A8 80E60F <1> AND DH,0FH

4277 000045AB 08FO <1> OT
                                                              MOV [CMD_BLOCK+4],AL ; CYLINDER HIGH ORDER 2 BITS
                                                                                                              ; DRIVE NUMBER
                                                                    mov al, [hf_m_s]
                                                                                                              ; HEAD NUMBER
                                                                    ;OR AL,80H or 20H
                                                                    OR AL,80h+20h ; ECC AND 512 BYTE SEC MOV [CMD_BLOCK+5],AL ; ECC/SIZE/DRIVE/HEAD
 4279 000045AD 0CA0
                                                    <1>
                                                                                                             ; ECC AND 512 BYTE SECTORS
                                                    <1>
 4280 000045AF 8845FD
                                                     <1> su4:
                                                     <1> ; POP ES ; **
 4282
                                                                   ;; 14/02/2015
;; POP AX
 4283
                                                     <1>
 4284
                                                     <1>
                                                                    ;;MOV [CMD_BLOCK+1],AL ; SECTOR COUNT
 4285
                                                     <1>
                                                                    ;;PUSH AX
 4286
                                                     <1>

;;MOV AL,AH ;GET INTO LOW BYTE
;XOR AH,AH ;ZERO HIGH BYTE
;;SAL AX,1 ;*2 FOR TABLE LOOKUP

pop ax; ***

mov [CMD_BLOCK+1], al

sub ebx, ebx

mov bl, ah

sub ebx, ebx

mov bl, ah

;xor bh, bh

;sal bx, 1

sal bx, 2; 32 bit offset (21/02/2015)

;MOV SI,AX ; PUT INTO SI FOR BRANCH

;;CMP AX,D1L ; TEST WITHIN RANGE

;;JNB short BAD_COMMAND_POP

;cmp bx, D1L

;mb short BAD_COMMAND

;xchg bx, si

xchg ebx, esi
;;;POP AX

; RESTORE A
                                                                       ;;MOV AL,AH
                                                                                                                       ; GET INTO LOW BYTE
 4287
                                                     <1>
 4288
 4289
 4290 000045B2 6658
 4291 000045B4 8845F9
 4292 000045B7 29DB
 4293 000045B9 88E3
 4295
 4296 000045BB 66C1E302
 4297
 4298
 4299
 4300
 4301 000045BF 83FB74
 4302 000045C2 7308
 4303
 4304 000045C4 87DE
                                                     <1>
                                                                 ;;;POP AX
 4305
                                                                                                             ; RESTORE AX
                                                                                                             ; AND DATA ADDRESS
 4306
                                                     <1>
                                                                  ;;;POP BX
 4307
                                                     <1>
 4308
                                                     <1>
                                                                    ;; PUSH CX
 4309
                                                      <1>
                                                                    ;; PUSH AX
                                                                                                             ; ADJUST ES:BX
                                                                                                              ; GET 3 HIGH ORDER NIBBLES OF BX
 4310
                                                     <1>
                                                                    ; MOV CX, BX
 4311
                                                     <1>
                                                                 ;SHR CX,4
                                                                 ; MOV AX, ES
; ADD AX, CX
 4312
                                                      <1>
 4313
                                                      <1>
                                                                    ; MOV ES, AX
 4314
                                                      <1>
                                                                  ;AND BX,000FH ; ES:BX CHANGED TO ES:000X
 4315
                                                      <1>
 4316
                                                      <1>
                                                                    ;;POP AX
                                                                    ;;POP CX
 4317
                                                      <1>
 4318
                                                      <1>
                                                                    ;;JMP word [CS:SI+D1]
 4319
                                                      <1>
                                                                    ;jmp word [SI+D1]
                                                                 jmp
 4320 000045C6 FFA6[13440000]
                                                      <1>
                                                                              dword [esi+D1]
 4321
                                                      <1> ;; BAD_COMMAND_POP:
                                                      <1>;; POP AX
 4322
 4323
                                                      <1> ;;
                                                                    POP
                                                                              BX
                                                    <1> BAD_COMMAND:
<1> MOV byte [DISK_STATUS1], BAD_CMD ; COMMAND ERROR
 4325 000045CC C605[5B590100]01
 4326 000045D3 B000
                                                      <1>
                                                                     MOV AL, 0
 4327 000045D5 C3
                                                      <1>
                                                                    RETn
 4328
                                                      <1>
 4329
                                                      <1> ;-----
                                                      <1> ; RESET THE DISK SYSTEM (AH=00H) :
 4330
 4331
                                                      <1> ;-----
 4332
                                                      <1>
                                                      <1>; 18-1-2015 : one controller reset (not other one)
 4333
 4334
                                                      <1>
 4335
                                                      <1> DISK RESET:
                                                      <1>
 4336 000045D6 FA
                                                                    CLI
 4337 000045D7 E4A1
                                                                     IN
                                                                              AL, INTB01 ; GET THE MASK REGISTER
                                                      <1>
                                                                    ;JMP $+2
 4338
                                                      <1>
                                                                     IODELAY
 4339
                                                      <1>
 4339 000045D9 EB00
                                                      <2> jmp short $+2
 4339 000045DB EB00
                                                      <2> jmp short $+2
                                                                                                          ; ENABLE FIXED DISK INTERRUPT
                                                                    ; AND AL, OBFH
 4340
                                                     <1>
                                                                     and al,3Fh
 4341 000045DD 243F
                                                    <1>
                                                                                                             ; 22/12/2014 (IRQ 14 & IRQ 15)
 4342 000045DF E6A1
                                                    <1>
                                                                    OUT INTB01,AL
                                                                  STI
 4343 000045E1 FB
                                                    <1>
                                                                                                            ; START INTERRUPTS
 4344
                                                     <1>
                                                                    ; 14/02/2015
                                                                  mov di, dx
 4345 000045E2 6689D7
                                                     <1>
                                                                    ; 04/01/2015
4346
                                                     <1>
 4347
                                                      <1>
                                                                     ;xor di,di
 4348
                                                      <1> drst0:
 4349 000045E5 B004
                                                      <1>
                                                              VOM
                                                                               AL,04H ; bit 2 - SRST
 4350
                                                      <1>
                                                                     ; MOV
                                                                              DX, HF REG PORT
 4351 000045E7 668B15[4A5D0000]
                                                      <1>
                                                                     VOM
                                                                               DX, [HF_REG_PORT]
                                                                                                             ; RESET
 4352 000045EE EE
                                                      <1>
                                                                     OUT
                                                                               DX,AL
                                                                     MOV
                                                                                                            ; DELAY COUNT
 4353
                                                                               CX,10
                                                      <1>;
                                                      <1> ; DRD: DEC
 4354
                                                                               CX
                                                                                                             ; WAIT 4.8 MICRO-SEC
 4355
                                                      <1> ;
                                                                     JNZ
                                                                              short DRD
4356
                                                     <1>
                                                                     ;mov cx,2
                                                                                                              ; wait for 30 micro seconds
 4357 000045EF B902000000
                                                     <1>
                                                                     mov ecx, 2 ; 21/02/2015
 4358 000045F4 E8FDD7FFFF
                                                                                                                       ; (Award Bios 1999 - WAIT REFRESH,
                                                     <1>
                                                                     call
                                                                               WAITF
 4359
                                                                                                                          ; 40 micro seconds)
                                                      <1>
 4360 000045F9 A0[5D590100]
                                                     <1>
                                                                 mov
                                                                             al,[CONTROL_BYTE]
```

```
AND AL, OFH
OUT DX, AL
CALL NOT_BUSY
JNZ short DRERR
4361 000045FE 240F
                            <1>
                                                         ; SET HEAD OPTION
                                                         ; TURN RESET OFF
4362 00004600 EE
                            <1>
4363 00004601 E86A040000
4364 00004606 7515
                           <1>
                           <1>
                                                         ; TIME OUT ON RESET
4365 00004608 668B15[485D0000] <1>
4366 0000460F FEC2
4367
4368
4369 00004611 B90A000000
4370
                           4371 00004616 EC
                                                         ; GET RESET STATUS
4372 00004617 3C01
4373
4374 00004619 740A
4375
                                                           ; BAD RESET STATUS
4376
                            <1>
                                          ; Drive/Head Register - bit 4
                            <1>
4377 0000461B E2F9
                                     loop drst1
                            <1> DRERR:
4378
4379 0000461D C605[5B590100]05
                                          byte [DISK STATUS1], BAD RESET; CARD FAILED
                            <1>
                                    MOV
4380 00004624 C3
                            <1>
                                     RETn
4381
                            <1> drst2:
                                 ; 14/02/2015
mov dx,di
4382
                            <1>
4383 00004625 6689FA
                            <1>
4384
                            <1> ;drst3:
                             <1>; 05/01/2015
4385
4386
                             <1>;
                                    shl di,1
4387
                             <1>;
                                     ; 04/01/2015
4388
                             <1>;
                                   mov ax,[di+hd_cports]
                             <1> ;
                                    cmp ax,[HF_REG_PORT] je short drst4
4389
4390
                             <1>;
                                    mov [HF_REG_PORT], ax
4391
                             <1>;
4392
                             <1>;
                                    ; 03/01/2015
                                   ; 03/01/2015
mov ax,[di+hd_ports]
                            <1> ;
4393
4394
                             <1>;
                                  ; 05/01/2014
shr di,1
; 04/01/1
                                     mov [HF_PORT], ax
                             <1>;
4395
4396
                             <1>;
                                     ; 04/01/2015
4397
                             <1>;
4398
                             <1>;
                                    jmp short drst0 ; reset other controller
                             <1> ;drst4:
4399
                             <1>;
                                   ; 05/01/2015
4400
4401
                             <1>;
                                     shr di,1
                                  mov
                                    mov al,[di+hd_dregs]
and al,10h; bit 4 only
4402
                             <1>;
                             <1>;
4403
4404
                             <1>;
                                     shr al,4; bit 4 \rightarrow bit 0
                            <1>;
4405
                                     mov [hf_m_s], al; (0 = master, 1 = slave)
4406
                            <1>
4407 00004628 A0[4C5D0000]
4408 0000462D A801
                            <1>
<1>
                                     mov al, [hf_m_s]; 18/01/2015
                                     test al,1
                            <1> ;
4409
                                     jnz short drst6
4410 0000462F 7516
                            <1>
                                     jnz short drst4
                                  AND byte [CMD_BLOCK+5], 0EFH; SET TO DRIVE 0
4411 00004631 8065FDEF
                            <1>
                            <1> ;drst5:
                            <1> drst3:
4413
4414 00004635 E867010000
                            <1> CALL INIT_DRV
                                                         ; SET MAX HEADS
4415
                            <1>
                                    ;mov dx,di
CALL HDISK_RECAL ; RECAL TO RESET SEEK SPEED
4416 0000463A E81F020000
                            <1>
                                     ; 04/01/2014
4417
                            <1>
4418
                            <1>;
                                  inc di
4419
                             <1>;
                                          dx,di
                                    mov
                            <1>;
                                     cmp dl,[HF_NUM]
4420
                                  jb
4421
                            <1> ;
                                          short drst3
                            <1> ;DRE:
                           <1> MOV byte [DISK_STATUS1],0 ; IGNORE ANY SET UP ERRORS
4423 0000463F C605[5B590100]00
4424 00004646 C3
                            <1>
                                     RETn
4425
                            <1> ;drst6:
4426
                            <1> drst4:
                                                ; Drive/Head Register - bit 4
                            ; prive/head kegister - bit 4 cl> OR byte [CMD_BLOCK+5],010H; SET TO DRIVE 1
4427 00004647 804DFD10
                            <1>
                                    jmp short drst5;
4428
                                     jmp
4429 0000464B EBE8
                            <1>
                                             short drst3
4430
                            <1>
4431
                             <1> ;-----
                             <1>; DISK STATUS ROUTINE (AH = 01H) :
4432
4433
                             <1> ;-----
4434
                             <1>
                            <1> RETURN STATUS:
4435
                            4436 0000464D A0[5B590100]
4437 00004652 C605[5B590100]00
4438 00004659 C3
                             <1>
                                   RETn
4439
                             <1>
                             <1> ;-----
4440
4441
                             <1>; DISK READ ROUTINE (AH = 02H):
4442
                             <1> ;------
4443
                             <1>
4444
                             <1> DISK_READ:
4445 0000465A C645FE20
                             JMP COMMANDI
4446 0000465E E986020000
                             <1>
                             <1>
4448
                             <1> ;------
                             <1>; DISK WRITE ROUTINE (AH = 03H):
4449
                             <1> ;-----
4450
4451
                             <1>
4452
                             <1> DISK WRITE:
4453 00004663 C645FE30
                             <1> MOV byte [CMD BLOCK+6], WRITE CMD
4454 00004667 E9D8020000
                             <1>
                                     JMP
                                           COMMANDO
                             <1>
4456
                             <1>;-----
                             <1>; DISK VERIFY (AH = 04H):
4457
                             <1> ;-----
4458
4459
                             <1>
4460
                            <1> DISK_VERF:
4461 0000466C C645FE40
                            <1> MOV
                                          byte [CMD_BLOCK+6], VERIFY_CMD
4462 00004670 E846030000
                            <1>
                                     CALL COMMAND
                           <1> JNZ short VERF_EXIT
<1> CALL _WAIT
<1> JNZ short VERF_EXIT
                                                            ; CONTROLLER STILL BUSY
4463 00004675 750C
4464 00004677 E8B8030000
                                                           ; (Original: CALL WAIT)
                                                          ; TIME OUT
4465 0000467C 7505
```

```
<1> CALL CHECK_STATUS
4466 0000467E E845040000
                                <1> VERF EXIT:
4468 00004683 C3
                               <1>
4469
                               <1>
4470
                                <1> ;-----
                                <1>; FORMATTING (AH = 05H):
4471
                                <1> ;-----
4472
4473
                               <1>
4474
                               <1> FMT_TRK:
                                                                 ; FORMAT TRACK
                                                                                 (AH = 005H)
                               4475 00004684 C645FE50
; POP ES
4485
                               <1>
                                        JMP CMD_OF ; GO EXECUTE THE COMMAND
4486 00004695 E9B1020000
                               <1>
                                <1>
                               <1>; 30/08/2020
4488
4489
                                <1>
4490
                                <1> ;-----
                                <1>; READ DASD TYPE (AH = 15H):
4491
4492
4493
                                <1>
                                <1> READ_DASD_TYPE:
4494
                                                             ; GET DRIVE PARAMETERS ; SAVE REGISTERS
                                <1> READ_D_T:
4495
4496 0000469A 1E
                                <1> PUSH DS
4497
                                <1>
                                         ; PUSH ES
                                        PUSH eBX
4498 0000469B 53
                               <1>
                                   ; push cs; pop ds
mov bx, KDATA
mov ds, bx
; mov es, h
4499
                               <1>
                                                              ; ESTABLISH ADDRESSING
4500
                               <1>
4501
4502 0000469C 66BB1000
                               <1>
                               <1>
                               <1>
                                        ;mov es, bx
MOV byte [DISK_STATUS1],0
4504
                               <1>
4505 000046A2 C605[5B590100]00 <1>
                                     MOV BL,[HF_NUM] ; GET NUMBER OF DRIVES
AND DL,7FH ; GET DRIVE NUMBER
CMP BL,DL
4506 000046A9 8AID[5C590100]

4507 000046AF 80E27F <1>

4508 000046B2 38D3 <1>

4509 000046B4 763C <1>

4510 000046B6 0FB6C2 <1>

4511 000046B9 E8BC040000 <1>

4512 <1>
4506 000046A9 8A1D[5C590100] <1>
                                         JBE short RDT_NOT_PRESENT ; RETURN DRIVE NOT PRESENT
                                        movzx eax, dl; \frac{28}{08}/2020
                                         CALL GET_VEC ; GET DISK PARAMETERS ADDRESS
                                        ; 28/08/2020 - TRDOS 386 v2
4513
                               <1>
                                         test byte [ebx+20], 40h; LBA bit (bit 6)
4514 000046BE F6431440
                               <1>
                                         jnz short RDT3 ; LBA disk (may be > 8GB)
4515 000046C2 751D
                               <1>
4516
                               <1>
                                        ;MOV AL,[ES:BX+2] mov al, [ebx+2]
4517
                               <1>
                                                               ; HEADS
4518 000046C4 8A4302
                               <1>
4519
                               <1>
                                        ;MOV CL, [ES:BX+14]
4520 000046C7 8A4B0E
                               <1>
                                              cl, [ebx+14]
                                        mov
4521 000046CA F6E9
                               <1>
                                        IMUL CL
                                                                 ; * NUMBER OF SECTORS
                                        ;MOV CX, [ES:BX]
4522
                               <1>
                                                                 ; MAX NUMBER OF CYLINDERS
4523 000046CC 668B0B
                               <1>
                                        mov cx, [ebx]
4524
                               <1>
                                        ; 02/01/2015
4525
                               <1>
4526
                               <1>
                                        ; ** leave the last cylinder as reserved for diagnostics **
4527
                               <1>
                                         ; (Also in Award BIOS - 1999, AHDSK.ASM, FUN15 -> sub ax, 1)
4528 000046CF 6649
                               <1>
                                                                ; LEAVE ONE FOR DIAGNOSTICS
                                         DEC CX
4529
                               <1>
                                                              ; NUMBER OF SECTORS
                                         IMUL CX
4530 000046D1 66F7E9
                               <1>
                                                                 ; HIGH ORDER HALF
4531 000046D4 6689D1
                               <1>
                                         MOV
                                              CX,DX
4532 000046D7 6689C2
                               <1>
                                        MOV DX, AX
                                                                 ; LOW ORDER HALF
4533
                               <1>
                                        ;SUB AX,AX
4534
                               <1>
                                         ; 28/08/2020
4535
                                        ;sub al, al
                               <1>
4536
                               <1> RDT4:
4537 000046DA 29C0
                                <1>
                                       sub
                                              eax, eax ; 28/08/2020
4538
                               <1>
4539 000046DC B403
                               <1>
                                         MOV
                                              AH,03H
                                                                 ; INDICATE FIXED DISK
                               <1>
                                         ; 30/08/2020 (clc is not needed here)
4540
4541
                               <1>
                                        ;and byte [esp+8], OFEh ; clear carry bit of eflags register
4542
                               <1> RDT2:
                                                                 ; RESTORE REGISTERS
4543 000046DE 5B
                                <1> POP
                                               eBX
4544
                                <1>
                                         ; POP ES
4545 000046DF 1F
                                        POP
                                <1>
                                              DS
                                    ; (*) CLC
4546
                                <1>
                                                                 ; CLEAR CARRY
                                <1>
                                         ;RETf
                                        ; (*) 29/05/2016
4548
                                <1>
4549
                                <1>
                                         ; (*) retf 4
4550
                                <1>
                                         ; 30/08/2020 (clc is not needed here)
4551
                                <1>
                                         ;and byte [esp+8], OFEh ; clear carry bit of eflags register
4552 000046E0 CF
                                <1>
                                         iretd
4553
                                <1>
                                <1> RDT3: ; 28/08/2020
4554
4555
                                        ; (use the result of INT 13h, function 48h as disk size)
                               <1>
4556
                               <1>
                                         ; eax = al = zero based hard disk number
4557
                                <1>
                                         ; 29/08/2020
4558
                               <1>
                                         ;add al, 2 ; hd0 = physical disk drive 2
4559 000046E1 C0E002
                               <1>
                                              al, 2 ; * 4
4560
                               <1> RDT5:
4561
                               <1>
                                         ;add eax, drv.size
                                        add eax, drv.size+8; 29/08/2020
4562 000046E4 05[8E5D0000]
                               <1>
4563 000046E9 668B10
                               <1>
                                        mov
                                              dx, [eax] ; low word of disk size
4564 000046EC 668B4802
                               <1>
                                         mov
                                               cx, [eax+2]; high word of disk size
                                        ;sub eax, eax
4565
                               <1>
4566 000046F0 EBE8
                               <1>
                                       jmp
                                              short RDT4
4567
                                <1>
                               <1> RDT_NOT_PRESENT:
4568
                                <1>
                                    ; 30/08/2020
4569
4570 000046F2 C605[5B590100]AA
                               <1>
                                         mov byte [DISK_STATUS1], NOT_RDY; DRIVE NOT READY
```

```
4571
                               <1>
4572
                               <1>
                                        ;SUB AX,AX
                                                               ; DRIVE NOT PRESENT RETURN
4573 000046F9 29C0
                                             eax, eax ; 30/08/2020
                               <1>
                                        sub
4574 000046FB 6689C1
                              <1>
                                        MOV CX, AX ; ZERO BLOCK COUNT
4575 000046FE 6689C2
                              <1>
                                       VOM
                                             DX,AX
4576
                              <1>
                                        ; 30/08/2020
                                        or byte [esp+16], 1; set carry bit of eflags register
4577 00004701 804C241001
                              <1>
                                             ; cf = 1 \rightarrow ah = 0, drive not ready, disk type = 0
4578
                               <1>
                                        JMP short RDT2
4579 00004706 EBD6
                               <1>
4580
                               <1>
                               <1> ; 28/05/2016
4581
                               <1> ; 27/05/2016 - TRDOS 386 (TRDOS v2.0)
4582
4583
                               <1>
                               <1> ;-----
4584
                               <1>; GET PARAMETERS (AH = 08H):
4585
4586
                               <1> ;-----
4587
4588
                               <1> GET PARM N:
4589
                               <1>
                                       ; ebx = user's buffer address for parameters table
                                               ; GET DRIVE PARAMETERS
4590
                               <1> ;GET PARM:
                                   PUSH DS
4591 00004708 1E
                                                               ; SAVE REGISTERS
                               <1>
4592 00004709 06
                                        PUSH ES
                               <1>
                                        PUSH eBX
4593 0000470A 53
                              <1>
4594
                              <1>
                                        ;MOV AX,ABS0
                                                              ; ESTABLISH ADDRESSING
4595
                               <1>
                                        ; MOV DS, AX
4596
                               <1>
                                        ;TEST DL,1
                                                                ; CHECK FOR DRIVE 1
                                       ;JZ short G0
4597
                               <1>
                                   ;LES BX,@HF1_TBL_VEC
4598
                               <1>
4599
                               <1>
                                        ;JMP SHORT G1
                               <1> ;G0: LES BX,@HF_TBL_VEC
4600
4601
                               <1> ;G1:
4602
                               <1>
                                       ; CALL DDS
                                                              ; ESTABLISH SEGMENT
                                        ; 22/12/2014
4603
                               <1>
4604
                               <1>
                                       ;push cs
4605
                               <1>
                                       ;pop ds
4606 0000470B 66BB1000
                              <1>
                                        mov
                                             bx, KDATA
4607 0000470F 8EDB
                              <1>
                                             ds, bx
                                        mov
                                             es, bx; 27/05/2016
4608 00004711 8EC3
                               <1>
                                        mov
4609
                               <1>
4610 00004713 80EA80
                              <1>
                                        SUB
                                             DL,80H
                                        ;CMP DL,MAX FILE ; TEST WITHIN RANGE
4611
                              <1>
                                             short G4; 29/08/2020 - BugFix
4612
                               <1>
                                        ;JAE
4613
                              <1>
                                        ; 30/08/2020
4614 00004716 3A15[5C590100] <1>
                                        cmp dl, [HF NUM]; is hard disk index < hard disk count?
4615 0000471C 736A
                              <1>
                                        jae
                                             short G\overline{4}; no, error ! drive not ready !
4616
                               <1>
4617 0000471E 31DB
                                        xor ebx, ebx; 21/02/2015
                              <1>
                                        ; 22/12/2014
                               <1>
4618
4619 00004720 88D3
                               <1>
                                        mov bl, dl
4620
                               <1>
                                        ;xor bh, bh
4621 00004722 C0E302
                               <1>
                                        shl bl, 2
                                                                ; convert index to offset
                                        ;add bx, HF_TBL_VEC add ebx, HF_TBL_VEC
4622
                               <1>
4623 00004725 81C3[60590100]
                               <1>
4624
                               <1>
                                        ; mov ax, [bx+2]
4625
                               <1>
                                             es, ax ; dpt segment bx, [bx] ; dpt offset
                                        ;mov es, ax
4626
                               <1>
                                        ; mov
                                        mov ebx, [ebx]; 32 bit offset
4627 0000472B 8B1B
                               <1>
4628
                               <1>
4629 0000472D C605[5B590100]00
                              <1>
                                        MOV byte [DISK_STATUS1],0
                              <1>
                                        ; MOV AX, [ES:BX] ; MAX NUMBER OF CYLINDERS
4630
4631 00004734 668B03
                              <1>
                                        mov ax, [ebx]
                               <1>
                                        ;;SUB AX,2
                                                                ; ADJUST FOR 0-N
                                        dec ax
4633 00004737 6648
                              <1>
                                                                ; max. cylinder number
4634 00004739 88C5
                              <1>
                                        MOV
                                             CH,AL
4635 0000473B 66250003
                              <1>
                                        AND
                                             AX,0300H
                                                               ; HIGH TWO BITS OF CYLINDER
4636 0000473F 66D1E8
                              <1>
                                        SHR
                                             AX,1
4637 00004742 66D1E8
                              <1>
                                        SHR AX,1
                                        ;OR AL,[ES:BX+14] ; SECTORS
4638
                              <1>
4639 00004745 0A430E
                              <1>
                                              al, [ebx+14]
                                        or
                                        MOV
4640 00004748 88C1
                              <1>
                                             CL,AL
4641
                              <1>
                                        ;MOV DH, [ES:BX+2]
                                                                ; HEADS
4642 0000474A 8A7302
                               <1>
                                        mov
                                              dh, [ebx+2]
4643 0000474D FECE
                              <1>
                                        DEC
                                                                ; 0-N RANGE
                                             DH
4644 0000474F 8A15[5C590100]
                              <1>
                                        MOV DL,[HF_NUM]
                                                                ; DRIVE COUNT
4645 00004755 6629C0
                               <1>
                                       SUB
                                             AX, AX
                                        ;27/12/2014
4646
                               <1>
4647
                               <1>
                                        ;mov di, bx
                                                              ; HDPT offset
4648
                               <1>
                                        ; 29/08/2020
4649
                               <1>
4650 00004758 833C2400
                               <1>
                                        cmp dword [esp], 0
4651 0000475C 7703
                               <1>
                                        ja short G7; ebx > 0
4652
                               <1>
                                        ; if EBX (user's buffer address) = 0, do not copy DPT
4653
                               <1>
4654 0000475E 5B
                               <1>
4655 0000475F EB24
                               <1>
                                        jmp short G5
                               <1> G7:
4656
                                        ; 27/05/2016
4657
                               <1>
4658
                               <1>
                                        ; return fixed disk parameters table to user
4659
                               <1>
                                        ; in user's buffer, which is pointed by EBX
                              <1>
4660
4661 00004761 873C24
                              <1>
                                        xchg edi, [esp]
                                                                ; ebx (input) -> edi, edi -> [esp]
4662 00004764 56
                              <1>
                                        push esi
4663 00004765 89DE
                              <1>
                                                                ; hard disk parameter table (32 bytes)
                                        mov
                                             esi, ebx
                                                                ; ebx = user's buffer address
4664 00004767 89FB
                              <1>
                                             ebx, edi
4665 00004769 51
                              <1>
                                       push ecx
4666 0000476A 50
                              <1>
                                        push eax
4667 0000476B B920000000
4668 00004770 E888A00000
                              <1>
                                              ecx, 32 ; 32 bytes
                                       mov
                                        call transfer to user buffer; trdosk6.s (16/05/2016)
                              <1>
4669 00004775 58
                              <1>
                                       pop
4670 00004776 59
                              <1>
                                       pop
                                              ecx
4671 00004777 5E
                              <1>
                                      pop
                                             esi
                                      pop
4672 00004778 5F
                              <1>
                                             edi
4673 00004779 730A
                              <1>
                                             short G5
                                       jnc
                              <1>
                                        ; 29/05/2016 (*)
4675 0000477B B8FF000000
                              <1>
                                      mov eax, OFFh; unknown error!
```

```
4676
                               <1> G6:
4677 00004780 804C241001
                               <1>
                                              byte [esp+16], 1; set carry bit of eflags register
                               <1> G5:
4678
4679
                               <1>
                                         ; 27/05/2016
4680
                               <1>
                                         ; POP eBX
                                                                 ; RESTORE REGISTERS
4681 00004785 07
                               <1>
                                         POP
                                              ES
4682 00004786 1F
                               <1>
                                        ;RETf 2
                               <1>
4683
4684
                               <1>
                                        ; (*) 29/05/2016
4685
                               <1>
                                        ; (*) retf 4
                                         ; (*) or byte [esp+8], 1 ; set carry bit of eflags register
4686
                               <1>
4687 00004787 CF
                               <1>
                               <1> G4:
4688
4689 00004788 C605[5B590100]07
                               <1>
                                         VOM
                                                byte [DISK_STATUS1], INIT_FAIL ; OPERATION FAILED
                                         ;mov ah, NOT_DRY; 30/08/2020 - 'drive not ready' error code
4690
                               <1>
4691 0000478F B407
                               <1>
                                         VOM
                                              AH, INIT_FAIL
4692 00004791 28C0
                               <1>
                                              AL,AL
                                        ;SUB DX,DX
4693
                               <1>
4694
                               <1>
                                        ; 30/08/2020
4695 00004793 8A15[5C590100]
                               <1>
                                        mov dl, [HF_NUM] ; disk count
4696 00004799 28F6
                               <1>
                                        sub dh, dh
4697 0000479B 6629C9
                               <1>
                                        SUB
                                              CX,CX
                                        ; 29/05/2016 (*)
4698
                               <1>
4699
                               <1>
                                        ;STC
                                                                 ; SET ERROR FLAG
                                        ;JMP
4700
                               <1>
                                              short G5
4701
                               <1>
                                        ; 29/08/2020 - BugFix
4702 0000479E 5B
                               <1>
                                        pop ebx
4703 0000479F EBDF
                                              short G6
                               <1>
                                        jmp
4704
                               <1>
                               <1> ;-----
4705
4706
                               <1>; INITIALIZE DRIVE (AH = 09H) :
4707
                                <1> ;-----
                                       ; 03/01/2015
4708
                               <1>
4709
                               <1>
                                       ; According to ATA-ATAPI specification v2.0 to v5.0
                                       ; logical sector per logical track
4710
                               <1>
4711
                               <1>
                                        ; and logical heads - 1 would be set but
                                        ; it is seen as it will be good
4712
4713
                               <1>
                                       ; if physical parameters will be set here
                                        ; because, number of heads <= 16.
4714
                               <1>
                                        ; (logical heads usually more than 16)
4715
                               <1>
4716
                               <1>
                                        ; NOTE: ATA logical parameters (software C, H, S)
4717
                                <1>
                                               == INT 13h physical parameters
4718
                               <1>
4719
                               <1> ; INIT DRV:
                               <1>;
4720
                                        MOV
                                              byte [CMD_BLOCK+6], SET_PARM_CMD
4721
                               <1>;
                                         CALL
                                              GET_VEC ; ES:BX -> PARAMETER BLOCK
                                                                 ; GET NUMBER OF HEADS
4722
                               <1>;
                                         VOM
                                              AL, [ES:BX+2]
                                                                 ; CONVERT TO 0-INDEX
                               <1>;
4723
                                         DEC
                                              AL
4724
                               <1>;
                                         MOV
                                               AH, [CMD_BLOCK+5]
                                                                ; GET SDH REGISTER
                                                                 ; CHANGE HEAD NUMBER
4725
                               <1>;
                                              AH, OFOH
                                         AND
4726
                               <1>;
                                        OR
                                               AH,AL
                                                                 ; TO MAX HEAD
4727
                                <1>;
                                        MOV
                                               [CMD BLOCK+5], AH
                                              AL, [ES:BX+14]
4728
                               <1>;
                                        MOV
                                                                 ; MAX SECTOR NUMBER
4729
                               <1>;
                                        MOV
                                              [CMD BLOCK+1],AL
4730
                               <1>;
                                        SUB
                                              AX, AX
4731
                               <1>;
                                        VOM
                                              [CMD_BLOCK+3],AL ; ZERO FLAGS
                                                                 ; TELL CONTROLLER
4732
                               <1>;
                                        CALL COMMAND
                               <1>;
                                         JNZ
4733
                                              short INIT_EXIT
                                                                 ; CONTROLLER BUSY ERROR
4734
                               <1>;
                                         CALL NOT_BUSY
                                                                 ; WAIT FOR IT TO BE DONE
                               <1> ;
                                                                   ; TIME OUT
4735
                                         JNZ short INIT EXIT
4736
                               <1>;
                                         CALL CHECK STATUS
4737
                                <1> ; INIT EXIT:
4738
                               <1>;
                                        RETn
4739
                                <1>
4740
                               <1> ; 04/01/2015
4741
                                <1>; 02/01/2015 - Derived from from AWARD BIOS 1999
                               <1>;
4742
                                                          AHDSK.ASM - INIT DRIVE
                               <1> INIT_DRV:
4743
                               <1>
4744
4745 000047A1 31C0
                               <1>
                                        xor eax, eax ; 21/02/2015
4746 000047A3 B00B
                               <1>
                                        mov al,11; Physical heads from translated HDPT
4747 000047A5 3825[70590100]
                               <1>
                                        cmp [LBAMode], ah ; 0
                                      ja short idrv0
mov al,2 ; Physical heads from standard HDPT
4748 000047AB 7702
                               <1>
4749 000047AD B002
                               <1>
4750
                               <1> idrv0:
                                      ; DL = drive number (0 based)
4751
                               <1>
4752 000047AF E8C6030000
                               <1>
                                        call GET VEC
                                        ;push bx
4753
                               <1>
4754 000047B4 53
                                        push ebx ; 21/02/2015
                               <1>
4755
                               <1>
                                        ;add bx,ax
                                         add ebx, eax
4756 000047B5 01C3
                               <1>
                                         ;; 05/01/2015
                                <1>
4758 000047B7 8A25[4C5D0000]
                              <1>
                                        mov
                                              ah, [hf_m_s]; drive number (0= master, 1= slave)
4759
                               <1>
                                         ;; and ah, 1
4760 000047BD C0E404
                               <1>
                                        shl ah,4
                                               ah,0A0h ; Drive/Head register - 10100000b (A0h)
4761 000047C0 80CCA0
                               <1>
                                        or
                              <1>
                                        ;mov al,[es:bx]
                                       mov al, [ebx] ; 21/02/2015
4763 000047C3 8A03
                              <1>
4764 000047C5 FEC8
                               <1>
                                              al
                                                   ; last head number
                                        dec
4765
                              <1>
                                        ; and al, OFh
4766 000047C7 08E0
                               <1>
                                        or
                                               al, ah ; lower 4 bits for head number
                               <1>
                                        ;
                                               byte [CMD BLOCK+6], SET PARM CMD
4768 000047C9 C645FE91
                               <1>
                                        mov
                                              [CMD BLOCK+5],al
4769 000047CD 8845FD
                              <1>
                                       mov
4770
                               <1>
                                        ;pop
                                              bx
4771 000047D0 5B
                               <1>
                                        pop
                                               ebx
                                               eax, eax; 21/02/2015
4772 000047D1 29C0
                              <1>
                                        sub
4773 000047D3 B004
                               <1>
                                               al,4 ; Physical sec per track from translated HDPT
                                        mov
4774 000047D5 803D[70590100]00 <1>
                                         cmp
                                               byte [LBAMode], 0
4775 000047DC 7702
                              <1>
                                               short idrv1
                                        jа
                                               al,14 ; Physical sec per track from standard HDPT
4776 000047DE B00E
                               <1>
                                       mov
4777
                               <1> idrv1:
4778
                               <1>
                                        ;xor
                                              ah,ah
4779
                               <1>
4780 000047E0 01C3
                               <1>
                                         add ebx, eax; 21/02/2015
```

```
4781
                          <1>
                                 ;mov al,[es:bx]
                                      ; sector number
al, [ebx]
4782
                          <1>
4783 000047E2 8A03
                          <1>
                                 mov
4784 000047E4 8845F9
                         <1>
                                 mov [CMD BLOCK+1],al
                         4785 000047E7 28C0
4786 000047E9 8845FB
4787 000047EC E8CA010000
4788 000047F1 750C
4789 000047F3 E878020000
4790 000047F8 7505
4791 000047FA E8C9020000
4792
                          <1> INIT EXIT:
4793 000047FF C3
                          <1>
                                 RETn
4794
                          <1>
                          <1> ;-----
4795
                          <1>; READ LONG (AH = 0AH) :
4796
                          <1> ;-----
4797
4798
                          <1>
4799
                          <1> RD_LONG:
                          <1> ;MOV @CMD_BLOCK+6,READ_CMD OR ECC MODE
4800
                                 mov byte [CMD_BLOCK+6], READ_CMD + ECC_MODE
4801 00004800 C645FE22
                         <1>
                                 JMP
4802 00004804 E9E0000000
                          <1>
                                        COMMANDI
4803
                          <1>
4804
                          <1> ;-----
                          <1>; WRITE LONG (AH = 0BH):
4805
4806
                          <1> ;-----
4807
                          <1>
                          <1> WR_LONG:
4808
                          4809
4810 00004809 C645FE32
4811 0000480D E932010000
4812
                          <1>
                          <1> ;-----
4813
                          <1>; SEEK (AH = 0CH):
4814
4815
                          <1> ;-----
4816
                          <1>
4817
                         <1> DISK_SEEK:
4818 00004812 C645FE70
4819 00004816 E8A0010000
4820 0000481B 751C
                                                         ; CONTROLLER BUSY ERROR
4821 0000481D E812020000
4822 00004822 7515
                                                         ; TIME OUT ON SEEK
4823 00004824 E89F020000
4824 00004829 803D[5B590100]40
4825 00004830 7507
4826 00004832 C605[5B590100]00
                          <1> DS_EXIT:
4828 00004839 C3
                          <1>
                                 RETn
4829
                          <1>
4830
                          <1> ;-----
                          <1>; TEST DISK READY (AH = 10H):
4831
4832
                          <1> ;-----
4833
                          <1>
4834
                          <1> TST_RDY:
                                                     ; WAIT FOR CONTROLLER
                         <1> CALL NOT_BUSY
4835 0000483A E831020000
4836 0000483F 751C
                          <1>
                                 JNZ
                                      short TR EX
AL, [CMD_BLOCK+5]; SELECT DRIVE
                                                  ; CHECK STATUS ONLY
4842 00004854 7507
                          <1>
                                 JNZ
                                      short TR EX
                          <1> JNZ 
<1> MOV
                                      byte [DISK_STATUS1],0 ; WIPE OUT DATA CORRECTED ERROR
4843 00004856 C605[5B590100]00
4844
                          <1> TR_EX:
4845 0000485D C3
                          <1>
                                 RETn
4846
                          <1>
                          <1> ;-----
4847
                          <1>; RECALIBRATE (AH = 11H):
4848
4849
                          <1> ;------
4850
                          <1>
4851
                          <1> HDISK_RECAL:
                          <1> MOV
                                         byte [CMD_BLOCK+6], RECAL CMD; 10h, 16
4852 0000485E C645FE10
                                 CALL COMMAND ; START THE OPERATION
4853 00004862 E854010000
                         <1>
4854 00004867 7523
4855 00004869 E8C6010000
4856 0000486E 7407
4857 00004870 E8BF010000
                                      short RECAL_EXIT ; TIME OUT TWO TIMES IS ERROR
4858 00004875 7515
                          <1> RECAL X:
4860 00004877 E84C020000
4859
                         <1> CALL CHECK_STATUS
                                      byte [DISK STATUS1], BAD SEEK; SEEK NOT COMPLETE
4861 0000487C 803D[5B590100]40 <1>
                                 CMP
4862 00004883 7507
                                      short RECAL_EXIT ; IS OK
                          <1>
                                 JNE
                          <1> MOV byte [DISK_STATUS1],0
4863 00004885 C605[5B590100]00
                          <1> RECAL EXIT:
                          <1> CMP
                                         byte [DISK STATUS1],0
4865 0000488C 803D[5B590100]00
4866 00004893 C3
                          <1>
                                  RETn
                          <1>
4868
                          <1> ;------
                          <1>; CONTROLLER DIAGNOSTIC (AH = 14H) :
4869
                          <1> ;-----
4870
4871
                          <1>
4872
                          <1> CTLR DIAGNOSTIC:
4873 00004894 FA
                                                           ; DISABLE INTERRUPTS WHILE CHANGING MASK
                          <1>
                                 CLI
                                 IN AL, INTB01 ; TURN ON SECOND INTERRUPT CHIP
4874 00004895 E4A1
                         <1>
                          <1>
                                 ; AND AL, OBFH
                                 and al, 3Fh
4876 00004897 243F
                         <1>
                                                          ; enable IRQ 14 & IRQ 15
                               ;JMP >+
IODELAY
                                 ;JMP $+2
4877
                         <1>
4878
                         <1>
4878 00004899 EB00
                         <2> jmp short $+2
                         <2> jmp short $+2
4878 0000489B EB00
                              OUT INTB01, AL
4879 0000489D E6A1
                         <1>
4880
                         <1>
                                 IODELAY
                     <2> jmp short $+2
<2> jmp short $+2
4880 0000489F EB00
4880 000048A1 EB00
4881 000048A3 E421
                         <1> IN AL, INTA01 ; LET INTERRUPTS PASS THRU TO
```

```
<1>
                                                           AND AL,0FBH ; SECOND CHIP ; JMP $+2
4882 000048A5 24FB
4883
                                              <1>
4884
                                             <1>
                                                           IODELAY
                                       4884 000048A7 EB00
4884 000048A9 EB00
                                                      OUT INTA01, AL
4885 000048AB E621
4886 000048AD FB
                                             <1>
                                                                                               ; WAIT FOR CARD
                                                                                               ; BAD CARD
                                                                                                ; START DIAGNOSE
                                                                                            ; WAIT FOR IT TO COMPLETE
                                                                                                         ; TIME OUT ON DIAGNOSTIC
## Company of the com
                                                                                               ; GET ERROR REGISTER
                                                                    AL,DX
                                                                                           ; SAVE IT
                                                                    [HF ERROR],AL
                                                                                            ; CHECK FOR ALL OK
                                                                     SHORT CD EXIT
                                             <1> CD_ERR: MOV AH, BAD_CNTLR
                                                            MOV [DISK_STATUS1], AH
4908 000048E8 C3
                                              <1>
                                                            RETn
4909
                                              <1>
                                               <1> ;-----
4910
4911
                                               <1> ; COMMANDI :
                                               <1> ; REPEATEDLY INPUTS DATA TILL
4912
4913
                                               <1>;
                                                        NSECTOR RETURNS ZERO
4914
                                              <1> ;-----
                                              <1> COMMANDI:
4915
                                    <1> CALL CHECK_DMA ; CHECK 64K BOUNDARY ERROR <1> JC short CMD_ABORT
4916 000048E9 E862020000
                                            <1> CMD I1:
                                    <1> CMD_I1:
<1> CALL _WAIT ; WAIT FOR DATA REQUEST INTERRUPT
<1> JNZ short TM_OUT ; TIME OUT
4923 000048F9 E836010000
4924 000048FE 7543
4931 0000490D FC
                                              <1>
                                                            CLD
                                                                                                ; GET THE SECTOR
4932 0000490E F3666D
                                             <1>
                                                                    INSW
                                                           REP
4940
4941 00004926 B904000000
4942 0000492B EC
                                                                                          ; GO SLOW FOR BOARD
4943
                                                                    [edi], al ; 21/02/2015
4944 0000492C 8807
                                         4945 0000492E 47
4946 0000492F E2FA
                                             <1> CMD_I3:
4947
                                              4948
4949 00004931 80C207
                                             <1>
                                                            add dl, 7
                                                       in al, dx
4950 00004934 EC
                                             <1>
                                                           in
4951 00004935 EC
                                              <1>
                                                                    al, dx
                                                         in
4952 00004936 EC
                                             <1>
                                                                    al, dx
                                             <1> ;
    CALL CHECK_STATUS
    Short CMD_ABO
    DEC byte [CMD_BLO
4953
4954 00004937 E88C010000
                                                                    short CMD_ABORT ; ERROR RETURNED
4955 0000493C 7505
                                                            DEC byte [CMD_BLOCK+1] ; CHECK FOR MORE
4956 0000493E FE4DF9
                                              <1>
                                                            ;JNZ SHORT CMD_I1
4957
4958 00004941 75BD
                                                            jnz
                                                                    short cmd_i1x ; 18/02/2016
                                              <1> CMD ABORT:
4959
4960 00004943 C3
                                               <1> TM_OUT: RETn
                                               <1>
                                               <1> ;------
4962
4963
                                               <1>; COMMANDO
                                               <1> ; REPEATEDLY OUTPUTS DATA TILL
4964
4965
                                               <1>;
                                                           NSECTOR RETURNS ZERO
                                               <1> ;-----
4966
4967
                                              <1> COMMANDO:
4968 00004944 E807020000
                                              <1>
                                                           CALL CHECK_DMA
                                                                                               ; CHECK 64K BOUNDARY ERROR
4969 00004949 72F8
                                                           JC short CMD ABORT
                                              <1>
                                             <1> CMD_OF: MOV eSI, eBX; 21/02/2015
4970 0000494B 89DE
                                                     CALL COMMAND
JNZ short CMD_ABORT
                                                                                              ; OUTPUT COMMAND
4971 0000494D E869000000
                                              <1>
4972 00004952 75EF
                                             <1>
4973 00004954 E844010000
                                                           CALL WAIT_DRQ

JC short TM_OUT
                                                                                             ; WAIT FOR DATA REQUEST
                                             <1>
4974 00004959 72E8
                                                                                             ; TOO LONG
                                              <1>
                                              <1> CMD 01: ; PUSH
4975
                                                                            DS
4976
                                               <1>
                                                          ; PUSH ES
                                                                                               ; MOVE ES TO DS
4977
                                               <1>
                                                            ; POP DS
                                                                                                ; PUT THE DATA OUT TO THE CARD
4978
                                               <1>
                                                            ;MOV CX,256
                                                           ; MOV DX, HF PORT
4979
                                               <1>
4980
                                               <1>
                                                         ; 01/02/2015
                                                         mov dx, [HF_PORT]
4981 0000495B 668B15[485D0000]
                                               <1>
4982
                                               <1>
                                                           ;push es
4983
                                               <1>
                                                            ;pop ds
4984
                                               <1>
                                                           ;mov cx, 256
```

```
<1>
 4985 00004962 B900010000
                                                                          ecx, 256 ; 21/02/2015
                                                                mov
 4986 00004967 FA
                                                  <1>
 4987 00004968 FC
                                                 <1>
                                                                 CLD
 4988 00004969 F3666F
                                            <1>
                                                                 REP
                                                                          OUTSW
 4989 0000496C FB
                                                 <1>
                                                                 STI
4990
                                                 <1>
                                                                 ;POP
                                                                        DS
                                                                                                       ; RESTORE DS
                                                                TEST byte [CMD_BLOCK+6], ECC_MODE; CHECK FOR NORMAL OUTPUT
                                                                                                       ; WAIT FOR DATA REQUEST
                                                                                                      ; OUTPUT THE ECC BYTES
                                     <1> CMD_O2: ;MOV AL, [ES:SI]
<1> mov al, [esi]
<1> OUT DX, AL
<1> INC eSI
<1> LOOP CMD O2
 4999
                                                 <1> CMD 02: ; MOV AL, [ES:SI]
 5000 00004986 8A06
 5001 00004988 EE
 5002 00004989 46
                                                <1>
<1>
 5003 0000498A E2FA
                                                                LOOP
                                                                        CMD 02
                                                 <1> CMD 03:
 5004
5005 0000498C E8A3000000
                                                TEST byte [HF_STATUS],ST_DRQ ; CHECK FOR MORE JNZ SHORT CMD_01
                                                                                                    ; CHECK RESIDUAL SECTOR COUNT
; OPERATION ABORTED - PARTIAL TRANSFER
                                                  <1> CMD 04:
 5022 000049BA C3
                                                  <1>
                                                               RETn
 5023
                                                  <1>
 5024
                                                  <1> ;-----
 5025
                                                  <1> ; COMMAND
                                                  <1>; THIS ROUTINE OUTPUTS THE COMMAND BLOCK
 5026
                                                  <1> ; OUTPUT
 5027
 5028
                                                  <1>; BL = STATUS
 5029
                                                  <1>;
                                                             BH = ERROR REGISTER
 5030
                                                  <1> ;------
 5031
                                                  <1>
                                                  <1> COMMAND:
 5032
                                                                ND:
PUSH eBX ; WAIT FOR SEEK COMPLETE AND READY
;; MOV CX, DELAY_2 ; SET INITIAL DELAY BEFORE TEST
                                                  <1>
 5033 000049BB 53
 5034
                                                  <1>
                                                5035
 5036
 5037 000049BC E879FEFFFF
 5038
 5039 000049C1 7419
 5040 000049C3 803D[5B590100]80
 5042
 5043
 5044 000049CA 7507
                                                  <1> CMD_TIMEOUT:
<1> MOV byte [DISK_STATUS1], BAD_CNTLR
 5045
 5046 000049CC C605[5B590100]20
 5047
                                                  <1> COMMAND4:
 5048 000049D3 5B
                                                  <1>
 5049 000049D4 803D[5B590100]00
                                                  <1>
                                                                 CMP byte [DISK_STATUS1],0 ; SET CONDITION CODE FOR CALLER
 5050 000049DB C3
                                                  <1>
                                                                 RETn
                                                  <1> COMMAND2:
                                                          POP
                                                               MOV byte [HF_INT_FLAG],0 ; RESET INTERRUPT FLAG
CLI ; INHIBIT INTERRUPTS WHILE CU
IN AL,INTB01
                                                                         eBX
 5052 000049DC 5B
                                                  <1>
 5053 000049DD 57
                                                  <1>
5055 000049DE 57

5054 000049DE C605[53590100]00 <1>

5055 000049E5 FA <1>

5056 000049E6 E4A1 <1>

5057 <1>
                                                                         ; INHIBIT INTERRUPTS WHILE CHANGING MASK
AL, INTB01 ; TURN ON SECOND INTERRUPT CHIP
                                                                ; AND AL, OBFH
                                                 <1>
 5057
 5058 000049E8 243F
                                                               and al, 3Fh
                                                 <1>
                                                                                                              ; Enable IRQ 14 & 15
                                                                ;JMP $+2
 5059
                                                  <1>
                                                 <1>
 5060
                                                                IODELAY
                                            <2> jmp short $+2
<2> jmp short $+2
 5060 000049EA EB00
 5060 000049EC EB00
                                                 <2> jmp short $+2
                                                               OUT INTB01,AL IN AL,INTA01
 5061 000049EE E6A1
                                                 <1>
                                                                                           ; LET INTERRUPTS PASS THRU TO ; SECOND CHIP
 5062 000049F0 E421
                                                 <1>
 5063 000049F2 24FB
                                                                AND AL, OFBH
                                                 <1>
                                                  <1>
                                                                ;JMP $+2
                                                 <1>
                                                                 IODELAY
 5065 000049F4 EB00
                                                  <2> jmp short $+2
 5065 000049F6 EB00
                                                 <2> jmp short $+2
                                                          OUT INTA01,AL
| STI 
                                                 <1>
 5066 000049F8 E621
                                                                                                                 ; INDEX THE COMMAND TABLE
                                                                TEST byte [CONTROL BYTE], OCOH; CHECK FOR RETRY SUPPRESSION
                                                                          AL, [CMD_BLOCK+6] ; YES-GET OPERATION CODE
                                                                         AL, OFOH ; GET RID OF MODIFIERS
                                                                                                      ; 20H-40H IS READ, WRITE, VERIFY
                                                                                                     ; VALID OPERATION FOR RETRY SUPPRESS
 5081
                                                <1>
                                            <1> COMMAND3: <1> MOV
 5082
                                                                         AL,[CMD_BLOCK+eDI] ; GET THE COMMAND STRING BYTE
 5083 00004A20 8A443DF8
 5084 00004A24 EE
                                                 <1>
                                                                 OUT DX, AL ; GIVE IT TO CONTROLLER
 5085
                                                  <1>
                                                                 IODELAY
```

```
<2> jmp short $+2
<2> jmp short $+2
<1> INC eDI ; NEXT BYTE IN COMMAND BLOCK
<1> INC DX ; NEXT DISK ADAPTER REGISTER
<1> cmp di, 7 ; 1/1/2015 ; ALL DONE?
<1> JNZ short COMMAND3 ; NO--GO DO NEXT ONE
<1> POP eDI
<1> RETN ; ZERO FLAG IS SET
5085 00004A25 EB00
5085 00004A27 EB00
5086 00004A29 47
5087 00004A2A 6642
5088 00004A2C 6683FF07
5089 00004A30 75EE
5090 00004A32 5F
5091 00004A33 C3
                               <1>
                                       RETn
                                                                  ; ZERO FLAG IS SET
5092
                                <1>
5093
                                <1>; CMD_TIMEOUT:
5094
                                <1> ; MOV byte [DISK_STATUS1],BAD_CNTLR
5095
                                <1> ; COMMAND4:
5096
                                <1> ; POP BX
                                          CMP [DISK_STATUS1], 0 ; SET CONDITION CODE FOR CALLER
5097
                                <1>;
5098
                                <1>;
                                         RETn
5099
                                <1>
                                <1> ;-----
5100
                                <1> ; WAIT FOR INTERRUPT :
5101
5102
                                <1> ;-----
                                <1> ; WAIT:
5103
                                <1> _WAIT:
5104
                                ; MAKE SURE INTERRUPTS ARE ON
5105 00004A34 FB
                                         ;SUB CX,CX
5106
                                                                  ; SET INITIAL DELAY BEFORE TEST
                                <1>
5107
                                         ;CLC
                                         ;MOV AX,9000H
;INT 15H
                                                                ; DEVICE WAIT INTERRUPT
5108
                                <1>
                                <1>
5109
                                       ;JC WT2 ; DEVICE TIMED OUT ;MOV BL,DELAY_1 ; SET DELAY COUNT
5110
                                <1>
                                <1>
5111
5112
                                <1>
5113
                                <1>
                                        ;mov bl, WAIT_HDU_INT_HI
                                <1> ;; 21/02/2015
<1> ;; mov bl, WAIT_HDU_INT_HI + 1
<1> ;; mov cx, WAIT_HDU_INT_LO
5114
5115
                                         ;; mov cx, WAIT HDU INT LO
5116
                                <1> mov ecx, WAIT_HDU_INT_LH
5117 00004A35 B915160500
                                                           ; (AWARD BIOS -> WAIT FOR MEM)
5118
                                <1>
                                <1> ;----
                                               WAIT LOOP
5119
5120
                                <1>
                                <1> WT1:
5121
                                     ;TEST byte [HF_INT_FLAG],80H ; TEST FOR INTERRUPT test byte [HF_INT_FLAG],0C0h
5122
                                <1>
5123 00004A3A F605[53590100]C0
                               <1>
                               <1> ; LOOPZ WT1
<1> ; JNZ short
<1> ; DEC BL
5124
                                               short WT3 ; INTERRUPT--LETS GO
5125 00004A41 7517
                                         ;DEC BL
5126
                                       ;JNZ short WT1 ; KEEP TRYING FOR A WHILE
5127
                                <1>
5128
                                <1>
                               <1> WT1_hi:
5129
                        5130 00004A43 E461
                                         test al, 10h ; transition on memory jnz short WT1_hi ; refresh.
5131 00004A45 A810
5132 00004A47 75FA
5134 00004A49 E461
5135 00004A4B A810
                               <1> test al, 10h
<1> jz short WT1_
<1> loop WT1
<1> ;;or bl, bl
<1> ;;jz short WT2
<1> ;;dec bl
5136 00004A4D 74FA
5137 00004A4F E2E9
5138
5139
5140
                                      ;;jmp short WT1
5141
                                <1>
                                         ;dec bl
;jnz short WT1
5142
                                <1>
5143
                                <1>
5144
                                <1>
5145 00004A51 C605[5B590100]80
                               <1> WT2: MOV
                                               byte [DISK STATUS1], TIME OUT ; REPORT TIME OUT ERROR
                                <1> JMP
5146 00004A58 EB0E
                                               SHORT WT4
5147 00004A5A C605[5B590100]00
                                               byte [DISK_STATUS1],0
                               <1> WT3: MOV
5148 00004A61 C605[53590100]00
                                               byte [HF INT FLAG],0
                                <1>
                                         VOM
5149 00004A68 803D[5B590100]00
                                <1> WT4: CMP
                                               byte [DISK_STATUS1],0
                                                                       ; SET CONDITION CODE FOR CALLER
5150 00004A6F C3
                                <1>
                                        RETn
5151
                                <1>
5152
                                <1> ;-----
                                <1>; WAIT FOR CONTROLLER NOT BUSY :
5153
                                <1> ;------
5154
                                <1> NOT BUSY:
5155
                                                                  ; MAKE SURE INTERRUPTS ARE ON
5156 00004A70 FB
                                <1> STI ; PUSI
5157
                                         ; PUSH eBX
                                                      ; SET INITIAL DELAY BEFORE TEST
                               5158
5159 00004A71 668B15[485D0000]
                                                                 ; Status port (HF PORT+7)
5160 00004A78 80C207
5161
                                                                   ; wait for 10 seconds
5162
                                <1>
5163
                                <1>
                                         ;mov cx, WAIT_HDU_INT_LO; 1615h
                                         ;;mov bl, WAIT_HDU_INT_HI; 05h
5164
                                <1>
                                         ;mov bl, WAIT_HDU_INT_HI + 1
mov ecx, WAIT_HDU_INT_LH ; 21/02/2015
5165
                                 <1>
5166 00004A7B B915160500
                                <1>
                                <1>
                                          mov
5168
                                <1> ;;
                                                 byte [wait_count], 0 ; Reset wait counter
                                <1> NB1:
5169
5170 00004A80 EC
                                          IN AL, DX
                                                                  ; CHECK STATUS
                                <1>
                                         ;TEST AL,ST_BUSY
5171
                               <1>
5172 00004A81 2480
                               <1>
                                         and al, ST_BUSY
                                         ;LOOPNZ NB1
5173
                               <1>
                                                                ; NOT BUSY--LETS GO
5174 00004A83 7410
                               <1>
                                          JZ short NB2
5175
                                <1>
                                         ; DEC BL
                                         ;JNZ short NB1
                                                                 ; KEEP TRYING FOR A WHILE
5176
                                <1>
5177
                               <1>
                               <1> NB1_hi: IN AL, SYS1 <1> TEST AL, 010H
5178 00004A85 E461
                                                                         ; wait for hi to lo
                               <1>
5179 00004A87 A810
                                                                         ; transition on memory
5180 00004A89 75FA
                                          JNZ SHORT NB1 hi ; refresh.
                              <1>
                             <1> NB1_lo: IN AL, SYS1
5181 00004A8B E461
                               <1>
                                         TEST AL, 010H
5182 00004A8D A810
                                         JZ short NB1 lo
5183 00004A8F 74FA
                               <1>
                                         LOOP NB1
5184 00004A91 E2ED
                               <1>
5185
                                <1>
                                         ;dec bl
5186
                                <1>
                                         ;jnz short NB1
5187
                                <1>
                                         ;
                                        cmp
5188
                                <1> ;;
                                                   byte [wait_count], 182 ; 10 seconds (182 timer ticks)
```

```
5189
                                             short NB1
                              <1> ;;
                                       jb
5190
                               <1>
                                       ;MOV [DISK_STATUS1],TIME_OUT ; REPORT TIME OUT ERROR
5191
                              <1>
5192
                              <1>
                                       ;JMP SHORT NB3
5193 00004A93 B080
                              <1>
                                       mov
                                             al, TIME OUT
5194
                              <1> NB2:
                                       ;MOV byte [DISK_STATUS1],0
5195
                              <1>
                              <1> ;NB3:
5196
                                   ; POP
                                            eBX
5197
                              <1>
                                  ;\
or
RETn
5198 00004A95 A2[5B590100]
                                             [DISK STATUS1], al ;;; will be set after return
                              <1>
                                       ;CMP byte [DISK_STATUS1],0 ; SET CONDITION CODE FOR CALLER
5199
                              <1>
5200 00004A9A 08C0
                                             al, al ; (zf = 0 \longrightarrow timeout)
                              <1>
5201 00004A9C C3
                              <1>
5202
                              <1>
5203
                              <1> ;-----
                              <1>; WAIT FOR DATA REQUEST :
5204
                               <1> ;-----
5205
5206
                               <1> WAIT DRQ:
                                   ;MOV CX,DELAY_3
5207
                               <1>
                                       ;MOV DX,HF PORT+7
5208
                              <1>
; (but it is written as 2000
5213
                              <1>
                                                              ; micro seconds in ATORGS.ASM file
; of Award Bios - 1999, D1A0622)
5214
                              <1>
5215
                              <1>
                       <1> mov ecx, WAIT_HDU_DRQ_LH; 21/02/2015
<1> WQ_1: IN AL,DX ; GET STATUS
5216 00004AA7 B9E8030000
                         5217 00004AAC EC
5218 00004AAD A808
5219 00004AAF 7516
                                       ;LOOP WQ_1
5220
                                                             ; KEEP TRYING FOR A SHORT WHILE
                              <1>
                              <1> WQ_hi:
5221
                         5222 00004AB1 E461
5223 00004AB3 A810
5224 00004AB5 75FA
5225 00004AB7 E461
                             <1> TEST AL,010H
<1> JZ SHORT WQ
<1> LOOP WQ_1
5226 00004AB9 A810
5227 00004ABB 74FA
                                             SHORT WQ_lo
5228 00004ABD E2ED
5229
                              <1>
                                      MOV
5230 00004ABF C605[5B590100]80 <1>
                                               byte [DISK STATUS1], TIME OUT ; ERROR
                              <1> MC <1> STC
5231 00004AC6 F9
5232
                              <1> WQ_OK:
                              <1> _ RETn
5233 00004AC7 C3
                              <1> ; WQ_OK:
5234
                                             ;CLC
5235
                               <1> ; RETn
5236
                              <1>
5237
                              <1> ;-----
                              <1>; CHECK FIXED DISK STATUS :
5238
                              <1> ;-----
5239
5240
                              <1> CHECK STATUS:
                             <1> CHECK_SIATOS:
<1> CALL CHECK_ST ; CHECK THE STATUS BYTE
<1> JNZ short CHECK_S1 ; AN ERROR WAS FOUND
<1> TEST AL,ST_ERROR ; WERE THERE ANY OTHER ERRORS
<1> JZ short CHECK_S1 ; NO ERROR REPORTED
<1> CALL CHECK_ER ; ERROR REPORTED
<1> CHECK_S1.
5241 00004AC8 E813000000
5242 00004ACD 7509
5243 00004ACF A801
5244 00004AD1 7405
5245 00004AD3 E849000000
                              <1> CHECK S1:
5247 00004AD8 803D[5B590100]00
                              <1> CMP
                                             byte [DISK_STATUS1],0 ; SET STATUS FOR CALLER
5248 00004ADF C3
                              <1>
                                       RETn
5249
                              <1>
5250
                               <1> ;-----
                               <1>; CHECK FIXED DISK STATUS BYTE :
5251
                               <1> ;------
5252
5253
                               <1> CHECK ST:
                                                           ; GET THE STATUS
                              <1> ; MOV DX, HF_PORT+7 <1> mov dx, [HF_PORT]
5254
5255 00004AE0 668B15[485D0000]
                              <1>
                                    add dl, 7
5256 00004AE7 80C207
                              <1>
5257
                              <1>
                                      ; 17/02/2016
5258
                              <1>
5259
                              <1>
                                     ; (http://wiki.osdev.org/ATA_PIO_Mode)
                                       ;"delay 400ns to allow drive to set new values of BSY and DRQ"
5260
                              <1>
5261 00004AEA EC
                              <1>
                                       IN AL, DX
5262
                              <1>
                                     ;in al, dx ; 100ns
                                       ;in al, dx; 100ns;in al, dx; 100ns
5263
                              <1>
5264
                              <1>
                                       NEWIODELAY; 18/02/2016 (AWARD BIOS - 1999, 'CKST' in AHSDK.ASM)
5265
                              <1>
5265 00004AEB E6EB
                              <2> out 0ebh,al
                              <1>
5266
5267 00004AED A2[51590100]
                                       MOV
                                            [HF STATUS],AL
                              <1>
                                       MOV AH, 0
5268 00004AF2 B400
                              <1>
5269 00004AF4 A880
                                                            ; IF STILL BUSY
                              <1>
                                       TEST AL, ST_BUSY
5270 00004AF6 751A
                              <1>
                                        JNZ
                                             short CKST EXIT
                                                                     ; REPORT OK
5271 00004AF8 B4CC
                              <1>
                                             AH, WRITE FAULT
                                       TEST AL,ST WRT FLT
                                                               ; CHECK FOR WRITE FAULT
5272 00004AFA A820
                              <1>
                                             short CKST EXIT
5273 00004AFC 7514
                              <1>
                                       JNZ
5274 00004AFE B4AA
                              <1>
                                       MOV
                                             AH, NOT RDY
5275 00004B00 A840
                              <1>
                                       TEST AL, ST_READY
                                                               ; CHECK FOR NOT READY
5276 00004B02 740E
                              <1>
                                       JZ
                                             short CKST EXIT
5277 00004B04 B440
                              <1>
                                       MOV
                                             AH, BAD SEEK
                                       TEST AL, ST_SEEK_COMPL
5278 00004B06 A810
                              <1>
                                                               ; CHECK FOR SEEK NOT COMPLETE
                              <1>
5279 00004B08 7408
                                       JZ
                                             short CKST EXIT
5280 00004B0A B411
                              <1>
                                       MOV
                                             AH, DATA_CORRECTED
5281 00004B0C A804
                              <1>
                                       TEST AL, ST_CORRCTD
                                                               ; CHECK FOR CORRECTED ECC
5282 00004B0E 7502
                              <1>
                                       JNZ
                                             short CKST EXIT
5283 00004B10 B400
                              <1>
                                       MOV
                                             AH,0
                              <1> CKST EXIT:
5285 00004B12 8825[5B590100]
                                       MOV
                                              [DISK_STATUS1],AH ; SET ERROR FLAG
                              <1>
5286 00004B18 80FC11
                              <1>
                                       CMP
                                              AH, DATA_CORRECTED ; KEEP GOING WITH DATA CORRECTED
                                             short CKST_EX1
5287 00004B1B 7403
                              <1>
                                       JZ
5288 00004B1D 80FC00
                              <1>
                                       CMP
                                             AH,0
5289
                              <1> CKST EX1:
5290 00004B20 C3
                              <1>
                                       RETn
5291
                               <1>
5292
                               <1> ;-----
```

```
5293
                            <1> ; CHECK FIXED DISK ERROR REGISTER :
 5294
                            <1> CHECK ER:
 5295
-
<1> ;MOV DX, HF_PORT+1 ; GET THE ERROR REGISTER
<1> mov dx, [HF_PORT] ;
                                                       ; MOVE NEXT ERROR BIT TO CARRY
5315
                            <1>
 5316
                            <1> ;-----
                            <1> ; CHECK DMA
 5317
                            <1> ; -CHECK ES:BX AND # SECTORS TO MAKE SURE THAT IT WILL :
 5318
 5319
                            <1>; FIT WITHOUT SEGMENT OVERFLOW.
                            <1> ; -ES:BX HAS BEEN REVISED TO THE FORMAT SSSS:000X :
 5320
                            <1>; -OK IF # SECTORS < 80H (7FH IF LONG READ OR WRITE) <1>; -OK IF # SECTORS = 80H (7FH) AND BX <= 00H (04H) :
 5321
 5322
                            <1>; -ERROR OTHERWISE :
 5323
                            <1> ;------
 5324
                <1> CHECK_DMA:
 5325
 5326 00004B50 6650
 5327 00004B52 66B80080
 5328 00004B56 F645FE02
 5329 00004B5A 7404
 5330 00004B5C 66B8047F
 5331 00004B60 3A65F9
 5332 00004B63 7706
 5333 00004B65 7208
 5334 00004B67 38D8
 5335 00004B69 7204
 5336 00004B6B F8
 5337 00004B6C 6658
 5338 00004B6E C3
5341 00004B77 6658
                            <1>
                                    POP AX
                                  RETn
 5342 00004B79 C3
                            <1>
 5343
                            <1>
 5344
                            <1> ;-----
                            <1>; SET UP ES:BX-> DISK PARMS :
 5345
 5346
                            <1> ;-----
 5347
                            <1>
 5348
                            <1> ; INPUT \rightarrow DL = 0 based drive number
                            <1> ; OUTPUT -> ES:BX = disk parameter table address
 5349
 5350
                            <1>
 5351
                            <1> GET_VEC:
                            <1> ;SUB AX,AX
 5352
                                                      ; GET DISK PARAMETER ADDRESS
 5353
                            <1>
                                   ; MOV ES, AX
                            <1> ;TEST DL,1 <1> .17
 5354
 5355
                                   ;JZ short GV_0
 5356
                            <1> ;
                                    LES BX,[HF1_TBL_VEC] ; ES:BX -> DRIVE PARAMETERS
                            <1>;
 5357
                                   JMP
                                        SHORT GV_EXIT
 5358
                            <1> ;GV_0:
                                        BX, [HF_TBL_VEC]
 5359
                            <1>;
                                                             ; ES:BX -> DRIVE PARAMETERS
                            <1>;
 5360
                                   ;xor bh, bh
 5361
                           <1> ;xor bh, bh <1> xor ebx, ebx
                            <1>
 5362 00004B7A 31DB
                                mov bl, dl
 5363 00004B7C 88D3
                            <1>
                                  ;;02/01/2015
 5364
                            <1>
 5365
                            <1>
                                   ;;shl bl, 1
                                                         ; port address offset
                                  ;;mov ax, [bx+hd_ports] ; Base port address (1F0h, 170h)
 5366
                            <1>
                                 ;;shl bl, 1 ; dpt pointer offset shl bl, 2 ;;
 5367
                            <1>
 5368 00004B7E C0E302
                            <1>
                                   ;add bx, HF_TBL_VEC
                            <1>
                                                              ; Disk parameter table pointer
                                  add ebx, HF_TBL_VEC; 21/02/2015
 5370 00004B81 81C3[60590100]
                            <1>
                                   ;push word [bx+2] ; dpt segment
 5371
                            <1>
5372
                            <1>
                                    ;pop es
 5373
                            <1>
                                    ;mov bx, [bx]
                                                       ; dpt offset
 5374 00004B87 8B1B
                            <1>
                                    mov
                                         ebx, [ebx]
 5375
                            <1> ; GV_EXIT:
 5376 00004B89 C3
                            <1>
 5377
                            <1>
                            <1> hdc1_int: ; 21/02/2015
 5378
                            <1>;--- HARDWARE INT 76H -- ( IRQ LEVEL 14 ) -------
 5379
 5380
                            <1>;
                            <1>;
 5381
                                    FIXED DISK INTERRUPT ROUTINE
 5382
                            <1>;
                            <1> ;-----
 5383
 5384
                            <1>
                            <1>; 22/12/2014
 5385
 5386
                            <1> ; IBM PC-XT Model 286 System BIOS Source Code - DISK.ASM (HD_INT)
 5387
                            <1>; '11/15/85'
                            <1> ; AWARD BIOS 1999 (D1A0622)
 5388
 5389
                            <1> ; Source Code - ATORGS.ASM (INT HDISK, INT HDISK1)
 5390
                            <1>
 5391
                            <1> ;int 76h:
5392
                           <1> HD INT:
                           <1>
                                PUSH AX
 5393 00004B8A 6650
 5394 00004B8C 1E
                           <1>
                                   PUSH DS
                           <1>
                                   ; CALL DDS
5395
 5396
                           <1> ; 21/02/2015 (32 bit, 386 pm modification)
 5397 00004B8D 66B81000
                           <1>
                                  mov ax, KDATA
```

```
5398 00004B91 8ED8
                           <1>
                                  mov ds, ax
5399
                            <1>
                                ;
;;MOV @HF_INT_FLAG,OFFH ; ALL DONE
;mov byte [CS:HF_INT_FLAG], OFFh
5400
                           <1>
5401
                           <1>
                                   mov byte [HF_INT_FLAG], OFFh
5402 00004B93 C605[53590100]FF <1>
5403
                           <1>
                                    push dx
5404 00004B9A 6652
                           <1>
                        ; Clear Controller (1) Clear_IRQ1415: (1) in all du
mov dx, HDC1_BASEPORT+7; Status Register (1F7h)
5406
                                                          ; (Award BIOS - 1999)
5407
5408 00004BA0 EC
5409 00004BA1 665A
                           <1>
                                    pop
                                         dx
                                    NEWIODELAY
5410
                           <1>
5410 00004BA3 E6EB
                           <2> out 0ebh,al

<1>  ;
<1>     MOV AL,EOI     ; NON-SPECIFIC END OF INTERRUPT
<1>     OUT INTB00,AL     ; FOR CONTROLLER #2
<1>     ;JMP $+2     ; WAIT
<1>     NEWIODELAY

5411
5412 00004BA5 B020
5413 00004BA7 E6A0
5414
5415
                        <2> out 0ebh,al
<1> OUT I

5415 00004BA9 E6EB
                           <1> OUT INTA00, AL ; FOR CONTROLLER #1
5416 00004BAB E620
                           5417 00004BAD 1F
5418
                                                         ; RE-ENABLE INTERRUPTS
5419
5420
                           <1> irq15_iret: ; 25/02/2015
5421
5422 00004BAE 6658
                           <1> POP AX
                                                        ; RETURN FROM INTERRUPT
                                    IRETd
5423 00004BB0 CF
                           <1>
5424
                            <1>
5425
                            <1> hdc2 int: ; 21/02/2015
5426
                            5427
                            <1>;
                                    FIXED DISK INTERRUPT ROUTINE
5428
                            <1>;
5429
                            <1>;
                            5430
5431
                            <1>
                            <1>; int 77h:
5432
5433
                            <1> HD1_INT:
5434 00004BB1 6650
                            <1>
5435
                           <1>
                                    ; Check if that is a spurious IRQ (from slave PIC)
                        5436
5437 00004BB3 B00B
5438 00004BB5 E6A0
5439 00004BB7 EB00
5440 00004BB9 EB00
5441 00004BBB E4A0
5442 00004BBD 2480
5443 00004BBF 74ED
                                         short irq15_iret ; Fake (spurious) IRQ, do not send EOI)
5444
                           <1>
                                  PUSH DS
5445 00004BC1 1E
                           <1>
5446
                           <1>
                                    ; CALL DDS
5447
                           <1>
                                    ; 21/02/2015 (32 bit, 386 pm modification)
5448 00004BC2 66B81000
                           <1>
                                   mov ax, KDATA
5449 00004BC6 8ED8
                           <1>
                                ;
;;MOV @HF_INT_FLAG,OFFH ; ALL DONE
;or byte [GS:UB TITE]
                                  mov ds, ax
5450
                            <1>
5451
                           <1>
                                    ;or byte [CS:HF INT FLAG],0C0h
5452
                           <1>
                                  or byte [HF_INT_FLAG], OCOh
5453 00004BC8 800D[53590100]C0 <1>
5454
                            <1>
                                    push dx
5455 00004BCF 6652
                           <1>
5456 00004BD1 66BA7701
                           <1>
                                         dx, HDC2_BASEPORT+7; Status Register (177h)
5457
                            <1>
                                                          ; Clear Controller (Award BIOS 1999)
5458 00004BD5 EBC9
                            <1>
                                         short Clear_IRQ1415
                                    jmp
5459
                            <1>
5460
                            <1>
                            <1> ;%include 'diskdata.inc' ; 11/03/2015
5461
                            <1> ;%include 'diskbss.inc' ; 11/03/2015
5462
5463
                            <1>
5464
                            <1>
5465
                            5466
                            <1> ;; END OF DISK I/O SYTEM ///
                               %include 'memory.s' ; 09/03/2015
2159
                            1
                            <1>; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.0 - memory.s
  2
                            <1> ; ------
  3
                            <1> ; Last Update: 22/07/2017
  4
  6
                            <1> ; Beginning: 24/01/2016
  7
  8
                            <1>; Assembler: NASM version 2.11 (trdos386.s)
                            <1> ; ------
  9
                            <1>; Turkish Rational DOS
 10
 11
                            <1>; Operating System Project v2.0 by ERDOGAN TAN (Beginning: 04/01/2016)
 12
                            <1>; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
 13
 14
                            <1>; memory.inc (18/10/2015)
                            15
                            <1>
 16
                            <1>; MEMORY.ASM - Retro UNIX 386 v1 MEMORY MANAGEMENT FUNCTIONS (PROCEDURES)
 17
                            <1>; Retro UNIX 386 v1 Kernel (unix386.s, v0.2.0.14) - MEMORY.INC
 18
 19
                            <1> ; Last Modification: 18/10/2015
 20
                            <1>; ////// MEMORY MANAGEMENT FUNCTIONS (PROCEDURES) //////////
 21
 22
                            <1>
 23
                            <1>::04/11/2014 (unix386.s)
 2.4
                            <1>; PDE_A_PRESENT
                                               equ
                                                                ; Present flag for PDE
                            <1>; PDE A WRITE equ
 25
                                               2
                                                          ; Writable (write permission) flag
 26
                            <1>; PDE_A_USER equ
                                              4
                                                          ; User (non-system/kernel) page flag
 27
                            <1> ;;
                            <1> ; PTE A PRESENT
                                                                ; Present flag for PTE (bit 0)
 28
                                               equ
                                               2
 29
                            <1>; PTE_A_WRITE equ
                                                          ; Writable (write permission) flag (bit 1)
                                                         ; User (non-system/kernel) page flag (bit 2)
 30
                            <1>; PTE A USER equ
                                               4
 31
                            <1> ; PTE A ACCESS equ32
                                                          ; Accessed flag (bit 5) ; 09/03/2015
 32
                            <1>
 33
                            <1> ; 27/04/2015
```

```
<1> PAGE_SIZE equ 4096
<1> PAGE_SHIFT equ 12
 35
                                                                     ; page size in bytes
                                                                     ; page table shift count
 36
                                 <1> PAGE_D_SHIFT equ 22; 12 + 10 ; page directory shift count
 37
 38
                                 39
                                 <1> PTE MASK
                                                 equ 03FFh
                                                                     ; page table entry mask
                                 <1> PTE DUPLICATED equ 200h
                                                                     ; duplicated page sign (AVL bit 0)
 40
                                 <1> PDE_A_CLEAR equ 0F000h
                                                                   ; to clear PDE attribute bits
 41
                                                                     ; to clear PTE attribute bits
 42
                                 <1> PTE_A_CLEAR equ 0F000h
                                 <1> LOGIC SECT SIZE equ 512
 43
                                                                           ; logical sector size
                                                                     ; major error: page fault
 44
                                 <1> ERR_MAJOR_PF equ 0E0h
                                 <1> ERR_MINOR_IM equ 4 ;15/10/2016 (1->4); insufficient (out of) memory
 45
                                 <1> ERR_MINOR_PV equ 6 ;15/10/2016 (1->4); protection violation
 46
 47
                                 <1> SWP_DISK_READ_ERR equ 40
                                 <1> SWP_DISK_NOT_PRESENT_ERR equ 41
 48
                                 <1> SWP SECTOR NOT PRESENT ERR equ 42
 49
                                 <1> SWP NO FREE SPACE ERR
 50
 51
                                 <1> SWP_DISK_WRITE_ERR
                                                                equ 44
                                 <1> SWP_NO_PAGE_TO_SWAP_ERR equ 45
 52
 53
                                 <1> PTE_A_ACCESS_BIT equ 5 ; Bit 5 (accessed flag)
                                 <1> SECTOR SHIFT
                                                      equ 3 ; sector shift (to convert page block number)
 54
 55
                                 <1> ; 12/07/2016
                                 <1> PTE_SHARED equ 400h
                                                                     ; AVL bit 1, direct memory access bit
 56
 57
                                 <1>
                                                                     ; (Indicates that the page is not allocated
                                                                      ; for the process, it is a shared or \ensuremath{\mathsf{system}}
 58
                                 <1>
 59
                                 <1>
                                                                             ; page, it must not be deallocated!)
 60
 61
                                 <1> ;; Retro Unix 386 v1 - paging method/principles
 62
                                 <1> ;; 10/10/2014
 63
 64
                                 <1> ;; RETRO UNIX 386 v1 - PAGING METHOD/PRINCIPLES
 65
                                 <1> ;;
                                 <1> ;; KERNEL PAGE MAP: 1 to 1 physical memory page map
 66
 67
                                 <1> ;;
                                           (virtual address = physical address)
 68
                                 <1> ;; KERNEL PAGE TABLES:
                                 <1> ;;
 69
                                           Kernel page directory and all page tables are
 70
                                 <1> ;;
                                           on memory as initialized, as equal to physical memory
 71
                                 <1> ;;
                                           layout. Kernel pages can/must not be swapped out/in.
 72
                                 <1> ;;
 73
                                 <1> ;;
                                           what for: User pages may be swapped out, when accessing
 74
                                 <1> ;;
                                           a page in kernel/system mode, if it would be swapped out,
 75
                                 <1> ;;
                                           kernel would have to swap it in! But it is also may be
                                           in use by a user process. (In system/kernel mode
 76
                                 <1> ;;
 77
                                 <1> ;;
                                           kernel can access all memory pages even if they are
 78
                                 <1> ;;
                                           reserved/allocated for user processes. Swap out/in would
                                 <1> ;;
 79
                                           cause conflicts.)
 80
                                 <1> ;;
                                           As result of these conditions,
 81
                                 <1> ;;
 82
                                 <1> ;;
                                           all kernel pages must be initialized as equal to
                                           physical layout for preventing page faults.
                                 <1> ;;
 83
 84
                                 <1> ;;
                                           Also, calling "allocate page" procedure after
 85
                                 <1> ;;
                                           a page fault can cause another page fault (double fault)
                                 <1> ;;
                                           if all kernel page tables would not be initialized.
 86
 87
                                 <1> ;;
                                           [first_page] = Beginning of users space, as offset to
 88
                                 <1> ;;
 89
                                 <1> ;;
                                           memory allocation table. (double word aligned)
 90
                                 <1> ;;
                                           [next page] = first/next free space to be searched
 91
                                 <1> ;;
 92
                                 <1> ;;
                                           as offset to memory allocation table. (dw aligned)
 93
                                 <1> ;;
 94
                                 <1> ;;
                                           [last_page] = End of memory (users space), as offset
 95
                                 <1> ;;
                                           to memory allocation table. (double word aligned)
                                 <1> ;;
 96
 97
                                 <1> ;; USER PAGE TABLES:
                                 <1> ;;
 98
                                           Demand paging (& 'copy on write' allocation method) ...
 99
                                 <1> ;;
                                                  'ready only' marked copies of the
100
                                 <1> ;;
                                                  parent process's page table entries (for
101
                                 <1> ;;
                                                  same physical memory).
102
                                 <1> ;;
                                                  (A page will be copied to a new page after
                                 <1> ;;
103
                                                  if it causes R/W page fault.)
104
                                 <1> ;;
105
                                 <1> ;;
                                           Every user process has own (different)
                                 <1> ;;
106
                                           page directory and page tables.
107
                                 <1> ;;
                                 <1> ;;
108
                                           Code starts at virtual address 0, always.
                                 <1> ;;
109
                                           (Initial value of EIP is 0 in user mode.)
110
                                 <1> ;;
                                           (Programs can be written/developed as simple
111
                                 <1> ;;
                                           flat memory programs.)
112
                                 <1> ;;
113
                                 <1> ;; MEMORY ALLOCATION STRATEGY:
114
                                 <1> ;;
                                           Memory page will be allocated by kernel only
                                 <1> ;;
                                                  (in kernel/system mode only).
115
                                           * After a
                                 <1> ;;
116
117
                                             - 'not present' page fault
                                 <1> ;;
118
                                 <1> ;;
                                             - 'writing attempt on read only page' page fault
                                           * For loading (opening, reading) a file or disk/drive
119
                                 <1>;;
120
                                 <1> ;;
                                           * As responce to 'allocate additional memory blocks'
                                            request by running process.
121
                                 <1> ;;
122
                                 <1> ;;
                                           * While creating a process, allocating a new buffer,
123
                                 <1> ;;
                                             new page tables etc.
124
                                 <1> ;;
125
                                 <1> ;;
                                           At first,
                                 <1> ;;
126
                                           - 'allocate page' procedure will be called;
                                 <1> ;,
127
                                              if it will return with a valid (>0) physical address
128
                                 <1> ;;
                                              (that means the relevant M.A.T. bit has been RESET)
                                              relevant memory page/block will be cleared (zeroed).
129
                                 <1> ;;
                                           - 'allocate page' will be called for allocating page
130
                                 <1> ;;
131
                                 <1> ;;
                                              directory, page table and running space (data/code).
                                           - every successful 'allocate page' call will decrease
132
                                 <1> ;;
133
                                 <1> ;;
                                             'free pages' count (pointer).
                                           - 'out of (insufficient) memory error' will be returned
134
                                 <1> ;;
135
                                 <1> ;;
                                             if 'free pages' points to a ZERO.
                                 <1> ;;
                                           - swapping out and swapping in (if it is not a new page)
136
                                             procedures will be called as responce to 'out of memory'
137
                                 <1> ;;
138
                                 <1> ::
                                             error except errors caused by attribute conflicts.
```

<1> ; 09/03/2015

34

```
140
                                <1> ;;
                                <1> ;;
141
                                          At second,
142
                                <1> ;;
                                          - page directory entry will be updated then page table
143
                                <1> ;;
                                            entry will be updated.
144
                                <1> ;;
                                <1> ;; MEMORY ALLOCATION TABLE FORMAT:
145
                                <1> ;;
                                         - M.A.T. has a size according to available memory as
146
147
                                <1> ;;
                                            follows:
                                                  - 1 (allocation) bit per 1 page (4096 bytes)
                                <1> ;;
148
149
                                <1> ;;
                                                  - a bit with value of 0 means allocated page
                                                  - a bit with value of 1 means a free page
150
                                <1> ;;
                                <1> ;,
151
                                          - 'free_pages' pointer holds count of free pages
152
                                <1> ;;
                                          depending on M.A.T.
153
                                <1> ;;
                                                (NOTE: Free page count will not be checked
                                <1> ;;
154
                                                again -on M.A.T.- after initialization.
                                                Kernel will trust on initial count.)
155
                                <1> ;;
                                <1> ;,
                                          - 'free pages' count will be decreased by allocation
156
157
                                <1> ;;
                                            and it will be increased by deallocation procedures.
158
                                <1> ;;
159
                                <1> ;;
                                          - Available memory will be calculated during
                                            the kernel's initialization stage (in real mode).
160
                                <1> ;;
                                <1> ;;
                                            Memory allocation table and kernel page tables
161
162
                                <1> ;;
                                            will be formatted/sized as result of available
                                <1> ;;
163
                                            memory calculation before paging is enabled.
164
                                <1> ;;
165
                                <1> ;; For 4GB Available/Present Memory: (max. possible memory size)
                                <1> ;;
166
                                         - Memory Allocation Table size will be 128 KB.
                                <1> ;;
                                          - Memory allocation for kernel page directory size
167
                                <1> ;;
168
                                           is always 4 KB. (in addition to total allocation size
169
                                <1> ;;
                                           for page tables)
170
                                <1> ;;
                                          - Memory allocation for kernel page tables (1024 tables)
                                <1> ;;
171
                                           is 4 MB (1024*4*1024 \text{ bytes}).
172
                                <1> ;;
                                          - User (available) space will be started
173
                                <1> ;;
                                          at 6th MB of the memory (after 1MB+4MB).
                                <1> ;;
174
                                          - The first 640 KB is for kernel's itself plus
175
                                <1> ;;
                                          memory allocation table and kernel's page directory
176
                                <1> ;;
                                            (D0000h-EFFFFh may be used as kernel space...)
                                          - B0000h to B7FFFh address space (32 KB) will be used
177
                                <1> ;;
                                          for buffers.
178
                                <1> ;;
                                <1> ;;
179
                                          - ROMBIOS, VIDEO BUFFER and VIDEO ROM space are reserved.
                                            (A0000h-AFFFFh, C0000h-CFFFFh, F0000h-FFFFFh)
180
                                <1> ;,
                                <1> ;;
181
                                          - Kernel page tables start at 100000h (2nd MB)
182
                                <1> ;;
                                <1> ;; For 1GB Available Memory:
183
184
                                <1> ;;
                                        - Memory Allocation Table size will be 32 KB.
185
                                <1> ;;
                                          - Memory allocation for kernel page directory size
186
                                <1> ;;
                                          is always 4 KB. (in addition to total allocation size
187
                                <1> ;;
                                           for page tables)
                                         - Memory allocation for kernel page tables (256 tables)
                                <1> ;;
188
189
                                <1> ;;
                                          is 1 MB (256*4*1024 \text{ bytes}).
190
                                <1> ;;
                                          - User (available) space will be started
                                <1> ;;
191
                                           at 3th MB of the memory (after 1MB+1MB).
192
                                <1> ;;
                                          - The first 640 KB is for kernel's itself plus
193
                                <1> ;;
                                          memory allocation table and kernel's page directory
194
                                <1> ;;
                                            (D0000h-EFFFFh may be used as kernel space...)
                                          - B0000h to B7FFFh address space (32 KB) will be used
195
                                <1> ;;
                                           for buffers.
196
                                <1> ;;
197
                                <1> ;;
                                          - ROMBIOS, VIDEO BUFFER and VIDEO ROM space are reserved.
                                <1> ;,
198
                                           (A0000h-AFFFFh, C0000h-CFFFFh, F0000h-FFFFFh)
199
                                <1> ;;
                                          - Kernel page tables start at 100000h (2nd MB).
200
                                <1> ;;
201
                                <1> ;;
202
                                <1>
203
                                <1>
                                204
205
                                <1> ;;
                                <1> ;; RETRO UNIX 386 v1 - Paging (Method for Copy On Write paging principle)
206
207
                                <1>;; DEMAND PAGING - PARENT&CHILD PAGE TABLE DUPLICATION PRINCIPLES (23/04/2015)
208
209
                                <1> ;; Main factor: "sys fork" system call
210
                                <1> ;;
                                <1> ;;
                                211
212
213
                                                           |---> child - duplicated PTEs, read only pages
214
                                <1> ;;
215
                                <1> ;;
216
                                <1> ;; AVL bit (0) of Page Table Entry is used as duplication sign
217
                                <1> ;;
218
                                <1>;; AVL Bit 0 [PTE Bit 9] = 'Duplicated PTE belongs to child' sign/flag (if it is set)
                                <1>;; Note: Dirty bit (PTE bit 6) may be used instead of AVL bit 0 (PTE bit 9)
219
220
                                             -while R/W bit is 0-.
221
222
                                <1> ;; Duplicate page tables with writable pages (the 1st sys fork in the process):
223
                                <1>;; # Parent's Page Table Entries are updated to point same pages as read only,
224
                                        as duplicated PTE bit -AVL bit 0, PTE bit 9- are reset/clear.
225
                                <1> ;; # Then Parent's Page Table is copied to Child's Page Table.
226
                                <1> ;; # Child's Page Table Entries are updated as duplicated child bit
227
                                         -AVL bit 0, PTE bit 9- is set.
                                <1> ;;
228
                                <1>;;
229
                                <1> ;; Duplicate page tables with read only pages (several sys fork system calls):
230
                                <1> ;; # Parent's read only pages are copied to new child pages.
                                <1> ;; Parent's PTE attributes are not changed.
231
232
                                         (Because, there is another parent-child fork before this fork! We must not
233
                                <1>;;
                                          destroy/mix previous fork result).
                                <1> ;; # Child's Page Table Entries (which are corresponding to Parent's
234
                                <1>;; read only pages) are set as writable (while duplicated PTE bit is clear).
                                <1> ;; # Parent's PTEs with writable page attribute are updated to point same pages
236
237
                                         as read only, (while) duplicated PTE bit is reset (clear).
238
                                <1> ;; # Parent's Page Table Entries (with writable page attribute) are duplicated
                                <1> ;; as Child's Page Table Entries without copying actual page.
239
240
                                <1> ;; # Child 's Page Table Entries (which are corresponding to Parent's writable
                                <1> ;; pages) are updated as duplicated PTE bit (AVL bit 0, PTE bit 9- is set.
241
242
243
                                <1> ;; !? WHAT FOR (duplication after duplication):
```

139

<1> ;;

(swapper functions)

```
<1> ;; In UNIX method for sys fork (a typical 'fork' application in /etc/init)
245
                                 <1> ;; program/executable code continues from specified location as child process,
246
                                 <1> ;; returns back previous code location as parent process, every child after
247
                                 <1> ;; every sys fork uses last image of code and data just prior the fork.
                                 <1> ;; Even if the parent code changes data, the child will not see the changed data
248
249
                                 <1> ;; after the fork. In Retro UNIX 8086 v1, parent's process segment (32KB)
                                 <1> ;; was copied to child's process segment (all of code and data) according to
250
251
                                 <1> ;; original UNIX v1 which copies all of parent process code and data -core-
252
                                 <1> ;; to child space -core- but swaps that core image -of child- on to disk.
253
                                 <1> ;; If I (Erdogan Tan) would use a method of to copy parent's core
254
                                 <1> ;; (complete running image of parent process) to the child process;
255
                                 <1>;; for big sizes, i would force Retro UNIX 386 v1 to spend many memory pages
256
                                 <1> ;; and times only for a sys fork. (It would excessive reservation for sys fork,
257
                                 <1> ;; because sys fork usually is prior to sys exec; sys exec always establishes
                                 <1> ;; a new/fresh core -running space-, by clearing all code/data content).
258
                                 <1> ;; 'Read Only' page flag ensures page fault handler is needed only for a few write
259
260
                                 <1> ;; attempts between sys fork and sys exec, not more... (I say so by thinking
                                 <1> ;; of "/etc/init" content, specially.) sys exec will clear page tables and
261
262
                                 <1> ;; new/fresh pages will be used to load and run new executable/program.
263
                                 <1> ;; That is what for i have preferred "copy on write", "duplication" method
                                 <1>;; for sharing same read only pages between parent and child processes.
264
265
                                 <1> ;; That is a pitty i have to use new private flag (AVL bit 0, "duplicated PTE
                                 <1> ;; belongs to child" sign) for cooperation on duplicated pages between a parent
266
267
                                 <1> ;; and it's child processes; otherwise parent process would destroy data belongs
268
                                 <1> ;; to its child or vice versa; or some pages would remain unclaimed
269
                                 <1> ;; -deallocation problem-.
270
                                 <1> ;; Note: to prevent conflicts, read only pages must not be swapped out...
271
                                 <1> ;;
272
                                 <1> ;; WHEN PARENT TRIES TO WRITE IT'S READ ONLY (DUPLICATED) PAGE:
273
                                 <1> ;; # Page fault handler will do those:
                                 <1> ;; - 'Duplicated PTE' flag (PTE bit 9) is checked (on the failed PTE).
274
275
                                         - If it is reset/clear, there is a child uses same page.
276
                                         - Parent's read only page -previous page- is copied to a new writable page.
                                 <1> ;;
277
                                 <1> ;;
                                         - Parent's PTE is updated as writable page, as unique page (AVL=0)
                                         - (Page fault handler whill check this PTE later, if child process causes to
278
                                 <1> ;;
279
                                 <1> ;;
                                            page fault due to write attempt on read only page. Of course, the previous
280
                                 <1> ;;
                                            read only page will be converted to writable and unique page which belongs
281
                                 <1> ;;
                                           to child process.)
282
                                 <1> ;; WHEN CHILD TRIES TO WRITE IT'S READ ONLY (DUPLICATED) PAGE:
283
                                 <1> ;; # Page fault handler will do those:
                                         - 'Duplicated PTE' flag (PTE bit 9) is checked (on the failed PTE).
284
                                 <1> ;;
285
                                 <1> ;;
                                         - If it is set, there is a parent uses -or was using- same page.
                                 <1> ;;
286
                                         - Same PTE address within parent's page table is checked if it has same page
287
                                 <1> ;;
                                          address or not.
288
                                 <1> ;;
                                         - If parent's PTE has same address, child will continue with a new writable page.
                                            Parent's PTE will point to same (previous) page as writable, unique (AVL=0).
289
                                 <1> ;;
290
                                 <1> ;;
                                         - If parent's PTE has different address, child will continue with it's
291
                                 <1> ;;
                                            own/same page but read only flag (0) will be changed to writable flag (1) and
292
                                 <1> ;;
                                            'duplicated PTE (belongs to child)' flag/sign will be cleared/reset.
293
                                 <1> ;;
294
                                 <1>;; NOTE: When a child process is terminated, read only flags of parent's page tables
295
                                 <1> ;;
                                             will be set as writable (and unique) in case of child process was using
                                 <1> ;;
                                             same pages with duplicated child PTE sign... Depending on sys fork and
296
297
                                 <1> ;;
                                              duplication method details, it is not possible multiple child processes
298
                                 <1> ;;
                                             were using same page with duplicated PTEs.
299
                                 <1> ;;
                                 300
301
                                 <1>
302
                                 <1> ;; 08/10/2014
303
                                 <1>;; 11/09/2014 - Retro UNIX 386 v1 PAGING (further) draft
                                 <1> ;;
304
                                                 by Erdogan Tan (Based on KolibriOS 'memory.inc')
305
306
                                 <1> ;; 'allocate_page' code is derived and modified from KolibriOS
307
                                 <1> ;; 'alloc_page' procedure in 'memory.inc'
308
                                 <1>;; (25/08/2014, Revision: 5057) file
309
                                 <1> ;; by KolibriOS Team (2004-2012)
310
                                 <1>
                                 <1> allocate page:
311
                                        ; \overline{0}1/07/2015
312
                                 <1>
                                          ; 05/05/2015
313
                                 <1>
314
                                 <1>
                                         ; 30/04/2015
                                         ; 16/10/2014
315
                                 <1>
                                          ; 08/10/2014
                                 <1>
316
317
                                 <1>
                                         ; 09/09/2014 (Retro UNIX 386 v1 - beginning)
318
                                 <1>
319
                                 <1>
                                          ; INPUT -> none
320
                                 <1>
                                          ; OUTPUT ->
321
                                 <1>
322
                                 <1>
                                                 EAX = PHYSICAL (real/flat) ADDRESS OF THE ALLOCATED PAGE
323
                                                 (corresponding MEMORY ALLOCATION TABLE bit is RESET)
                                 <1>
324
                                 <1>
325
                                 <1>
                                                 CF = 1 and EAX = 0
326
                                 <1>
                                                           if there is not a free page to be allocated
327
                                 <1>
328
                                 <1>
                                           ; Modified Registers -> none (except EAX)
329
                                 <1>
                                                 eax, [free_pages]
330 00004BD7 A1[C8580100]
                                 <1>
                                          mov
331 00004BDC 21C0
                                 <1>
                                           and
                                                 eax, eax
332 00004BDE 7438
                                 <1>
                                           jΖ
                                                 short out_of_memory
333
                                <1>
334 00004BE0 53
                                 <1>
                                           push
                                                ebx
335 00004BE1 51
                                 <1>
                                           push
                                                 ecx
336
                                 <1>
                                                 ebx, MEM ALLOC TBL ; Memory Allocation Table offset
337 00004BE2 BB00001000
                                 <1>
338 00004BE7 89D9
                                 <1>
                                           mov
                                                 ecx, ebx
                                                                   ; NOTE: 32 (first page) is initial
339
                                 <1>
340
                                 <1>
                                                                   ; value of [next page].
341
                                 <1>
                                                                   ; It points to the first available
                                                                   ; page block for users (ring 3) ...
342
                                 <1>
                                                                   ; (MAT offset 32 = 1024/32)
343
                                 <1>
344
                                 <1>
                                                                   ; (at the of the first 4 MB)
                                                 ebx, [next_page] ; Free page searching starts from here
345 00004BE9 031D[CC580100]
                                 <1>
                                           add
346
                                 <1>
                                                               ; next free page >> 5
347 00004BEF 030D[D0580100]
                                 <1>
                                                 ecx, [last_page] ; Free page searching ends here
                                 <1>
348
                                                               ; (total_pages - 1) >> 5
```

244

```
349
                                 <1> al_p_scan:
350 00004BF5 39CB
                                 <1>
                                           cmp
                                                  ebx, ecx
351 00004BF7 770A
                                 <1>
                                            jа
                                                  short al_p_notfound
352
                                 <1>
353
                                 <1>
                                           ; 01/07/2015
354
                                 <1>
                                           ; AMD64 Architecture Programmer's Manual
355
                                 <1>
356
                                 <1>
                                           ; General-Purpose and System Instructions
357
                                 <1>
358
                                 <1>
                                           ; BSF - Bit Scan Forward
359
                                 <1>
360
                                  <1>
                                               Searches the value in a register or a memory location
361
                                 <1>
                                                (second operand) for the least-significant set bit.
362
                                 <1>
                                               If a set bit is found, the instruction clears the zero flag (ZF)
363
                                 <1>
                                               and stores the index of the least-significant set bit in a destination
364
                                 <1>
                                               register (first operand). If the second operand contains 0,
                                               the instruction sets ZF to 1 and does not change the contents of the
365
                                  <1>
                                 <1>
                                               destination register. The bit index is an unsigned offset from bit 0
366
                                               of the searched value
367
                                 <1>
368
                                 <1>
369 00004BF9 0FBC03
                                 <1>
                                            bsf
                                                  eax, [ebx] ; Scans source operand for first bit set (1).
370
                                  <1>
                                                            ; Clear ZF if a bit is found set (1) and
371
                                 <1>
                                                            ; loads the destination with an index to
372
                                 <1>
                                                            ; first set bit. (0 -> 31)
373
                                 <1>
                                                            ; Sets ZF to 1 if no bits are found set.
374 00004BFC 7525
                                 <1>
                                                   short al_p_found ; ZF = 0 \rightarrow a free page has been found
                                            jnz
375
                                 <1>
                                                          ; NOTE: a Memory Allocation Table bit
376
                                 <1>
377
                                 <1>
                                                                  with value of 1 means
378
                                 <1>
                                                                  the corresponding page is free
379
                                 <1>
                                                                  (Retro UNIX 386 v1 feature only!)
380 00004BFE 83C304
                                  <1>
                                            add
                                                  ebx, 4
                                                          ; We return back for searching next page block
                                 <1>
381
382
                                 <1>
                                                          ; NOTE: [free_pages] is not ZERO; so,
383
                                 <1>
                                                                we always will find at least 1 free page here.
384 00004C01 EBF2
                                 <1>
                                                      short al_p_scan
385
                                 <1>
386
                                 <1> al_p_notfound:
387 00004C03 81E900001000
                                 <1>
                                                  ecx, MEM_ALLOC_TBL
                                            sub
                                                  [next_page], ecx ; next/first free page = last page
388 00004C09 890D[CC580100]
                                 <1>
                                           mov
                                 <1>
                                                                 ; (deallocate_page procedure will change it)
390 00004C0F 31C0
                                 <1>
                                           xor
                                                  eax, eax
391 00004C11 A3[C8580100]
                                 <1>
                                                  [free_pages], eax ; 0
                                           mov
392 00004C16 59
                                 <1>
                                           pop
393 00004C17 5B
                                 <1>
                                                  ebx
                                           pop
394
                                 <1>
                                 <1> out of memory:
396 00004C18 E85B040000
                                           call swap out
                                 <1>
                                                  short al p ok ; [free pages] = 0, re-allocation by swap out
397 00004C1D 7325
                                 <1>
                                 <1>
                                           ;
399 00004C1F 29C0
                                 <1>
                                            sub
                                                  eax, eax; 0
400 00004C21 F9
                                 <1>
                                            stc
401 00004C22 C3
                                 <1>
                                           retn
402
                                 <1>
403
                                 <1> al_p_found:
                                           mov
404 00004C23 89D9
                                 <1>
                                                  ecx, ebx
405 00004C25 81E900001000
                                 <1>
                                            sub
                                                  ecx, MEM ALLOC TBL
406 00004C2B 890D[CC580100]
                                 <1>
                                                  [next_page], ecx ; Set first free page searching start
                                            mov
                                  <1>
                                                                ; address/offset (to the next)
407
408 00004C31 FF0D[C8580100]
                                                      dword [free_pages]; 1 page has been allocated (X = X-1)
                                 <1>
                                              dec
409
                                 <1>
410 00004C37 0FB303
                                  <1>
                                                  [ebx], eax
                                                                 ; The destination bit indexed by the source value
                                 <1>
411
                                                                 ; is copied into the Carry Flag and then cleared
412
                                  <1>
                                                                 ; in the destination.
413
                                  <1>
414
                                 <1>
                                                                 ; Reset the bit which is corresponding to the
415
                                 <1>
                                                                 ; (just) allocated page.
                                           ; 01/07/2015 (4*8 = 32, 1 allocation byte = 8 pages)
416
                                 <1>
417 00004C3A C1E103
                                 <1>
                                                 ecx, 3
                                                                ; (page block offset * 32) + page index
                                            shl
418 00004C3D 01C8
                                 <1>
                                                                 ; = page number
                                            add
                                                 eax, ecx
419 00004C3F C1E00C
                                 <1>
                                            shl
                                                 eax, 12
                                                                       ; physical address of the page (flat/real value)
420
                                 <1>
                                           ; EAX = physical address of memory page
421
                                 <1>
                                           ; NOTE: The relevant page directory and page table entry will be updated
422
                                 <1>
423
                                 <1>
                                                  according to this EAX value...
                                           ;
424 00004C42 59
                                 <1>
                                           pop
                                                  ecx
425 00004C43 5B
                                 <1>
                                                  ebx
                                           pop
                                 <1> al_p_ok:
426
427 00004C44 C3
                                  <1>
428
                                 <1>
429
                                 <1>
                                  <1> make page dir:
430
431
                                 <1>
                                           ; 18/04/2015
432
                                  <1>
                                           ; 12/04/2015
433
                                  <1>
                                           ; 23/10/2014
434
                                 <1>
                                           ; 16/10/2014
435
                                  <1>
                                           ; 09/10/2014 ; (Retro UNIX 386 v1 - beginning)
                                 <1>
436
                                          ; INPUT ->
437
                                 <1>
438
                                 <1>
                                           ;
                                                 none
                                           ; OUTPUT ->
439
                                 <1>
                                                  (EAX = 0)
440
                                  <1>
                                                  cf = 1 \rightarrow insufficient (out of) memory error
441
                                 <1>
442
                                 <1>
                                                  cf = 0 ->
443
                                 <1>
                                                  u.pgdir = page directory (physical) address of the current
444
                                 <1>
                                                           process/user.
445
                                 <1>
446
                                 <1>
                                           ; Modified Registers -> EAX
447
                                 <1>
448 00004C45 E88DFFFFFF
                                 <1>
                                           call allocate page
449 00004C4A 7216
                                 <1>
                                                  short mkpd_error
                                 <1>
451 00004C4C A3[B8030300]
                                           mov
                                 <1>
                                                   [u.pgdir], eax
                                                                     ; Page dir address for current user/process
452
                                 <1>
                                                                  ; (Physical address)
453
                                 <1> clear_page:
```

```
; 18/04/2015
454
                                <1>
455
                                 <1>
                                          ; 09/10/2014 ; (Retro UNIX 386 v1 - beginning)
456
                                <1>
457
                                <1>
                                         ; INPUT ->
458
                                <1>
                                               EAX = physical address of the page
                                          ; OUTPUT ->
459
                                <1>
460
                                <1>
                                          ; all bytes of the page will be cleared
461
                                <1>
462
                                <1>
                                          ; Modified Registers -> none
463
                                <1>
464 00004C51 57
                                <1>
                                         push edi
465 00004C52 51
                                <1>
                                          push ecx
466 00004C53 50
                                <1>
                                          push eax
467 00004C54 B900040000
                                <1>
                                                ecx, PAGE_SIZE / 4
                                         mov
468 00004C59 89C7
                                <1>
                                         mov
                                                edi, eax
                                                eax, eax
469 00004C5B 31C0
                                <1>
                                          xor
470 00004C5D F3AB
                               <1>
                                          rep
                                                stosd
471 00004C5F 58
                                <1>
                                          pop
                                                eax
472 00004C60 59
                                <1>
                                          pop
473 00004C61 5F
                                <1>
                                                edi
                                          pop
                                <1> mkpd error:
474
                                <1> mkpt error:
475
476 00004C62 C3
                                <1>
                                          retn
477
                                <1>
478
                                <1> make_page_table:
479
                                <1>
                                       ; 23/06/2015
480
                                <1>
                                          ; 18/04/2015
                                        ; 12/04/2015
481
                                <1>
                                        ; 16/10/2014
; 09/10/2014 ; (Retro UNIX 386 v1 - beginning)
482
                                <1>
483
                                <1>
484
                                <1>
                                        ; INPUT ->
485
                                <1>
                                                EBX = virtual (linear) address
                                <1>
486
487
                                <1>
                                                ECX = page table attributes (lower 12 bits)
488
                                <1>
                                                       (higher 20 bits must be ZERO)
489
                                <1>
                                                       (bit 0 must be 1)
490
                                <1>
                                          ;
                                                u.pgdir = page directory (physical) address
                                       ; OUTPUT ->
491
                                <1>
492
                                <1>
                                                EDX = Page directory entry address
                                                EAX = Page table address
493
                                <1>
494
                                <1>
                                                cf = 1 -> insufficient (out of) memory error
                                         ;
                                                cf = 0 -> page table address in the PDE (EDX)
495
                                <1>
                                <1>
496
                                          ; Modified Registers -> EAX, EDX
497
                                <1>
498
                                <1>
                                          call allocate_page
499 00004C63 E86FFFFFF
                                <1>
                                          jc short mkpt_error
500 00004C68 72F8
                                <1>
501 00004C6A E811000000
                                          call set_pde
                                <1>
502 00004C6F EBE0
                                <1>
                                          jmp
                                               short clear_page
503
                                <1>
                                <1> make_page:
504
                                      ; 24/07/2015
505
                                <1>
                                          ; 23/06/2015 ; (Retro UNIX 386 v1 - beginning)
506
                                <1>
507
                                <1>
                                        ; INPUT ->
508
                                <1>
509
                                <1>
                                                EBX = virtual (linear) address
510
                                <1>
                                                ECX = page attributes (lower 12 bits)
                                <1>
511
                                                       (higher 20 bits must be ZERO)
512
                                <1>
                                                       (bit 0 must be 1)
513
                                <1>
                                                u.pgdir = page directory (physical) address
                                          ;
514
                                <1>
                                        ; OUTPUT ->
515
                                <1>
                                        ;
                                                EBX = Virtual address
516
                                <1>
                                                (EDX = PTE value)
                                          ;
517
                                <1>
                                                EAX = Physical address
                                                cf = 1 -> insufficient (out of) memory error
518
                                <1>
                                          ;
519
                                <1>
520
                                <1>
                                          ; Modified Registers -> EAX, EDX
521
                                <1>
522 00004C71 E861FFFFFF
                                <1>
                                          call allocate_page
523 00004C76 7207
                                <1>
                                          jс
                                                short mkp_err
                                          call set_pte
524 00004C78 E821000000
                                <1>
525 00004C7D 73D2
                                <1>
                                          jnc
                                                short clear_page ; 18/04/2015
                                <1> mkp_err:
526
527 00004C7F C3
                                <1>
                                <1>
528
529
                                <1>
530
                                <1> set_pde:
                                                 ; Set page directory entry (PDE)
                                       ; 20/07/2015
531
                                <1>
                                          ; 18/04/2015
532
                                 <1>
533
                                          ; 12/04/2015
                                <1>
534
                                <1>
                                        ; 23/10/2014
                                        ; 10/10/2014 ; (Retro UNIX 386 v1 - beginning)
535
                                 <1>
536
                                 <1>
537
                                 <1>
                                          ; INPUT ->
538
                                 <1>
                                                EAX = physical address
539
                                 <1>
                                                       (use present value if EAX = 0)
540
                                 <1>
                                                 EBX = virtual (linear) address
541
                                <1>
                                                 ECX = page table attributes (lower 12 bits)
542
                                <1>
                                                       (higher 20 bits must be ZERO)
543
                                <1>
                                                       (bit 0 must be 1)
                                                 u.pgdir = page directory (physical) address
544
                                <1>
                                          ; OUTPUT ->
545
                                 <1>
                                                EDX = PDE address
                                <1>
546
547
                                <1>
                                                 EAX = page table address (physical)
548
                                 <1>
                                                 ; (CF=1 -> Invalid page address)
549
                                <1>
550
                                <1>
                                          ; Modified Registers -> EDX
551
                                <1>
552 00004C80 89DA
                                <1>
                                          mov
                                                 edx, ebx
553 00004C82 C1EA16
                                                 edx, PAGE D SHIFT; 22
                                <1>
                                          shr
                                                 edx, 2 ; offset to page directory (1024*4)
554 00004C85 C1E202
                                <1>
                                          shl
555 00004C88 0315[B8030300]
                                <1>
                                          add
                                                 edx, [u.pgdir]
                                <1>
557 00004C8E 21C0
                                <1>
                                          and
                                                 eax, eax
558 00004C90 7506
                                <1>
                                          jnz
                                                short spde_1
```

```
mov eax, [edx] ; old PDE value
559
                                <1>
560 00004C92 8B02
                                <1>
561
                               <1>
                                         ;test al, 1
                                         jz short spde 2
                               <1>
563 00004C94 662500F0
                               <1>
                                               ax, PDE_A_CLEAR ; OF000h ; clear lower 12 bits
                                         and
564
                               <1> spde 1:
                               <1>
                                               cx, OFFFh
565
                                         ;and
566 00004C98 8902
                                         mov
                                               [edx], eax
                               <1>
567 00004C9A 66090A
                               <1>
                                         or
                                                [edx], cx
                                     retn
568 00004C9D C3
                               <1>
569
                                <1> ;spde_2: ; error
570
                                <1> ; stc
                                <1> ;
571
                                         retn
572
                                <1>
                                <1> set_pte:
573
                                              ; Set page table entry (PTE)
                                <1> ; 24/07/2015
<1> ; 20/07/2015
574
                                         ; 20/07/2015
575
                                       ; 23/06/2015
576
                                <1>
                                        ; 18/04/2015
577
                                <1>
                                         ; 12/04/2015
578
                                <1>
579
                                <1>
                                        ; 10/10/2014 ; (Retro UNIX 386 v1 - beginning)
580
                                <1>
                                        ; input ->
581
                                <1>
582
                                <1>
                                                EAX = physical page address
583
                                <1>
                                                     (use present value if EAX = 0)
584
                                <1>
                                                EBX = virtual (linear) address
                                                ECX = page attributes (lower 12 bits)
585
                                <1>
                                                (higher 20 bits must be ZERO)
                                <1>
586
587
                                <1>
                                                      (bit 0 must be 1)
                                         ; u.pgdir = page directory (physical) address
588
                                <1>
                                         ; OUTPUT ->
589
                                <1>
                                         ; EAX = physical page address
590
                                <1>
                                                (EDX = PTE value)
591
                                <1>
592
                                <1>
                                              EBX = virtual address
593
                                <1>
                                               CF = 1 \rightarrow error
594
                                <1>
                                <1>
596
                                <1>
                                         ; Modified Registers -> EAX, EDX
597
                                <1>
598 00004C9E 50
                                <1>
                                         push eax
599 00004C9F A1[B8030300]
                               <1>
                                         mov eax, [u.pgdir]; 20/07/2015
600 00004CA4 E837000000
                                <1>
                                         call get pde
                                                ; EDX = PDE address
601
                               <1>
602
                               <1>
                                                ; EAX = PDE value
603 00004CA9 5A
                                               edx ; physical page address
                               <1>
                                         pop
604 00004CAA 722A
                               <1>
                                         jс
                                                short spte_err ; PDE not present
                               <1>
606 00004CAC 53
                                         push ebx ; 24/07/2015
                               <1>
607 00004CAD 662500F0
                               <1>
                                                ax, PDE_A_CLEAR ; OF000h ; clear lower 12 bits
                               <1>
                                                  ; EDX = PT address (physical)
609 00004CB1 C1EB0C
                                                ebx, PAGE_SHIFT ; 12
                               <1>
                                          shr
610 00004CB4 81E3FF030000
                               <1>
                                         and
                                                ebx, PTE MASK; 03FFh
                                                ; clear higher 10 bits (PD bits)
611
                               <1>
                                                ebx, 2 ; offset to page table (1024*4)
612 00004CBA C1E302
                               <1>
                                         shl
613 00004CBD 01C3
                               <1>
                                         add
                                                ebx, eax
614
                               <1>
615 00004CBF 8B03
                             <1>
                                         mov
                                               eax, [ebx] ; Old PTE value
616 00004CC1 A801
                               <1>
                                     test al, 1
617 00004CC3 740C
                               <1>
                                       jz
or
                                                short spte_0
618 00004CC5 09D2
                               <1>
                                                edx, edx
                               <1> jnz short spte_1
<1> and ax, PTE_A_CLI
<1> mov edx, eax
<1> jmp short spte_2
619 00004CC7 750F
                                                ax, PTE_A_CLEAR; 0F000h; clear lower 12 bits
620 00004CC9 662500F0
621 00004CCD 89C2
622 00004CCF EB09
623
                                <1> spte_0:
624
                                <1> ; If this PTE contains a swap (disk) address,
                                         ; it can be updated by using 'swap_in' procedure
625
                               <1>
                               <1> ; only!
<1> and eax, eax
<1> jz short spte_1
<1> ; 24/07/2015
626
627 00004CD1 21C0
628 00004CD3 7403
629
                                       ; swapped page ! (on disk)
630
                                <1>
631 00004CD5 5B
                               <1>
                                         pop
                                                ebx
632
                               <1> spte_err:
                                        stc
633 00004CD6 F9
                                <1>
634 00004CD7 C3
                               <1>
                                         retn
                                <1> spte_1:
636 00004CD8 89D0
                                <1> mov
                                                eax, edx
637
                                <1> spte_2:
638 00004CDA 09CA
                                <1> or
                                                edx, ecx
639
                                <1>
                                          ; 23/06/2015
640 00004CDC 8913
                                                [ebx], edx ; PTE value in EDX
                                <1>
                                          mov
                                          ; 24/07/2015
641
                                <1>
642 00004CDE 5B
                                <1>
                                          pop ebx
643 00004CDF C3
                                <1>
                                          retn
644
                                <1>
                                <1> get_pde:
645
                                                ; Get present value of the relevant PDE
                                        ; 20/07/2015
646
                                <1>
647
                                <1>
                                         ; 18/04/2015
                                         ; 12/04/2015
                                <1>
648
                                         ; 10/10/2014 ; (Retro UNIX 386 v1 - beginning)
649
                                <1>
650
                                <1>
                                         ; INPUT ->
                                <1>
651
652
                                <1>
                                                EBX = virtual (linear) address
653
                                <1>
                                                EAX = page directory (physical) address
                                         ;
                                         ; OUTPUT ->
654
                                <1>
                                                EDX = Page directory entry address
655
                                <1>
                                         ;
                                                EAX = Page directory entry value
                                <1>
656
657
                                <1>
                                                CF = 1 \rightarrow PDE not present or invalid ?
                                         ; Modified Registers -> EDX, EAX
658
                                <1>
659
                                <1>
660 00004CE0 89DA
                                <1>
                                                edx, ebx
                                         mov
                                                edx, PAGE_D_SHIFT ; 22 (12+10)
661 00004CE2 C1EA16
                                <1>
                                          shr
662 00004CE5 C1E202
                                <1>
                                                edx, 2; offset to page directory (1024*4)
663 00004CE8 01C2
                                <1>
                                                edx, eax; page directory address (physical)
                                          add
```

```
664 00004CEA 8B02
                                <1>
                                         mov
                                               eax, [edx]
665 00004CEC A801
                                <1>
                                          test al, PDE A PRESENT; page table is present or not!
666 00004CEE 751F
                                <1>
                                          jnz
                                                short gpte_retn
667 00004CF0 F9
                                <1>
                                <1> gpde_retn:
668
669 00004CF1 C3
                                <1>
                                          retn
670
                                <1>
671
                                <1> get_pte:
672
                                <1>
                                                 ; Get present value of the relevant PTE
                                          ; 29/07/2015
673
                                <1>
                                        ; 20/07/2015
674
                                <1>
                                          ; 18/04/2015
675
                                <1>
                                          ; 12/04/2015
676
                                <1>
677
                                <1>
                                         ; 10/10/2014 ; (Retro UNIX 386 v1 - beginning)
678
                                <1>
                                        ; INPUT ->
; EBX
679
                                <1>
680
                                                EBX = virtual (linear) address
                                <1>
                                                EAX = page directory (physical) address
681
682
                                <1>
                                          ; OUTPUT ->
683
                                <1>
                                          ; EDX = Page table entry address (if CF=0)
684
                                <1>
                                                      Page directory entry address (if CF=1)
                                        ;
                                                       (Bit 0 value is 0 if PT is not present)
685
                                <1>
                                                EAX = Page table entry value (page address)
686
                                <1>
                                          ;
687
                                <1>
                                                CF = 1 -> PDE not present or invalid ?
                                          ; Modified Registers -> EAX, EDX
                                <1>
688
689
                                <1>
690 00004CF2 E8E9FFFFFF
                                <1>
                                         call get pde
                                                                  ; page table is not present
691 00004CF7 72F8
                                <1>
                                        jс
                                                short gpde_retn
692
                                <1>
                                          ;jnc short gpte_1
693
                                <1>
                                          ;retn
694
                                <1> ;gpte_1:
                                      and
                                                ax, PDE_A_CLEAR ; 0F000h ; clear lower 12 bits
695 00004CF9 662500F0
                                <1>
696 00004CFD 89DA
                                          mov
                                <1>
                                                edx, ebx
                               <1> shr edx, PAGE_SHIFT; 12
<1> and edx, PTE_MASK; 03FFh
<1> clear highe
697 00004CFF C1EA0C
698 00004D02 81E2FF030000
699
                                <1>
                                                       ; clear higher 10 bits (PD bits)
                               <1> shl
<1> add
<1> mov
700 00004D08 C1E202
                                                edx, 2; offset from start of page table (1024*4)
                                                edx, eax
701 00004D0B 01C2
702 00004D0D 8B02
                                <1>
                                          mov
                                                 eax, [edx]
                                <1> gpte_retn:
703
704 00004D0F C3
                                <1>
                                         retn
705
                                <1>
706
                                <1> deallocate_page_dir:
707
                                <1> ; 15/09/2015
                                        ; 05/08/2015
708
                                <1>
                                        ; 30/04/2015
; 28/04/2015
709
                                <1>
710
                                <1>
                                        ; 17/10/2014
711
                                <1>
712
                                <1>
                                          ; 12/10/2014 (Retro UNIX 386 v1 - beginning)
713
                                <1>
714
                                <1>
                                        ; INPUT ->
                                        ;
715
                                <1>
                                                EAX = PHYSICAL ADDRESS OF THE PAGE DIRECTORY (CHILD)
                                                 EBX = PHYSICAL ADDRESS OF THE PARENT'S PAGE DIRECTORY
716
                                <1>
717
                                <1>
                                          ; OUTPUT ->
718
                                <1>
                                                All of page tables in the page directory
719
                                <1>
                                                 and page dir's itself will be deallocated
                                                 except 'read only' duplicated pages (will be converted
720
                                <1>
                                          ;
                                                 to writable pages).
721
                                <1>
722
                                <1>
723
                                <1>
                                          ; Modified Registers -> EAX
724
                                <1>
                                <1>
726 00004D10 56
                                <1>
                                          push esi
727 00004D11 51
                                <1>
                                          push ecx
728 00004D12 50
                                <1>
                                          push eax
729 00004D13 89C6
                                <1>
                                          mov
                                                esi, eax
                                          xor ecx, ecx
730 00004D15 31C9
                                <1>
731
                                <1>
                                          ; The 1st PDE points to Kernel Page Table 0 (the 1st 4MB),
732
                                <1>
                                          ; it must not be deallocated
733 00004D17 890E
                                <1>
                                         mov [esi], ecx; 0; clear PDE 0
734
                                <1> dapd_0:
735 00004D19 AD
                                <1>
                                          lodsd
736 00004D1A A801
                                <1>
                                          test al, PDE_A_PRESENT; bit 0, present flag (must be 1)
737 00004D1C 7409
                                <1>
                                                 short dapd_1
                                     and ax, PDE_A_CLEAR; 0F0 call deallocate_page_table
                                                ax, PDE A CLEAR; 0F000h; clear lower 12 (attribute) bits
738 00004D1E 662500F0
                                <1>
739 00004D22 E812000000
                                <1>
                                <1> dapd_1:
741 00004D27 41
                                                 ecx ; page directory entry index
                                <1> inc
                                                 ecx, PAGE SIZE / 4 ; 1024
742 00004D28 81F900040000
                                <1>
                                          cmp
743 00004D2E 72E9
                                                 short dapd 0
                                <1>
                                          jЬ
744
                                <1> dapd_2:
745 00004D30 58
                                <1>
                                          pop
                                                 eax
746 00004D31 E87F000000
                                <1>
                                          call
                                               deallocate_page
                                                                    ; deallocate the page dir's itself
747 00004D36 59
                                <1>
                                          pop
748 00004D37 5E
                                <1>
                                                 esi
                                          pop
749 00004D38 C3
                                <1>
                                          retn
                                <1>
751
                                <1> deallocate_page_table:
752
                                <1>
                                          ; 12/07/2016
753
                                <1>
                                          ; 19/09/2015
754
                                <1>
                                          ; 15/09/2015
755
                                 <1>
                                          ; 05/08/2015
                                          ; 30/04/2015
756
                                <1>
757
                                <1>
                                          ; 28/04/2015
758
                                <1>
                                          ; 24/10/2014
759
                                <1>
                                          ; 23/10/2014
760
                                 <1>
                                          ; 12/10/2014 (Retro UNIX 386 v1 - beginning)
761
                                <1>
762
                                <1>
                                          ; INPUT ->
763
                                <1>
                                               EAX = PHYSICAL (real/flat) ADDRESS OF THE PAGE TABLE
                                                 EBX = PHYSICAL ADDRESS OF THE PARENT'S PAGE DIRECTORY
764
                                <1>
765
                                 <1>
                                                 (ECX = page directory entry index)
                                          ; OUTPUT ->
766
                                <1>
767
                                 <1>
                                                 All of pages in the page table and page table's itself
768
                                 <1>
                                                 will be deallocated except 'read only' duplicated pages
```

```
769
                                <1>
                                                (will be converted to writable pages).
770
                                <1>
771
                                <1>
                                          ; Modified Registers -> EAX
772
                                <1>
773 00004D39 56
                                <1>
                                          push esi
774 00004D3A 57
                                <1>
                                          push edi
775 00004D3B 52
                               <1>
                                          push edx
                                          push eax; *
776 00004D3C 50
                                <1>
777 00004D3D 89C6
                                <1>
                                                esi, eax
                                          mov
778 00004D3F 31FF
                               <1>
                                               edi, edi ; 0
                                          xor
779
                                <1> dapt_0:
780 00004D41 AD
                                <1>
                                          lodsd
                                          test al, PTE_A_PRESENT ; bit 0, present flag (must be 1)
781 00004D42 A801
                                <1>
782 00004D44 7441
                                <1>
                                                short dapt_1
783
                                <1>
                                          test al, PTE_A_WRITE ; bit 1, writable (r/w) flag
784 00004D46 A802
                                <1>
                                                               ; (must be 1)
                                <1>
                                          jnz short dapt 3
786 00004D48 754C
                                <1>
787
                                <1>
                                          ; Read only -duplicated- page (belongs to a parent or a child)
788 00004D4A 66A90002
                                <1>
                                          test ax, PTE_DUPLICATED; Was this page duplicated
789
                                <1>
                                                                ; as child's page ?
790 00004D4E 7451
                                              short dapt 4; Clear PTE but don't deallocate the page!
                                <1>
                                          jz
791
                                          ; check the parent's PTE value is read only & same page or not..
                                <1>
792
                                <1>
                                          ; ECX = page directory entry index (0-1023)
793 00004D50 53
                                <1>
                                          push ebx
794 00004D51 51
                                <1>
                                          push ecx
795 00004D52 66C1E102
                               <1>
                                          shl cx, 2; *4
796 00004D56 01CB
                               <1>
                                          add ebx, ecx; PDE offset (for the parent)
797 00004D58 8B0B
                                <1>
                                          mov
                                                ecx, [ebx]
798 00004D5A F6C101
                                         test cl, PDE_A_PRESENT; present (valid) or not?
                               <1>
799 00004D5D 7435
                               <1>
                                          jz
                                                short dapt_2 ; parent process does not use this page
800 00004D5F 6681E100F0
                                               cx, PDE A CLEAR; 0F000h; Clear attribute bits
                                <1>
                                          and
801
                                <1>
                                         ; EDI = page table entry index (0-1023)
802 00004D64 89FA
                               <1>
                                         mov edx, edi
803 00004D66 66C1E202
                               <1>
                                         shl
                                               dx, 2 ; *4
804 00004D6A 01CA
                               <1>
                                          add
                                               edx, ecx; PTE offset (for the parent)
805 00004D6C 8B1A
                               <1>
                                         mov
                                               ebx, [edx]
806 00004D6E F6C301
                               <1>
                                          test bl, PTE_A_PRESENT ; present or not ?
807 00004D71 7421
                                <1>
                                                short dapt_2 ; parent process does not use this page
                                          jΖ
808 00004D73 662500F0
                               <1>
                                          and
                                                ax, PTE A CLEAR; OF000h; Clear attribute bits
809 00004D77 6681E300F0
                               <1>
                                          and bx, PTE_A_CLEAR; 0F000h; Clear attribute bits
810 00004D7C 39D8
                                <1>
                                          cmp
                                               eax, ebx ; parent's and child's pages are same ?
                                                short dapt_2 ; not same page
811 00004D7E 7514
                                <1>
                                          jne
812
                                <1>
                                                             ; deallocate the child's page
                                          or
813 00004D80 800A02
                                <1>
                                                   byte [edx], PTE_A_WRITE ; convert to writable page (parent)
814 00004D83 59
                                <1>
                                          pop
                                                ecx
815 00004D84 5B
                               <1>
                                                ebx
                                          pop
816 00004D85 EB1A
                                <1>
                                                short dapt_4
                                          jmp
                                <1> dapt 1:
817
818 00004D87 09C0
                                <1>
                                                           ; swapped page ?
                                          or
                                                eax, eax
819 00004D89 741D
                                <1>
                                                short dapt_5 ; no
820
                                <1>
                                                            ; yes
821 00004D8B D1E8
                                <1>
                                          shr
                                                eax, 1
                                          call
822 00004D8D E8CA040000
                                <1>
                                               unlink_swap_block ; Deallocate swapped page block
                                <1>
                                                               ; on the swap disk (or in file)
823
824 00004D92 EB14
                                <1>
                                                short dapt_5
                                <1> dapt_2:
826 00004D94 59
                                <1>
                                          pop
                                                ecx
827 00004D95 5B
                                <1>
                                          pop
                                                ebx
                                <1> dapt_3:
828
829
                                <1>
                                          ; 12/07/2016
830 00004D96 66A90004
                                <1>
                                          test ax, PTE SHARED; shared or direct memory access indicator
831 00004D9A 7505
                                <1>
                                                short dapt 4 ; AVL bit 1 = 1, do not deallocate this page!
                                          jnz
                                <1>
832
833
                                <1>
                                          ;and ax, PTE_A_CLEAR; 0F000h; clear lower 12 (attribute) bits
834 00004D9C E814000000
                                <1>
                                          call
                                                deallocate_page ; set the mem allocation bit of this page
                                <1> dapt_4:
836 00004DA1 C746FC00000000
                                <1>
                                                dword [esi-4], 0 ; clear/reset PTE (child, dupl. as parent)
                                          mov
                                <1> dapt_5:
837
838 00004DA8 47
                                <1>
                                                edi ; page table entry index
                                          inc
839 00004DA9 81FF00040000
                                <1>
                                          cmp
                                                edi, PAGE_SIZE / 4 ; 1024
840 00004DAF 7290
                                <1>
                                          jb
                                                short dapt 0
                                <1>
841
                                          ;
                                                eax ; *
842 00004DB1 58
                                <1>
                                          pop
843 00004DB2 5A
                                <1>
                                          pop
                                                edx
844 00004DB3 5F
                                <1>
                                                edi
                                          pop
845 00004DB4 5E
                                <1>
                                          pop
                                                esi
846
                                <1>
                                                                   ; deallocate the page table's itself
847
                                <1>
                                          ;call deallocate_page
848
                                <1>
                                          :retn
849
                                <1>
850
                                <1> deallocate page:
                                         ; 15/09/2015
851
                                <1>
852
                                <1>
                                          ; 28/04/2015
853
                                <1>
                                         ; 10/03/2015
                                          ; 17/10/2014
854
                                <1>
855
                                <1>
                                         ; 12/10/2014 (Retro UNIX 386 v1 - beginning)
856
                                <1>
857
                                <1>
                                         ; INPUT ->
                                               EAX = PHYSICAL (real/flat) ADDRESS OF THE ALLOCATED PAGE
858
                                <1>
                                          ; OUTPUT ->
859
                                <1>
860
                                <1>
                                                [free pages] is increased
                                <1>
                                                (corresponding MEMORY ALLOCATION TABLE bit is SET)
861
862
                                <1>
                                                CF = 1 if the page is already deallocated
863
                                <1>
                                                       (or not allocated) before.
864
                                <1>
                                          ; Modified Registers -> EAX
865
                                <1>
                                <1>
866
867 00004DB5 53
                                <1>
                                          push
868 00004DB6 52
                                          push
                                <1>
                                               edx
869
                                <1>
870 00004DB7 C1E80C
                                <1>
                                          shr
                                                eax, PAGE SHIFT
                                                                    ; shift physical address to
871
                                <1>
                                                                  ; 12 bits right
872
                                <1>
                                                                  ; to get page number
873 00004DBA 89C2
                                <1>
                                                edx, eax
                                          mov
```

```
874
                                                                                   <1>
                                                                                                            ; 15/09/2015
875 00004DBC C1EA03
                                                                                   <1>
                                                                                                                          edx, 3
                                                                                                                                                                         ; to get offset to M.A.T.
                                                                                                                                                                          ; (1 allocation bit = 1 page)
876
                                                                                   <1>
877
                                                                                   <1>
                                                                                                                                                                         ; (1 allocation bytes = 8 pages)
                                                                                                                                                                          ; clear lower 2 bits
878 00004DBF 80E2FC
                                                                                   <1>
                                                                                                            and
                                                                                                                             dl, OFCh
879
                                                                                   <1>
                                                                                                                                                                           ; (to get 32 bit position)
880
                                                                                   <1>
881 00004DC2 BB00001000
                                                                                                                             ebx, MEM_ALLOC_TBL ; Memory Allocation Table address
                                                                                   <1>
                                                                                                            mov
882 00004DC7 01D3
                                                                                   <1>
                                                                                                            add
                                                                                                                             ebx, edx
883 00004DC9 83E01F
                                                                                   <1>
                                                                                                                            eax, 1Fh
                                                                                                            and
                                                                                                                                                                          ; lower 5 bits only
884
                                                                                   <1>
                                                                                                                                                                           ; (allocation bit position)
885 00004DCC 3B15[CC580100]
                                                                                   <1>
                                                                                                                             edx, [next page]
                                                                                                                                                                                ; is the new free page address lower
                                                                                                            cmp
886
                                                                                   <1>
                                                                                                                                                                           ; than the address in 'next_page' ?
887
                                                                                   <1>
                                                                                                                                                                           ; (next/first free page value)
888 00004DD2 7306
                                                                                   <1>
                                                                                                            jnb
                                                                                                                             short dap 1
                                                                                                                                                                          ; no
889 00004DD4 8915[CC580100]
                                                                                   <1>
                                                                                                            mov
                                                                                                                             [next_page], edx
                                                                                                                                                                                ; yes
                                                                                   <1> dap_1:
891 00004DDA 0FAB03
                                                                                                                                                                           ; unlink/release/deallocate page
                                                                                   <1>
                                                                                                            bts
                                                                                                                             [ebx], eax
892
                                                                                   <1>
                                                                                                                                                                           ; set relevant bit to 1.
893
                                                                                   <1>
                                                                                                                                                                           ; set CF to the previous bit value
894
                                                                                   <1>
                                                                                                            ;cmc
                                                                                                                                                                           ; complement carry flag
895
                                                                                   <1>
                                                                                                            ;jc
                                                                                                                             short dap 2
                                                                                                                                                                          ; do not increase free pages count
                                                                                   <1>
                                                                                                                                                                           ; if the page is already deallocated % \left( 1\right) =\left( 1\right) \left( 1\right) \left(
896
897
                                                                                   <1>
                                                                                                                                     dword [free_pages]
898 00004DDD FF05[C8580100]
                                                                                   <1>
                                                                                                                 inc
899
                                                                                   <1> dap_2:
900 00004DE3 5A
                                                                                   <1>
                                                                                                                             edx
                                                                                                           pop
901 00004DE4 5B
                                                                                   <1>
                                                                                                            pop
                                                                                                                             ebx
902 00004DE5 C3
                                                                                   <1>
903
                                                                                   <1>
904
                                                                                   905
                                                                                   <1> ;;
                                                                                   <1> ;; Copyright (C) KolibriOS team 2004-2012. All rights reserved. ;;
906
907
                                                                                   <1> ;; Distributed under terms of the GNU General Public License
                                                                                                                                                                                                                                                           ;;
                                                                                   <1>;;
908
                                                                                                                                                                                                                                                               ;;
909
                                                                                   910
911
                                                                                   <1> ;;$Revision: 5057 $
912
                                                                                   <1>
913
                                                                                   <1>
914
                                                                                   <1> ;;align 4
915
                                                                                   <1> ;;proc alloc_page
916
                                                                                   <1>
917
                                                                                   <1> ;;
                                                                                                                      pushfd
                                                                                   <1> ;;
918
                                                                                                                      cli
919
                                                                                   <1> ;;
                                                                                                                      push
                                                                                   <1> ;;;//-
920
921
                                                                                   <1> ;;
                                                                                                                                           [pg_data.pages_free], 1
                                                                                                                       cmp
922
                                                                                   <1> ;;
                                                                                                                       jle
                                                                                                                                           .out_of_memory
923
                                                                                   <1> ;;;//-
924
                                                                                   <1> ;;
925
                                                                                   <1> ;;
                                                                                                                      mov
                                                                                                                                           ebx, [page start]
                                                                                   <1> ;;
926
                                                                                                                      mov
                                                                                                                                           ecx, [page_end]
927
                                                                                   <1> ;;.11:
928
                                                                                   <1> ;;
                                                                                                                      bsf
                                                                                                                                           eax, [ebx];
929
                                                                                   <1> ;;
                                                                                                                       jnz
                                                                                                                                           .found
930
                                                                                   <1> ;;
                                                                                                                       add
                                                                                                                                           ebx, 4
                                                                                   <1> ;;
931
                                                                                                                                           ebx, ecx
                                                                                                                       cmp
932
                                                                                   <1> ;;
                                                                                                                       jb
                                                                                                                                            .11
933
                                                                                   <1> ;;
                                                                                                                      pop
                                                                                                                                           ebx
934
                                                                                   <1> ;;
                                                                                                                      popfd
935
                                                                                   <1> ;;
                                                                                                                       xor
                                                                                                                                           eax, eax
                                                                                   <1> ;;
936
                                                                                                                      ret
937
                                                                                   <1> ;;.found:
938
                                                                                   <1> ;;;//-
939
                                                                                   <1> ;;
                                                                                                                       dec
                                                                                                                                            [pg_data.pages_free]
940
                                                                                   <1> ;;
                                                                                                                       jΖ
                                                                                                                                           .out_of_memory
                                                                                   <1> ;;;//-
941
942
                                                                                   <1> ;;
                                                                                                                      btr
                                                                                                                                            [ebx], eax
                                                                                   <1> ;;
943
                                                                                                                                            [page_start], ebx
                                                                                                                      mov
                                                                                                                                           ebx, sys pgmap
944
                                                                                   <1> ;;
                                                                                                                       sub
945
                                                                                   <1> ;;
                                                                                                                       lea
                                                                                                                                           eax, [eax+ebx*8]
                                                                                   <1> ;;
946
                                                                                                                                           eax, 12
                                                                                                                       shl
947
                                                                                   <1> ;;;//-
                                                                                                                           dec [pg_data.pages_free]
                                                                                   <1> ;;
948
                                                                                                                      pop
                                                                                                                                           ebx
949
                                                                                   <1> ;;
                                                                                                                      popfd
950
                                                                                   <1>;;
                                                                                                                       ret
                                                                                   <1> ;;;//-
951
952
                                                                                    <1> ;;.out_of_memory:
953
                                                                                   <1> ;;
                                                                                                                                           [pg_data.pages_free], 1
                                                                                                                      mov
954
                                                                                   <1> ;;
                                                                                                                       xor
                                                                                                                                           eax, eax
955
                                                                                    <1> ;;
                                                                                                                                           ebx
                                                                                                                       pop
956
                                                                                   <1> ;;
                                                                                                                      popfd
957
                                                                                   <1> ;;
                                                                                   <1> ;;;//-
958
959
                                                                                   <1> ;;endp
960
                                                                                   <1>
961
                                                                                   <1> duplicate_page_dir:
                                                                                                           ; 2\overline{1}/09/\overline{2}015
962
                                                                                   <1>
963
                                                                                   <1>
                                                                                                           ; 31/08/2015
                                                                                                          ; 20/07/2015
964
                                                                                   <1>
965
                                                                                   <1>
                                                                                                          ; 28/04/2015
                                                                                                           ; 27/04/2015
966
                                                                                   <1>
967
                                                                                   <1>
                                                                                                          ; 18/04/2015
968
                                                                                   <1>
                                                                                                          ; 12/04/2015
969
                                                                                   <1>
                                                                                                           ; 18/10/2014
970
                                                                                   <1>
                                                                                                          ; 16/10/2014 (Retro UNIX 386 v1 - beginning)
971
                                                                                   <1>
                                                                                                           ; INPUT ->
972
                                                                                   <1>
                                                                                                                          [u.pgdir] = PHYSICAL (real/flat) ADDRESS of the parent's
973
                                                                                   <1>
                                                                                                           ;
                                                                                                                                                      page directory.
974
                                                                                   <1>
975
                                                                                   <1>
                                                                                                            ; OUTPUT ->
                                                                                                                           EAX = PHYSICAL (real/flat) ADDRESS of the child's
976
                                                                                   <1>
                                                                                                            ;
977
                                                                                   <1>
                                                                                                                                           page directory.
978
                                                                                   <1>
                                                                                                                            (New page directory with new page table entries.)
```

```
979
                                 <1>
                                                  (New page tables with read only copies of the parent's
 980
                                 <1>
                                                  pages.)
                                           ;
 981
                                 <1>
                                                  EAX = 0 \rightarrow Error (CF = 1)
                                           ;
 982
                                 <1>
 983
                                 <1>
                                           ; Modified Registers -> none (except EAX)
 984
                                 <1>
 985 00004DE6 E8ECFDFFFF
                                 <1>
                                           call allocate page
 986 00004DEB 723E
                                 <1>
                                                 short dpd_err
                                           jс
987
                                 <1>
                                           push ebp ; 20/07/2015
 988 00004DED 55
                                 <1>
 989 00004DEE 56
                                 <1>
                                           push esi
 990 00004DEF 57
                                 <1>
                                           push edi
 991 00004DF0 53
                                 <1>
                                           push ebx
992 00004DF1 51
                                 <1>
                                           push ecx
                                           mov
 993 00004DF2 8B35[B8030300]
                                 <1>
                                                 esi, [u.pgdir]
994 00004DF8 89C7
                                 <1>
                                           mov
                                                 edi, eax
 995 00004DFA 50
                                           push eax; save child's page directory address
                                 <1>
                                           ; 31/08/2015
996
                                 <1>
                                           ; copy PDE 0 from the parent's page dir to the child's page dir
997
                                 <1>
                                           ; (use same system space for all user page tables)
998
                                 <1>
999 00004DFB A5
                                           movsd
                                 <1>
1000 00004DFC BD00004000
                                                 ebp, 1024*4096; pass the 1st 4MB (system space)
                                 <1>
                                           mov
1001 00004E01 B9FF030000
                                                 ecx, (PAGE_SIZE / 4) - 1; 1023
                                 <1>
                                           mov
                                 <1> dpd_0:
1003 00004E06 AD
                                 <1>
                                           lodsd
1004
                                 <1>
                                           ;or eax, eax
1005
                                 <1>
                                            jnz short dpd 1;
                                           test al, PDE_A_PRESENT; bit 0 = 1
1006 00004E07 A801
                                 <1>
1007 00004E09 7508
                                 <1>
                                                 short dpd_1
                                           ; 20/07/2015 (virtual address at the end of the page table)
1008
                                 <1>
1009 00004E0B 81C500004000
                                 <1>
                                           add ebp, 1024*4096; page size * PTE count
1010 00004E11 EB0F
                                 <1>
                                           jmp
                                                 short dpd 2
                                 <1> dpd_1:
1011
1012 00004E13 662500F0
                                 <1>
                                                 ax, PDE_A_CLEAR ; OF000h ; clear attribute bits
1013 00004E17 89C3
                                 <1>
                                           mov
                                                 ebx, eax
1014
                                 <1>
                                           ; EBX = Parent's page table address
1015 00004E19 E81F000000
                                 <1>
                                           call duplicate page table
1016 00004E1E 720C
                                 <1>
                                           jc short dpd_p_err
1017
                                 <1>
                                           ; EAX = Child's page table address
1018 00004E20 0C07
                                 <1>
                                           or al, PDE_A_PRESENT + PDE_A_WRITE + PDE_A_USER
1019
                                 <1>
                                                         ; set bit 0, bit 1 and bit 2 to 1
                                                         ; (present, writable, user)
1020
                                 <1>
1021
                                 <1> dpd_2:
1022 00004E22 AB
                                 <1>
                                           stosd
1023 00004E23 E2E1
                                 <1>
                                           loop dpd_0
1024
                                 <1>
1025 00004E25 58
                                                  eax ; restore child's page directory address
                                 <1>
                                           pop
                                 <1> dpd_3:
1026
1027 00004E26 59
                                 <1>
                                           pop
                                                 ecx
1028 00004E27 5B
                                 <1>
                                                 ebx
                                           pop
1029 00004E28 5F
                                 <1>
                                                  edi
                                           pop
1030 00004E29 5E
                                 <1>
                                                  esi
                                           pop
1031 00004E2A 5D
                                                 ebp ; 20/07/2015
                                 <1>
                                           pop
1032
                                 <1> dpd_err:
1033 00004E2B C3
                                 <1>
                                           retn
1034
                                 <1> dpd_p_err:
                                 <1>
                                           ; release the allocated pages missing (recover free space)
1036 00004E2C 58
                                 <1>
                                           pop eax ; the new page directory address (physical)
1037 00004E2D 8B1D[B8030300]
                                 <1>
                                                  ebx, [u.pgdir] ; parent's page directory address
1038 00004E33 E8D8FEFFFF
                                 <1>
                                           call deallocate page dir
1039 00004E38 29C0
                                 <1>
                                           sub
                                                 eax, eax ; 0
1040 00004E3A F9
                                 <1>
                                           stc
1041 00004E3B EBE9
                                 <1>
                                                 short dpd_3
                                           jmp
1042
                                 <1>
1043
                                 <1> duplicate_page_table:
                                        ; 20/02/\overline{2}017
1044
                                 <1>
                                           ; 21/09/2015
1045
                                 <1>
                                          ; 20/07/2015
1046
                                 <1>
                                           ; 05/05/2015
1047
                                 <1>
                                           ; 28/04/2015
1048
                                 <1>
1049
                                 <1>
                                           ; 27/04/2015
1050
                                 <1>
                                           ; 18/04/2015
                                           ; 18/10/2014
1051
                                 <1>
                                           ; 16/10/2014 (Retro UNIX 386 v1 - beginning)
1052
                                 <1>
1053
                                 <1>
1054
                                 <1>
                                           ; INPUT ->
1055
                                  <1>
                                                 EBX = PHYSICAL (real/flat) ADDRESS of the parent's page table.
1056
                                 <1>
                                                  20/02/2017
1057
                                  <1>
                                                 EBP = Linear address of the page (from 'duplicate_page_dir')
1058
                                 <1>
                                                       (Linear address = CORE + user's virtual address)
                                           ; OUTPUT ->
1059
                                  <1>
                                                  EAX = PHYSICAL (real/flat) ADDRESS of the child's page table.
1060
                                  <1>
                                                        (with 'read only' attribute of page table entries)
1061
                                  <1>
1062
                                  <1>
                                                  20/02/2017
1063
                                  <1>
                                                  EBP = Next linear page address (for 'duplicate_page_dir')
1064
                                 <1>
                                                 CF = 1 \rightarrow error
1065
                                 <1>
1066
                                 <1>
1067
                                 <1>
                                           ; Modified Registers -> EBP (except EAX)
                                 <1>
1068
1069 00004E3D E895FDFFFF
                                 <1>
                                           call allocate_page
1070 00004E42 726A
                                 <1>
                                           jс
                                                 short dpt_err
1071
                                 <1>
1072 00004E44 50
                                           push eax; *
                                 <1>
1073 00004E45 56
                                           push esi
                                 <1>
1074 00004E46 57
                                 <1>
                                           push
                                                 edi
1075 00004E47 52
                                           push
                                 <1>
                                                edx
1076 00004E48 51
                                 <1>
                                           push
                                                 ecx
1077
                                 <1>
1078 00004E49 89DE
                                 <1>
                                                  esi, ebx
                                           mov
1079 00004E4B 89C7
                                 <1>
                                           mov
                                                  edi, eax
1080 00004E4D 89C2
                                 <1>
                                           mov
                                                  edx, eax
1081 00004E4F 81C200100000
                                 <1>
                                           add
                                                 edx, PAGE_SIZE
1082
                                 <1> dpt_0:
1083 00004E55 AD
                                 <1>
                                           lodsd
```

```
1084 00004E56 21C0
                                <1>
                                          and eax, eax
                                          jz short dpt_3
test al, PTE_A_PRESENT; bit 0 = 1
1085 00004E58 7444
                                <1>
1086 00004E5A A801
                                <1>
                                          jnz short dpt_1
1087 00004E5C 7507
                                <1>
1088
                                <1>
                                          ; 20/07/2015
                                <1>
1089
                                           ; ebp = virtual (linear) address of the memory page
1090 00004E5E E83F050000
                                           call reload page ; 28/04/2015
                                <1>
1091 00004E63 7244
                                <1>
                                         jc short dpt_p_err
1092
                                <1> dpt_1:
                                         ; 21/09/2015
1093
                                <1>
1094 00004E65 89C1
                                <1>
                                          mov ecx, eax
1095 00004E67 662500F0
                                                ax, PTE A CLEAR ; OF000h ; clear attribute bits
                                <1>
                                      and ax, FTE_A_CLEAR; 0F000h; clear a
test cl, PTE_A_WRITE; writable page?
jnz short dpt_2
; Read only (parent) page
                                          and
1096 00004E6B F6C102
                                <1>
1097 00004E6E 7525
                                <1>
1098
                                <1>
1099
                                          ; - there is a third process which uses this page -
                                <1>
                                       ; Allocate a new page call allocate_page
                                          ; Allocate a new page for the child process
1100
                                <1>
1101 00004E70 E862FDFFFF
                                <1>
                                                 short dpt p err
1102 00004E75 7232
                                <1>
                                          jс
                                          push edi
1103 00004E77 57
                                <1>
1104 00004E78 56
                                          push esi
                                <1>
1105 00004E79 89CE
                                <1>
                                          mov
                                                 esi, ecx
1106 00004E7B 89C7
                                <1>
                                                edi, eax
                                          mov
1107 00004E7D B900040000
                                <1>
                                                 ecx, PAGE_SIZE/4
                                                movsd ; copy page (4096 bytes)
1108 00004E82 F3A5
                                <1>
                                          rep
1109 00004E84 5E
                                <1>
                                          pop
                                                 esi
1110 00004E85 5F
                                <1>
                                                 edi
                                          pop
1111
                                <1>
1112 00004E86 53
                                <1>
                                          push ebx
1113 00004E87 50
                                <1>
                                          push eax
1114
                                <1>
                                          ; 20/07/2015
1115 00004E88 89EB
                                 <1>
                                          mov ebx, ebp
                                <1>
                                          ; ebx = virtual (linear) address of the memory page
1116
1117 00004E8A E887030000
                                <1>
                                          call add_to_swap_queue
                                        pop eax
pop ebx
1118 00004E8F 58
                                 <1>
1119 00004E90 5B
                                <1>
                                          pop
                                          ; 21/09/2015
                                <1>
1121 00004E91 0C07
                                 <1>
                                           or
                                                al, PTE_A_USER+PTE_A_WRITE+PTE_A_PRESENT
1122
                                 <1>
                                                 ; user + writable + present page
1123 00004E93 EB09
                                 <1>
                                                short dpt_3
                                           jmp
                                 <1> dpt_2:
1124
                                                 ax, PTE A USER+PTE A PRESENT
1125
                                 <1>
                                          ;or
                                                 al, PTE A USER+PTE A PRESENT
1126 00004E95 0C05
                                 <1>
                                           or
                                                    ; (read only page!)
1127
                                 <1>
                                                 [esi-4], eax; update parent's PTE
1128 00004E97 8946FC
                                 <1>
                                           mov
1129 00004E9A 660D0002
                                 <1>
                                           or
                                                  ax, PTE_DUPLICATED ; (read only page & duplicated PTE!)
                                 <1> dpt_3:
1131 00004E9E AB
                                          stosd ; EDI points to child's PTE
                                 <1>
1132
                                 <1>
1133 00004E9F 81C500100000
                                <1>
                                           add
                                                 ebp, 4096; 20/07/2015 (next page)
1134
                                 <1>
1135 00004EA5 39D7
                                 <1>
                                           cmp
                                                 edi, edx
1136 00004EA7 72AC
                                 <1>
                                           jb
                                                 short dpt_0
1137
                                 <1> dpt_p_err:
1138 00004EA9 59
                                 <1>
                                          pop
                                                 ecx
1139 00004EAA 5A
                                <1>
                                                 edx
                                           pop
1140 00004EAB 5F
                                <1>
                                                 edi
                                           pop
1141 00004EAC 5E
                                <1>
                                           pop
                                                 esi
1142 00004EAD 58
                                 <1>
                                          pop
                                                 eax ; *
                                 <1> dpt_err:
1143
1144 00004EAE C3
                                 <1>
1145
                                 <1>
                                 <1> page_fault_handler:; CPU EXCEPTION 0Eh (14) : Page Fault !
1146
                                        ; 21/09/2015
1147
                                 <1>
1148
                                 <1>
                                          ; 19/09/2015
                                          ; 17/09/2015
1149
                                 <1>
                                          ; 28/08/2015
1150
                                 <1>
                                         ; 20/07/2015
                                 <1>
1151
1152
                                 <1>
                                          ; 28/06/2015
                                          ; 03/05/2015
1153
                                 <1>
                                          ; 30/04/2015
1154
                                 <1>
                                          ; 18/04/2015
1155
                                 <1>
                                          ; 12/04/2015
1156
                                 <1>
                                         ; 30/10/2014
1157
                                 <1>
                                          ; 11/09/2014
1158
                                 <1>
                                          ; 10/09/2014 (Retro UNIX 386 v1 - beginning)
1159
                                 <1>
1160
1161
                                 <1>
                                          ; Note: This is not an interrupt/exception handler.
1162
                                 <1>
                                                 This is a 'page fault remedy' subroutine
                                                 which will be called by standard/uniform
1163
                                 <1>
1164
                                 <1>
                                                 exception handler.
1165
                                 <1>
                                           ; INPUT ->
1166
                                 <1>
1167
                                 <1>
                                                 [error_code] = 32 bit ERROR CODE (lower 5 bits are valid)
                                 <1>
1168
                                                 cr2 = the virtual (linear) address
1169
                                 <1>
                                                       which has caused to page fault (19/09/2015)
1170
                                 <1>
1171
                                 <1>
1172
                                 <1>
                                          ; OUTPUT ->
1173
                                 <1>
                                                 (corresponding PAGE TABLE ENTRY is mapped/set)
                                                 EAX = 0 \rightarrow no error
1174
                                 <1>
1175
                                 <1>
                                                 EAX > 0 \rightarrow error code in EAX (also CF = 1)
1176
                                 <1>
                                          ; Modified Registers -> none (except EAX)
1177
                                 <1>
1178
                                 <1>
1179
                                 <1>
1180
                                 <1>
                                            ; ERROR CODE:
                                                 31 .... 4 3 2 1 0
                                 <1>
1181
1182
                                 <1>
1183
                                 <1>
                                                 | Reserved | I | R | U | W | P |
1184
                                 <1>
                                                 +---+--
1185
                                 <1>
                                           1186
                                 <1>
                                                      a page-protection violation. When not set,
1187
                                 <1>
1188
                                 <1>
                                                       it was caused by a non-present page.
```

```
; W : WRITE - When set, the page fault was caused by
1189
                              <1>
                                     ; a page write. When not set, it was caused by a page read.
1190
                              <1>
1191
                              <1>
                                       ; U : USER - When set, the page fault was caused
1192
                              <1>
                                      ; while CPL = 3.
; This does not necessarily mean that
1193
                              <1>
                                       ; This does not necessarily mean that; the page fault was a privilege violation.
1194
                              <1>
1195
                                     ; R : RESERVD - When set, the page fault was caused by
1196
                              <1>
1197
                              <1>
                                       ; WRITE reading a 1 in a reserved field.
                                       ; I : INSTRUC - When set, the page fault was caused by
1198
                              <1>
1199
                              <1>
                                       ; FETCH an instruction fetch
1200
                              <1>
                                       ;; x86 (32 bit) VIRTUAL ADDRESS TRANSLATION
1201
                              <1>
                                       ; 31 22 12 11
1202
                              <1>
1203
                              <1>
                                                 -----
                                           ; | PAGE DIR. ENTRY # | PAGE TAB. ENTRY # | OFFSET
1204
                              <1>
1205
1206
                              <1>
1207
                              <1>
1208
                              <1>
                                       ;; CR3 REGISTER (Control Register 3)
                                                                           12 5 4 3 2 0
1209
                              <1>
1210
                              <1>
                                                                | | | | | | | |
                                            ; |
1211
                              <1>
1212
                              <1>
                                           ; | PAGE DIRECTORY TABLE BASE ADDRESS | reserved |C|W|rsvrd|
                                                                                 | |D|T| |
1213
                              <1>
                                           ; |
1214
                              <1>
1215
                              <1>
                                            PWT - WRITE THROUGH
PCD - CACHE DISABLE
1216
                              <1>
1217
                              <1>
1218
                              <1>
1219
                              <1>
1220
                              <1>
                                       ;; x86 PAGE DIRECTORY ENTRY (4 KByte Page)
                                                                 12 11 9 8 7 6 5 4 3 2 1 0
1221
                              <1>
                                       1222
                              <1>
1223
                              <1>
1224
                              <1>
1225
                              <1>
1226
                              <1>
1227
                              <1>
                                       ; P - PRESENT
; R/W - READ/WRITE
; U/S - USER/SUPERVISOR
; PWT - WRITE THROUGH
1228
                              <1>
1229
                              <1>
1230
                              <1>
1231
                              <1>
                                       ; PCD - CACHE DISABLE
1232
                              <1>
                                       ; A - ACCESSED
; D - DIRTY (IGNORED)
1233
                              <1>
1234
                              <1>
                                     ; PAT - PAGE ATTRIBUTE TABLE INDEX (CACHE BEHAVIOR)
; G - GLOBAL (IGNORED)
1235
                              <1>
1236
                              <1>
                                              AVL - AVAILABLE FOR SYSTEMS PROGRAMMER USE
1237
                              <1>
1238
                              <1>
1239
                              <1>
1240
                              <1>
                                       ;; x86 PAGE TABLE ENTRY (4 KByte Page)
                                                                          12 11 9 8 7 6 5 4 3 2 1 0
1241
                              <1>
                                       ; 31
                                                                  1242
                              <1>
1243
                              <1>
                                            ; |
                                                  1244
                              <1>
1245
                              <1>
1246
                              <1>
1247
                              <1>
                                      ; P - PRESENT; R/W - READ/WRITE; U/S - USER/SUPERVISOR; PWT - WRITE THROUGH
1248
                              <1>
1249
                              <1>
1250
                              <1>
1251
                              <1>
1252
                              <1>
                                            PCD - CACHE DISABLE
                                            A - ACCESSED D - DIRTY
1253
                              <1>
                                       ; PAT - PAGE ATTRIBUTE TABLE INDEX (CACHE BEHAVIOR)
; G - GLOBAL
1254
                              <1>
1255
                              <1>
1256
                              <1>
1257
                              <1>
                                               AVL - AVAILABLE FOR SYSTEMS PROGRAMMER USE
1258
                              <1>
1259
                              <1>
                                       ;; 80386 PAGE TABLE ENTRY (4 KByte Page)
1260
                              <1>
                                       ; 31
                                                                    12 11 9 8 7 6 5 4 3 2 1 0
1261
                              <1>
                                                    1262
                              <1>
                                            ; | PAGE FRAME BASE ADDRESS 31..12 | AVL |0|0|D|A|0|0|/|/|P|; |
1263
                              <1>
1264
                              <1>
1265
                              <1>
1266
                              <1>
1267
                              <1>
1268
                                                Р
                                                     - PRESENT
                              <1>
                                             R/W - READ/WRITE
U/S - USER/SUPERVISOR
D - DIRTY
1269
                              <1>
1270
                               <1>
1271
1272
                              <1>
                                                      - AVAILABLE FOR SYSTEMS PROGRAMMER USE
1273
                              <1>
                                                NOTE: 0 INDICATES INTEL RESERVED. DO NOT DEFINE.
1274
                              <1>
1275
                              <1>
1276
                              <1>
1277
                              <1>
                                       ;; Invalid Page Table Entry
1278
                              <1>
                                       ; 31
1279
                              <1>
1280
                              <1>
                                             ; |
                                                                                                       |0|
1281
                              <1>
                                                                     AVAILABLE
                                             ;
1282
                              <1>
1283
                              <1>
1284
                              <1>
1285
                              <1>
1286 00004EAF 53
                              <1>
                                       push ebx
1287 00004EB0 52
                              <1>
                                       push
                                             edx
1288 00004EB1 51
                              <1>
                                       push ecx
1289
                              <1>
1290
                              <1>
                                       ; 21/09/2015 (debugging)
1291 00004EB2 FF05[CC030300]
                             <1>
                                       inc dword [u.pfcount] ; page fault count for running process
1292 00004EB8 FF05[80050300]
                             <1>
                                       inc dword [PF_Count] ; total page fault count
1293
                              <1>
                                       ; 28/06/2015
```

```
1294
                                 <1>
                                           ;mov
                                                edx, [error_code] ; Lower 5 bits are valid
1295 00004EBE 8A15[78050300]
                                 <1>
                                           mov
                                                 dl, [error code]
1296
                                 <1>
1297 00004EC4 F6C201
                                 <1>
                                           test dl, 1 ; page fault was caused by a non-present page
                                 <1>
1298
                                                       ; sian
1299 00004EC7 7422
                                 <1>
                                                  short pfh alloc np
1300
                                 <1>
1301
                                 <1>
                                           ; If it is not a 'write on read only page' type page fault
1302
                                 <1>
                                           ; major page fault error with minor reason must be returned without
1303
                                           ; fixing the problem. 'sys_exit with error' will be needed
                                 <1>
1304
                                 <1>
                                           ; after return here!
1305
                                  <1>
                                           ; Page fault will be remedied, by copying page contents
1306
                                 <1>
                                           ; to newly allocated page with write permission;
1307
                                           ; sys fork -> sys exec -> copy on write, demand paging method is
                                 <1>
1308
                                 <1>
                                           ; used for working with minimum possible memory usage.
                                           ; sys_fork will duplicate page directory and tables of parent
1309
                                 <1>
                                           ; process with 'read only' flag. If the child process attempts to
1310
                                  <1>
                                 <1>
                                           ; write on these read only pages, page fault will be directed here
1311
                                           ; for allocating a new page with same data/content.
1312
                                 <1>
1313
                                 <1>
1314
                                 <1>
                                           ; IMPORTANT : Retro UNIX 386 v1 (and SINGLIX and TR-DOS)
1315
                                  <1>
                                           ; will not force to separate CODE and DATA space
1316
                                 <1>
                                           ; in a process/program...
1317
                                 <1>
                                           ; CODE segment/section may contain DATA!
1318
                                 <1>
                                           ; It is flat, smoth and simplest programming method already as in
1319
                                 <1>
                                           ; Retro UNIX 8086 v1 and MS-DOS programs.
1320
                                 <1>
1321 00004EC9 F6C202
                                 <1>
                                           test dl, 2 ; page fault was caused by a page write
                                 <1>
1322
                                                        ; sign
                                                     pfh_p_err
1323 00004ECC 0F84AB000000
                                 <1>
                                             iΖ
                                           ; 31/08/2015
1324
                                 <1>
1325 00004ED2 F6C204
                                  <1>
                                           test dl, 4 ; page fault was caused while CPL = 3 (user mode)
                                                        ; sign. (U+W+P = 4+2+1 = 7)
                                 <1>
1326
1327 00004ED5 0F84A2000000
                                 <1>
                                             jz pfh pv err
1328
                                  <1>
                                           ;
1329
                                 <1>
                                           ; make a new page and copy the parent's page content
1330
                                  <1>
                                           ; as the child's new page content
1331
                                 <1>
1332 00004EDB 0F20D3
                                 <1>
                                                  ebx, cr2 ; CR2 contains the linear address
                                                   ; which has caused to page fault
                                 <1>
1333
                                                  copy page
1334 00004EDE E8A2000000
                                 <1>
                                           call
1335 00004EE3 0F828D000000
                                 <1>
                                           jс
                                                     pfh_im_err ; insufficient memory
1336
                                 <1>
1337 00004EE9 EB7D
                                 <1>
                                             jmp
                                                     pfh_cpp_ok
1338
                                 <1>
                                 <1> pfh_alloc_np:
1339
1340 00004EEB E8E7FCFFFF
                                 <1>
                                           call allocate page; (allocate a new page)
1341 00004EF0 0F8280000000
                                                                  ; 'insufficient memory' error
                                 <1>
                                            jс
                                                     pfh_im_err
1342
                                 <1> pfh_chk_cpl:
                                           ; EAX = Physical (base) address of the allocated (new) page
1343
                                 <1>
                                                 ; (Lower 12 bits are ZERO, because
1344
                                 <1>
1345
                                 <1>
                                                      the address is on a page boundary)
                                                 dl, 4 ; CPL = 3 ?
1346 00004EF6 80E204
                                 <1>
                                           and
                                                  short pfh_um
1347 00004EF9 7505
                                 <1>
1348
                                 <1>
                                                        ; Page fault handler for kernel/system mode (CPL=0)
1349 00004EFB 0F20DB
                                 <1>
                                                  ebx, cr3; CR3 (Control Register 3) contains physical address
1350
                                 <1>
                                                         ; of the current/active page directory
1351
                                 <1>
                                                         ; (Always kernel/system mode page directory, here!)
1352
                                 <1>
                                                         ; Note: Lower 12 bits are 0. (page boundary)
1353 00004EFE EB06
                                 <1>
                                           qmŗ
                                                  short pfh_get_pde
1354
                                 <1>
                                 <1> pfh um:
                                                               ; Page fault handler for user/appl. mode (CPL=3)
1356 00004F00 8B1D[B8030300]
                                                  ebx, [u.pgdir]; Page directory of current/active process
                                 <1>
                                           mov
1357
                                 <1>
                                                       ; Physical address of the USER's page directory
1358
                                 <1>
                                                        ; Note: Lower 12 bits are 0. (page boundary)
1359
                                 <1> pfh_get_pde:
1360 00004F06 80CA03
                                 <1>
                                           or dl, 3; USER + WRITE + PRESENT or SYSTEM + WRITE + PRESENT
1361 00004F09 0F20D1
                                                  ecx, cr2; CR2 contains the virtual address
                                 <1>
                                           mov
                                 <1>
                                                         ; which has been caused to page fault
1362
1363
                                 <1>
                                                  ecx, 20
1364 00004F0C C1E914
                                 <1>
                                           shr
                                                              ; shift 20 bits right
1365 00004F0F 80E1FC
                                 <1>
                                                 cl, OFCh; mask lower 2 bits to get PDE offset
                                           and
1366
                                 <1>
1367 00004F12 01CB
                                 <1>
                                           add
                                                  ebx, ecx; now, EBX points to the relevant page dir entry
1368 00004F14 8B0B
                                 <1>
                                                 ecx, [ebx] ; physical (base) address of the page table
                                           mov
1369 00004F16 F6C101
                                 <1>
                                           test
                                                 cl, 1; check bit 0 is set (1) or not (0).
                                                  short pfh_set_pde ; Page directory entry is not valid,
1370 00004F19 740B
                                 <1>
                                           jz
1371
                                                                ; set/validate page directory entry
                                 <1>
                                                  cx, PDE A CLEAR; OF000h; Clear attribute bits
1372 00004F1B 6681E100F0
                                 <1>
1373 00004F20 89CB
                                                  ebx, ecx; Physical address of the page table
                                 <1>
                                           mov
1374 00004F22 89C1
                                 <1>
                                           mov
                                                  ecx, eax; new page address (physical)
1375 00004F24 EB16
                                  <1>
                                                  short pfh_get_pte
                                           qmţ
1376
                                  <1> pfh_set_pde:
1377
                                           ;; NOTE: Page directories and page tables never be swapped out!
                                  <1>
1378
                                 <1>
                                                  (So, we know this PDE is empty or invalid)
                                           ;;
1379
                                 <1>
                                                  al, dl ; lower 3 bits are used as U/S, R/W, P flags
1380 00004F26 08D0
                                 <1>
                                          or
                                         mov
1381 00004F28 8903
                                 <1>
                                                 [ebx], eax; Let's put the new page directory entry here!
                                                  al, al ; clear lower (3..8) bits
1382 00004F2A 30C0
                                 <1>
                                           xor
1383 00004F2C 89C3
                                <1>
                                           mov
                                                 ebx, eax
1384 00004F2E E8A4FCFFFF
                                <1>
                                           call allocate_page ; (allocate a new page)
1385 00004F33 7241
                                 <1>
                                           jс
                                                 short pfh_im_err ; 'insufficient memory' error
                                 <1> pfh spde 1:
1386
1387
                                 <1>
                                          ; EAX = Physical (base) address of the allocated (new) page
1388 00004F35 89C1
                                 <1>
                                           mov
                                                 ecx, eax
1389 00004F37 E815FDFFFF
                                <1>
                                           call clear_page ; Clear page content
                                 <1> pfh_get_pte:
1391 00004F3C 0F20D0
                                           mov eax, cr2; virtual address
                                 <1>
1392
                                 <1>
                                                        ; which has been caused to page fault
1393 00004F3F 89C7
                                                  edi, eax ; 20/07/2015
                                <1>
                                           mov
                                                              ; shift 12 bit right to get
1394 00004F41 C1E80C
                                <1>
                                           shr
                                                  eax, 12
1395
                                 <1>
                                                         ; higher 20 bits of the page fault address
1396 00004F44 25FF030000
                                <1>
                                           and
                                                  eax, 3FFh; mask PDE# bits, the result is PTE# (0 to 1023)
1397 00004F49 C1E002
                                                  eax, 2; shift 2 bits left to get PTE offset
                                 <1>
                                           shl
1398 00004F4C 01C3
                                 <1>
                                                 ebx, eax; now, EBX points to the relevant page table entry
                                           add
```

```
<1>
1399 00004F4E 8B03
                                                eax, [ebx] ; get previous value of pte
1400
                                <1>
                                                ; bit 0 of EAX is always 0 (otherwise we would not be here)
1401 00004F50 21C0
                                               eax, eax
                                <1>
                                          and
                                          jz
1402 00004F52 7410
                                <1>
                                                short pfh_gpte_1
                                          ; 20/07/2015
1403
                                <1>
1404 00004F54 87D9
                                <1>
                                          xchg ebx, ecx; new page address (physical)
                                         push ebp ; 20/07/2015
1405 00004F56 55
                                <1>
1406 00004F57 0F20D5
                                <1>
                                         mov ebp, cr2
1407
                                <1>
                                                ; ECX = physical address of the page table entry
1408
                                <1>
                                                ; EBX = Memory page address (physical!)
1409
                                <1>
                                                ; EAX = Swap disk (offset) address
1410
                                <1>
                                                 ; EBP = virtual address (page fault address)
1411 00004F5A E8B7000000
                                <1>
                                          call swap_in
1412 00004F5F 5D
                                <1>
                                         pop
                                               ebp
1413 00004F60 7210
                                <1>
                                          jс
                                                short pfh err retn
                                          xchg ecx, ebx
1414 00004F62 87CB
                                <1>
                                <1>
                                                ; EBX = physical address of the page table entry
1416
                                                ; ECX = new page
                                <1>
1417
                                <1> pfh_gpte_1:
                                <1> or cl, dl; lower 3 bits are used as U/S, R/W, P flags
1418 00004F64 08D1
1419 00004F66 890B
                                <1>
                                          mov [ebx], ecx; Let's put the new page table entry here!
                                <1> pfh cpp ok:
                                         ; 20/07/2015
1421
                                <1>
1422 00004F68 0F20D3
                                <1>
                                          mov ebx, cr2
1423 00004F6B E8A6020000
                                         call add_to_swap_queue
                                <1>
1424
                                <1>
1425
                                <1>
                                         ; The new PTE (which contains the new page) will be added to
1426
                                <1>
                                         ; the swap queue, here.
1427
                                <1>
                                          ; (Later, if memory will become insufficient,
                                          ; one page will be swapped out which is at the head of
1428
                                <1>
1429
                                <1>
                                         ; the swap queue by using FIFO and access check methods.)
1430
                                 <1>
1431 00004F70 31C0
                                <1>
                                          xor
                                              eax, eax ; 0
1432
                                <1>
1433
                                <1> pfh_err_retn:
1434 00004F72 59
                                <1>
                                          pop ecx
1435 00004F73 5A
                                <1>
                                                edx
                                          pop
1436 00004F74 5B
                                <1>
                                                ebx
                                          pop
1437 00004F75 C3
                                <1>
                                          retn
1438
                                <1>
                                <1> pfh im err:
1439
1440 00004F76 B8E4000000
                                <1>
                                               eax, ERR MAJOR PF + ERR MINOR IM ; Error code in AX
                                                      ; Major (Primary) Error: Page Fault
1441
                                <1>
                                                       ; Minor (Secondary) Error: Insufficient Memory!
1442
                                <1>
1443 00004F7B EBF5
                                <1>
                                                short pfh_err_retn
                                          jmp
1444
                                <1>
1445
                                <1>
                                <1> pfh_p_err: ; 09/03/2015
1446
                                <1> pfh_pv_err:
1447
1448
                                <1>
                                         ; Page fault was caused by a protection-violation
1449 00004F7D B8E6000000
                                <1>
                                          mov eax, ERR_MAJOR_PF + ERR_MINOR_PV ; Error code in AX
1450
                                <1>
                                                      ; Major (Primary) Error: Page Fault
1451
                                <1>
                                                       ; Minor (Secondary) Error: Protection violation !
1452 00004F82 F9
                                <1>
                                          stc
1453 00004F83 EBED
                                <1>
                                          jmp
                                                short pfh_err_retn
1454
                                <1>
                                <1> copy_page:
1455
                                      ; 22/09/2015
1456
                                 <1>
1457
                                 <1>
                                          ; 21/09/2015
                                         ; 19/09/2015
1458
                                 <1>
                                        ; 07/09/2015
1459
                                 <1>
                                        ; 31/08/2015
1460
                                 <1>
1461
                                         ; 20/07/2015
                                 <1>
                                        ; 05/05/2015
1462
                                 <1>
                                        ; 03/05/2015
1463
                                 <1>
1464
                                 <1>
                                          ; 18/04/2015
1465
                                 <1>
                                         ; 12/04/2015
                                        ; 30/10/2014
1466
                                 <1>
1467
                                 <1>
                                         ; 18/10/2014 (Retro UNIX 386 v1 - beginning)
1468
                                 <1>
                                         ; INPUT ->
1469
                                 <1>
1470
                                 <1>
                                                EBX = Virtual (linear) address of source page
                                          ;
1471
                                 <1>
                                                    (Page fault address)
                                          ; OUTPUT ->
1472
                                 <1>
1473
                                 <1>
                                                EAX = PHYSICAL (real/flat) ADDRESS OF THE ALLOCATED PAGE
1474
                                 <1>
                                                (corresponding PAGE TABLE ENTRY is mapped/set)
1475
                                 <1>
                                                EAX = 0 (CF = 1)
1476
                                 <1>
                                                      if there is not a free page to be allocated
1477
                                 <1>
                                                 (page content of the source page will be copied
1478
                                 <1>
                                                onto the target/new page)
1479
                                 <1>
1480
                                 <1>
                                          ; Modified Registers -> ecx, ebx (except EAX)
                                <1>
1482 00004F85 56
                                <1>
                                          push esi
1483 00004F86 57
                                <1>
                                          push edi
1484
                                <1>
                                          ;push ebx
                               <1>
                                          ;push ecx
                              <1>
<1>
1486 00004F87 31F6
                                          xor esi, esi
1487 00004F89 C1EB0C
                                                ebx, 12 ; shift 12 bits right to get PDE & PTE numbers
                                          shr
                               <1>
1488 00004F8C 89D9
                                                ecx, ebx; save page fault address (as 12 bit shifted)
                                          mov
                              <1>
1489 00004F8E C1EB08
                                          shr
                                                ebx, 8 ; shift 8 bits right and then
1490 00004F91 80E3FC
                                <1>
                                          and
                                                bl, OFCh; mask lower 2 bits to get PDE offset
                               <1>
                                                edi, ebx ; save it for the parent of current process
1491 00004F94 89DF
                                          mov
1492 00004F96 031D[B8030300] <1>
                                                ebx, [u.pgdir]; EBX points to the relevant page dir entry
                                        add
                                                eax, [ebx]; physical (base) address of the page table
1493 00004F9C 8B03
                               <1>
                                         mov
                               <1>
1494 00004F9E 662500F0
                                          and
                                                ax, PTE_A_CLEAR ; OF000h ; clear attribute bits
1495 00004FA2 89CB
                               <1>
                                          mov
                                                ebx, ecx ; (restore higher 20 bits of page fault address)
1496 00004FA4 81E3FF030000
                               <1>
                                          and
                                                ebx, 3FFh ; mask PDE# bits, the result is PTE# (0 to 1023)
1497 00004FAA 66C1E302
                                <1>
                                          shl
                                                bx, 2 ; shift 2 bits left to get PTE offset
1498 00004FAE 01C3
                                          add
                                                ebx, eax ; EBX points to the relevant page table entry
                               <1>
                               <1>
                                         ; 07/09/2015
1499
                               <1>
<1>
                                         test word [ebx], PTE_DUPLICATED; (Does current process share this
1500 00004FB0 66F7030002
1501
                                                                  ; read only page as a child process?)
1502 00004FB5 7509
                               <1>
                                                 short cpp 0 ; yes
1503 00004FB7 8B0B
                                <1>
                                                ecx, [ebx] ; PTE value
                                          mov
```

```
1504 00004FB9 6681E100F0
                              <1>
                                             cx, PTE_A_CLEAR ; OF000h ; clear page attributes
                                        and
1505 00004FBE EB32
                               <1>
                                        jmp
                                              short cpp 1
                               <1> cpp_0:
1506
                              <1>
1507 00004FC0 89FE
                                              esi, edi
1508 00004FC2 0335[BC030300]
                              <1>
                                        add
                                              esi, [u.ppgdir] ; the parent's page directory entry
1509 00004FC8 8B06
                              <1>
                                        mov
                                               eax, [esi] ; physical (base) address of the page table
ax, PTE_A_CLEAR ; OF000h ; clear attribute bits
                                              esi, ecx ; (restore higher 20 bits of page fault address) esi, 3FFh ; mask PDE# bits, the result is PTE# (0 to 1023)
                                        shl si, 2 ; shift 2 bits left to get PTE offset
                                        add esi, eax ; EDX points to the relevant page table entry
                                               ecx, [esi] ; PTE value of the parent process
                                        ; 21/09/2015
1517 00004FDE 8B03
                          <1>
<1>
<1>
                                              eax, [ebx] ; PTE value of the child process
                              <1>
<1>
                                      and
                                              ax, PTE A CLEAR ; OF000h ; clear page attributes
1518 00004FE0 662500F0
1519
                            <1>
<1>
<1>
1520 00004FE4 F6C101
                                        test cl, PTE A PRESENT; is it a present/valid page?
1521 00004FE7 7424
                                              short cpp_3; the parent's page is not same page
                                        jz
1522
                               <1>
1523 00004FE9 6681E100F0
                                        and cx, PTE A CLEAR; 0F000h; clear page attributes
                              <1>
1524 00004FEE 39C8
                              <1>
                                        cmp eax, ecx ; Same page?
                                              short cpp_3 ; Parent page and child page are not same
1525 00004FF0 751B
                               <1>
                                        jne
1526
                               <1>
                                                     ; Convert child's page to writable page
                              <1> cpp_1:
1527
                              <1> call allocate_page
1528 00004FF2 E8E0FBFFFF
                                              short cpp_4; 'insufficient memory' error
1529 00004FF7 721A
                              <1>
                                        jс
                              <1>
                                        and esi, esi ; check ESI is valid or not
1530 00004FF9 21F6
                              <1>
                                      jz short cpp_2
1531 00004FFB 7405
1532
                               <1>
                                              ; Convert read only page to writable page
1533
                              <1>
                                              ; (for the parent of the current process)
                              <1> ;and word [esi], PTE_A_CLEAR; 0F000h
<1> ; 22/09/2015
<1> mov [esi] ecv
1534
1536 00004FFD 890E
                              <1>
                                        mov [esi], ecx
1537 00004FFF 800E07
                              <1>
                                      or byte [esi], PTE_A_PRESENT + PTE_A_WRITE + PTE_A_USER
                                                       \frac{1}{1+2+4} = 7
1538
                               <1>
                               <1> cpp_2:
1539
                              1540 00005002 89C7
                                        ; 07/09/2015
1541
                              <1>
                                        mov esi, ecx; the page address of the parent process
1542 00005004 89CE
                               <1>
                        1543 00005006 B900040000
1544 0000500B F3A5
1545
1546 0000500D 0C07
                              <1> or <1> mov
                                              al, PTE_A_PRESENT + PTE_A_WRITE + PTE_A_USER ; 1+2+4 = 7
1547 0000500F 8903
                              <1>
                                              [ebx], eax ; Update PTE
1548 00005011 28C0
                               <1>
                                        sub
                                              al, al ; clear attributes
1549
                               <1> cpp_4:
1550
                               <1> ;pop ecx
1551
                               <1>
                                        ;pop ebx
1552 00005013 5F
                               <1>
                                        pop
                                              edi
1553 00005014 5E
                               <1>
                                        pop
                                              esi
1554 00005015 C3
                               <1>
                                       retn
1555
                               <1>
                               <1> ;; 28/04/2015
1556
1557
                               <1> ;; 24/10/2014
1558
                               <1> ;; 21/10/2014 (Retro UNIX 386 v1 - beginning)
1559
                               <1> ;; SWAP_PAGE_QUEUE (4096 bytes)
1560
                               <1>;; 0000 0001 0002 0003 .... 1020 1021 1022 1023
1561
1562
                                1563
                               <1>;; | pg1 | pg2 | pg3 | pg4 | .... |pg1021|pg1022|pg1023|pg1024|
1564
                               <1> ;; +----+----+-----+
1565
                                <1> ;;
1566
                               <1> ;; [swpq_last] = 0 to 4096 (step 4) -> the last position on the queue
1567
                               <1> ;;
1568
                               <1> ;; Method:
1569
                                <1> ;;
                                        Swap page queue is a list of allocated pages with physical
1570
                               <1> ;;
                                         addresses (system mode virtual adresses = physical addresses).
                                        It is used for 'swap_in' and 'swap_out' procedures.
1571
                               <1> ;;
1572
                               <1> ;;
                                        When a new page is being allocated, swap queue is updated
1573
                               <1> ;;
                                        by 'swap_queue_shift' procedure, header of the queue (offset 0)
                                        is checked for 'accessed' flag. If the 1st page on the queue
1574
                               <1> ;;
1575
                                <1> ;;
                                        is 'accessed' or 'read only', it is dropped from the list;
                               <1> ;;
                                        other pages from the 2nd to the last (in [swpq_last]) shifted
1576
1577
                               <1> ;;
                                         to head then the 2nd page becomes the 1st and '[swpq_last]'
1578
                               <1> ;;
                                        offset value becomes it's previous offset value - 4.
                               <1> ;;
                                        If the 1st page of the swap page queue is not 'accessed'
1579
1580
                               <1> ;;
                                        the queue/list is not shifted.
1581
                               <1> ;;
                                        After the queue/list shift, newly allocated page is added
1582
                                <1> ;;
                                         to the tail of the queue at the [swpq_count*4] position.
                                        But, if [swpq_count] > 1023, the newly allocated page
1583
                               <1> ;;
1584
                               <1> ;;
                                         will not be added to the tail of swap page queue.
1585
                                <1> ;;
                               <1> ;;
                                         During 'swap_out' procedure, swap page queue is checked for
1586
1587
                                         the first non-accessed, writable page in the list,
                                <1> ;;
1588
                                <1> ;;
                                         from the head to the tail. The list is shifted to left
                               <1> ;;
1589
                                         (to the head) till a non-accessed page will be found in the list.
                                <1>;;
                                         Then, this page is swapped out (to disk) and then it is dropped
1590
1591
                               <1> ;;
                                         from the list by a final swap queue shift. [swpq_count] value
                                         is changed. If all pages on the queue' are 'accessed',
1592
                               <1> ;;
1593
                               <1> ;;
                                         'insufficient memory' error will be returned ('swap_out'
1594
                                <1> ;;
                                         procedure will be failed)...
                                <1>;;
1595
                               <1> ;;
                                         Note: If the 1st page of the queue is an 'accessed' page,
1596
                                         'accessed' flag of the page will be reset (0) and that page
1597
                               <1> ;;
1598
                                <1> ;;
                                         (PTE) will be added to the tail of the queue after
                               <1> ;;
                                         the check, if [swpq_count] < 1023. If [swpq_count] = 1024
1599
1600
                                <1> ;;
                                         the queue will be rotated and the PTE in the head will be
1601
                               <1> ;;
                                         added to the tail after resetting 'accessed' bit.
1602
                                <1> ;;
1603
                               <1> ;;
1604
                               <1> ;;
1605
                                <1>;; SWAP DISK/FILE (with 4096 bytes swapped page blocks)
1606
                               <1> ;;
                                <1>;; 00000000 00000004 00000008 0000000C ... size-8 size-4
1607
1608
```

```
1611
                                 <1> ;;
1612
                                 <1> ;; [swpd_next] = the first free block address in swapped page records
1613
                                                      for next free block search by 'swap out' procedure.
                                 <1> ;;
1614
                                 <1> ;; [swpd_size] = swap disk/file size in sectors (512 bytes)
1615
                                 <1> ;;
                                             NOTE: max. possible swap disk size is 1024 GB
                                                  (entire swap space must be accessed by using
1616
                                 <1> ;;
1617
                                 <1> ;;
                                                  31 bit offset address)
                                 <1> ;; [swpd free] = free block (4096 bytes) count in swap disk/file space
1618
1619
                                 <1> ;; [swpd_start] = absolute/start address of the swap disk/file
1620
                                                  O for file, or beginning sector of the swap partition
1621
                                 <1> ;; [swp_drv] = logical drive description table addr. of swap disk/file
1622
                                 <1> ;;
                                 <1> ;;
1623
1624
                                 <1> ;; Method:
                                           When the memory (ram) becomes insufficient, page allocation
1625
                                 <1> ;;
                                 <1> ;;
                                           procedure swaps out a page from memory to the swap disk
1626
1627
                                 <1> ;;
                                           (partition) or swap file to get a new free page at the memory.
                                 <1> ;;
1628
                                           Swapping out is performed by using swap page queue.
1629
                                 <1> ;;
1630
                                 <1> ;;
                                          Allocation block size of swap disk/file is equal to page size
                                 <1> ;;
1631
                                           (4096 bytes). Swapping address (in sectors) is recorded
                                 <1> ;;
1632
                                           into relevant page file entry as 31 bit physical (logical)
                                           offset address as 1 bit shifted to left for present flag (0).
1633
                                 <1> ;;
1634
                                 <1> ;;
                                           Swapped page address is between 1 and swap disk/file size - 4.
1635
                                 <1> ;;
                                           Absolute physical (logical) address of the swapped page is
1636
                                 <1> ;;
                                           calculated by adding offset value to the swap partition's
1637
                                 <1> ;;
                                           start address. If the swap device (disk) is a virtual disk
                                 <1> ;;
                                           or it is a file, start address of the swap disk/volume is 0,
1638
1639
                                 <1> ;;
                                           and offset value is equal to absolute (physical or logical)
1640
                                 <1> ;;
                                           address/position. (It has not to be ZERO if the swap partition
                                 <1> ;;
1641
                                           is in a partitioned virtual hard disk.)
1642
                                 <1> ;;
                                           Note: Swap addresses are always specified/declared in sectors,
1643
                                 <1> ;;
1644
                                 <1> ;;
                                           not in bytes or
                                                             in blocks/zones/clusters (4096 bytes) as unit.
1645
                                 <1> ;;
1646
                                 <1> ;;
                                           Swap disk/file allocation is mapped via 'Swap Allocation Table'
1647
                                 <1> ;;
                                           at memory as similar to 'Memory Allocation Table'.
1648
                                 <1> ;;
                                           Every bit of Swap Allocation Table repsesents one swap block
1649
                                 <1> ;;
                                           (equal to page size) respectively. Bit 0 of the S.A.T. byte 0
1650
                                 <1> ;;
                                 <1> ;;
1651
                                           is reserved for swap disk/file block 0 as descriptor block
1652
                                 <1> ;;
                                           (also for compatibility with PTE). If bit value is ZERO,
1653
                                 <1> ;;
                                           it means relevant (respective) block is in use, and,
                                 <1> ;;
1654
                                           of course, if bit value is 1, it means relevant (respective)
1655
                                 <1> ;;
                                           swap disk/file block is free.
                                           For example: bit 1 of the byte 128 repsesents block 1025
1656
                                 <1> ;;
1657
                                 <1> ;;
                                           (128*8+1) or sector (offset) 8200 on the swap disk or
                                           byte (offset/position) 4198400 in the swap file.
1658
                                 <1> ;;
1659
                                 <1> ;;
                                           4GB swap space is represented via 128KB Swap Allocation Table.
1660
                                 <1> ;;
                                           Initial layout of Swap Allocation Table is as follows:
                                 <1> ;;
                                           ______
1661
1662
                                 <1> ;;
                                           1663
                                 <1> ;;
                                           ______
1664
                                 <1> ;;
                                           (0 is reserved block, 1s represent free blocks respectively.)
1665
                                 <1> ;;
                                           (Note: Allocation cell/unit of the table is bit, not byte)
                                 <1> ;;
1666
1667
                                 <1> ;;
                                 <1> ;;
1668
1669
                                 <1> ;;
                                           'swap_out' procedure checks 'free_swap_blocks' count at first,
1670
                                 <1> ;;
                                           then it searches Swap Allocation Table if free count is not
1671
                                 <1> ;;
                                           zero. From begining the [swpd next] dword value, the first bit
1672
                                 <1> ;;
                                           position with value of 1 on the table is converted to swap
1673
                                 <1> ;;
                                           disk/file offset address, in sectors (not 4096 bytes block).
1674
                                 <1> ;;
                                           'ldrv_write' procedure is called with ldrv (logical drive
1675
                                 <1> ;;
                                           number of physical swap disk or virtual swap disk)
1676
                                 <1> ;;
                                           number, sector offset (not absolute sector -LBA- number),
1677
                                 <1> ;;
                                           and sector count (8, 512*8 = 4096) and buffer adress
                                 <1> ;;
1678
                                           (memory page). That will be a direct disk write procedure.
1679
                                 <1> ;;
                                           (for preventing late memory allocation, significant waiting).
1680
                                 <1> ;;
                                           If disk write procedure returns with error or free count of
                                 <1> ;;
1681
                                           swap blocks is ZERO, 'swap_out' procedure will return with
1682
                                 <1> ;;
                                           'insufficient memory error' (cf=1).
                                 <1> ;;
1683
1684
                                 <1> ;;
                                           (Note: Even if free swap disk/file blocks was not zero,
1685
                                 <1> ;;
                                           any disk write error will not be fixed by 'swap out' procedure,
                                           in other words, 'swap_out' will not check the table for other
1686
                                 <1> ;;
1687
                                 <1> ;;
                                           free blocks after a disk write error. It will return to
1688
                                 <1> ;;
                                           the caller with error (CF=1) which means swapping is failed.
1689
                                 <1> ;;
1690
                                 <1> ;;
                                           After writing the page on to swap disk/file address/sector,
                                 <1> ;;
                                           'swap_out' procesure returns with that swap (offset) sector
1691
1692
                                 <1> ;;
                                           address (cf=0).
                                 <1> ;;
1693
                                 <1> ;;
1694
                                 <1>;;
1695
1696
                                 <1> ;;
                                           'swap_in' procedure loads addressed (relevant) swap disk or
                                           file sectors at specified memory page. Then page allocation
1697
                                 <1> ;;
                                           procedure updates relevant page table entry with 'present'
1698
                                 <1> ;;
1699
                                 <1>;;
                                           attribute. If swap disk or file reading fails there is nothing
1700
                                 <1>;;
                                           to do, except to terminate the process which is the owner of
                                 <1> ;;
1701
                                           the swapped page.
1702
                                 <1> ;;
1703
                                 <1> ;;
                                           'swap in' procedure sets the relevant/respective bit value
                                 <1> ;;
1704
                                           in the Swap Allocation Table (as free block). 'swap_in' also
                                 <1> ;;
1705
                                           updates [swpd_first] pointer if it is required.
1706
                                 <1> ;;
1707
                                 <1> ;;
                                           1708
                                 <1> ;;
                                 <1> ;;
1709
                                           Note: If [swap_enabled] value is ZERO, that means there is not
1710
                                 <1> ;;
                                           a swap disk or swap file in use... 'swap in' and 'swap out'
1711
                                 <1> ;;
                                           procedures ans 'swap page que' procedures will not be active...
1712
                                 <1> ;;
                                           'Insufficient memory' error will be returned by 'swap_out'
1713
                                 <1>;;
                                           and 'general protection fault' will be returned by 'swap in'
```

<1> ;; |descriptr| page(1) | page(2) | page(3) | ... |page(n-1) | page(n) |

1609

1610

```
1714
                                 <1> ;;
                                           procedure, if it is called mistakenly (a wrong value in a PTE).
1715
                                 <1> ;;
1716
                                 <1>
1717
                                 <1> swap_in:
                                         ; 31/08/2015
1718
                                 <1>
1719
                                 <1>
                                           ; 20/07/2015
                                          ; 28/04/2015
1720
                                 <1>
                                         ; 18/04/2015
1721
                                 <1>
1722
                                 <1>
                                           ; 24/10/2014 (Retro UNIX 386 v1 - beginning)
1723
                                 <1>
1724
                                 <1>
                                         ; INPUT ->
1725
                                                 EBX = PHYSICAL (real/flat) ADDRESS OF THE MEMORY PAGE
                                  <1>
1726
                                 <1>
                                                  EBP = VIRTUAL (LINEAR) ADDRESS (page fault address)
1727
                                 <1>
                                                  EAX = Offset Address for the swapped page on the
1728
                                 <1>
                                                       swap disk or in the swap file.
1729
                                 <1>
                                           ; OUTPUT ->
1730
                                 <1>
                                                 EAX = 0 if loading at memory has been successful
1731
                                 <1>
1732
                                 <1>
1733
                                 <1>
                                                  CF = 1 -> swap disk reading error (disk/file not present
1734
                                 <1>
                                                          or sector not present or drive not ready
                                                      EAX = Error code
1735
                                  <1>
1736
                                                      [u.error] = EAX
                                 <1>
1737
                                 <1>
                                                               = The last error code for the process
1738
                                 <1>
                                                                 (will be reset after returning to user)
1739
                                 <1>
1740
                                 <1>
                                           ; Modified Registers -> EAX
1741
                                 <1>
1742
                                 <1>
1743 00005016 833D[62050300]00
                                 <1>
                                                     dword [swp_drv], 0
                                            cmp
1744 0000501D 7646
                                 <1>
                                                  short swpin_dnp_err
                                 <1>
1746 0000501F 3B05[66050300]
                                 <1>
                                                  eax, [swpd_size]
                                           cmp
1747 00005025 734A
                                 <1>
                                                 short swpin_snp_err
                                           jnb
1748
                                 <1>
1749 00005027 56
                                 <1>
                                           push esi
1750 00005028 53
                                           push ebx
                                 <1>
1751 00005029 51
                                 <1>
                                           push ecx
                                                 esi, [swp_drv]
1752 0000502A 8B35[62050300]
                                 <1>
                                           mov ecx, PAGE_SIZE / LOGIC_SECT_SIZE ; 8 !
1753 00005030 B908000000
                                 <1>
1754
                                 <1>
                                                ; Note: Even if corresponding physical disk's sector
1755
                                 <1>
                                                 ; size different than 512 bytes, logical disk sector
1756
                                                 ; size is 512 bytes and disk reading procedure
                                 <1>
                                                 ; will be performed for reading 4096 bytes
1757
                                 <1>
1758
                                 <1>
                                                  ; (2*2048, 8*512).
1759
                                 <1>
                                           ; ESI = Logical disk description table address
1760
                                 <1>
                                           ; EBX = Memory page (buffer) address (physical!)
1761
                                 <1>
                                           ; EAX = Sector adress (offset address, logical sector number)
1762
                                 <1>
                                           ; ECX = Sector count ; 8 sectors
1763 00005035 50
                                 <1>
                                           push eax
1764 00005036 E8AF020000
                                 <1>
                                           call logical_disk_read
1765 0000503B 58
                                 <1>
                                           pop
                                                 eax
1766 0000503C 730C
                                 <1>
                                           jnc
                                                 short swpin_read_ok
1767
                                 <1>
1768 0000503E B828000000
                                 <1>
                                                  eax, SWP_DISK_READ_ERR ; drive not ready or read error
                                           mov
1769 00005043 A3[C8030300]
                                 <1>
                                           mov
                                                  [u.error], eax
1770 00005048 EB17
                                 <1>
                                                  short swpin_retn
                                           jmp
1771
                                 <1>
1772
                                 <1> swpin_read_ok:
                                         ; EAX = Offset address (logical sector number)
1773
                                 <1>
1774 0000504A E80D020000
                                 <1>
                                           call unlink_swap_block ; Deallocate swap block
1775
                                 <1>
1776
                                 <1>
                                           ; EBX = Memory page (buffer) address (physical!)
1777
                                 <1>
                                           ; 20/07/2015
1778 0000504F 89EB
                                 <1>
                                           mov ebx, ebp; virtual address (page fault address)
1779 00005051 6681E300F0
                                 <1>
                                           and bx, ~PAGE_OFF; ~OFFFh; reset bits, 0 to 11
1780 00005056 8A1D[B3030300]
                                 <1>
                                           mov bl, [u.uno]; current process number
                                           ; EBX = Virtual (Linear) address & process number combination
1781
                                 <1>
1782 0000505C E8DB000000
                                 <1>
                                           call swap_queue_shift
1783
                                 <1>
                                           ; eax = 0 ; 10/06/2016 (if ebx input > 0, eax output = 0)
1784
                                 <1>
                                           ; sub eax, eax ; 0 ; Error Code = 0 (no error)
1785
                                           ; zf = 1
                                 <1>
                                 <1> swpin_retn:
1786
1787 00005061 59
                                 <1>
                                           pop
                                                 ecx
1788 00005062 5B
                                 <1>
                                           pop
                                                  ebx
1789 00005063 5E
                                 <1>
                                                  esi
                                           pop
1790 00005064 C3
                                 <1>
                                           retn
1791
                                 <1>
1792
                                 <1> swpin_dnp_err:
1793 00005065 B829000000
                                 <1> mov eax, SWP_DISK_NOT_PRESENT_ERR
1794
                                 <1> swpin_err_retn:
                                           mov [u.error], eax
1795 0000506A A3[C8030300]
                                  <1>
1796 0000506F F9
                                 <1>
                                           stc
1797 00005070 C3
                                 <1>
1798
                                 <1>
1799
                                 <1> swpin_snp_err:
1800 00005071 B82A000000
                                 <1>
                                           mov eax, SWP SECTOR NOT PRESENT ERR
1801 00005076 EBF2
                                 <1>
                                           jmp
                                                 short swpin_err_retn
1802
                                 <1>
1803
                                 <1> swap_out:
                                           ; 10/06/2016
1804
                                 <1>
1805
                                  <1>
                                           ; 07/06/2016
                                 <1>
1806
                                             ; 23/05/2016
1807
                                 <1>
                                           ; 19/05/2016 - TRDOS 386 (TRDOS v2.0)
                                           ; 24/10/2014 - 31/08/2015 (Retro UNIX 386 v1)
1808
                                  <1>
1809
                                 <1>
1810
                                  <1>
                                           ; INPUT ->
                                 <1>
1811
                                          ;
                                                  none
1812
                                 <1>
1813
                                           ; OUTPUT ->
                                 <1>
1814
                                 <1>
                                                 EAX = Physical page address (which is swapped out
1815
                                  <1>
                                                       for allocating a new page)
1816
                                 <1>
                                                  CF = 1 -> swap disk writing error (disk/file not present
                                           ;
1817
                                  <1>
                                                         or sector not present or drive not ready
1818
                                  <1>
                                                      EAX = Error code
```

```
1819
                                 <1>
                                                      [u.error] = EAX
                                                      = The last error code for the process
1820
                                 <1>
1821
                                 <1>
                                                                (will be reset after returning to user)
1822
                                 <1>
1823
                                 <1>
                                          ; Modified Registers -> none (except EAX)
1824
                                 <1>
1825 00005078 66833D[60050300]01 <1>
                                          cmp word [swpq count], 1
                                                   swpout_im_err ; 'insufficient memory'
1826 00005080 0F82AF000000
                                 <1>
                                          jс
1827
                                 <1>
1828
                                 <1>
                                            ;cmp dword [swp drv], 1
1829
                                 <1>
                                          ;jc short swpout_dnp_err ; 'swap disk/file not present'
1830
                                 <1>
1831 00005086 833D[6A050300]01
                                                    dword [swpd_free], 1
                                 <1>
                                            cmp
1832 0000508D 0F828F000000
                                 <1>
                                                    swpout_nfspc_err ; 'no free space on swap disk'
1833
                                 <1>
1834 00005093 53
                                 <1>
                                          push ebx; *
                                <1> swpout 1:
1835
                                          ; 10/06/2016
1836
                                <1>
1837 00005094 31DB
                                <1>
                                                ebx, ebx; shift the queue and return a PTE value
                                          call swap_queue_shift
1838 00005096 E8A1000000
                                <1>
1839 0000509B 21C0
                                <1>
                                          and eax, eax; 0 = \text{empty queue (improper entries)}
1840 0000509D 0F848A000000
                                <1>
                                          jz swpout_npts_err ; There is not any proper PTE
1841
                                <1>
                                                                  ; pointer in the swap queue
                                      ; EAX = PTE value of the page
; EBX = PTE address of the page
1842
                                <1>
1843
                                 <1>
1844 000050A3 662500F0
                                 <1>
                                          and ax, PTE_A_CLEAR; 0F000h; clear attribute bits
1845
                                 <1>
                                          ; 07/06/2016
1846
                                 <1>
1847
                                 <1>
                                          ; 19/05/2016
                                          ; check this page is in timer events or not
1848
                                 <1>
1849
                                 <1>
1850
                                 <1> swpout timer page 0:
                                          push edx; **
1851 000050A7 52
                                 <1>
1852
                                 <1>
1853
                                 <1>
                                          ; 07/06/2016
1854 000050A8 803D[53650100]00
                                <1>
                                          cmp
                                                byte [timer_events], 0
1855 000050AF 762F
                                 <1>
                                                short swpout 2
                                          jna
1856
                                 <1>
1857 000050B1 8A15[53650100]
                                 <1>
                                                dl, [timer_events]
1858
                                <1>
1859 000050B7 51
                                          push ecx; ***
                                 <1>
                                          push ebx ; ****
1860 000050B8 53
                                 <1>
1861 000050B9 BB[60040300]
                                <1>
                                          mov ebx, timer_set ; beginning address of timer event
1862
                                <1>
                                                             ; structures
                                 <1> swpout_timer_page_1:
1863
1864 000050BE 8A0B
                                <1> mov cl, [ebx]
1865 000050C0 08C9
                                                cl, cl; 0 = free, >0 = process number
                                <1>
                                          or
                                          jz short swpout_timer_page_3
mov ecx, [ebx+12]; response
1866 000050C2 7415
                                <1>
1867 000050C4 8B4B0C
                                <1>
                                                ecx, [ebx+12]; response (signal return) address
                                          and cx, PTE_A_CLEAR ; clear offset part (right 12 bits)
1868 000050C7 6681E100F0
                                <1>
1869
                                <1>
                                                             ; of the response byte address, to
1870
                                 <1>
                                                             ; get beginning of the page address)
1871 000050CC 39C8
                                <1>
                                          cmp
                                                eax, ecx
1872 000050CE 7505
                                 <1>
                                          jne short swpout_timer_page_2 ; not same page
1873
                                 <1>
1874
                                 <1>
                                          ; !same page!
1875
                                 <1>
                                          ; NOTE: // 19/05/2016 // - TRDOS 386 feature only ! -
1876
                                 <1>
1877
                                 <1>
                                          ; This page will be used by the kernel to put timer event
1878
                                 <1>
                                          ; response (signal return) byte at the requested address;
1879
                                 <1>
                                          ; in order to prevent a possible wrong write (while
1880
                                 <1>
                                          ; this page is swapped out) on physical memory,
1881
                                 <1>
                                          ; we must protect this page against to be swapped out!
1882
                                 <1>
                                          pop ebx ; ****
1883 000050D0 5B
                                 <1>
                                                ecx ; ***
1884 000050D1 59
                                 <1>
                                          pop
                                          pop edx; **
1885 000050D2 5A
                                 <1>
1886 000050D3 EBBF
                                 <1>
                                                short swpout_1 ; do not swap out this page !
                                          jmp
1887
                                 <1>
1888
                                 <1> swpout_timer_page_2:
1889
                                 <1>
                                       ; 07/06/2016
1890 000050D5 FECA
                                 <1>
                                          dec dl
1891 000050D7 7405
                                 <1>
                                          jz
                                                short swpout_timer_page_4
                                 <1> swpout timer page 3:
1892
                                          ;cmp ebx, timer_set + 240 ; last timer event (15*16)
1893
                                 <1>
                                          ;jnb short swpout_timer_page_4
1894
                                 <1>
1895 000050D9 83C310
                                 <1>
                                          add ebx, 16
1896 000050DC EBE0
                                 <1>
                                          jmp short swpout_timer_page_1
1897
                                 <1>
1898
                                 <1> swpout_timer_page_4:
1899 000050DE 5B
                                      pop ebx; ****
                                 <1>
1900 000050DF 59
                                                 ecx ; ***
                                 <1>
                                           pop
                                 <1> swpout_2:
1902 000050E0 89DA
                                                                     ; Page table entry address
                                 <1>
                                                 edx, ebx
1903 000050E2 89C3
                                 <1>
                                                 ebx, eax
                                          mov
                                                                     ; Buffer (Page) Address
1904
                                 <1>
1905 000050E4 E8A6010000
                                <1>
                                          call
                                                 link swap block
1906 000050E9 7304
                                 <1>
                                                 short swpout_3
                                          jnc
                                                                           ; It may not be needed here
                                                                     ; because [swpd free] value
1907
                                 <1>
1908
                                 <1>
                                                                     ; was checked at the beginging.
1909 000050EB 5A
                                                 edx ; **
                                 <1>
                                          pop
1910 000050EC 5B
                                 <1>
                                          pop
                                                 ebx ; *
1911 000050ED EB33
                                 <1>
                                                 short swpout nfspc err
                                           jmp
                                 <1> swpout_3:
1913 000050EF A900000080
                                 <1>
                                                 eax, 80000000h; test bit 31 (this may not be needed!)
                                           test
1914 000050F4 752C
                                 <1>
                                           jnz
                                                 short swpout_nfspc_err ; 10/06/2016 (bit 31 = 1 !)
1915
                                 <1>
                                          ;
1916 000050F6 56
                                                esi ; **
                                 <1>
                                          push
1917 000050F7 51
                                 <1>
                                          push
                                                 ecx ; ***
1918 000050F8 50
                                          push eax ; sector address ; (31 bit !, bit 31 = 0)
                                <1>
1919 000050F9 8B35[62050300]
                                          mov
                                <1>
                                                 esi, [swp_drv]
1920 000050FF B908000000
                                 <1>
                                                 ecx, PAGE SIZE / LOGIC SECT SIZE ; 8 !
1921
                                 <1>
                                                 ; Note: Even if corresponding physical disk's sector
1922
                                 <1>
                                                 ; size different than 512 bytes, logical disk sector
1923
                                 <1>
                                                 ; size is 512 bytes and disk writing procedure
```

```
1924
                                 <1>
                                                ; will be performed for writing 4096 bytes
1925
                                 <1>
                                                ; (2*2048, 8*512).
                                         ; ESI = Logical disk description table address
1926
                                 <1>
1927
                                <1>
                                        ; EBX = Buffer (Page) address
                                        ; EAX = Sector adress (offset address, logical sector number)
1928
                                 <1>
1929
                                <1>
                                          ; ECX = Sector count ; 8 sectors
                                         ; edx = PTE address
                                <1>
                                        call logical_disk write
1931 00005104 E8E2010000
                                <1>
1932
                                <1>
                                          ; edx = PTE address
1933 00005109 59
                                <1>
                                         pop ecx; sector address
1934 0000510A 730C
                                <1>
                                         jnc short swpout_write_ok
1935
                                <1>
                                          ;
1936
                                <1>
                                          ;; call
                                                       unlink_swap_block; this block must be left as 'in use'
                                <1> swpout dw err:
1937
1938 0000510C B82C000000
                                <1> mov eax, SWP_DISK_WRITE_ERR ; drive not ready or write error
1939 00005111 A3[C8030300]
                                                [u.error], eax
                                <1>
                                          mov
                                               short swpout retn
1940 00005116 EB06
                                <1>
                                          jmp
1941
                                <1>
                                        ;
1942
                                <1> swpout_write_ok:
1943
                                <1>
                                     ; EBX = Buffer (page) address
                                          ; EDX = Page Table Entry address
1944
                                <1>
                                          ; ECX = Swap disk sector (file block) address (31 bit)
1945
                                <1>
1946 00005118 D1E1
                                <1>
                                         shl ecx, 1 ; 31 bit sector address from bit 1 to bit 31
1947 0000511A 890A
                                <1>
                                          mov [edx], ecx
1948
                                <1>
                                                ; bit 0 = 0 (swapped page)
1949 0000511C 89D8
                                <1>
                                          mov
                                                eax, ebx
1950
                                <1> swpout_retn:
                                <1> pop ecx; ***
1951 0000511E 59
1952 0000511F 5E
                                <1>
                                                esi ; **
                                          pop
1953 00005120 5B
                                <1>
                                               ebx ; *
                                          pop
1954 00005121 C3
                                <1>
                                          retn
1955
                                <1>
1956
                                <1> ;swpout_dnp_err:
1957
                                <1> ; mov eax, SWP_DISK_NOT_PRESENT_ERR ; disk not present
                                               short swpout_err_retn
1958
                                <1>;
                                          jmp
1959
                                <1> swpout_nfspc_err:
1960 00005122 B82B000000
                                <1> mov eax, SWP_NO_FREE_SPACE_ERR ; no free space
1961
                                <1> swpout_err_retn:
1962 00005127 A3[C8030300]
                                <1>
                                       mov [u.error], eax
                                <1>
1963
                                          ;stc
1964 0000512C C3
                                <1>
                                          retn
1965
                                <1> swpout_npts_err:
1966 0000512D B82D000000
                                1967 00005132 5B
                                          pop ebx
                                <1>
1968 00005133 EBF2
                                               short swpout_err_retn
                                <1>
                                          qmj
1969
                                <1> swpout_im_err:
1970 00005135 B804000000
                                <1>
                                     mov eax, ERR MINOR IM; insufficient (out of) memory
1971 0000513A EBEB
                                <1>
                                          jmp
                                               short swpout_err_retn
1972
                                <1>
                                <1> swap_queue_shift:
1973
1974
                                <1> ; 26/03/2017
1975
                                 <1>
                                          ; 10/06/2016
                                          ; 09/06/2016 - TRDOS 386 (TRDOS v2.0)
1976
                                <1>
1977
                                 <1>
                                        ; 23/10/2014 - 20/07/2015 (Retro UNIX 386 v1)
1978
                                 <1>
                                       ; INPUT ->
1979
                                 <1>
                                                EBX = Virtual (linear) address (bit 12 to 31)
1980
                                 <1>
1981
                                 <1>
                                                 and process number combination (bit 0 to 11)
1982
                                 <1>
                                                EBX = 0 \rightarrow shift/drop from the head (offset 0)
1983
                                 <1>
                                          ;
                                          ; OUTPUT ->
1984
                                 <1>
1985
                                 <1>
                                                If EBX input > 0
                                          ;
1986
                                 <1>
                                                   the queue will be shifted 4 bytes (dword),
1987
                                 <1>
                                                   from the tail to the head, up to entry offset
1988
                                 <1>
                                                   which points to EBX input value or nothing
1989
                                 <1>
                                                   to do if EBX value is not found on the queue.
1990
                                 <1>
                                                   (The entry -with EBX value- will be removed
                                                   from the queue if it is found.)
1991
                                 <1>
1992
                                 <1>
1993
                                 <1>
                                                   EAX = 0
                                          ;
1994
                                 <1>
                                                If EBX input = 0
1995
                                 <1>
                                          ;
1996
                                 <1>
                                                   the queue will be shifted 4 bytes (dword),
1997
                                 <1>
                                                   from the tail to the head, if the PTE address
                                                   which is pointed in head of the queue is marked
1998
                                 <1>
                                                   as "accessed" or it is marked as "non present".
1999
                                 <1>
2000
                                                   (If "accessed" flag of the PTE -which is pointed
2001
                                 <1>
                                                   in the head- is set -to 1-, it will be reset
                                                   -to 0- and then, the queue will be rotated
2002
                                 <1>
                                                   -without dropping pointer of the PTE from
2003
                                 <1>
2004
                                 <1>
                                                   the queue- for 4 bytes on head to tail direction.
2005
                                 <1>
                                                    Pointer in the head will be moved into the tail,
                                                    other PTEs will be shifted on head direction.)
2006
                                 <1>
2007
                                 <1>
2008
                                 <1>
                                                    Swap queue will be shifted up to the first
                                                    'present' or 'non accessed' page will be found
2009
                                 <1>
2010
                                 <1>
                                                    (as pointed) on the queue head (then it will be
                                          ;
2011
                                 <1>
                                                       removed/dropped from the queue).
                                            ;
2012
                                 <1>
                                                    EAX (> 0) = PTE value of the page which is
2013
                                 <1>
                                          ;
2014
                                 <1>
                                                        (it's pointer -virtual address-) dropped
                                                        (removed) from swap queue.
2015
                                 <1>
2016
                                 <1>
                                                    EBX = PTE address of the page (if EAX > 0)
                                                         which is (it's pointer -virtual address-)
2017
                                 <1>
2018
                                 <1>
                                                        dropped (removed) from swap queue.
2019
                                 <1>
2020
                                 <1>
                                                   EAX = 0 \rightarrow empty swap queue !
2021
                                 <1>
2022
                                 <1>
                                          ; Modified Registers -> EAX, EBX
2023
                                 <1>
2024 0000513C 0FB705[60050300]
                                <1>
                                          movzx
                                                eax, word [swpq_count] ; Max. 1024
2025 00005143 6621C0
                                 <1>
                                          and
                                                ax, ax
2026 00005146 7431
                                <1>
                                          jΖ
                                                 short swpqs_retn
2027 00005148 57
                                 <1>
                                          push edi
2028 00005149 56
                                 <1>
                                          push esi
```

```
push ecx
mov esi,
                               <1>
2029 0000514A 51
2030 0000514B BE00E00800
                               <1>
                                               esi, swap_queue
2031 00005150 89C1
                               <1>
                                         mov
                                               ecx, eax
                                     or
jz
2032 00005152 09DB
                             <1>
                                         or
                                               ebx, ebx
2033 00005154 7424
                               <1>
                                               short swpqs 7
                               <1> swpqs_1:
2034
2035 00005156 AD
                               <1> lodsd
                               <1>
2036 00005157 39D8
                                         cmp eax, ebx
                              cmp eax, ebx
je short sw
<1> loop swpqs_1
<1> ; 10/06/2016
<1> sub eax, eax
<1> jmp short sw
2037 00005159 7406
                                               short swpqs_2
2038 0000515B E2F9
2039
2040 0000515D 29C0
                                              eax, eax
2041 0000515F EB15
                                               short swpqs_6
2042
                               <1> swpqs_2:
                               <1>
2043 00005161 89F7
                                               edi, esi
                                         mov
2044 00005163 83EF04
                               <1>
                                         sub
                                               edi, 4
                               <1> swpqs_3:
2046 00005166 66FF0D[60050300] <1>
                                         dec
                                               word [swpq_count]
2047 0000516D 7403
                               <1>
                                         jz
                                               short swpqs_5
                               <1> swpqs_4:
2048
                               <1>
2049 0000516F 49
                                         dec
                                               ecx
                              <1> rep <1> swpqs_5:
2050 00005170 F3A5
                                               movsd ; shift up (to the head)
2051
                            <1> xor <1> mov <1> swpqs_6:
2052 00005172 31C0
                                               eax, eax
2053 00005174 8907
                                               [edi], eax
2054
2055 00005176 59
                              <1> pop
                                               ecx
                               <1>
2056 00005177 5E
                                         pop
                                               esi
                               <1>
2057 00005178 5F
                                        pop
                                               edi
                               <1> swpqs_retn:
2058
2059 00005179 C3
                               <1> retn
2060
                               <1> swpqs_7:
2061 0000517A 89F7
                               <1>
                                               edi, esi ; head
                                        mov
2062 0000517C AD
                               <1>
                                         lodsd
                                         ; 20/07/2015
2063
                               <1>
2064 0000517D 89C3
                               <1>
                                         mov ebx, eax
2065 0000517F 81E300F0FFFF
2066
                                               ebx, ~PAGE_OFF ; ~OFFFh
                               <1>
2066
                               <1>
                                               ; ebx = virtual address (at page boundary)
                                               eax, PAGE OFF ; OFFFh
2067 00005185 25FF0F0000
                               <1>
                               <1>
                                               ; ax = process number (1 to 4095)
2068
                              2069 0000518A 3A05[B3030300]
                                               al, [u.uno]
                                         cmp
                                               ; Max. 16 (nproc) processes for Retro UNIX 386 v1
2071 00005190 7507
                                         jne
                                               short swpqs_8
2072 00005192 A1[B8030300]
                                              eax, [u.pgdir]
                                         mov
                                         jmp
2073 00005197 EB28
                                               short swpqs_9
2074
                                <1> swpqs_8:
                               <1> ; 09/06/2016
2075
2076 00005199 80B8[AF000300]00
                                         cmp byte [eax+p.stat-1], 0
                               <1>
2077 000051A0 76C4
                                               short swpqs 3 ; free (or terminated) process
                                <1>
2078 000051A2 80B8[AF000300]02
                                               byte [eax+p.stat-1], 2; waiting
                               <1>
                                         cmp
2079 000051A9 77BB
                               <1>
                                               short swpqs_3 ; zombie (3) or undefined ?
2080
                               <1>
2081
                               <1>
                                     ;shl ax, 2 shl al, 2
                                         ;shl ax, 2
                                         mov
                                               eax, [eax+p.upage-4]
2084 000051B4 09C0
                               <1>
                                         or
                                               eax, eax
                                         jz
2085 000051B6 74AE
                               <1>
                                               short swpqs_3 ; invalid upage
                                         add
                                               eax, u.pgdir - user
2086 000051B8 83C05C
                               <1>
                                               ; u.pgdir value for the process
2087
                               <1>
2088
                               <1>
                                                      ; is in [eax]
2089 000051BB 8B00
                               <1>
                                         mov
                                               eax, [eax]
2090 000051BD 21C0
                               <1>
                                         and
                                               eax, eax
2091 000051BF 74A5
                               <1>
                                               short swpqs_3 ; invalid page directory
                                         jz
                               <1> swpqs_9:
2092
                                     edx
; eax = page d.
; ebx = virtual
call get_pte
mov ebx
2093 000051C1 52
                               <1>
2094
                               <1>
                                         ; eax = page directory
                                         ; ebx = virtual address
                               <1>
                               <1>
2096 000051C2 E82BFBFFFF
                                        mov ebx, edx ; PTE address pop edx
2097 000051C7 89D3
                               <1>
2098 000051C9 5A
                               <1>
                                         ; 10/06/2016
2099
                               <1>
                                         jc short swpqs_13; empty PDE
2100 000051CA 723A
                               <1>
                               <1>
                                         ; EAX = PTE value
2101
                                         test al, PTE_A_PRESENT; bit 0 = 1
2102 000051CC A801
                               <1>
                                         jz short swpqs_13 ; Drop non-present page
2103 000051CE 7436
                               <1>
2104
                               <1>
                                                           ; from the queue (head)
2105 000051D0 A802
                               <1>
                                         test al, PTE A WRITE; bit 1 = 0 (read only)
                                         jz short swpqs_13 ; Drop read only page
2106 000051D2 7432
                                <1>
2107
                                <1>
                                                            ; from the queue (head)
                                         ; test al, PTE A ACCESS ; bit 5 = 1 (Accessed)
2108
                                <1>
2109
                                <1>
                                         ;jnz short swpqs_11 ; present
                                                             ; accessed page
2110
                                <1>
                                         btr eax, PTE_A_ACCESS_BIT ; reset 'accessed' bit
2111 000051D4 0FBAF005
                                <1>
2112 000051D8 7210
                                <1>
                                               short swpqs 11 ; accessed page
                                <1>
2113
2114 000051DA 49
                                <1>
                                         dec
                                                ecx
2115 000051DB 66890D[60050300]
                                               [swpq_count], cx
                               <1>
                                         mov
2116 000051E2 7402
                               <1>
                                         jz
                                                short swpqs_10
2117
                               <1>
                                               ; esi = head + 4
2118
                               <1>
                                               ; edi = head
2119 000051E4 F3A5
                                <1>
                                         rep
                                               movsd ; n = 1 to k-1, [n - 1] = [n]
                                <1> swpqs 10:
2120
2121 000051E6 890F
                                <1>
                                         mov
                                                [edi], ecx ; 0
2122 000051E8 EB8C
                                               short swpqs 6 ; 26/03/2017
                               <1>
2123
                                <1>
                                <1> swpqs_11:
2124
2125 000051EA 8903
                               <1>
                                                           ; save changed attribute
                                         mov
                                               [ebx], eax
                               <1>
                                         ; Rotation (head -> tail)
2126
2127 000051EC 49
                               <1>
                                         dec ecx
                                                    ; entry count -> last entry number
2128 000051ED 74F7
                               <1>
                                         jΖ
                                               short swpqs 10
                                               ; esi = head + 4
2129
                               <1>
2130
                               <1>
                                               ; edi = head
2131 000051EF 8B07
                                               eax, [edi] ; 20/07/2015
                               <1>
                                         mov
                                               movsd ; n = 1 to k-1, [n - 1] = [n]
2132 000051F1 F3A5
                               <1>
                                               [edi], eax; head -> tail; [k] = [1]
2133 000051F3 8907
                                <1>
                                         mov
```

```
2134
2135 000051F5 668B0D[60050300]
                               <1>
                                        mov cx, [swpq count]
2136
                               <1>
2137
                               <1> swpqs_12:
2138 000051FC BE00E00800
                               2139 00005201 E974FFFFF
                                         jmp swpqs 7
                               <1>
                               <1>
                               <1> swpqs_13:
2141
                               <1>
2142 00005206 49
                                        dec
                                               ecx
2143 00005207 66890D[60050300] <1>
                                        mov
                                              [swpq_count], cx
2144 0000520E 0F845EFFFFFF
                               <1>
                                        jz swpqs_5
2145 00005214 EBE6
                                <1>
                                         jmp short swpqs_12
                               <1>
2146
2147
                               <1> add_to_swap_queue:
                               <1>; temporary - 16/09/2015
2148
2149 00005216 C3
                               <1> retn
                                       ; 20/02/2017
2150
                               <1>
                                       ; 20/07/2015
2151
                                <1>
2152
                                <1>
                                        ; 24/10/2014 (Retro UNIX 386 v1 - beginning)
2153
                                <1>
2154
                                <1>
                                       ; Adds new page to swap queue
                                       ; (page directories and page tables must not be added
; to swap queue)
2155
                                <1>
2156
                                <1>
2157
                                <1>
                                       ; INPUT ->
                                <1>
2158
2159
                                <1>
                                              EBX = Linear (Virtual) addr for current process
2160
                                <1>
                                               [u.uno]
                                <1>
                                               20/02/2017
2161
2162
                                <1>
                                               (Linear address = CORE + user's virtual address)
2163
                                <1>
                                     ; OUTPUT ->
2164
                                <1>
2165
                                <1>
                                        ; EAX = [swpq_count]
                                                    (after the PTE has been added)
2166
                                <1>
2167
                                <1>
                                               EAX = 0 \rightarrow Swap queue is full, (1024 entries)
                                               the PTE could not be added.
2168
                                <1>
2169
                                <1>
                                        ; Modified Registers -> EAX
2170
                                <1>
2171
                                <1>
                                        push ebx
2172 00005217 53
                               <1>
2173 00005218 6681E300F0
                               <1>
                                        and bx, ~PAGE_OFF; ~OFFFh; reset bits, 0 to 11
2174 0000521D 8A1D[B3030300]
                                       mov bl, [u.uno]; current process number
                               <1>
                                       call swap_queue_shift; drop from the queue if
2175 00005223 E814FFFFFF
                               <1>
2176
                               <1>
                                                            ; it is already on the queue
                                              ; then add it to the tail of the queue
2177
                               <1>
                                        movzx eax, word [swpq_count]
2178 00005228 0FB705[60050300]
                               <1>
2179 0000522F 663D0004
                               <1>
                                         cmp ax, 1024
2180 00005233 7205
                               <1>
                                        jb
                                              short atsq 1
                              <1>
2181 00005235 6629C0
                                       sub ax, ax
2182 00005238 5B
                               <1>
                                        pop
                                              ebx
                                       retn
                            <1>
2183 00005239 C3
2184
                              <1> atsq_1:
                        2185 0000523A 56
2186 0000523B BE00E00800
                                              esi, swap_queue
2187 00005240 6621C0
2188 00005243 740A
                                              short atsq_2
2189 00005245 66C1E002
                                              ax, 2 ; convert to offset
2190 00005249 01C6
                                        add esi, eax
2191 0000524B 66C1E802
2192
                               <1> atsq_2:
                               <1> inc
2193 0000524F 6640
                               <1>
2194 00005251 891E
                                        mov
                                              [esi], ebx ; Virtual address + [u.uno] combination
                                    mov
2195 00005253 66A3[60050300]
                               <1>
                                               [swpq_count], ax
2196 00005259 5E
                               <1>
                                        pop
                                               esi
                                       pop
2197 0000525A 5B
                               <1>
2198 0000525B C3
                                <1>
                                        retn
2199
                               <1>
2200
                                <1> unlink swap block:
                                     ; 15/<del>0</del>9/2015
2201
                                <1>
2202
                                <1>
                                        ; 30/04/2015
2203
                                <1>
                                        ; 18/04/2015
2204
                                <1>
                                       ; 24/10/2014 (Retro UNIX 386 v1 - beginning)
2205
                                <1>
                                        ; INPUT ->
2206
                                <1>
                                       ; EAX = swap disk/file offset address
2207
                                <1>
2208
                                <1>
                                                   (bit 1 to bit 31)
                                        ; OUTPUT ->
2209
                                <1>
2210
                                <1>
                                            [swpd free] is increased
                                               (corresponding SWAP DISK ALLOC. TABLE bit is SET)
2211
                                <1>
2212
                                <1>
2213
                                <1>
                                         ; Modified Registers -> EAX
2214
                                <1>
2215 0000525C 53
                                         push ebx
                                <1>
2216 0000525D 52
                                <1>
                                         push
                                              edx
2217
                                <1>
2218 0000525E C1E804
                                <1>
                                               eax, SECTOR_SHIFT+1 ;3+1; shift sector address to
                                         shr
                                                                ; 3 bits right
2219
                               <1>
2220
                               <1>
                                                                ; to get swap block/page number
2221 00005261 89C2
                               <1>
                                               edx, eax
                                         mov
2222
                               <1>
                                         ; 15/09/2015
2223 00005263 C1EA03
                               <1>
                                              edx, 3
                                                                ; to get offset to S.A.T.
                                         shr
2224
                                <1>
                                                                ; (1 allocation bit = 1 page)
2225
                                <1>
                                                                ; (1 allocation bytes = 8 pages)
2226 00005266 80E2FC
                                                                ; clear lower 2 bits
                               <1>
                                               dl, OFCh
                                         and
2227
                                <1>
                                                                ; (to get 32 bit position)
2228
                                <1>
                                         ;
2229 00005269 BB00000D00
                               <1>
                                         mov
                                               ebx, swap_alloc_table ; Swap Allocation Table address
2230 0000526E 01D3
                                <1>
                                         add
                                               ebx, edx
                                               eax, 1Fh
2231 00005270 83E01F
                                <1>
                                         and
                                                                ; lower 5 bits only
2232
                                <1>
                                                                 ; (allocation bit position)
2233 00005273 3B05[6E050300]
                                <1>
                                                                  ; is the new free block addr. lower
                                         cmp
                                               eax, [swpd_next]
2234
                                <1>
                                                                ; than the address in 'swpd_next' ?
                                                                ; (next/first free block value)
2235
                                <1>
2236 00005279 7305
                                <1>
                                         jnb
                                               short uswpbl_1
                                                                    ; no
2237 0000527B A3[6E050300]
                                <1>
                                               [swpd next], eax
                                         mov
                                <1> uswpbl 1:
2238
```

<1>

```
2239 00005280 0FAB03
                                  <1>
                                                                     ; unlink/release/deallocate block
                                            bts
                                                   [ebx], eax
2240
                                  <1>
                                                                      ; set relevant bit to 1.
2241
                                  <1>
                                                                     ; set CF to the previous bit value
2242 00005283 F5
                                  <1>
                                            cmc
                                                                     ; complement carry flag
2243 00005284 7206
                                  <1>
                                                                       ; do not increase swfd free count
                                            iс
                                                   short uswpbl 2
2244
                                  <1>
                                                                      ; if the block is already deallocated
                                                                      ; before.
2245
                                  <1>
2246 00005286 FF05[6A050300]
                                                      dword [swpd_free]
                                  <1>
                                              inc
2247
                                  <1> uswpbl 2:
2248 0000528C 5A
                                  <1>
                                                   edx
                                            pop
2249 0000528D 5B
                                  <1>
                                                   ebx
                                            pop
2250 0000528E C3
                                  <1>
                                            retn
2251
                                  <1>
2252
                                  <1> link_swap_block:
                                           ; 01/07/2015
2253
                                  <1>
2254
                                  <1>
                                            ; 18/04/2015
                                            ; 24/10/2014 (Retro UNIX 386 v1 - beginning)
2255
                                  <1>
2256
                                  <1>
                                            ; INPUT -> none
2257
                                  <1>
2258
                                  <1>
                                            ; OUTPUT ->
2259
                                  <1>
                                                  EAX = OFFSET ADDRESS OF THE ALLOCATED BLOCK (4096 bytes)
2260
                                  <1>
                                            ;
2261
                                  <1>
                                                        in sectors (corresponding
2262
                                  <1>
                                                         SWAP DISK ALLOCATION TABLE bit is RESET)
2263
                                  <1>
2264
                                  <1>
                                                  CF = 1 and EAX = 0
2265
                                  <1>
                                                            if there is not a free block to be allocated
                                  <1>
2266
2267
                                  <1>
                                            ; Modified Registers -> none (except EAX)
                                  <1>
2268
2269
                                  <1>
2270
                                  <1>
                                                  eax, [swpd_free]
                                            ; mov
2271
                                  <1>
                                            ;and
                                                   eax, eax
2272
                                  <1>
                                                   short out of swpspc
                                            ;jz
2273
                                  <1>
                                            ;
2274 0000528F 53
                                  <1>
                                            push
                                                   ebx
2275 00005290 51
                                  <1>
                                            push
                                                   ecx
2276
                                  <1>
2277 00005291 BB00000D00
                                  <1>
                                                   ebx, swap_alloc_table ; Swap Allocation Table offset
                                            mov
2278 00005296 89D9
                                  <1>
                                                   ecx, ebx
                                            mov
2279 00005298 031D[6E050300]
                                                   ebx, [swpd next] ; Free block searching starts from here
                                  <1>
                                            add
2280
                                  <1>
                                                                 ; next free swap block >> 5
2281 0000529E 030D[72050300]
                                  <1>
                                            add
                                                   ecx, [swpd_last] ; Free block searching ends here
                                                                ; (total swap blocks - 1) >> 5
2282
                                  <1>
2283
                                  <1> lswbl scan:
2284 000052A4 39CB
                                  <1>
                                            cmp
                                                   ebx, ecx
2285 000052A6 770A
                                                   short lswbl notfound
                                  <1>
                                            jа
                                  <1>
2286
2287 000052A8 0FBC03
                                  <1>
                                            bsf
                                                   eax, [ebx] ; Scans source operand for first bit set (1).
2288
                                  <1>
                                                            ; Clears ZF if a bit is found set (1) and
2289
                                  <1>
                                                             ; loads the destination with an index to
2290
                                  <1>
                                                             ; first set bit. (0 -> 31)
                                                             ; Sets ZF to 1 if no bits are found set.
2291
                                  <1>
2292
                                  <1>
                                            ; 01/07/2015
2293 000052AB 751C
                                  <1>
                                            jnz short lswbl_found ; ZF = 0 -> a free block has been found
2294
                                  <1>
2295
                                  <1>
                                                           ; NOTE: a Swap Disk Allocation Table bit
                                                                  with value of 1 means
2296
                                  <1>
2297
                                  <1>
                                                                  the corresponding page is free
2298
                                  <1>
                                                                  (Retro UNIX 386 v1 feaure only!)
2299 000052AD 83C304
                                  <1>
                                            add
                                                   ebx, 4
2300
                                  <1>
                                                           ; We return back for searching next page block
2301
                                  <1>
                                                           ; NOTE: [swpd_free] is not ZERO; so,
2302
                                  <1>
                                                               we always will find at least 1 free block here.
                                                         short lswbl_scan
2303 000052B0 EBF2
                                  <1>
                                            qmŗ
2304
                                  <1>
                                  <1> lswbl notfound:
2306 000052B2 81E900000D00
                                  <1>
                                            sub
                                                   ecx, swap_alloc_table
                                                   [swpd_next], ecx; next/first free page = last page
2307 000052B8 890D[6E050300]
                                  <1>
                                            mov
2308
                                  <1>
                                                                 ; (unlink_swap_block procedure will change it)
2309 000052BE 31C0
                                  <1>
                                            xor
                                                   eax, eax
2310 000052C0 A3[6A050300]
                                  <1>
                                            mov
                                                   [swpd_free], eax
2311 000052C5 F9
                                  <1>
                                            stc
2312
                                  <1> lswbl ok:
2313 000052C6 59
                                            pop
                                  <1>
                                                   ecx
2314 000052C7 5B
                                  <1>
                                                   ebx
                                            pop
2315 000052C8 C3
                                  <1>
                                            retn
2316
                                  <1>
                                  <1> ;out of swpspc:
2317
2318
                                  <1>;
                                           stc
                                  <1> ;
2319
                                            retn
2320
                                   <1>
                                  <1> lswbl_found:
2321
2322 000052C9 89D9
                                                  ecx, ebx
                                  <1>
2323 000052CB 81E900000D00
                                  <1>
                                                   ecx, swap alloc table
                                            sub
2324 000052D1 890D[6E050300]
                                  <1>
                                            mov
                                                   [swpd_next], ecx; Set first free block searching start
                                  <1>
                                                                 ; address/offset (to the next)
2326 000052D7 FF0D[6A050300]
                                  <1>
                                                       dword [swpd_free] ; 1 block has been allocated (X = X-1)
                                              dec
2327
                                  <1>
                                  <1>
2328 000052DD 0FB303
                                                   [ebx], eax
                                                                 ; The destination bit indexed by the source value
                                            btr
                                                                 ; is copied into the Carry Flag and then cleared
2329
                                  <1>
                                                                 ; in the destination.
2330
                                  <1>
2331
                                  <1>
2332
                                  <1>
                                                                 ; Reset the bit which is corresponding to the
2333
                                  <1>
                                                                 ; (just) allocated block.
                                                   ecx, 5
                                                                 ; (block offset * 32) + block index
2334 000052E0 C1E105
                                  <1>
                                            shl
                                                                 ; = block number
2335 000052E3 01C8
                                  <1>
                                            add
                                                   eax, ecx
2336 000052E5 C1E003
                                  <1>
                                                   eax, SECTOR_SHIFT; 3, sector (offset) address of the block
                                            shl
2337
                                  <1>
                                                                 ; 1 block = 8 sectors
2338
                                  <1>
                                            ; EAX = offset address of swap disk/file sector (beginning of the block)
2339
                                  <1>
2340
                                  <1>
2341
                                  <1>
                                            ; NOTE: The relevant page table entry will be updated
2342
                                  <1>
                                                    according to this EAX value...
2343
                                  <1>
```

```
2344 000052E8 EBDC
                                  <1>
                                            jmp short lswbl_ok
2345
                                  <1>
2346
                                  <1> logical_disk_read:
                                         ; 20/07/2015
2347
                                  <1>
2348
                                  <1>
                                           ; 09/03/2015 (temporary code here)
2349
                                  <1>
                                           ; INPUT ->
2350
                                  <1>
2351
                                  <1>
                                                  ESI = Logical disk description table address
2352
                                  <1>
                                                  EBX = Memory page (buffer) address (physical!)
2353
                                  <1>
                                                  EAX = Sector adress (offset address, logical sector number)
2354
                                  <1>
                                                  ECX = Sector count
2355
                                  <1>
                                            ;
2356
                                  <1>
2357 000052EA C3
                                  <1>
2358
                                  <1>
2359
                                  <1> logical_disk_write:
                                          ; 20/07/2015
2360
                                  <1>
                                           ; 09/03/2015 (temporary code here)
2361
                                  <1>
2362
                                  <1>
2363
                                  <1>
                                           ; INPUT ->
                                                  ESI = Logical disk description table address
2364
                                  <1>
2365
                                  <1>
                                                  EBX = Memory page (buffer) address (physical!)
2366
                                  <1>
                                                  EAX = Sector adress (offset address, logical sector number)
2367
                                  <1>
                                                  ECX = Sector count
                                  <1>
2368
2369 000052EB C3
                                  <1>
                                            retn
2370
                                  <1>
2371
                                  <1> get_physical_addr:
                                          ; 26/03/2017
2372
                                  <1>
2373
                                           ; 20/02/2017
                                  <1>
2374
                                  <1>
                                          ; 27/05/2016 - TRDOS 386 (TRDOS v2.0)
                                          ; 18/10/2015
2375
                                  <1>
                                           ; 29/07/2015
2376
                                  <1>
                                          ; 20/07/2015
2377
                                  <1>
                                          ; 04/06/2015
2378
                                  <1>
2379
                                  <1>
                                           ; 20/05/2015
                                           ; 28/04/2015
2380
                                  <1>
                                          ; 18/04/2015
2381
                                  <1>
                                           ; Get physical address
2382
                                  <1>
                                                 (allocates a new page for user if it is not present)
2383
                                  <1>
2384
                                  <1>
                                          ; (This subroutine is needed for mapping user's virtual
2385
                                  <1>
2386
                                  <1>
                                           ; (buffer) address to physical address (of the buffer).)
2387
                                  <1>
                                           ; ('sys write', 'sys read' system calls...)
2388
                                  <1>
                                           ; INPUT ->
2389
                                  <1>
                                           ; EBX = virtual address
2390
                                  <1>
2391
                                                  u.pgdir = page directory (physical) address
                                  <1>
2392
                                  <1>
2393
                                  <1>
                                           ; OUTPUT ->
2394
                                  <1>
                                                 EAX = physical address
2395
                                  <1>
                                                  EBX = linear address
                                                  EDX = physical address of the page frame
2396
                                  <1>
2397
                                  <1>
                                                        (with attribute bits)
2398
                                  <1>
                                                  ECX = byte count within the page frame
                                           ;
2399
                                  <1>
2400
                                  <1>
                                           ; Modified Registers -> EAX, EBX, ECX, EDX
2401
                                  <1>
2402 000052EC 81C300004000
                                  <1>
                                            add ebx, CORE; 18/10/2015
                                  <1> get_physical_addr_x: ; 27/05/2016
2403
2404 000052F2 A1[B8030300]
                                  <1>
                                           mov eax, [u.pgdir]
2405 000052F7 E8F6F9FFFF
                                  <1>
                                            call get pte
2406
                                  <1>
                                                  ; EDX = Page table entry address (if CF=0)
2407
                                  <1>
                                                           Page directory entry address (if CF=1)
2408
                                  <1>
                                                          (Bit 0 value is 0 if PT is not present)
2409
                                  <1>
                                                  ; EAX = Page table entry value (page address)
                                                  ; CF = 1 \rightarrow PDE not present or invalid ?
2410
                                  <1>
2411 000052FC 731C
                                  <1>
                                            jnc
                                                 short gpa_1
2412
                                  <1>
                                            call allocate_page
2413 000052FE E8D4F8FFFF
                                  <1>
                                                  short gpa_im_err ; 'insufficient memory' error
2414 00005303 7248
                                  <1>
                                            jс
2415
                                  <1> gpa 0:
2416 00005305 E847F9FFFF
                                  <1>
                                           call clear page
2417
                                  <1>
                                           ; EAX = Physical (base) address of the allocated (new) page
2418 0000530A 0C07
                                  <1>
                                                  al, PDE A PRESENT + PDE A WRITE + PDE A USER; 4+2+1 = 7
2419
                                  <1>
                                                           ; lower 3 bits are used as U/S, R/W, P flags
2420
                                  <1>
                                                            ; (user, writable, present page)
2421 0000530C 8902
                                  <1>
                                           mov
                                                  [edx], eax ; Let's put the new page directory entry here !
2422 0000530E A1[B8030300]
                                  <1>
                                            mov
                                                  eax, [u.pgdir]
2423 00005313 E8DAF9FFFF
                                  <1>
                                            call get_pte
2424 00005318 7233
                                  <1>
                                                  short gpa_im_err ; 'insufficient memory' error
2425
                                  <1> gpa_1:
                                            ; EAX = PTE value, EDX = PTE address
                                  <1>
2427 0000531A A801
                                  <1>
                                            test al, PTE_A_PRESENT
2428 0000531C 751F
                                  <1>
                                            jnz short gpa_3 ; 26/03/2017
2429 0000531E 09C0
                                  <1>
                                            or
                                                  eax, eax
2430 00005320 7456
                                  <1>
                                                  short gpa_7 ; Allocate a new page
                                            jΖ
2431
                                  <1>
                                           ; 20/07/2015
2432 00005322 55
                                  <1>
                                           push ebp
2433 00005323 89DD
                                  <1>
                                                 ebp, ebx; virtual (linear) address
                                           mov
2434
                                  <1>
                                            ; reload swapped page
2435 00005325 E878000000
                                  <1>
                                            call reload_page ; 28/04/2015
2436 0000532A 5D
                                  <1>
                                            pop
                                                  ebp
2437 0000532B 724A
                                  <1>
                                                  short gpa_retn
                                            jс
2438
                                  <1> gpa_2:
2439
                                  <1>
                                            ; 26/03/2017
                                            ; 20/02/2017
2440
                                  <1>
2441
                                  <1>
                                            ; If a page will contain a Signal Response Byte
2442
                                  <1>
                                           ; it must not be swapped out, because
2443
                                  <1>
                                           ; timer service or irq callback service
2444
                                  <1>
                                            ; will write a signal return/response byte
2445
                                  <1>
                                           ; directly by using physical address of Signal
2446
                                  <1>
                                            ; Response Byte. (Even if process is not running,
                                            ; or it is running with swapped out pages.)
2447
                                  <1>
2448
                                  <1>
```

```
2449
                                <1>
                                          ; 'no_page_swap' will be set by 'systimer' or
2450
                                 <1>
                                          ; 'syscalbac' sistem functions/calls. (*)
2451
                                <1>
                                          cmp byte [no_page_swap], 0
2452 0000532D 803D[926A0100]00
                                <1>
2453 00005334 761D
                                <1>
                                          jna short gpa_4; this page can be swapped out
2454
                                <1>
                                          ; this page must not be swapped out
                                         ; but 'no page swap' must be reset here
2455
                                <1>
                                         ; imediately for other callers (*)
2456
                                <1>
2457
                                <1>
                                          ; (otherwise, swap queue would not be long enough)
2458 00005336 E84B000000
                                <1>
                                          call gpa 8 ; 26/03/2017
2459 0000533B EB1D
                                <1>
                                          jmp short gpa_5
                                <1> gpa 3:
2460
                                         ; 26/03/2017
2461
                                <1>
2462 0000533D 803D[926A0100]00
                               <1>
                                          cmp byte [no_page_swap], 0
2463 00005344 7618
                                <1>
                                               short gpa_6 ; this page can be swapped out
                                          jna
2464 00005346 E83B000000
                                <1>
                                          call gpa_8
2465 0000534B EB11
                                <1>
                                          jmp
                                               short gpa 6
2466
                                <1>
2467
                                <1> gpa_im_err:
                                <1>
2468 0000534D B804000000
                                        mov eax, ERR_MINOR_IM ; Insufficient memory (minor) error!
2469
                                <1>
                                                             ; Major error = 0 (No protection fault)
2470 00005352 C3
                                <1>
                                         retn
2471
                                <1> gpa_4:
                                         ; 20/07/2015
2472
                                <1>
                                         ; 20/05/2015
2473
                                <1>
2474
                                <1>
                                          ; add this page to swap queue
2475 00005353 50
                                <1>
                                         push eax
                                        ; EBX = Linear (CORE+virtual) address ; 20/02/2017
                                <1>
2476
2477 00005354 E8BDFEFFFF
                                <1>
                                          call add_to_swap_queue
2478 00005359 58
                                <1>
                                         pop
                                               eax
                                <1> gpa_5:
2479
2480
                                <1>
                                                ; PTE address in EDX
2481
                                <1>
                                                ; virtual address in EBX
2482
                                <1>
                                         ; EAX = memory page address
                                          or al, PTE_A_PRESENT + PTE_A_USER + PTE_A WRITE
2483 0000535A 0C07
                                <1>
                                                               ; present flag, bit 0 = 1
2484
                                <1>
2485
                                <1>
                                                               ; user flag, bit 2 = 1
2486
                                <1>
                                                              ; writable flag, bit 1 = 1
2487 0000535C 8902
                                <1>
                                                [edx], eax ; Update PTE value
                                          mov
                                <1> gpa_6:
2488
                                     ; 18/10/2015
2489
                                <1>
2490 0000535E 89D9
                                <1>
                                         mov
                                               ecx, ebx
                                               ecx, PAGE OFF
2491 00005360 81E1FF0F0000
                               <1>
                                         and
2492 00005366 89C2
                               <1>
                                               edx, eax
                                         mov
2493 00005368 662500F0
                                <1>
                                         and
                                               ax, PTE_A_CLEAR
2494 0000536C 01C8
                                                eax, ecx
                                <1>
                                          add
2495 0000536E F7D9
                                               ecx ; 1 -> -1 (0FFFFFFFFh), 4095 (0FFFh) -> -4095
                                <1>
                                         neg
2496 00005370 81C100100000
                                <1>
                                         add
                                                ecx, PAGE_SIZE
2497 00005376 F8
                                <1>
                                         clc
                                <1> gpa_retn:
2498
2499 00005377 C3
                                <1>
                                         retn
2500
                                <1> gpa 7:
                                <1>
2501 00005378 E85AF8FFFF
                                         call
                                               allocate_page
2502 0000537D 72CE
                                <1>
                                               short gpa_im_err ; 'insufficient memory' error
                                          jс
2503 0000537F E8CDF8FFFF
                                <1>
                                         call clear_page
2504 00005384 EBA7
                                <1>
                                         jmp short gpa_2
                                <1>
                                <1> gpa_8: ; 26/03/2017
2506
                                      mov byte [no_page_swap], 0
2507 00005386 C605[926A0100]00
                                <1>
                                         push ebx
2508 0000538D 53
                                <1>
                                         push eax ; 26/03/2017
2509 0000538E 50
                                <1>
2510 0000538F 6681E300F0
                                <1>
                                         and bx, ~PAGE OFF; ~OFFFh; reset bits, 0 to 11
2511 00005394 8A1D[B3030300]
                                <1>
                                         mov bl, [u.uno]; current process number
2512 0000539A E89DFDFFFF
                                <1>
                                         call swap_queue_shift ; drop from the queue if
2513
                                <1>
                                                             ; it is already on the queue
2514 0000539F 58
                                <1>
                                                eax ; 26/03/2017
2515 000053A0 5B
                                         pop ebx
                                <1>
2516 000053A1 C3
                                <1>
                                         retn
2517
                                <1>
                                <1> reload_page:
2518
                                        ; 20/07/2015
2519
                                <1>
2520
                                <1>
                                          ; 28/04/2015 (Retro UNIX 386 v1 - beginning)
2521
                                <1>
2522
                                <1>
                                         ; Reload (Restore) swapped page at memory
2523
                                <1>
                                         ; INPUT ->
2524
                                <1>
2525
                                <1>
                                               EBP = Virtual (linear) memory address
                                                EAX = PTE value (swap disk sector address)
2526
                                <1>
2527
                                <1>
                                                (Swap disk sector address = bit 1 to bit 31 of EAX)
2528
                                <1>
                                          ; OUTPUT ->
2529
                                <1>
                                                EAX = PHYSICAL (real/flat) ADDRESS OF RELOADED PAGE
2530
                                 <1>
                                                CF = 1 and EAX = error code
2531
                                <1>
2532
                                <1>
2533
                                <1>
                                          ; Modified Registers -> none (except EAX)
2534
                                <1>
                                          shr
2535 000053A2 D1E8
                               <1>
                                                eax, 1 ; Convert PTE value to swap disk address
                               <1>
2536 000053A4 53
                                          push ebx ;
2537 000053A5 89C3
                               <1>
                                               ebx, eax; Swap disk (offset) address
                                         mov
                                         call allocate_page
2538 000053A7 E82BF8FFFF
                               <1>
2539 000053AC 720C
                               <1>
                                        jc short rlp_im_err
2540 000053AE 93
                                <1>
                                         xchg eax, ebx
                                         ; EBX = Physical memory (page) address
2541
                               <1>
                                         ; EAX = Swap disk (offset) address
2542
                               <1>
                                         ; EBP = Virtual (linear) memory address
2543
                                <1>
2544 000053AF E862FCFFFF
                               <1>
                                         call swap_in
2545 000053B4 720B
                               <1>
                                          jс
                                                short rlp_swp_err ; (swap disk/file read error)
2546 000053B6 89D8
                               <1>
                                        mov
                                                eax, ebx
                                <1> rlp_retn:
2547
2548 000053B8 5B
                                <1>
                                         pop
                                                ebx
2549 000053B9 C3
                                <1>
                                          retn
2550
                                <1>
                                <1> rlp_im_err:
2551
2552 000053BA B804000000
                                <1>
                                         mov eax, ERR_MINOR_IM; Insufficient memory (minor) error!
2553
                                <1>
                                                            ; Major error = 0 (No protection fault)
```

```
2554 000053BF EBF7
                                 <1>
                                           jmp short rlp_retn
2555
                                 <1>
2556
                                 <1> rlp_swp_err:
2557 000053C1 B828000000
                                 <1>
                                           mov eax, SWP DISK READ ERR; Swap disk read error!
2558 000053C6 EBF0
                                 <1>
                                                short rlp retn
                                           jmp
2559
                                 <1>
2560
                                 <1>
                                 <1> copy_page_dir:
2561
2562
                                 <1>
                                          ; 19/09/2015
                                           ; temporary - 07/09/2015
2563
                                 <1>
2564
                                 <1>
                                           ; 07/09/2015 (Retro UNIX 386 v1 - beginning)
2565
                                 <1>
                                           ; INPUT ->
2566
                                 <1>
2567
                                 <1>
                                           ; [u.pgdir] = PHYSICAL (real/flat) ADDRESS of the parent's
2568
                                 <1>
                                                            page directory.
                                           ; OUTPUT ->
2569
                                 <1>
                                                 EAX = PHYSICAL (real/flat) ADDRESS of the child's
2570
                                 <1>
2571
                                 <1>
                                                       page directory.
2572
                                 <1>
                                                  (New page directory with new page table entries.)
2573
                                 <1>
                                                 (New page tables with read only copies of the parent's
                                           ;
2574
                                 <1>
                                                 pages.)
2575
                                 <1>
                                                 EAX = 0 \rightarrow Error (CF = 1)
2576
                                 <1>
2577
                                 <1>
                                           ; Modified Registers -> none (except EAX)
2578
                                 <1>
2579 000053C8 E80AF8FFFF
                                 <1>
                                           call allocate_page
2580 000053CD 723E
                                 <1>
                                                 short cpd_err
                                           jс
2581
                                 <1>
                                           ;
2582 000053CF 55
                                 <1>
                                           push ebp; 20/07/2015
                                           push esi
2583 000053D0 56
                                 <1>
2584 000053D1 57
                                 <1>
                                           push edi
2585 000053D2 53
                                 <1>
                                           push ebx
2586 000053D3 51
                                 <1>
                                           push ecx
2587 000053D4 8B35[B8030300]
                                 <1>
                                           mov esi, [u.pgdir]
2588 000053DA 89C7
                                 <1>
                                                 edi, eax
                                           mov
2589 000053DC 50
                                 <1>
                                           push eax ; save child's page directory address
                                           ; copy PDE 0 from the parent's page dir to the child's page dir
2590
                                 <1>
2591
                                 <1>
                                           ; (use same system space for all user page tables)
2592 000053DD A5
                                 <1>
                                           movsd
2593 000053DE BD00004000
                                 <1>
                                           mov ebp, 1024*4096; pass the 1st 4MB (system space)
2594 000053E3 B9FF030000
                                 <1>
                                           mov
                                                 ecx, (PAGE_SIZE / 4) - 1 ; 1023
                                 <1> cpd 0:
2595
2596 000053E8 AD
                                 <1>
                                           lodsd
                                           ;or eax, eax
2597
                                 <1>
2598
                                 <1>
                                           ;jnz short cpd_1
2599 000053E9 A801
                                           test al, PDE_A_PRESENT; bit 0 = 1
jnz short cpd_1
                                 <1>
2600 000053EB 7508
                                 <1>
                                           ; (virtual address at the end of the page table)
2601
                                 <1>
2602 000053ED 81C500004000
                                           add ebp, 1024*4096; page size * PTE count
                                 <1>
                                                 short cpd_2
2603 000053F3 EB0F
                                 <1>
                                           jmp
2604
                                 <1> cpd_1:
                                          and ax, PDE_A_CLEAR ; 0F000h ; clear attribute bits
mov ebx, eax
2605 000053F5 662500F0
                                 <1>
2606 000053F9 89C3
                                 <1>
2607
                                 <1>
                                           ; EBX = Parent's page table address
2608 000053FB E81F000000
                                 <1>
                                           call copy_page_table
2609 00005400 720C
                                 <1>
                                           jc short cpd_p_err
2610
                                 <1>
                                           ; EAX = Child's page table address
2611 00005402 0C07
                                           or al, PDE_A_PRESENT + PDE_A_WRITE + PDE_A_USER
                                 <1>
2612
                                 <1>
                                                         ; set bit 0, bit 1 and bit 2 to 1
                                                         ; (present, writable, user)
2613
                                 <1>
2614
                                 <1> cpd_2:
2615 00005404 AB
                                 <1>
                                           stosd
2616 00005405 E2E1
                                 <1>
                                           loop cpd_0
2617
                                 <1>
2618 00005407 58
                                 <1>
                                                 eax ; restore child's page directory address
                                           pop
2619
                                 <1> cpd_3:
2620 00005408 59
                                 <1>
                                                 ecx
                                           pop
2621 00005409 5B
                                 <1>
                                           pop
                                                 ebx
2622 0000540A 5F
                                 <1>
                                           pop
                                                  edi
2623 0000540B 5E
                                 <1>
                                           pop
                                                  esi
2624 0000540C 5D
                                 <1>
                                           pop
2625
                                 <1> cpd err:
2626 0000540D C3
                                 <1>
                                          retn
                                 <1> cpd_p_err:
2627
                                        ; release the allocated pages missing (recover free space)
2628
                                 <1>
2629 0000540E 58
                                                 eax ; the new page directory address (physical)
                                 <1>
2630 0000540F 8B1D[B8030300]
                                 <1>
                                                 ebx, [u.pgdir]; parent's page directory address
                                           mov
                                           call deallocate_page_dir
2631 00005415 E8F6F8FFFF
                                 <1>
2632 0000541A 29C0
                                 <1>
                                           sub
                                                 eax, eax ; 0
2633 0000541C F9
                                 <1>
                                           stc
2634 0000541D EBE9
                                 <1>
                                           jmp
                                                 short cpd_3
2635
                                  <1>
2636
                                  <1> copy_page_table:
                                           ; 19/09/2015
2637
                                           ; temporary - 07/09/2015
2638
                                  <1>
                                           ; 07/09/2015 (Retro UNIX 386 v1 - beginning)
2639
                                  <1>
2640
                                  <1>
                                          ; INPUT ->
2641
                                 <1>
2642
                                  <1>
                                                  EBX = PHYSICAL (real/flat) ADDRESS of the parent's page table.
2643
                                  <1>
                                                  EBP = page table entry index (from 'copy_page_dir')
                                           ; OUTPUT ->
2644
                                 <1>
                                                 EAX = PHYSICAL (real/flat) ADDRESS of the child's page table.
2645
                                  <1>
2646
                                 <1>
                                                  EBP = (recent) page table index (for 'add_to_swap_queue')
2647
                                 <1>
                                                  CF = 1 \rightarrow error
2648
                                  <1>
2649
                                 <1>
                                           ; Modified Registers -> EBP (except EAX)
                                 <1>
2651 0000541F E8B3F7FFFF
                                 <1>
                                           call allocate_page
2652 00005424 725A
                                 <1>
                                           jс
                                                 short cpt_err
2653
                                 <1>
2654 00005426 50
                                           push eax; *
                                 <1>
2655
                                 <1>
                                           ;push ebx
2656 00005427 56
                                 <1>
                                           push esi
2657 00005428 57
                                           push edi
                                 <1>
2658 00005429 52
                                 <1>
                                           push edx
```

```
2659 0000542A 51
                               <1>
                                         push ecx
2660
                                <1>
2661 0000542B 89DE
                               <1>
                                          mov
                                                esi, ebx
2662 0000542D 89C7
                               <1>
                                          mov
                                               edi, eax
2663 0000542F 89C2
                               <1>
                                                edx, eax
                                          mov
                               <1>
2664 00005431 81C200100000
                                          add
                                               edx, PAGE SIZE
                               <1> cpt 0:
                                     lodsd
2666 00005437 AD
                                <1>
                                          test al, PTE_A_PRESENT; bit 0 = 1
2667 00005438 A801
                               <1>
                                          jnz short cpt 1
2668 0000543A 750B
                               <1>
                               <1>
2669 0000543C 21C0
                                     and eax, eax
jz short cpt_2
; ebp = virtual (1
2670 0000543E 7430
                                <1>
2671
2671
2672 00005440 E85DFFFFFF <1>
                               <1>
                                         ; ebp = virtual (linear) address of the memory page
                                          call reload_page ; 28/04/2015
                                         jc short cpt_p_err
2674
                               <1> cpt_1:
                               2675 00005447 662500F0
2676 0000544B 89C1
2677
                                <1>
                                         ; Allocate a new page for the child process
                          <1><1><1><1></1>
2678 0000544D E885F7FFFF
                                         call allocate page
                                     jc short cpt_p_err
push edi
push esi
2679 00005452 7227
2680 00005454 57
                               <1>
                                         push esi
                               <1>
2681 00005455 56
mov esi, ecx
                                         mov
                                               edi, eax
                                                ecx, PAGE_SIZE/4
                                         mov
                                         rep
                                               movsd ; copy page (4096 bytes)
                                       pop
                                               esi
2687 00005462 5F
                               <1>
                                                edi
                                         pop
2688
                               <1>
2689 00005463 53
                                       push ebx
                               <1>
2690 00005464 50
                                <1>
                                         push eax
2691 00005465 89EB
                               <1>
                                         mov
                                               ebx, ebp
2692
                               <1>
                                         ; ebx = virtual address of the memory page
                                         call add_to_swap_queue
2693 00005467 E8AAFDFFFF
                                <1>
2694 0000546C 58
                                <1>
                                         pop
                                               eax
2695 0000546D 5B
                               <1>
                                               ebx
                                         pop
2696
                                <1>
2697
                                <1>
                                          ;or
                                                ax, PTE_A_USER+PTE_A_PRESENT
2698 0000546E 0C07
                                <1>
                                                al, PTE_A_USER+PTE_A_WRITE+PTE_A_PRESENT
                                          or
2699
                                <1> cpt_2:
2700 00005470 AB
                                <1>
                                          stosd ; EDI points to child's PTE
2701
                                <1>
2702 00005471 81C500100000
                                <1>
                                          add ebp, 4096; 20/07/2015 (next page)
2703
                                <1>
                                          ;
2704 00005477 39D7
                                               edi, edx
                                <1>
                                          cmp
2705 00005479 72BC
                                               short cpt 0
                               <1>
                                          jb
2706
                                <1> cpt_p_err:
2707 0000547B 59
                                <1>
                                         pop
                                                ecx
2708 0000547C 5A
                               <1>
                                                edx
                                          pop
2709 0000547D 5F
                               <1>
                                                edi
                                         pop
2710 0000547E 5E
                                <1>
                                                esi
                                          pop
                               <1>
2711
                                          ;pop
                                               ebx
2712 0000547F 58
                                <1>
                                         pop
                                                eax ; *
2713
                                <1> cpt_err:
2714 00005480 C3
                                <1>
                                         retn
2715
                                <1>
                                <1> allocate_memory_block:
2716
                                      ; 01/05/2017
2717
                                <1>
2718
                                <1>
                                         ; 28/04/2017
                                       ; 25/04/2017
2719
                                <1>
                                        ; 01/04/2016, 02/04/2016, 03/04/2016
2720
                                <1>
2721
                                <1>
                                         ; 13/03/2016, 14/03/2016
2722
                                <1>
                                        ; 12/03/2016 (TRDOS 386 = TRDOS v2.0)
2723
                                <1>
                                         ; Allocating contiguous memory pages (in the kernel's memory space)
2724
                                <1>
                                         ; INPUT ->
2725
                                <1>
2726
                                <1>
                                               EAX = Beginning address (physical)
                                        ;
2727
                                <1>
                                                EAX = 0 -> Allocate memory block from the first proper aperture
2728
                                <1>
                                                ECX = Number of bytes to be allocated
                                          ;
2729
                                <1>
                                         ; OUTPUT ->
2730
                                 <1>
                                              1) cf = 0 \rightarrow successful
2731
                                <1>
                                                EAX = Beginning (physical) address of the allocated memory block
2732
                                <1>
2733
                                <1>
                                                ECX = Number of allocated bytes (rounded up to page borders)
                                                2) cf = 1 -> unsuccessful
2734
                                <1>
2735
                                                2.1) If EAX > 0 ->
2736
                                <1>
                                                      (Number of requested pages is more than # of free pages
                                                       but contiguous free pages -the aperture- is not enough!)
2737
                                 <1>
2738
                                 <1>
                                                      EAX = Beginning address of available aperture
                                                          (one of all aperture with max. aperture size/length)
2739
                                 <1>
                                                      {\tt ECX} = Size of available aperture (memory block) in bytes
2740
                                 <1>
                                                 2.2) If EAX = 0 \rightarrow Out of memory error
2741
                                 <1>
2742
                                 <1>
                                                           (number of free pages is less than requested number)
                                                      ECX = Total number of free bytes (free pages * 4096)
2743
                                 <1>
2744
                                                          (It is not number of contiguous free bytes)
                                 <1>
2745
                                 <1>
2746
                                <1>
                                          ; (Modified Registers -> EAX, ECX)
2747
                                <1>
2748
                                <1>
                                          ; PURPOSE: Loading a file at memory for copying or running etc.
2749
                                <1>
                                          ; If this procedure returns with cf is set, ECX contains maximum
2750
                                 <1>
                                          ; available space and EAX contains the beginning address of it.
2751
                                <1>
                                          ; If EAX has zero, ECX contains total number of free bytes.
2752
                                <1>
                                          ; If requested block has been successfully allocated (by rounding up to
                                          ; the last page border), it must be deallocated later by using
2753
                                 <1>
2754
                                <1>
                                          ; 'deallocate_memory_block' procedure.
2755
                                 <1>
2756 00005481 52
                                          push edx; *
                                <1>
2757 00005482 BAFF0F0000
                                <1>
                                                edx, PAGE_SIZE - 1 ; 4095
2758 00005487 01D0
                                <1>
                                                eax, edx
                                          add
2759 00005489 01D1
                                <1>
                                          add
                                                ecx, edx
2760 0000548B C1E90C
                                 <1>
                                          shr
                                                ecx, PAGE SHIFT
                                                                       ; 12
2761
                                <1>
2762
                                <1>
                                          ; ECX = number of contiguous pages to be allocated
2763 0000548E 8B15[C8580100]
                                <1>
                                          mov edx, [free_pages]
```

```
; 01/05/2017
2764
                                  <1>
2765
                                  <1>
                                            ;or
                                                 ecx, ecx
2766
                                  <1>
                                            ;jz
                                                 short amb3
2767
                                  <1>
                                            ; If ECX=0, set cf to 1 and return with max. available mem block size
2768
                                  <1>
2769 00005494 39D1
                                  <1>
                                                   ecx, edx
2770 00005496 7760
                                  <1>
                                                  short amb 3
                                            jа
2771
                                  <1>
2772 00005498 C1E80C
                                  <1>
                                                  eax, PAGE_SHIFT
                                            shr
2773
                                  <1>
2774 0000549B 89C2
                                            mov
                                  <1>
                                                  edx, eax
                                                                    ; page number
2775 0000549D C1EA03
                                  <1>
                                            shr
                                                  edx, 3
                                                                    ; to get offset to M.A.T.
2776
                                  <1>
                                                                    ; (1 allocation bit = 1 page)
2777
                                  <1>
                                                                    ; (1 allocation bytes = 8 pages)
                                                                    ; clear lower 2 bits
2778 000054A0 80E2FC
                                  <1>
                                                  dl. OFCh
                                            and
2779
                                  <1>
                                                                    ; (to get 32 bit position)
2780 000054A3 53
                                                  ebx ; **
                                  <1>
                                            push
                                  <1> amb 0:
2781
                                                   [mem ipg count], ecx; initial (reset) value of page count
2782 000054A4 890D[7C640100]
                                  <1>
                                            mov
2783 000054AA 890D[80640100]
                                  <1>
                                            mov
                                                   [mem_pg_count], ecx
                                                   ecx, ecx; 0
2784 000054B0 31C9
                                  <1>
                                            xor
2785 000054B2 890D[84640100]
                                  <1>
                                                   [mem aperture], ecx; 0
                                            mov
2786 000054B8 890D[88640100]
                                  <1>
                                            mov
                                                   [mem_max_aperture], ecx ; 0
                                  <1>
2788 000054BE BB00001000
                                                  ebx, MEM_ALLOC_TBL ; Memory Allocation Table address.
                                  <1>
                                            mov
2789 000054C3 3B15[CC580100]
                                  <1>
                                                                        ; Is the beginning page address lower
                                            cmp
                                                   edx, [next_page]
2790
                                  <1>
                                                                    ; than the address in 'next page' ?
2791
                                  <1>
                                                                     ; (the first/next free page of user space)
2792 000054C9 7208
                                  <1>
                                            jb
                                                   short amb 1
2793 000054CB 3B15[D0580100]
                                  <1>
                                                  edx, [last_page]
                                                                       ; is the beginning page address higher
                                            cmp
2794
                                  <1>
                                                                     ; than the address in 'last_page' ?
                                                                     ; (end of the memory)
2795
                                  <1>
2796 000054D1 7606
                                  <1>
                                                  short amb 2
                                            jna
                                                                     ; no
2797
                                  <1> amb_1:
2798 000054D3 8B15[CC580100]
                                  <1>
                                           mov
                                                  edx, [next_page]
                                                                     ; M.A.T. offset (1 M.A.T. byte = 8 pages)
2799
                                  <1> amb_2:
2800 000054D9 01D3
                                  <1>
                                            add
                                                  ebx, edx
2801
                                  <1>
                                            ; 28/04/2017
2802
                                  <1>
2803
                                  <1>
                                            ;xor ecx, ecx
2804 000054DB 0FBC0B
                                  <1>
                                            bsf
                                                  ecx, [ebx]
                                                                    ; 0 to 31
2805 000054DE 89D0
                                  <1>
                                            mov
                                                  eax, edx
                                                                    ; *8
2806 000054E0 C1E003
                                  <1>
                                            shl
                                                  eax, 3
2807 000054E3 01C8
                                  <1>
                                                                    ; beginning page number
                                                  eax, ecx
2808
                                  <1>
2809 000054E5 A3[8C640100]
                                                   [mem_pg_pos], eax ; beginning page no (for curr. mem. aperture)
                                  <1>
                                            mov
                                                  [mem_max_pg_pos], eax; beginning page no for max. mem. aperture
2810 000054EA A3[90640100]
                                  <1>
                                            mov
2811
                                  <1>
                                                  eax, 1Fh
2812 000054EF 83E01F
                                  <1>
                                            and
                                                                     ; lower 5 bits only (0 to 31)
2813
                                  <1>
                                                                     ; (allocation bit position)
                                                                     ; 0
2814 000054F2 750E
                                  <1>
                                            jnz
                                                  short amb_4
2815 000054F4 B120
                                  <1>
                                                  cl, 32
                                            mov
2816 000054F6 EB4B
                                 <1>
                                            jmp
                                                  short amb_10
2817
                                 <1>
2818
                                  <1> amb_3:
                                                  ; out_of_memory
2819 000054F8 31C0
                                 <1>
                                                   eax, eax ; 0
                                            xor
2820 000054FA 89D1
                                  <1>
                                                  ecx, edx; free pages
                                            mov
2821 000054FC C1E10C
                                 <1>
                                            shl
                                                  ecx, PAGE_SHIFT
2822 000054FF 5A
                                 <1>
                                            pop
                                                  edx ; *
2823 00005500 F9
                                 <1>
                                            stc
2824 00005501 C3
                                 <1>
                                            retn
                                  <1> amb 4:
2826 00005502 8B13
                                                  edx, [ebx]
                                 <1>
                                            mov
2827 00005504 88C1
                                 <1>
                                                  cl, al ; 1 to 31
                                                  edx, cl
2828 00005506 D3EA
                                 <1>
                                            shr
2829 00005508 89D0
                                  <1>
                                                  eax, edx
                                            mov
2830
                                  <1> amb 5:
2831 0000550A D1E8
                                                  eax, 1 ; (***)
                                 <1>
                                            shr
2832 0000550C 7317
                                  <1>
                                            jnc
                                                  short amb_7
2833 0000550E FF05[84640100]
                                  <1>
                                                  dword [mem_aperture]
                                            inc
2834 00005514 FF0D[80640100]
                                  <1>
                                            dec
                                                  dword [mem_pg_count]
2835 0000551A 7470
                                  <1>
                                            jz
                                                  short amb 15
2836
                                  <1> amb_6:
2837
                                  <1>
                                           ; 28/04/2017
2838 0000551C FEC1
                                  <1>
                                            inc
                                                 cl
2839 0000551E 80F920
                                  <1>
                                            cmp
                                                  cl, 32
2840 00005521 730D
                                  <1>
                                                  short amb 9
                                            jnb
2841 00005523 EBE5
                                  <1>
                                            jmp
                                                  short amb_5
2842
                                  <1> amb 7:
2843 00005525 50
                                                  eax ; (***) allocation bits (in shifted status)
                                  <1>
                                            push
2844 00005526 E81B010000
                                  <1>
                                            call
                                                  amb 26 ; set maximum memory aperture (free memory block size)
2845 0000552B 58
                                  <1>
                                                  eax ; (***)
                                            pop
2846 0000552C EBEE
                                  <1>
                                                  short amb_6
                                            jmp
                                  <1> amb_8:
2847
                                           ; 28/04/2017
                                 <1>
2848
2849 0000552E B120
                                 <1>
                                           mov
                                                 cl, 32
2850
                                 <1> amb 9:
2851 00005530 89DA
                                 <1>
                                           mov
                                                  edx, ebx
2852 00005532 81EA00001000
                                 <1>
                                                  edx, MEM_ALLOC_TBL
                                            sub
2853 00005538 3B15[D0580100]
                                 <1>
                                                  edx, [last_page]
                                            cmp
                                                  short amb 14; contiguous pages not enough
2854 0000553E 7336
                                 <1>
                                            jnb
2855 00005540 83C304
                                 <1>
                                            add
                                                  ebx, 4
                                 <1> amb_10:
2856
                                 <1> mov
2857 00005543 8B03
                                                  eax, [ebx]
2858 00005545 21C0
                                 <1>
                                           and
                                                  eax, eax
2859 00005547 7408
                                 <1>
                                           jz
                                                  short amb_11 ; there is not a free page bit in this alloc dword
                                                  eax ; OFFFFFFFFh -> 0
2860 00005549 40
                                <1>
                                           inc
2861 0000554A 740C
                                 <1>
                                                  short amb_12; all of bits are set (32 free pages)
                                           jz
2862 0000554C 48
                                 <1>
                                           dec
2863 0000554D 28C9
                                                  cl, cl; 0
                                 <1>
                                           sub
2864 0000554F EBB9
                                 <1>
                                          jmp
                                                  short amb_5
2865
                                 <1> amb 11:
2866 00005551 E8F0000000
                                 <1>
                                           call
                                                  amb_26; set maximum memory aperture (free memory block size)
2867 00005556 EBD8
                                 <1>
                                                  short amb 9
                                            jmp
                                  <1> amb 12:
2868
```

```
2869 00005558 390D[80640100]
                               <1>
                                         cmp
                                               [mem_pg_count], ecx; 32
2870 0000555E 7306
                                <1>
                                         jnb
                                               short amb 13
                                               ecx, [mem_pg_count]
2871 00005560 8B0D[80640100]
                               <1>
                                         mov
2872
                               <1> amb 13:
2873 00005566 010D[84640100]
                                <1> add
                                               [mem aperture], ecx
2874 0000556C 290D[80640100]
                               <1>
                                         sub
                                                [mem_pg_count], ecx
                               <1> jna
<1> jmp
2875 00005572 7618
                                               short amb 15
2876 00005574 EBBA
                                               short amb 9 ; 01/05/2017
2877
                               <1> amb_14:
                               <1> call amb_26; 28/04/2017
2878 00005576 E8CB000000
2879 0000557B A1[90640100]
                               <1>
                                         mov
                                               eax, [mem_max_pg_pos] ; begin address of max. mem aperture
2880 00005580 8B0D[88640100]
                               <1>
                                               ecx, [mem max aperture]; max. (largest) memory aperture
2881 00005586 F9
                               <1>
                                         stc
2882 00005587 E9AF000000
                                         jmp
                               <1>
                                                   amb 25
2883
                                <1>
                                <1> amb_15: ; OK !
2884
2885 0000558C A1[8C640100]
                                        mov eax, [mem pg pos] ; Beginning address as page number
                               <1>
2886 00005591 8B0D[84640100]
                               <1>
                                         mov ecx, [mem_aperture] ; Free contiguous page count (>=1)
                                <1> amb 16:
2887
2888
                               <1> ; allocate contiguous memory pages (via memory allocation table bits)
2889 00005597 89C2
                               <1>
                                         mov edx, eax
2890
                                <1>
                                         ; 25/04/2017
                                                         ; 8 pages in one allocation byte
; clear lower 2 bits
2891 00005599 C1EA03
                               <1>
                                         shr edx, 3
2892 0000559C 80E2FC
                               <1>
                                         and dl, OFCh
2893
                                <1>
                                                             ; (for dword/32bit positioning)
2894
                                <1>
                               2895 0000559F BB00001000
2896 000055A4 01D3
2897 000055A6 83E01F
2898
2899 000055A9 BA20000000
2900 000055AE 28C2
                               <1>
                                         sub
                                               dl, al
2901 000055B0 39CA
                               <1>
                                               edx, ecx
                                         cmp
                                                             ; ecx >= 1
2902 000055B2 7602
                               <1>
                                         jna short amb 17
                                               edx, ecx
2903 000055B4 89CA
                               <1>
                                         mov
                               <1> amb_17:
2904
2905 000055B6 29D1
                                         sub
                               <1>
                                               ecx, edx
2906 000055B8 51
                                         push ecx; ***
                               <1>
2907 000055B9 89D1
                               <1>
                                               ecx, edx
                                         mov
2908
                               <1> amb_18:
2909 000055BB 0FB303
                               <1>
                                                [ebx], eax
                                                             ; The destination bit indexed by the source value
2910
                                <1>
                                                             ; is copied into the Carry Flag and then cleared
                                                             ; in the destination.
2911
                               <1>
2912 000055BE FF0D[C8580100] <1>
2913 000055C4 49 <1>
                                               dword [free pages]; 1 page has been allocated (X = X-1)
                                         dec
                                         dec
                                               ecx
                               <1>
2914 000055C5 7404
                                         jz
                                               short amb_19
                                               al
2915 000055C7 FEC0
                               <1>
                                         inc
2916 000055C9 EBF0
                               <1>
                                         jmp short amb_18
                               <1> amb 19:
2917
                               <1> pop
                                               ecx ; ***
2918 000055CB 59
2919 000055CC 21C9
                                               ecx, ecx; 0 ?
                               <1>
                                         and
2920 000055CE 741E
                               <1>
                                         jz
                                               short amb 22
                                         ; 01/04/2016
                               <1>
2921
2922 000055D0 B020
                               <1>
                                               al, 32
2923
                               <1> amb_20:
                               <1>
2924 000055D2 83C304
                                         add
                                               ebx, 4
2925 000055D5 39C1
                               <1>
                                         cmp
                                               ecx, eax ; 32
2926 000055D7 7305
                               <1>
                                         jnb
                                               short amb_21
2927
                               <1>
                                         ; ECX < 32
2928 000055D9 28C0
                               <1>
                                         sub al, al; 0
                                         push eax ; 0 ***
2929 000055DB 50
                               <1>
2930 000055DC EBDD
                                <1>
                                         jmp short amb 18
2931
                               <1> amb_21:
2932 000055DE 2905[C8580100]
                               <1> sub
                                               [free_pages], eax ; [free_pages] = [free_pages] - 32
                                                                   ; reset 32 bits
2933 000055E4 C7030000000
                                <1>
                                         mov
                                               dword [ebx], 0
2934 000055EA 29C1
                                <1>
                                         sub
                                               ecx, eax ; 32
2935 000055EC 75E4
                                <1>
                                         jnz
                                               short amb 20
2936
                                <1> amb_22:
                                <1>
2937 000055EE A1[8C640100]
                                               eax, [mem_pg_pos] ; Beginning address as page number
                                               ecx, [mem_aperture] ; Free contiguous page count
2938 000055F3 8B0D[84640100]
                                <1>
                                         mov
2939
                                <1>
                                         ; [next_page] update
2940 000055F9 89C2
                                <1>
                                         mov edx, eax
2941
                                <1>
                                         ; 03/04/2016
                                         shr edx, 3
2942 000055FB C1EA03
                                <1>
                                                                ; to get offset to M.A.T.
                                                                ; (1 allocation bit = 1 page)
2943
                                <1>
                                                                 ; (1 allocation bytes = 8 pages)
2944
                                <1>
2945 000055FE 80E2FC
                                <1>
                                               dl, 0FCh
                                                                ; clear lower 2 bits
                                                                 ; (to get 32 bit position)
2946
                                <1>
                                               edx, [next page]; first free page pointer offset
2947 00005601 3B15[CC580100]
                                <1>
2948 00005607 7732
                                               short amb \frac{1}{25}
                                <1>
                                         jа
2949 00005609 BB00001000
                                                ebx, MEM_ALLOC TBL
                                <1>
                                         mov
                                                dword [ebx+edx], 0
2950 0000560E 833C1300
                                <1>
                                         cmp
2951 00005612 7721
                                <1>
                                                short amb_24
                                         jа
2952 00005614 89C2
                                <1>
                                                edx, eax
                                <1>
2953 00005616 01CA
                                         add
                                                edx, ecx
2954 00005618 C1EA03
                                <1>
                                         shr
                                                edx, 3
2955 0000561B 80E2FC
                                <1>
                                                dl, OFCh
                                         and
2956
                                <1> amb_23:
2957 0000561E 833C1300
                                <1>
                                                dword [ebx+edx], 0
                                         cmp
2958 00005622 7711
                                <1>
                                                short amb 24
                                         jа
                                         add
2959 00005624 83C204
                                <1>
                                                edx, 4
2960 00005627 3B15[D0580100]
                                <1>
                                         cmp
                                                edx, [last_page]
                                                                   ; last page pointer offset
2961 0000562D 76EF
                                <1>
                                         jna
                                                short amb_23
                                                edx, [first page]
2962 0000562F 8B15[D4580100]
                                <1>
                                                                   ; (for) beginning of user's space
                                <1> amb_24:
2963
2964 00005635 8915[CC580100]
                                <1>
                                         mov
                                                [next_page], edx
                                <1> amb 25:
2966 0000563B 9C
                                <1>
                                         pushf
                                                eax, PAGE SHIFT
                                                                       ; convert to phy. address in bytes
2967 0000563C C1E00C
                                <1>
                                         shl
2968 0000563F C1E10C
                                                ecx, PAGE_SHIFT
                                <1>
                                                                       ; convert to byte counts
                                         shl
2969 00005642 9D
                                <1>
                                         popf
2970 00005643 5B
                                <1>
                                                ebx ; **
                                         pop
                                                edx ; *
2971 00005644 5A
                                <1>
                                         pop
2972 00005645 C3
                                <1>
2973
                                <1>
```

```
2974
                                 <1> amb_26:
                                                ; set maximum free memory aperture (free memory block size)
                                <1>
2975 00005646 89DA
                                          mov
                                                edx, ebx; current address
2976 00005648 81EA00001000
                                                edx, MEM_ALLOC_TBL ; MAT beginning address
                                <1>
                                          sub
2977
                                <1>
                                          ; 02/04/2016
2978 0000564E C1E203
                                <1>
                                          shl
                                                edx, 3; MAT byte offset * 8 = page number base
2979 00005651 01CA
                                <1>
                                          add
                                                edx, ecx; current page number (ecx = 0 to 32)
                                <1>
                                          ;
2981 00005653 A1[84640100]
                                        mov
                                                eax, [mem_aperture]
                                <1>
                                     and
2982 00005658 21C0
                                <1>
                                                eax, eax
                                        jz short amb_27
mov dword [mem ax
2983 0000565A 7421
                                <1>
2984 0000565C C705[84640100]0000- <1>
                                                  dword [mem_aperture], 0
2984 00005664 0000
                                <1>
2985 00005666 3B05[88640100]
                                <1>
                                          cmp
                                                eax, [mem_max_aperture]
2986 0000566C 760F
                                <1>
                                        jna short amb_27
2987 0000566E A3[88640100]
                                          mov
                                                [mem max_aperture], eax
                                <1>
2988
                                 <1>
                                          ; 25/04/2017
                                          mov eax, [mem_pg_pos]
2989 00005673 A1[8C640100]
                                 <1>
                                          ; EAX = Beginning page number of the max. aperture
2990
                                 <1>
                                                [mem max pg pos], eax
2991 00005678 A3[90640100]
                                 <1>
2992
                                 <1> amb 27:
2993 0000567D 8915[8C640100]
                                <1>
                                          mov
                                                 [mem_pg_pos], edx ; current page
                                 <1>
2995 00005683 A1[7C640100]
                                                 eax, [mem_ipg_count] ; initial (reset) value of page count
                                 <1>
                                          mov
2996 00005688 A3[80640100]
                                 <1>
                                                 [mem_pg_count], eax
                                 <1>
2997
2998 0000568D C3
                                 <1>
                                           retn
2999
                                 <1>
3000
                                 <1> deallocate_memory_block:
3001
                                 <1>
                                          ; 14/03/2016 (TRDOS 386 = TRDOS v2.0)
3002
                                 <1>
3003
                                 <1>
                                          ; Deallocating contiguous memory pages (in the kernel's memory space)
3004
                                 <1>
                                          ; INPUT ->
3005
                                 <1>
3006
                                 <1>
                                                EAX = Beginning address (physical)
3007
                                 <1>
                                                 ECX = Number of bytes to be deallocated
3008
                                 <1>
                                          ; OUTPUT ->
3009
                                 <1>
3010
                                 <1>
                                                 Memory Allocation Table bits will be updated
                                                 [free pages] will be changed (increased)
3011
                                 <1>
3012
                                 <1>
3013
                                 <1>
                                          ; (Modified Registers -> EAX, ECX)
3014
                                 <1>
                                          ; PURPOSE: Unloading/Freeing a file -or an allocated memory block-
3015
                                 <1>
                                          ; at memory after copying, running, saving, reading, writing etc.
3016
                                 <1>
3017
                                 <1>
3018
                                <1>
3019 0000568E 52
                                          push edx; *
                                <1>
                                          push ebx; **
3020 0000568F 53
                                <1>
3021
                                <1>
                                                eax, PAGE SHIFT
3022 00005690 C1E80C
                                <1>
                                           shr
                                                                         ; 12
3023 00005693 C1E90C
                                <1>
                                           shr ecx, PAGE_SHIFT
                                                                         ; 12
3024
                                 <1>
3025
                                <1>
                                          ; EAX = Beginning page number
3026
                                <1>
                                          ; ECX = Number of contiguous pages to be deallocated
3027
                                 <1> damb 0:
3028
                                 <1>
                                          ; deallocate contiguous memory pages (via memory allocation table bits)
3029 00005696 89C2
                                 <1>
                                          mov edx, eax
3030 00005698 C1EA03
                                <1>
                                          shr edx, 3
                                                                  ; to get offset to M.A.T.
3031
                                 <1>
                                                                  ; (1 allocation bit = 1 page)
3032
                                <1>
                                                                  ; (1 allocation bytes = 8 pages)
3033 0000569B 80E2FC
                                <1>
                                                dl, OFCh
                                                                  ; clear lower 2 bits
3034
                                 <1>
                                                                  ; (to get 32 bit position)
3035 0000569E 3B15[CC580100]
                                <1>
                                                 edx, [next_page] ; next free page
                                          cmp
3036 000056A4 7306
                                <1>
                                                short damb 1
                                          jnb
3037 000056A6 8915[CC580100]
                                <1>
                                                 [next_page], edx
                                          mov
3038
                                 <1> damb_1:
3039 000056AC BB00001000
                                <1>
                                          mov
                                                 ebx, MEM ALLOC TBL
                                          add
3040 000056B1 01D3
                                <1>
                                                 ebx, edx
3041 000056B3 83E01F
                                <1>
                                                eax, 1Fh ; 31
                                          and
3042
                                <1>
                                          ; 03/04/2016
3043
                                <1>
3044 000056B6 BA20000000
                                                edx, 32
                                <1>
                                          mov
3045 000056BB 28C2
                                                dl, al
                                <1>
                                          sub
3046 000056BD 39CA
                                <1>
                                          cmp
                                                edx, ecx
                                                short damb 2
3047 000056BF 7602
                                <1>
                                          jna
3048 000056C1 89CA
                                <1>
                                          mov
                                                 edx, ecx
                                <1> damb 2:
3050 000056C3 29D1
                                <1>
                                          sub
                                                ecx, edx
3051 000056C5 51
                                                ecx ; ***
                                <1>
                                          push
3052 000056C6 89D1
                                 <1>
                                                ecx, edx
                                          mov
3053
                                 <1> damb_3:
                                                               ; unlink/release/deallocate page
                                                 [ebx], eax
3054 000056C8 0FAB03
                                 <1>
                                          bts
3055
                                 <1>
                                                                   ; set relevant bit to 1.
                                 <1>
                                                                   ; set CF to the previous bit value
3056
3057 000056CB FF05[C8580100]
                                 <1>
                                          inc
                                                  dword [free_pages] ; 1 page has been deallocated (X = X+1)
3058 000056D1 49
                                 <1>
                                          dec
                                                 ecx
3059 000056D2 7404
                                 <1>
                                          jг
                                                 short damb 4
3060 000056D4 FECO
                                <1>
                                          inc
                                                 al
3061 000056D6 EBF0
                                                 short damb 3
                                <1>
                                           jmp
                                <1> damb 4:
3062
                                                 ecx ; ***
3063 000056D8 59
                                <1>
                                          pop
3064 000056D9 21C9
                                                 ecx, ecx ; 0 ?
                                 <1>
                                           and
3065 000056DB 741E
                                                 short damb_7
                                 <1>
                                           jz
                                <1>
                                          ; 03/04/2016
3067 000056DD B020
                                 <1>
                                          mov
                                                al, 32
                                 <1> damb 5:
3068
                                          add
3069 000056DF 83C304
                                <1>
                                                 ebx, 4
3070 000056E2 39C1
                                <1>
                                           cmp
                                                 ecx, eax ; 32
3071 000056E4 7305
                                <1>
                                          jnb
                                                 short damb_6
3072
                                <1>
                                          ; ECX < 32
3073 000056E6 28C0
                                <1>
                                          sub
                                                al, al ; 0
3074 000056E8 50
                                 <1>
                                          push eax ; 0 ***
3075 000056E9 EBDD
                                 <1>
                                          jmp
                                                 short damb_3
                                 <1> damb 6:
3077 000056EB 0105[C8580100]
                                <1>
                                                [free pages], eax; [free pages] = [free pages] + 32
                                          add
```

```
dword [ebx], OFFFFFFFFh; set 32 bits
3078 000056F1 C703FFFFFFF
                               <1>
                                        mov
3079 000056F7 29C1
                                <1>
                                          sub
                                                ecx, eax ; 32
                               <1> sub 
<1> jnz
3080 000056F9 75E4
                                                short damb_5
3081
                                <1> damb 7:
3082 000056FB 5B
                                <1>
                                                ebx ; **
                                          pop
3083 000056FC 5A
                                <1>
                                          pop
                                                edx ; *
3084 000056FD C3
                                <1>
3085
                                <1>
3086
                                <1> direct_memory_access:
                                        ; 22/07/2017
3087
                                <1>
3088
                                <1>
                                         ; 12/05/2017
3089
                                 <1>
                                         ; 16/07/2016
                                          ; 12/07/2016 (TRDOS 386 = TRDOS v2.0)
3090
                                <1>
3091
                                 <1>
                                        ; This processure will be called to map
3092
                                 <1>
                                        ; user's (ring 3) page tables to access phsical
3093
                                 <1>
                                          ; (flat/linear) memory addresses, directly (without
3094
                                 <1>
                                          ; kernel's data transfer functions).
3095
                                 <1>
3096
                                 <1>
                                          ; Purpose: Video memory access and shared memory access.
3097
                                 <1>
                                         ; INPUT ->
3098
                                 <1>
                                                EAX = Beginning address (physical).
3099
                                 <1>
                                                EBX = User's buffer address; 12/05/2017
3100
                                 <1>
3101
                                 <1>
                                                ECX = Number of contiguous pages to be mapped.
3102
                                 <1>
                                          ; OUTPUT ->
3103
                                 <1>
                                                User's page directory and pages tables
3104
                                 <1>
                                                will be updated.
3105
                                 <1>
3106
                                 <1>
                                                If an old page table entry has valid page address,
                                                that page will be deallocated just before PTE will
3107
                                 <1>
3108
                                 <1>
                                                be changed for direct (1 to 1) memory page access.
3109
                                 <1>
                                          ;
                                                If old PTE value points to a swapped page,
3110
                                 <1>
                                          ;
3111
                                 <1>
                                                   that page (block) will be unlinked on swap disk.
3112
                                 <1>
                                          ;
3113
                                 <1>
                                                Newly allocated pages (except page tables) will not
                                                be applied to Memory Allocation Table.
3114
                                                AVL bit 1 (PTE bit 10) of page table entry will be
3115
                                 <1>
                                                 used to indicate shared (direct) memory page; then,
3116
                                 <1>
3117
                                 <1>
                                                this page will not be deallocated later during
3118
                                 <1>
                                                process termination. (Memory Allocation Table and
                                                free memory count will not be affected.
3119
                                 <1>
                                                (Except deallocating page table's itself.)
3120
                                 <1>
3121
                                 <1>
3122
                                 <1>
                                                 CF = 1 -> error (EAX = error code)
3123
                                 <1>
                                                 CF = 0 -> success (EAX = beginning address)
3124
                                 <1>
3125
                                 <1>
                                          ;; (Modified Registers -> none)
3126
                                 <1>
                                          ; Modified registers: ebp, edx, ecx, ebx, esi, edi
3127
                                 <1>
3128
                                 <1>
3129
                                 <1>
                                          ; push ebp
3130
                                <1>
                                          ;push ebx
3131
                                <1>
                                          ;push ecx
3132
                                <1>
                                          ;push edx
3133 000056FE 662500F0
                                <1>
                                          and ax, PTE_A_CLEAR ; clear page offset
3134 00005702 50
                                <1>
                                          push eax
3135
                                <1>
                                          ;and ecx, ecx; page count
3136
                                <1>
                                                dmem_acc_7 ; 'insufficient memory' error
                                          ;jz
                                          mov
3137 00005703 89C5
                                <1>
                                                ebp, eax
3138 00005705 81C300004000
                                <1>
                                          add ebx, CORE ; 12/05/2017
                                <1> dmem acc 0:
3140 0000570B 891D[7C6F0100]
                                <1>
                                                 [base_addr], ebx; 12/05/2017
                                          mov
3141 00005711 A1[B8030300]
                                <1>
                                                eax, [u.pgdir]; page dir address (physical)
3142 00005716 E8D7F5FFFF
                                <1>
                                          call get_pte
3143
                                <1>
                                                ; EDX = Page table entry address (if CF=0)
                                                 ; Page directory entry address (if CF=1)
3144
                                <1>
3145
                                <1>
                                                        (Bit 0 value is 0 if PT is not present)
3146
                                <1>
                                                ; EAX = Page table entry value (page address)
                                                ; CF = 1 -> PDE not present or invalid ?
3147
                                <1>
3148 0000571B 7324
                                <1>
                                                short dmem_acc_1
3149
                                <1>
3150 0000571D E8B5F4FFFF
                                          call
                                <1>
                                               allocate_page
3151 00005722 0F82AB000000
                                <1>
                                          jс
                                                   dmem_acc_7 ; 'insufficient memory' error
                                <1>
3152
3153 00005728 E824F5FFFF
                                          call clear page
                                <1>
                                <1>
                                          ; EAX = Physical (base) address of the allocated (new) page
                                                al, PDE A PRESENT + PDE A WRITE + PDE A USER; 4+2+1 = 7
3155 0000572D 0C07
                                <1>
                                                        ; lower 3 bits are used as \overline{U/S}, R/W, P flags
3156
                                <1>
3157
                                                          ; (user, writable, present page)
                                <1>
3158 0000572F 8902
                                <1>
                                          mov
                                                [edx], eax ; Let's put the new page directory entry here !
3159 00005731 A1[B8030300]
                                 <1>
                                          mov
                                                 eax, [u.pgdir]
3160 00005736 E8B7F5FFFF
                               <1>
                                          call get_pte
3161 0000573B 0F8292000000
                                <1>
                                                  dmem_acc_7 ; 'insufficient memory' error
                                          jс
                                <1> dmem acc 1:
3162
3163
                                <1>
                                       ; EAX = PTE value, EDX = PTE address
3164 00005741 A801
                                          test al, PTE A PRESENT
                               <1>
                               <1>
<1>
3165 00005743 750D
                                        jnz short dmem_acc_2
3166 00005745 09C0
                                                eax, eax
                                          or
                                          j z
3167 00005747 7468
                               <1>
                                                short dmem_acc_6 ; Change PTE
                                          shr eax, 1 ; swap disk block (8 sectors) address
3168 00005749 D1E8
                               <1>
3169
                                <1>
                                          ; unlink swap disk block
                                          call unlink swap block
3170 0000574B E80CFBFFFF
                               <1>
                                          jmp short dmem acc 6
3171 00005750 EB5F
                               <1>
3172
                                <1>
                                <1> dmem_acc_2:
3173
3174 00005752 A802
                                <1>
                                          test al, PTE_A_WRITE ; bit 1, writable (r/w) flag
3175
                                                 ; (must be 1)
                                <1>
                                          jnz short dmem acc 4
3176 00005754 7550
                                <1>
                                          ; Read only -duplicated- page (belongs to a parent or a child)
3177
                                <1>
3178 00005756 66A90002
                               <1>
                                          test ax, PTE_DUPLICATED; Was this page duplicated
3179
                                <1>
                                                               ; as child's page ?
3180 0000575A 7455
                                                short dmem acc 5; Change PTE but don't deallocate the page!
                                <1>
                                          jz
3181
                                <1>
3182
                                 <1>
                                          ;push edi
```

```
3183
                                <1>
                                          ;push esi
3184
                                <1>
3185 0000575C 51
                                <1>
                                          push ecx
3186
                                <1>
                                          ;push ebx
3187 0000575D 8B1D[BC030300]
                                <1>
                                          mov ebx, [u.ppgdir] ; parent's page dir address (physical)
3188
                                <1>
                                <1>
                                         ; check the parent's PTE value is read only & same page or not..
3190 00005763 89EF
                                         mov edi, ebp
                                <1>
3191 00005765 C1EF16
                                <1>
                                          shr
                                                edi, PAGE_D_SHIFT ; 22
                                <1>
                                         ; EDI = page directory entry index (0-1023)
3193 00005768 89EE
                                <1>
                                          mov esi, ebp
                                         shr esi, PAGE_SHIFT; 12 and esi, PTE_MASK
3194 0000576A C1EE0C
                                <1>
3195 0000576D 81E6FF030000
                               <1>
3196
                                <1>
                                         ; ESI = page table entry index (0-1023)
3197
                                <1>
3198 00005773 66C1E702
                                               di, 2 ; * 4
                               <1>
                                         shl
                                         add ebx, edi; PDE offset (for the parent)
3199 00005777 01FB
                               <1>
3200 00005779 8B0F
                                         mov
                                <1>
                                               ecx, [edi]
3201 0000577B F6C101
                                <1>
                                          test cl, PDE_A_PRESENT; present (valid) or not?
3202 0000577E 7425
                               <1>
                                          jz
                                                short dmem acc 3 ; parent process does not use this page
3203 00005780 6681E100F0
                               <1>
                                          and
                                               cx, PDE_A_CLEAR ; OF000h ; Clear attribute bits
3204 00005785 66C1E602
                                <1>
                                          shl
                                               si, 2 ; *4
                                         add esi, ecx; PTE offset (for the parent)
3205 00005789 01CE
                               <1>
3206 0000578B 8B1E
                               <1>
                                          mov
                                               ebx, [esi]
                                         test bl, PTE_A_PRESENT ; present or not ?
3207 0000578D F6C301
                               <1>
                                                short dmem_acc_3 ; parent process does not use this page
3208 00005790 7413
                               <1>
                                          jz
3209 00005792 662500F0
                               <1>
                                         and
                                                ax, PTE A CLEAR; OF000h; Clear attribute bits
3210 00005796 6681E300F0
                               <1>
                                         and bx, PTE_A_CLEAR ; 0F000h ; Clear attribute bits
3211 0000579B 39D8
                                <1>
                                                eax, ebx ; parent's and child's pages are same ?
                                          cmp
                                       jne
                                                short dmem_acc_3 ; not same page
3212 0000579D 7506
                               <1>
                                                ; deallocate the child's page
3213
                                <1>
                                         or
3214 0000579F 800E02
                                <1>
                                                   byte [esi], PTE_A_WRITE ; convert to writable page (parent)
3215
                                <1>
                                          ;pop ebx
3216 000057A2 59
                                <1>
                                          pop
3217 000057A3 EB0C
                                <1>
                                          jmp
                                               short dmem_acc_5
                                <1> dmem acc 3:
3218
3219
                                <1>
                                         ;pop ebx
3220 000057A5 59
                                <1>
                                          pop
3221
                                <1> dmem acc 4:
3222 000057A6 66A90004
                                         test ax, PTE SHARED; shared or direct memory access indicator
                                <1>
3223 000057AA 7505
                                <1>
                                                short dmem_acc_5 ; AVL bit 1 = 1, do not deallocate this page!
3224
                                <1>
3225
                                               ax, PTE_A_CLEAR; 0F000h; clear lower 12 (attribute) bits
                                <1>
                                          ;and
3226 000057AC E804F6FFFF
                                <1>
                                          call deallocate page
3227
                                <1> dmem_acc_5:
3228
                                <1>
                                          ;pop esi
3229
                                <1>
                                          ;pop edi
                                <1> dmem_acc_6:
3230
3231 000057B1 89E8
                                <1>
                                         mov eax, ebp; physical page (offset=0) address
3232
                                <1>
                                         ; EAX = memory page address
3233
                                <1>
                                         ; EDX = PTE entry address (physical)
                                          or ax, PTE_A_PRESENT+PTE_A_USER+PTE_A_WRITE+PTE SHARED
3234 000057B3 660D0704
                                <1>
                                                      ; present flag, bit 0 = 1
3235
                                <1>
3236
                                <1>
                                                      ; user flag, bit 2 = 1
3237
                                <1>
                                                      ; writable flag, bit 1 = 1
3238
                                <1>
                                                      ; direct memory access flag, bit 10 = 1
                                <1>
                                                      ; (This page must not be deallocated!)
3240 000057B7 8902
                                               [edx], eax ; Update PTE value
                                <1>
                                         mov
3241 000057B9 49
                                <1>
                                                ecx; remain count of contiguous pages
                                          dec
3242 000057BA 741E
                                          jz
                                               short dmem_acc 8
                                <1>
3243 000057BC 81C500100000
                                <1>
                                         add ebp, PAGE_SIZE; next physical page address
3244
                                <1>
                                         ; 22/07/2017
3245
                                <1>
                                         ;mov eax, ebp
3246
                                <1>
                                         ; 12/05/2017
3247 000057C2 8B1D[7C6F0100]
                                <1>
                                          mov ebx, [base_addr] ; linear address (virtual+CORE)
                                                                ; next linear address
3248 000057C8 81C300100000
                                <1>
                                          add
                                               ebx, PAGE_SIZE
3249 000057CE E938FFFFFF
                                <1>
                                         jmp dmem_acc_0
3250
                                <1> dmem_acc_7: ; ERROR !
3251 000057D3 C7042404000000
                                <1>
                                         mov dword [esp], ERR_MINOR_IM
                                               ; Insufficient memory (minor) error!
3252
                                <1>
3253
                                <1>
                                                ; Major error = 0 (No protection fault)
3254
                                <1>
                                         ; cf = 1
                                <1> dmem_acc_8:
3255
3256 000057DA 58
                                <1>
                                      pop eax
3257
                                <1>
                                          ;pop edx
3258
                                <1>
                                          ;pop
                                                ecx
3259
                                <1>
                                                ebx
                                          ;pop
3260
                                <1>
                                          ;pop
                                                ebp
3261 000057DB C3
                                <1>
3262
                                <1>
                                <1> deallocate_user_pages:
3263
3264
                                <1>
                                          ; 20/05/2017
                                          ; 15/05/2017
3265
                                <1>
3266
                                <1>
                                          ; 20/02/2017
3267
                                <1>
                                          ; 19/02/2017 (TRDOS 386 = TRDOS v2.0)
3268
                                <1>
                                         ; Deallocate virtually contiguous user pages (memory block)
3269
                                 <1>
3270
                                <1>
                                          ; (caller: 'sysdalloc' system call)
3271
                                <1>
3272
                                <1>
                                          ; INPUT ->
3273
                                <1>
                                                EBX = VIRTUAL ADDRESS (beginning address)
3274
                                 <1>
                                                ECX = byte count
                                                [u.pgdir] = user's page directory
3275
                                <1>
3276
                                <1>
                                                [u.ppdir] = parent's page directory
3277
                                <1>
                                          ; OUTPUT ->
3278
                                <1>
3279
                                <1>
                                             If CF = 0
                                                EAX = Deallocated memory bytes
3280
                                <1>
                                                  (Even if shared or read only pages will not be
3281
                                <1>
3282
                                                   deallocated on M.A.T., this byte count will be
                                <1>
3283
                                <1>
                                                   returned as virtually deallocated bytes; in fact
3284
                                <1>
                                                   virtually deallocated user pages * 4096.)
                                                EBX = Virtual address (as rounded up)
3285
                                <1>
3286
                                <1>
                                             If CF = 1
3287
                                <1>
                                               EAX = 0 (there is not any deallocated pages)
```

```
3288
                                 <1>
3289
                                 <1>
                                          ; Note: Empty page tables will not be deallocated!!!
3290
                                 <1>
                                                (they will be deallocated at process termination stage)
3291
                                 <1>
3292
                                 <1>
                                          ; Modified Registers -> EAX, EDX, ESI, EDI, EBX, ECX, EBP
3293
                                <1>
3294 000057DC 89DE
                                <1>
                                                esi, ebx
                                          mov
3295 000057DE 89F7
                                                edi, esi
                                <1>
                                          mov
3296 000057E0 01CF
                                <1>
                                                edi, ecx
                                          add
3297 000057E2 81C6FF0F0000
3298 000057E8 C1EE0C
                                          add
                                               esi, PAGE SIZE - 1 ; 4095 (round up)
                               <1>
3298 000057E8 C1EE0C
                                <1>
                                          shr esi, PAGE_SHIFT
3299 000057EB C1EF0C
                                          shr
mov
                                <1>
                                                edi, PAGE SHIFT
3300 000057EE 89F8
3301 000057F0 29F0
3302 000057F2 0F86D5000000
                                <1>
                                                eax, edi ; end page
                                          sub eax, esi; end page - start page
                               <1>
                               <1>
                                          jna
                                                da_u_pd_err ; < 1
3303 000057F8 89F3
                                <1>
                                          mov
                                                ebx, esi
3304 000057FA C1E30C
                               <1>
                                          shl
                                                ebx, PAGE_SHIFT ; virtual address (as rounded up)
                                          push ebx ; *
3305 000057FD 53
                                <1>
3306 000057FE 89C1
                                <1>
                                                ecx, eax ; page count
                                          mov
3307 00005800 C1E00C
                                <1>
                                          shl
                                                eax, PAGE_SHIFT ; byte count as adjusted
3308 00005803 50
                                <1>
                                          push eax ; **
                                         mov
3309 00005804 8B1D[B8030300]
                                <1>
                                                ebx, [u.pgdir] ; physical addr of user's page dir
3310 0000580A 81C600040000
                                                esi, CORE/PAGE_SIZE
                                <1>
3311 00005810 89F7
                                <1>
                                         mov
                                                edi, esi
                                       and edi, PTE_MASK; PTE entry in the page table
3312 00005812 81E7FF030000
                                <1>
3313 00005818 57
                                <1>
                                          push edi ; *** ; PTE index (of page directory)
                                         shr esi, PAGE_D_SHIFT - PAGE SHIFT ; 22-12=10
3314 00005819 C1EE0A
                               <1>
3315 0000581C 89F2
                               <1>
                                          mov edx, esi
                                <1>
                                          ; EDX = PDE index
3316
                             <1><1>
3317 0000581E C1E602
                                          shl esi, 2; convert PDE index to dword offset
3318 00005821 01DE
                               <1>
                                          add
                                                esi, ebx ; add page directory address
3319
                                <1> da u pd 1:
3320 00005823 AD
                                <1>
                                         lodsd
3321
                                <1>
3322 00005824 89F5
                                <1>
                                              ebp, esi ; 20/02/2017
                                         mov
3323
                                <1>
                                          ; EBP = next PDE address
3324
                               <1>
3325 00005826 A801
                                <1>
                                          test al, PDE_A_PRESENT; bit 0, present flag (must be 1)
                                          jz da_u_pd_3; 20/05/2017
and ax, PDE_A_CLEAR; 0F000h; clear lower 12 (attribute) bits
3326 00005828 0F8494000000
                                <1>
3327 0000582E 662500F0
                                <1>
3328
                                <1>
                                          ; EAX = PHYSICAL (flat) ADDRESS OF THE PAGE TABLE
3329 00005832 8B3C24
                                <1>
                                         mov edi, [esp] ; ***
                                          ; EDI = PTE index (of complete page directory)
3330
                                <1>
                                          ; and edi, PTE MASK ; PTE entry in the page table
3331
                               <1>
3332 00005835 C1E702
                                <1>
                                          shl edi, 2; convert PTE index to dword offset
3333 00005838 89FE
                             <1>
<1>
<1>
                                          mov
                                                esi, edi ; PTE offset in page table (0-4092)
3334 0000583A 01C6
                                          add esi, eax; now, esi points to requested PTE
3335
                               <1> da_u_pt_0:
3336 0000583C AD
                                <1>
                                          lodsd
3337 0000583D A801
                               <1>
                                          test al, PTE_A_PRESENT; bit 0, present flag (must be 1)
3338 0000583F 743F
                               <1>
                                          jz short da_u_pt_1
3339
                                <1>
3340 00005841 A802
                                <1>
                                          test al, PTE_A_WRITE ; bit 1, writable (r/w) flag
                                                              ; (must be 1)
3341
                                <1>
3342 00005843 7549
                                <1>
                                          jnz short da_u_pt_3
3343
                                <1>
                                          ; Read only -duplicated- page (belongs to a parent or a child)
                                          test ax, PTE_DUPLICATED; Was this page duplicated
3344 00005845 66A90002
                                <1>
3345
                                 <1>
                                                           ; as child's page ?
3346 00005849 744E
                                 <1>
                                                short da_u_pt_4; Clear PTE but don't deallocate the page!
                                          jΖ
3347
                                <1>
3348
                                <1>
                                          ; check the parent's PTE value is read only & same page or not..
3349
                                 <1>
                                          ; EDX = page directory entry index (0-1023)
                                          push edx; ****
3350 0000584B 52
                                <1>
3351
                                <1>
                                          ; EDI = page table entry offset (0-4092)
                                          mov ebx, [u.ppgdir] ; page directory of the parent process
3352 0000584C 8B1D[BC030300]
                                <1>
3353 00005852 66C1E202
                                <1>
                                          shl
                                                dx, 2; *4
                                          add ebx, edx; PDE address (for the parent)
3354 00005856 01D3
                                <1>
3355 00005858 8B13
                                <1>
                                          mov edx, [ebx]; page table address
3356 0000585A F6C201
                                <1>
                                          test dl, PDE_A_PRESENT ; present (valid) or not ?
                                                short da_u_pt_2 ; parent process does not use this page
                                          jz
3357 0000585D 742E
                                <1>
                         <1>
<1>
<1>
                                          and dx, PDE_A_CLEAR; OF000h; Clear attribute bits
3358 0000585F 6681E200F0
3359
                                <1>
                                          ; EDI = page table entry offset (0-4092)
3360 00005864 01D7
                                                            ; PTE address (for the parent)
                                <1>
                                          add edi, edx
3361 00005866 8B1F
                               <1>
                                          mov ebx, [edi]
3362 00005868 F6C301
                                <1>
                                          test bl, PTE_A_PRESENT; present or not?
3363 0000586B 7420
                                <1>
                                          jz
                                                short da_u_pt_2 ; parent process does not use this page
3364 0000586D 662500F0
                                <1>
                                          and
                                                ax, PTE A CLEAR; OF000h; Clear attribute bits
                                          and bx, PTE_A_CLEAR ; 0F000h ; Clear attribute bits
3365 00005871 6681E300F0
                                <1>
3366 00005876 39D8
                                <1>
                                                eax, ebx ; parent's and child's pages are same ?
                                          cmp
                                          jne short da_u_pt_2 ; not same page
3367 00005878 7513
                                <1>
                                                byte [edi], PTE_A_WRITE ; convert to writable page (parent)
edx ; ****
3368
                                <1>
3369 0000587A 800F02
                                 <1>
                                            or
3370 0000587D 5A
                                <1>
                                          pop
3371 0000587E EB19
                                 <1>
                                                 short da_u_pt_4
                                          jmp
                                 <1> da_u_pt_1:
3372
3373 00005880 09C0
                                                 eax, eax ; swapped page ?
                                <1>
                                          or
3374 00005882 741C
                                <1>
                                                 short da_u_pt_5 ; no
                                          jΖ
3375
                                <1>
3376 00005884 D1E8
                                <1>
                                          shr
                                                 eax, 1
                                                unlink_swap_block ; Deallocate swapped page block
3377 00005886 E8D1F9FFFF
                                <1>
                                          call
3378
                                <1>
                                                              ; on the swap disk (or in file)
3379 0000588B EB13
                                 <1>
                                          qmŗ
                                                 short da_u_pt_5
                                <1> da_u_pt_2:
3380
3381 0000588D 5A
                                <1>
                                         pop
                                <1> da_u_pt_3:
3382
                                          test
3383 0000588E 66A90004
                                <1>
                                                ax, PTE_SHARED
                                                                   ; shared or direct memory access indicator
                                                                 ; AVL bit 1 = 1, do not deallocate this page!
3384 00005892 7505
                                 <1>
                                                 short da u pt 4
                                          jnz
3385
                                <1>
                                                 ax, PTE A CLEAR; 0F000h; clear lower 12 (attribute) bits
3386
                                 <1>
                                          ;and
                                                deallocate page; set the mem allocation bit of this page
3387 00005894 E81CF5FFFF
                                 <1>
                                         call
3388
                                 <1> da_u_pt_4:
3389 00005899 C746FC00000000
                                 <1>
                                                dword [esi-4], 0 ; clear/reset PTE (child, dupl. as parent)
                                         mov
3390
                                 <1> da_u_pt_5:
                                         ; 20/05/2017
                                 <1>
3391
3392 000058A0 58
                                 <1>
                                          pop eax; *** PTE index (of page directory)
```

```
3393 000058A1 49
                                  <1>
                                             dec ecx ; remain page count
                                             jz short da_u_pd_
inc eax; next PTE
3394 000058A2 7426
                                  <1>
                                                    short da u pd 4
                                  <1>
3395 000058A4 40
3396 000058A5 6625FF03
                               <1>
                                         and ax, PTE_MASK; PTE entry index in the page table
3397 000058A9 50
                                  <1>
                                           push eax ; *** (save again)
3398
                                   <1>
                                             ;mov edi, eax
                                            ;and di, PTE_MASK
3399
                                  <1>

<!> ;and di, PTE_MASK
<!> ;cmp edi, PAGE_SIZE / 4 ; 1024
<!> ;jnb short da_u_pd_2
<!> mov edi, eax
<!> shl edi, 2 ; convert index to dword offset
<!> ;test ax, PTE_MASK ; 3FFh
<!> or eax, eax
<!> jnz short da_u_pt_0 ; 1-1023
<!> da u pd_2 ;
3400
3401
3402 000058AA 89C7
3403 000058AC C1E702
3405 000058AF 09C0
3406 000058B1 7589
                                   <1> da_u_pd_2:
3407
3408 000058B3 42
                                  edx
                                             ; 20/05/2017
                                  <1>
                                  <1> and dx, PTE_MASK; 3FFh
<1> jz short da_u_pd_4; 0
<1> ;cmp edx, 1024
<1> ;jnb short da_u_pd_4
<1> mov
3410 000058B4 6681E2FF03
3411 000058B9 740F
                                                    short da_u_pd_4 ; 0 (1024)
3412
3413
                             <1> mov
<1> jmp
                                                    esi, ebp; 20/02/2017
3414 000058BB 89EE
3415 000058BD E961FFFFF
                                                    da_u_pd_1
                                  <1> da_u_pd_3:
                                  <1> ; 15/05/2017 (empty page directory entry)
3417
3418 000058C2 81E900040000
                                  <1>
                                              sub ecx, 1024
                                           ja
3419 000058C8 77E9
                                  <1>
                                                    short da_u_pd_2 ; 20/05/2017
                                  <1> da_u_pd_4:
3420
3421 000058CA 58
                                  <1>
                                                    eax ; **
                                             pop
3422 000058CB 5B
                                                    ebx ; *
                                  <1>
                                              pop
3423 000058CC C3
                                  <1>
3424
                                   <1>
                                   <1> da_u_pd_err:
3425
3426 000058CD 31C0
                                  <1>
                                          xor eax, eax
3427 000058CF F9
                                   <1>
                                              stc
3428 000058D0 C3
                                   <1>
                                             retn
3429
                                   <1>
                                   <1> allocate_user_pages:
3430
                                         ; \overline{2}0/05\overline{/2}0\overline{17}
3431
                                   <1>
                                             ; 01/05/2017, 02/05/2017, 15/05/2017
3432
                                   <1>
                                           ; 04/03/2017
3433
                                   <1>
                                            ; 20/02/2017 (TRDOS 386 = TRDOS v2.0)
3434
                                   <1>
3435
                                   <1>
3436
                                   <1>
                                           ; Allocate physically contiguous user pages (memory block)
3437
                                   <1>
                                           ; (caller: 'sysalloc' system call)
3438
                                   <1>
                                             ; Note: This procedure does not alloc a page's itself
3439
                                   <1>
3440
                                   <1>
                                                    (page bit) on Memory Allocation Table.
                                            ;
                                                     (allocate memory block is needed before this proc)
3441
                                   <1>
3442
                                   <1>
                                             ; INPUT ->
3443
                                   <1>
3444
                                   <1>
                                                    EAX = PHYSICAL ADDRESS (beginning address)
                                                    EBX = VIRTUAL ADDRESS (beginning address)
3445
                                   <1>
3446
                                   <1>
                                                    ECX = byte count (>=4096)
3447
                                   <1>
                                                    [u.pgdir] = user's page directory
3448
                                   <1>
3449
                                   <1>
                                                    Note: All addresses are (must be) already adjusted
                                             ;
3450
                                   <1>
                                                    to page borders, otherwise, lower 12bits of addresses
3451
                                   <1>
                                                    and byte count would be truncated.
3452
                                   <1>
                                             ;
                                            ; OUTPUT ->
3453
                                   <1>
3454
                                   <1>
3455
                                   <1>
3456
                                   <1>
                                                    CF = 1 -> insufficient memory error
3457
                                   <1>
3458
                                   <1>
                                              ; Note: All pages will be allocated in physical page order
3459
                                   <1>
                                              ; from the beginning page address.
3460
                                   <1>
                                                    * A new page table will be added to the page dir
3461
                                   <1>
                                                      when the requested PDE is invalid.
3462
                                   <1>
                                                     * Those pages will not be added to swap queue
3463
                                   <1>
                                                    because main purpose of this allocation is to
3464
                                   <1>
                                                      set a direct memory access (DMA controller) buffer.
                                                     (Swapping out a page in a DMA buffer would be wrong!)
3465
                                   <1>
3466
                                   <1>
                                                    * Previous content of page tables (PTEs) would be
3467
                                   <1>
                                                      (should be) deallocated before entering this
3468
                                   <1>
                                                      procedure. So, new page table entries (PTEs)
                                                    directly will be written without checking
3469
                                   <1>
3470
                                                      their previous content.
                                   <1>
3471
                                   <1>
                                                    * Only solution to increase free memory by removing
                                                      that non-swappable memory block is to terminate
3472
                                   <1>
3473
                                   <1>
                                                       the process or to wait until the process will
3474
                                    <1>
                                                       deallocate that memory block as itself. ('sysdalloc')
3475
                                   <1>
                                                       (No problem, if the process does not grab all of
3476
                                   <1>
                                                       -very big amount of- free memory by using
3477
                                   <1>
                                                       'sysalloc' system call!?)
3478
                                   <1>
                                                       (Even if the process has grabbed all of free memory,
3479
                                   <1>
                                                       no problem if the process is not running in
3480
                                   <1>
                                                       multitasking mode. No problem in multitasking
                                                       mode if there is not another process which is running
3481
                                   <1>
3482
                                   <1>
                                                       or waiting or sleeping for an event as it's pages
                                                       are swapped-out. But a new process can not start to
3483
                                   <1>
3484
                                   <1>
                                                       run if all of free memory has beeen allocated
                                                       by running processes. Deallocation -'sysdalloc'-
3485
                                   <1>
3486
                                   <1>
                                                       or terminate a running process is needed
3487
                                   <1>
                                                       in order to run a new process.)
3488
                                   <1>
3489
                                   <1>
                                              ; Modified Registers -> EAX, EDX, ESI, EDI, EBX, ECX, EBP
3490
                                   <1>
3491
                                   <1>
3492
                                              ; 01/05/2017
                                   <1>
3493 000058D1 662500F0
                                   <1>
                                              and
                                                    ax, ~PAGE_OFF
3494 000058D5 6681E300F0
                                   <1>
                                              and
                                                    bx, ~PAGE OFF
                                              ; 02/05/2017
                                   <1>
3496 000058DA BD00F0FFFF
                                   <1>
                                                    ebp, OFFFFF000h; 4 Giga Bytes - 4096 Bytes (for Stack)
3497 000058DF C1E90C
                                   <1>
                                                    ecx, PAGE_SHIFT ; page count
                                              shr
```

```
3498 000058E2 83F901
                              <1>
                                           cmp
                                                 ecx, 1
3499 000058E5 7251
                                <1>
                                           jb
                                                 short a_u_im_retn
                                <1>
3500 000058E7 89C2
                                           mov
                                                 edx, eax
                              <1>
3501 000058E9 01CA
                                           add
                                                 edx, ecx
                                <1>
<1>
                                           jс
3502 000058EB 724B
                                                 short a_u_im_retn
3503 000058ED 39D5
                                           cmp
                                                 ebp, edx
3504 000058EF 7247
                                <1>
                                           jb
                                                 short a u im retn
3505 000058F1 89DA
                                <1>
                                                 edx, ebx
                                           mov
3506 000058F3 81C200004000
                                <1>
                                           add
                                                 edx, CORE
                                                 short a_u_im_retn
3507 000058F9 723D
                                <1>
                                           jс
3508 000058FB 01CA
                                <1>
                                           add
                                                 edx, ecx
3509 000058FD 7239
                                 <1>
                                           jс
                                                 short a u im retn
3510 000058FF 39D5
                                <1>
                                                 ebp, edx
                                           cmp
3511 00005901 7235
                                <1>
                                                 short a_u_im_retn
                                           jb
3512
                                 <1>
                                          ;
3513 00005903 89C5
                                <1>
                                          mov
                                                 ebp, eax ; physical address
3514 00005905 89DE
                                <1>
                                          mov
                                                 esi, ebx
3515 00005907 81C600004000
                                                esi, CORE ; start of user's memory (4M)
                                <1>
                                           add
                                                 esi, PAGE SHIFT; higher 20 bits of the linear address
3516 0000590D C1EE0C
                                 <1>
                                           shr
3517
                                <1>
                                          ;shr ecx, PAGE SHIFT ; page count
                                                 ebx, [u.pgdir] ; physical addr of user's page dir
3518 00005910 8B1D[B8030300]
                                <1>
                                          mov
3519 00005916 89F7
3520 00005918 81E7FF030000
                                 <1>
                                          mov
                                                 edi, esi
                                <1>
                                          and
                                                edi, PTE_MASK ; PTE entry index in the page table
3521 0000591E 57
                                <1>
                                          push edi ; * ; PTE index (in page directory)
                                          shr esi, PAGE_D_SHIFT - PAGE_SHIFT; 22-12=10
3522 0000591F C1EE0A
                                <1>
3523 00005922 89F2
                                <1>
                                           mov
                                                 edx, esi
                                <1>
                                          ; EDX = PDE index
                                <1>
                                           shl esi, 2 ; convert PDE index to dword offset
3525 00005924 C1E602
3526 00005927 01DE
                                <1>
                                                 esi, ebx; add page directory address
                                           add
3527
                                <1> a_u_pd_0:
3528 00005929 AD
                                 <1>
                                           lodsd
3529
                                 <1>
3530 0000592A 89F3
                                 <1>
                                                 ebx, esi ; next PDE address
                                           mov
3531
                                <1>
                                           test al, PDE_A_PRESENT; bit 0, present flag (must be 1)
3532 0000592C A801
                                 <1>
3533 0000592E 7513
                                <1>
                                           jnz
                                                short a_u_pd_2
                                 <1>
                                          ; empty PDE (it does not point to valid page table address)
3535
                                 <1>
                                           call allocate page ; (allocate a new page table)
3536 00005930 E8A2F2FFFF
                                 <1>
                                           jnc short a_u_pd_1; OK... now, we have a new page table.
3537 00005935 7302
                                 <1>
3538
                                 <1>
                                           ; cf = 1
3539
                                 <1>
                                           ; There is not a free memory page to allocate a new page table !!!
3540 00005937 5E
                                 <1>
                                           pop esi; '
3541
                                 <1> a_u_im_retn:
3542 00005938 C3
                                           retn ; return to 'sysalloc' with 'insufficient memory' error
                                 <1>
3543
                                 <1>
                                 <1> a u pd 1: ; clear the new page table content
3544
                                        ; EAX = Physical (base) address of the new page table
3545
                                 <1>
3546 00005939 E813F3FFFF
                                <1>
                                           call clear_page ; Clear page content
3547
                                <1>
                                          ;
                                                 al, PDE_A_PRESENT + PDE_A_WRITE + PDE_A_USER
3548 0000593E 0C07
                                <1>
                                                 ; set \overline{\text{bit}} 0, bit 1 and \overline{\text{bit}} 2 to 1
3549
                                 <1>
                                                  ; (present, writable, user)
                                <1>
3550
3551 00005940 8946FC
                                <1>
                                                [esi-4], eax
3552
                                 <1> a_u_pd_2:
3553 00005943 662500F0
                                <1>
                                          and
                                                 ax, PDE_A_CLEAR; 0F000h; clear lower 12 (attribute) bits
                                          ; EAX = PHYSICAL (flat) ADDRESS OF THE PAGE TABLE
                                <1>
3555 00005947 8B3C24
                                 <1>
                                          mov edi, [esp] ; *
3556
                                 <1>
                                          ; EDI = PTE index (of page directory)
3557
                                <1>
                                          ;and edi, PTE_MASK; PTE entry index in the page table
                                        ; EBX = next PDE address
3558
                                <1>
                                          mov esi, edi; PTE index in page table (0-1023) shl edi, 2; convert PTE index to dword offset
3559 0000594A 89FE
                                 <1>
3560 0000594C C1E702
                                <1>
3561 0000594F 01C7
                                <1>
                                           add edi, eax; now, edi points to requested PTE
3562
                                 <1> a_u_pt_0:
3563
                                 <1>
                                           ; 02/05/2017
3564 00005951 8B07
                                <1>
                                           mov eax, [edi]
3565
                                 <1>
3566 00005953 A801
                                 <1>
                                           test al, PTE_A_PRESENT; bit 0, present flag (must be 1)
3567 00005955 7445
                                <1>
                                                 short a_u_pt_1
                                           jΖ
3568
                                 <1>
3569 00005957 A802
                                 <1>
                                          test al, PTE A WRITE ; bit 1, writable (r/w) flag
3570
                                 <1>
                                                                ; (must be 1)
3571 00005959 7550
                                 <1>
                                           jnz short a u pt 3
                                          ; Read only -duplicated- page (belongs to a parent or a child)
                                 <1>
3572
3573 0000595B 66A90002
                                           test ax, PTE_DUPLICATED; Was this page duplicated
                                 <1>
                                                                ; as child's page ?
                                 <1>
3575 0000595F 7455
                                 <1>
                                           jΖ
                                               short a_u_pt_4 ; Clear PTE but don't deallocate the page!
3576
                                 <1>
3577
                                 <1>
                                           ; check the parent's PTE value is read only & same page or not..
3578
                                 <1>
                                           ; EDX = page directory entry index (0-1023)
                                           push edx; **
push ebx; ***
3579 00005961 52
                                  <1>
3580 00005962 53
                                 <1>
                                 <1>
                                           ; ESI = page table entry index (0-1023)
3582
                                 <1>
                                           ;push esi ; **** ; 20/05/2017
3583 00005963 8B1D[BC030300]
                                <1>
                                           mov ebx, [u.ppgdir]; page directory of the parent process
3584 00005969 66C1E202
                                                 dx, 2; *4
                                <1>
3585 0000596D 01D3
                                <1>
                                           add ebx, edx; PTE address, 0 (for the parent)
3586 0000596F 8B13
                                <1>
                                                 edx, [ebx] ; page table address
                                           mov
3587 00005971 F6C201
                                           test dl, PDE_A_PRESENT ; present (valid) or not ?
                                <1>
                                                 short a_u_pt_2 ; parent process does not use this page
3588 00005974 7433
                                <1>
                                          jz
3589 00005976 6681E200F0
                                <1>
                                           and
                                                 dx, PDE A CLEAR; OF000h; Clear attribute bits
                                           shl si, 2; \star 4
3590 0000597B 66C1E602
                                <1>
                                <1>
                                           ; ESI = page table entry offset (0-4092)
3592 0000597F 01D6
                                <1>
                                           add esi, edx ; PTE address (for the parent)
                                <1>
3593 00005981 8B1E
                                           mov
                                                 ebx, [esi]
3594 00005983 F6C301
                               <1>
                                           test bl, PTE_A_PRESENT ; present or not ?
3590 UU0005988 662500F0
3597 0000598C 6681E300F0
3598 00005991 39D8
3599 00005993 7514
                                <1>
                                                 short a_u_pt_2 ; parent process does not use this page
                                           jz
                                <1>
                                                 ax, PTE_A_CLEAR ; OF000h ; Clear attribute bits
                                <1>
                                           and bx, PTE_A_CLEAR; 0F000h; Clear attribute bits
                                <1>
                                           cmp eax, ebx ; parent's and child's pages are same ?
                                <1>
                                                 short a_u_pt_2 ; not same page
                                           jne
                                           ; deallocate the child's page or byte [esi], PTE_A_WRITE; convert to writable page (parent)
3600
                                <1>
3601 00005995 800E02
                                <1>
                                 <1>
                                           ;pop esi; ****; 20/05/2017
3602
```

```
3603 00005998 5B
                          <1>
                                        ebx ; ***
                                 pop
                        <1> pop
<1> jmp
3604 00005999 5A
                                        edx ; **
3605 0000599A EB1A
                                        short a_u_pt_4
3606
                          <1> a_u_pt_1:
                          <1> or <1> jz
                                        eax, eax ; swapped page ?
3607 0000599C 09C0
                                        short a_u_pt_4 ; no
                                        unlink_swap_block ; Deallocate swapped page block
                                               ; on the swap disk (or in file)
                                        short a_u_pt_4
                          <1> ;pop esi; ****; 20/05/2017
                       <1> pop
<1> pop
<1> a_u_pt_3:
                                        ebx ; ***
3616 000059A9 5B
3617 000059AA 5A
                                        edx ; **
3618

<1> test ax, PTE_SHARED ; shared or direct memory access indicator
<1> jnz short a u pt 4 ; AVL bit 1 = 1, do not deallocate this page.

3619 000059AB 66A90004
                          <1>
                                        short a_u_pt_4 ; AVL bit 1 = 1, do not deallocate this page!
3620 000059AF 7505
                                   jnz
3621
                           <1>
                          <1>
3622
                                   ;and
                                       ax, PTE A CLEAR; 0F000h; clear lower 12 (attribute) bits
al, PTE_A_PRESENT + PTE_A_WRITE + PTE_A_USER; 04/03/2017
                           3642
                           <1>;
                                   cmp edx, 1024
                                        short a_u_pd_4; 02/05/2017 (error!, ecx > 0)
                           <1> ; jnb
3643
                          <1> mov
<1> jmp
3644 000059CF 89DE
                                        esi, ebx ; the next PDE address
3645 000059D1 E953FFFFFF
                                        a_u_pd_0
3646
                           <1> a_u_pd_4:
                           <1> ; 02/05/2017 <1> ; stc
3647
3648
3649
                           <1> a_u_pd_5:
3650
                           <1> ; 20/05/2017
                                   ;pop edi; *
3651
                           <1>
3652 000059D6 C3
                           <1>
                                   retn
3653
                           <1>
3654
                           <1>
                           <1> ; /// End Of MEMORY MANAGEMENT FUNCTIONS ///
3655
3656
                           <1>
3657
                           <1> ;; Data:
3658
                           <1>
                           <1>; 09/03/2015
3659
                           <1> ; swpq_count: dw 0 ; count of pages on the swap que
3660
                           3661
3662
                           <1> ; swpd free: dd 0 ; free page blocks (4096 bytes) on swap disk/drive (logical)
3663
                           <1> ;swpd_next: dd 0 ; next free page block
3664
3665
                           <1> ;swpd_last: dd 0 ; last swap page block
2160
                              %include 'timer.s' ; 17/01/2015
                            1
                           <1>; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.0 - timer.s
  2
  3
                           4
                            <1> ; Last Update: 15/01/2017
   5
   6
                            <1> ; Beginning: 17/01/2016
   7
                           <1>; Assembler: NASM version 2.11 (trdos386.s)
   8
                            <1>; -------
  9
  10
                            <1>; Turkish Rational DOS
                           <1>; Operating System Project v2.0 by ERDOGAN TAN (Beginning: 04/01/2016)
  11
  12
  13
                           <1>; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
  14
                            <1>;
                           <1>; Derived from 'IBM PC-AT' BIOS source code (1985)
  15
                            16
                            <1>
                            <1>; TRDOS 386 (TRDOS v2.0) Kernel - TIMER & REAL TIME CLOCK (BIOS) FUNCTIONS
  18
  19
  20
                            <1>; IBM PC-AT BIOS Source Code ('BIOS2.ASM')
                            <1>; TITLE BIOS2 ---- 06/10/85 BIOS INTERRUPT ROUTINES
  21
  22
  23
                           <1>:
  24
                           <1>; ////// TIMER (& REAL TIME CLOCK) FUNCTIONS ///////////
  25
                           <1>
  26
                           <1> int1Ah:
                                ; 29/01/2016
  27
                           <1>
                                   ; 17/01/2016 (TRDOS 386 = TRDOS v2.0)
  28
                           <1>
  29 000059D7 9C
                           <1>
  30 000059D8 0E
                                 push cs
call TIME_OF_DAY_1
                           <1>
  31 000059D9 E801000000
                           <1>
                                  retn
  32 000059DE C3
                           <1>
  33
                           <1>
  34
                           <1> ;--- INT 1A H -- (TIME OF DAY) ------
                           <1>; THIS BIOS ROUTINE ALLOWS THE CLOCKS TO BE SET OR READ
  35
  36
                           <1>;
  37
                           <1> ; PARAMETERS:
                           <1>; (AH) = 00H READ THE CURRENT SETTING AND RETURN WITH,
  38
  39
                           <1>;
                                                 (CX) = HIGH PORTION OF COUNT
  40
                           <1>;
                                                 (DX) = LOW PORTION OF COUNT
```

```
(AL) = 0 TIMER HAS NOT PASSED 24 HOURS SINCE LAST READ:
 41
                                <1>;
 42
                                 <1>;
                                                                  1 IF ON ANOTHER DAY. (RESET TO ZERO AFTER READ) :
 43
                                <1>;
 44
                                 <1>;
                                           (AH) = 01H SET THE CURRENT CLOCK USING,
 45
                                                     (CX) = HIGH PORTION OF COUNT
                                 <1>;
 46
                                 <1>;
                                                      (DX) = LOW PORTION OF COUNT.
                                 <1>;
                                                    NOTE: COUNTS OCCUR AT THE RATE OF 1193180/65536 COUNTS/SECOND:
                                <1>;
 48
 49
                                <1>;
                                                                (OR ABOUT 18.2 PER SECOND -- SEE EQUATES)
 50
                                <1>;
 51
                                 <1>;
                                           (AH) = 02H READ THE REAL TIME CLOCK AND RETURN WITH,
                                                           (CH) = HOURS IN BCD (00-23)
 52
                                 <1>;
                                <1>;
 53
                                                           (CL) = MINUTES IN BCD (00-59)
                                <1>;
                                                            (DH) = SECONDS IN BCD (00-59)
 54
                                                           (DL) = DAYLIGHT SAVINGS ENABLE (00-01)
 55
                                <1>;
                                <1> ;
 56
                                 <1>;
                                          (AH) = 03H SET THE REAL TIME CLOCK USING,
                                                          (CH) = HOURS IN BCD (00-23)
 58
                                <1>;
                                <1>;
                                                           (CL) = MINUTES IN BCD (00-59)
 59
                                                          (DH) = SECONDS IN BCD (00-59)
 60
                                <1>;
                                <1> ;
 61
                                                          (DL) = 01 IF DAYLIGHT SAVINGS ENABLE OPTION, ELSE 00.
                                <1>;
 62
                                                  NOTE: (DL) = 00 IF DAYLIGHT SAVINGS TIME ENABLE IS NOT ENABLED. :
 63
                                <1>;
                                                        (DL) = 01 ENABLES TWO SPECIAL UPDATES THE LAST SUNDAY IN :
                                                     APRIL (1:59:59 --> 3:00:00 AM) AND THE LAST SUNDAY IN
 65
                                 <1>;
                                <1>;
                                                        OCTOBER (1:59:59 --> 1:00:00 AM) THE FIRST TIME.
 66
 67
                                 <1>;
                                           (AH) = 04H READ THE DATE FROM THE REAL TIME CLOCK AND RETURN WITH,
 68
                                <1>;
 69
                                                           (CH) = CENTURY IN BCD (19 OR 20)
                                 <1>;
                                                           (CL) = YEAR IN BCD (00-99)
 70
                                 <1>;
                                <1> ;
 71
                                                            (DH) = MONTH IN BCD (01-12)
                                <1>;
 72
                                                           (DL) = DAY IN BCD (01-31).
                                <1>;
 73
 74
                                 <1>;
                                           (AH) = 05H SET THE DATE INTO THE REAL TIME CLOCK USING,
                                <1>;
                                                          (CH) = CENTURY IN BCD (19 OR 20)
 75
 76
                                <1>;
                                                           (CL) = YEAR IN BCD (00-99)
                                <1>;
 77
                                                           (DH) = MONTH IN BCD (01-12)
 78
                                                           (DL) = DAY IN BCD (01-31).
                                <1>;
 79
                                 <1>;
                                           (AH) = 06H SET THE ALARM TO INTERRUPT AT SPECIFIED TIME,
 80
                                <1>;
                                <1> ;
 81
                                                           (CH) = HOURS IN BCD (00-23 (OR FFH))
                                <1>;
 82
                                                           (CL) = MINUTES IN BCD (00-59 \text{ (OR FFH)})
                                <1>;
                                                          (DH) = SECONDS IN BCD (00-59 (OR FFH))
 83
                                 <1>;
 84
 85
                                 <1>;
                                          (AH) = 07H RESET THE ALARM INTERRUPT FUNCTION.
 86
                                 <1>;
                                 <1>; NOTES: FOR ALL RETURNS CY= 0 FOR SUCCESSFUL OPERATION.
 87
                                          FOR (AH) = 2, 4, 6 - CARRY FLAG SET IF REAL TIME CLOCK NOT OPERATING. :
 88
                                <1>;
                                <1>;
 89
                                             FOR (AH) = 6 - CARRY FLAG SET IF ALARM ALREADY ENABLED. :
                                           FOR THE ALARM FUNCTION (AH = 6) THE USER MUST SUPPLY A ROUTINE AND
 90
                                 <1>;
                                            INTERCEPT THE CORRECT ADDRESS IN THE VECTOR TABLE FOR INTERRUPT 4AH.:
 91
                                 <1>;
 92
                                <1>;
                                              USE OFFH FOR ANY "DO NOT CARE" POSITION FOR INTERVAL INTERRUPTS.
                                           INTERRUPTS ARE DISABLED DURING DATA MODIFICATION.
 93
                                <1>;
 94
                                           AH & AL ARE RETURNED MODIFIED AND NOT DEFINED EXCEPT WHERE INDICATED .:
                                 <1>;
 95
                                 <1> ;------
 96
                                 <1>
                                 <1> ; 15/01/2017
                                 <1> ; 14/01/2017
 98
 99
                                 <1>; 07/01/2017
                                 <1>; 02/01/2017
100
101
                                 <1>; 29/05/2016
102
                                 <1>; 29/01/2016
                                <1> ; 17/01/2016 (TRDOS 386 = TRDOS v2.0)
103
104
                                 <1>
105
                                <1> ; 29/05/2016
                                 <1>; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
106
107
                                 <1> int35h: ; Date/Time functions
108
                                <1>
                                <1> TIME_OF_DAY_1:
109
                                                                    ; INTERRUPTS BACK ON
                                        ;sti
110
                                <1>
                                          ; 29/05/2016
111
                                <1>
112 000059DF 80642408FE
                                          and byte [esp+8], 111111110b ; clear carry bit of eflags register
                                <1>
113
                                <1>
                                              ah, (RTC_TBE-RTC_TB)/4 ; CHECK IF COMMAND IN VALID RANGE (0-7)
114 000059E4 80FC08
                                <1>
                                          cmc ; COMPLEMENT CARRY FOR ERROR EXIT ; (*) jc short TIME_9 ; EXIT WITH CARRY = 1 IF NOT VALID
115 000059E7 F5
                                <1>
                                          cmc
116
                                <1>
                                                short _TIME_9 ; 29/05/2016
117 000059E8 721A
                                <1>
                                          jc
118
                                <1>
119 000059EA 1E
                                <1>
                                          push ds
120 000059EB 56
                                          push esi
                                <1>
121 000059EC 66BE1000
                                          mov
                                <1>
                                                si, KDATA
                                                                 ; kernel data segment
122 000059F0 8EDE
                                <1>
                                          mov
                                                 ds, si
123
                                <1>
                                 <1>
                                          ;;15/01/2017
124
125
                                <1>
                                          ; 14/01/2017
                                          ; 02/01/2017
126
                                <1>
127
                                <1>
                                          ;; mov byte [intflg], 35h ; date & time interrupt
                                <1>
128
                                          ;sti
129
                                <1>
130 000059F2 C0E402
                                <1>
                                          shl ah, 2
                                                                   ; convert function to dword offset
                                                                          ; PLACE INTO ADDRESSING REGISTER
131 000059F5 0FB6F4
                                <1>
                                          movzx esi, ah
                                <1>
                                          ;cli
                                                                   ; NO INTERRUPTS DURING TIME FUNCTIONS
133 000059F8 FF96[0A5A0000]
                                <1>
                                                 [esi+RTC_TB]
                                                                    ; VECTOR TO FUNCTION REQUESTED WITH CY=0
                                          call
                                <1>
                                                                   ; RETURN WITH CARRY FLAG SET FOR RESULT
134
                                                                   ; INTERRUPTS BACK ON
135
                                <1>
                                          ;sti
136 000059FE B400
                                <1>
                                          mov
                                                 ah, 0
                                                                   ; CLEAR (AH) TO ZERO
137 00005A00 5E
                                <1>
                                                                   ; RECOVER USERS REGISTER
                                                esi
                                          pop
138 00005A01 1F
                                <1>
                                                                   ; RECOVER USERS SEGMENT SELECTOR
                                                ds
                                         pop
139
                                <1>
                                          ;;15/01/2017
140
                                <1>
                                          ; 02/01/2017
141
                                <1>
                                          ;;mov byte [ss:intflg], 0 ; 07/01/2017
142
                                <1>
143
                                <1>
144
                                <1> ;TIME 9:
145
                                <1>
                                                                    ; RETURN WITH CY= 0 IF NO ERROR
```

```
; (*) 29/05/2016
146
                               <1>
                               <1>
                                        ; (*) retf 4 ; skip eflags on stack
                                    ; (°, 1001 )
jnc short _TIME_10
148 00005A02 7305
                               <1>
149
                               <1> TIME 9:
                                     ; 29/05/2016 -set carry flag on stack-
150
                               <1>
151
                               <1>
                                        ; [esp] = EIP
152
                               <1>
                                       ; [esp+4] = CS
                                      ; [esp+8] = E-FLAGS
153
                               <1>
154 00005A04 804C240801
                               <1>
                                       or byte [esp+8], 1
                                                                 ; set carry bit of eflags register
                                        ; [esp+12] = ESP (user)
155
                               <1>
                                    ; [esp+16] = SS (User)
156
                               <1>
157
                               <1> _TIME_10:
158 00005A09 CF
                               <1>
                                        iretd
159
                               <1>
160
                               <1>
                                       ; (*) 29/05/2016 - 'ref 4' intruction causes to stack fault
161
                               <1>
                                        ; (OUTER-PRIVILEGE-LEVEL)
162
                                       ; INTEL 80386 PROGRAMMER'S REFERENCE MANUAL 1986
                                       ; // RETF instruction:
                               <1>
163
164
                               <1>
                                       ; IF OperandMode=32 THEN
165
                               <1>
                                       ; Load CS:EIP from stack;
166
                               <1>
                                            Set CS RPL to CPL;
167
                               <1>
                                       ;
                                            Increment eSP by 8 plus the immediate offset if it exists;
                               <1>
168
                                        ;
169
                               <1>
                                       ; Load SS:eSP from stack;
                                      ; ELSE (* OperandMode=16 *)
170
                               <1>
171
                               <1>
                                        ; Load CS: IP from stack;
172
                               <1>
                                            Set CS RPL to CPL;
                                            Increment eSP by 4 plus the immediate offset if it exists;
                               <1>
173
174
                               <1>
                                            Load SS:eSP from stack;
                                       ;
175
                               <1>
                                       ; FI;
176
                               <1>
177
                               <1>
                                        ; //
178
                                                                ; ROUTINE VECTOR TABLE (AH) =
                               <1>
179
                               <1> RTC_TB:
                                                                ; 0 = READ CURRENT CLOCK COUNT
                                              RTC 00
180 00005A0A [2A5A0000]
                              <1>
                                        dd
                                              RTC 10
                                                                ; 1 = SET CLOCK COUNT
181 00005A0E [3D5A0000]
                              <1>
                                        dd
182 00005A12 [4B5A0000]
                              <1> dd
<1> dd
<1> dd
<1> dd
<1> dd
                                              RTC 20
                                                                ; 2 = READ THE REAL TIME CLOCK TIME
183 00005A16 [7A5A0000]
                                                                ; 3 = SET REAL TIME CLOCK TIME
                                              RTC_30
184 00005A1A [BC5A0000]
                                              RTC_40
                                                                ; 4 = READ THE REAL TIME CLOCK DATE
                                              RTC_50
                                                                ; 5 = SET REAL TIME CLOCK DATE
185 00005A1E [E95A0000]
186 00005A22 [365B0000]
                                       dd
                                              RTC_60
                                                                ; 6 = SET THE REAL TIME CLOCK ALARM
                              <1>
                                                                ; 7 = RESET ALARM
187 00005A26 [895B0000]
                               <1>
                                              RTC_70
188
                               <1>
                                              equ $
189
                               <1> RTC_TBE
190
                               <1>
                                             ; READ TIME COUNT al, [TIMER_OFL] ; GET THE OVERFLOW FLAG
191
                               <1> RTC_00:
192 00005A2A A0[44590100]
                              <1> mov
193 00005A2F C605[44590100]00
                                        mov byte [TIMER_OFL], 0; AND THEN RESET THE OVERFLOW FLAG
                              <1>
194 00005A36 8B0D[40590100]
                              <1>
                                        mov
                                               ecx, [TIMER_LH] ; GET COUNT OF TIME
195 00005A3C C3
                               <1>
                                        retn
196
                               <1>
                              <1> RTC_10: ; SET TIME COUNT 
<1> mov [TIMER_LH], ecx ; SET TIME COUNT
197
198 00005A3D 890D[40590100]
199 00005A43 C605[44590100]00
                              <1>
                                        mov byte [TIMER_OFL], 0; RESET OVERFLOW FLAG
200 00005A4A C3
                              <1>
                                                                ; RETURN WITH NO CARRY
                                        retn
201
                              <1>
                              <1> RTC 20:
                                                                ; GET RTC TIME
                                       call UPD IPR
203 00005A4B E8EB010000
                              <1>
                                                                ; CHECK FOR UPDATE IN PROCESS
                                              short RTC_29
204 00005A50 7227
                              <1>
                                                                ; EXIT IF ERROR (CY= 1)
                                   jс
                              <1>
206 00005A52 B000
                              <1>
207 00005A54 E8FD010000
                              <1>
                              <1>
208 00005A59 88C6
209 00005A5B B00B
                              <1>
                                                                    ; ADDRESS ALARM REGISTER
                                                                ; READ CURRENT VALUE OF DSE BIT
210 00005A5D E8F4010000
                              <1>
211 00005A62 2401
                              <1>
                                       and
                                             al, 00000001b
                                                                ; MASK FOR VALID DSE BIT
                                    mov dl, al
                              mov al, CMOS_MINUTES ; SET [DL] TO ZERO FOR NO DSE 1

<1> mov al, CMOS_MINUTES ; SET ADDRESS OF MINUTES

<1> call CMOS_READ ; GET MINUTES

<1> mov cl, al ; SAVE

<1> mov al, CMOS_HOURS ; SET ADDRESS OF HOURS

<1> call CMOS READ ; GET HOURS
212 00005A64 88C2
                              <1>
                                                                ; SET [DL] TO ZERO FOR NO DSE BIT
213 00005A66 B002
214 00005A68 E8E9010000
215 00005A6D 88C1
216 00005A6F B004
217 00005A71 E8E0010000
                              <1>
                                       call CMOS_READ ; GET HOURS
                                     mc
clc
218 00005A76 88C5
                              <1>
                                                                ; SAVE
                                       mov
                                             ch, al
219 00005A78 F8
                                                                ; SET CY= 0
                              <1>
220
                              <1> RTC_29:
221 00005A79 C3
                                                                ; RETURN WITH RESULT IN CARRY FLAG
                              <1>
                                       retn
222
                               <1>
                               <1> RTC 30:
                                                                ; SET RTC TIME
223
                                    call UPD IPR
224 00005A7A E8BC010000
                                                                 ; CHECK FOR UPDATE IN PROCESS
                               <1>
225 00005A7F 7305
                                        jnc short RTC_35
                                                                ; GO AROUND IF CLOCK OPERATING
                               <1>
226 00005A81 E817010000
                                     call RTC_STA
                               <1>
                                                                     ; ELSE TRY INITIALIZING CLOCK
                               <1> RTC_35:
                          <1>
                                                                ; GET TIME BYTE - SECONDS
228 00005A86 88F4
                                              ah, dh
                                        mov
229 00005A88 B000
                              <1>
                                              al, CMOS_SECONDS ; ADDRESS SECONDS
230 00005A8A E8E0010000
                              <1>
                                       call CMOS_WRITE ; UPDATE SECONDS
                                                                ; GET TIME BYTE - MINUTES
                              <1>
231 00005A8F 88CC
                                       mov
                                              ah, cl
250
                               <1> RTC 40:
                                                              ; GET RTC DATE
```

```
call UPD_IPR ; CHECK FOR UPDA
jc short RTC_49 ; EXIT IF ERROR (CY= 1)

mov al, CMOS_DAY_MONTH ; ADDRESS DAY OF MONTH
call CMOS_READ ; READ DAY OF MONTH
mov dl, al ; SAVE
                                                                                 <1>
   251 00005ABC E87A010000
                                                                                                                                                                                                                                              ; CHECK FOR UPDATE IN PROCESS
                                                                                                                                                          short RTC_49 ; EXIT IF ERROR (CY= 1)
   252 00005AC1 7225
                                                                                                        <1>
                                                                                                       <1>
cl, al ; SAVE al, CMOS_CENTURY ; ADDRESS CENTURY LOCATION
   268 00005AE8 C3
                                                                                                                                                                                                                         ; RETURN WITH RESULTS IN CARRY FLAG
| Second 
   269
                                                                                                        <1>
                                                                                                                        RTC_50: ; SET RTC DATE

call UPD_IPR ; CHECK FOR UPDATE IN PROCESS

jnc short RTC_55 ; GO AROUND IF NO ERROR

call RTC_STA ; ELSE INITIALIZE CLOCK

RTC_55:
                                                                                                                                                                                                                        ; RETURN CY=0
   297 00005B35 C3
                                                                                                      <1>
                                                                                                                                      retn
| STATE | STAT
                                                                                                       <1> RTC_69:
   331 00005B84 66B80000
                                                                                                                           mov ax, 0 ; CLEAR AX REGISTER
                                                                                                      <1>
                                                                                                                                                                                                                  ; RETURN WITH RESULTS IN CARRY FLAC
   332 00005B88 C3
                                                                                                          <1>
                                                                                                                                        retn
   333
                                                                                                          <1>
                                                                                                        334
   335
                                                                 ; mov ah, al

mov ax, CMOS_REG_B * 257 ; ADDRESS ALARM REGISTER (TO BOTH AH, AL)

call CMOS_READ ; READ ALARM REGISTER

c1> and al, 57h ; TURN OFF ALARM ENABLE

c1> xchg ah, al ; SAVE DATA AND RECOVER ADDRESS

call CMOS_WRITE ; RESTORE NEW VALUE

c1> clc ; SET CY= 0

c1> retn ; RETURN WITH NO CARRY

c1> RETURN WITH NO CARRY
   336
   337 00005B89 66B80B0B
   338 00005B8D E8C4000000
   339 00005B92 2457
   340 00005B94 86E0
   341 00005B96 E8D4000000
   342 00005B9B F8
   343 00005B9C C3
                                                                                                       <1>
   344
                                                                                                                                                                                         ; INITIALIZE REAL TIME CLOCK
   345
                                                                                                        <1> RTC STA:
                                                                                                                           ;mov al, CMOS_REG_A
   346
                                                                                                        <1>
                                                                                                                                                                                                                                              ; ADDRESS REGISTER A AND LOAD DATA MASK
                                                                                                                                      ;mov ah, 26h
                                                                                                      <1>
                                                                                                                        mov
                                                                                 <1><1>
                                                                                                                                                          ax, (26h*100h)+CMOS_REG_A
   348 00005B9D 66B80A26
   349 00005BA1 E8C9000000
                                                                                                      <1>
                                                                                                                                     call CMOS_WRITE ; INITIALIZE STATUS REGISTER A
                                                                                                                                                                                                                    ; SET "SET BIT" FOR CLOCK INITIALIZATION
                                                                                                                                     ;mov al, CMOS_REG_B
                                                                                                     <1>
```

```
mov al, CMOS_REG_D ; ADDRESS REGISTER D call CMOS_READ ; READ REGISTER D TO INITIALIZE
356 00005BB6 B00D
                                            <1>
357 00005BB8 E899000000
                                            <1>
358 00005BBD C3
                                            <1>
                                                         retn
359
                                            <1>
360
                                            <1> ; 17/01/2016 (TRDOS 386 = TRDOS v2.0)
361
                                            <1>
                                             362
                                             <1> ; ALARM INTERRUPT HANDLER (RTC)
363
364
                                                            THIS ROUTINE HANDLES THE PERIODIC AND ALARM INTERRUPTS FROM THE CMOS :
                                             <1>;
                                                             TIMER. INPUT FREQUENCY IS 1.024 KHZ OR APPROXIMATELY 1024 INTERRUPTS :
365
366
                                             <1>;
                                                            EVERY SECOND FOR THE PERIODIC INTERRUPT. FOR THE ALARM FUNCTION,
                                                             THE INTERRUPT WILL OCCUR AT THE DESIGNATED TIME.
367
                                             <1>;
368
                                            <1> ;
369
                                            <1>;
                                                            INTERRUPTS ARE ENABLED WHEN THE EVENT OR ALARM FUNCTION IS ACTIVATED. :
370
                                            <1>;
                                                            FOR THE EVENT INTERRUPT, THE HANDLER WILL DECREMENT THE WAIT COUNTER :
                                                            AND WHEN IT EXPIRES WILL SET THE DESIGNATED LOCATION TO 80H. FOR
371
                                            <1>;
                                                            THE ALARM INTERRUPT. THE USER MUST PROVIDE A ROUTINE TO INTERCEPT
372
                                            <1>;
                                                            THE CORRECT ADDRESS FROM THE VECTOR TABLE INVOKED BY INTERRUPT 4AH
373
                                                            PRIOR TO SETTING THE REAL TIME CLOCK ALARM (INT 1AH, AH= 06H).
374
                                            <1>;
                                            <1> ;-----
375
376
                                            <1>
                                            <1> RTC A INT: ; 07/01/2017
                                                                                      ; ALARM INTERRUPT ; LEAVE INTERRUPTS DISABLED
                                            <1> ; RTC INT:
378
                                                         push ds
379 00005BBE 1E
                                            <1>
                                                                                           ; SAVE REGISTERS
380 00005BBF 50
                                           <1>
                                           <1>
                                                          push eax
381 00005BC0 57
                                                         push edi
                                           <1> RTC_I_1:
                                                                                            ; CHECK FOR SECOND INTERRUPT
                                           <1> mov
                                                                 ax, 256*(CMOS_REG_B+NMI)+CMOS_REG_C+NMI; ALARM AND STATUS
383 00005BC1 66B88C8B
384 00005BC5 E670
                                            <1>
                                                                 CMOS_PORT, al ; WRITE ALARM FLAG MASK ADDRESS
                                      <1> out

<1> nop

<1> jmp

<1> in

<1> test

<1> jz
                                                         out
385 00005BC7 90
                                                                                             ; I/O DELAY
                                                         in al, CMOS_DATA ; READ AND RESET INTERRUPT REQUEST FLAGS test al, 01100000b ; CHECK FOR EITHER INTERRUPT PENDING ; EXIT IF NOT A VALUE RECOMMENDATION ; EXPLANTABLE RECOMMENDATION ; EXIT IF NOT A VALUE RECOMMENDATION ; EXPLANTABLE RECOMMENDATION
386 00005BC8 EB00
387 00005BCA E471
388 00005BCC A860
389 00005BCE 745D
                                                                                                     ; EXIT IF NOT A VALID RTC INTERRUPT
390
                                            <1>
                                      <1> xchc
<1> out
<1> nop
<1> jmp
<1> in
                                                                ah, al
391 00005BD0 86E0
                                                         xchg
                                                                  ah, al ; SAVE FLAGS AND GET ENABLE ADDRE.
CMOS_PORT, al ; WRITE ALARM ENABLE MASK ADDRESS
                                                                                            ; SAVE FLAGS AND GET ENABLE ADDRESS
392 00005BD2 E670
                                                         out
                                                                  short $+2
393 00005BD4 90
                                                                                            ; I/O DELAY
394 00005BD5 EB00
395 00005BD7 E471
                                                                                           ; READ CURRENT ALARM ENABLE MASK
                                                                  al, CMOS_DATA
                                           <1> and
                                                                                         ; ALLOW ONLY SOURCES THAT ARE ENABLED
396 00005BD9 20E0
                                                                  al, ah
                                                                                         ; CHECK FOR PERIODIC INTERRUPT
                                                         test al, 01000000b
397 00005BDB A840
                                           <1>
                                           <1>
                                                                  short RTC_I_5
                                                                                            ; SKIP IF NOT A PERIODIC INTERRUPT
398 00005BDD 743B
                                                         jz
399
                                            <1>
                                            <1> ;----
400
                                                                  DECREMENT WAIT COUNT BY INTERRUPT INTERVAL
401
                                            <1>
402 00005BDF 66BF1000
                                            <1>
                                                                  di, KDATA
                                                                                          ; kernel data segment
                                                         mov
403 00005BE3 8EDF
                                            <1>
                                                       mov
                                                                   ds, di
404
                                            <1>
405 00005BE5 812D[38590100]D003- <1>
                                                                  dword [RTC LH], 976; DECREMENT COUNT BY 1/1024
                                                         sub
405 00005BED 0000
                                           <1>
406 00005BEF 7329
                                            <1>
                                                                  short RTC I 5 ; SKIP TILL 32 BIT WORD LESS THAN ZERO
                                                         jnc
407
                                      <1>
408
                                                                  TURN OFF PERIODIC INTERRUPT ENABLE
409
410 00005BF1 6650
                                                                                             ; SAVE INTERRUPT FLAG MASK
411 00005BF3 66B88B8B
                                                                  ax, 257* (CMOS REG B+NMI) ; INTERRUPT ENABLE REGISTER
412 00005BF7 E670
                                                                 CMOS_PORT, al ; WRITE ADDRESS TO CMOS CLOCK
413 00005BF9 90
                                                                 short $+2
                                                                                            ; I/O DELAY
414 00005BFA EB00
                                                                al, CMOS_DATA ; READ CURRENT ENABLES
al, 0BFh ; TURN OFF PIE
al, ah ; GET CMOS ADDRESS AND SAVE VALUE
CMOS_PORT, al ; ADDRESS REGISTER B
al, ah ; GET NEW INTERRUPT ENABLE MASK
CMOS_DATA, al ; SET MASK IN INTERRUPT ENABLE REGISTER
415 00005BFC E471
416 00005BFE 24BF
417 00005C00 86C4
                            418 00005C02 E670
byte [RTC_WAIT_FLAG], 0 ; SET FUNCTION ACTIVE FLAG OFF
                                                                  edi, [USER_FLAG] ; SET UP (DS:DI) TO POINT TO USER FLAG
423 00005C15 C60780
                                            <1>
                                                                  byte [edi], 80h
                                                                                             ; TURN ON USERS FLAG
424 00005C18 6658
                                                                                            ; GET INTERRUPT SOURCE BACK
                                           <1>
                                                         pop
                                                                  ax
425
                                           <1> RTC_I_5:
                                                                 al, 00100000b
                                                                                            ; TEST FOR ALARM INTERRUPT
426 00005C1A A820
                                            <1>
                                                         test
427 00005C1C 740D
                                                                   short RTC_I_7
                                                                                             ; SKIP USER INTERRUPT CALL IF NOT ALARM
                                           <1>
                                                         jz
428
                                           <1>
                                                      mov
429 00005C1E B00D
                                            <1>
                                                                  al, CMOS REG D
                                                                                                      ; POINT TO DEFAULT READ ONLY REGISTER
                                                    out
sti
                                                                                            ; ENABLE NMI AND CMOS ADDRESS TO DEFAULT
430 00005C20 E670
                                           <1>
                                                                   CMOS_PORT, al
431 00005C22 FB
                                           <1>
                                                                                             ; INTERRUPTS BACK ON NOW
432 00005C23 52
                                            <1>
                                                    push
                                                                 edx
                                                                                             ; TRANSFER TO USER ROUTINE
433 00005C24 E8309E0000
                                            <1>
                                                                  INT4Ah
                                                         call
434 00005C29 5A
                                                         pop
                                            <1>
                                                                  edx
                                                                                             ; BLOCK INTERRUPT FOR RETRY
435 00005C2A FA
                                            <1>
                                                         cli
                                                                                             ; RESTART ROUTINE TO HANDLE DELAYED
                                            <1> RTC_I_7:
                                                                                             ; ENTRY AND SECOND EVENT BEFORE DONE
437 00005C2B EB94
                                            <1>
                                                                  short RTC_I_1
                                                          jmp
                                            <1>
                                            <1> RTC_I_9:
439
                                                                                            ; EXIT - NO PENDING INTERRUPTS
                                                                  al, CMOS_REG D
440 00005C2D B00D
                                            <1>
                                                         mov
                                                                                                     ; POINT TO DEFAULT READ ONLY REGISTER
                                                                  CMOS PORT, al
                                                                                             ; ENABLE NMI AND CMOS ADDRESS TO DEFAULT
441 00005C2F E670
                                           <1>
                                                         out
442 00005C31 B020
                                           <1>
                                                                                              ; END OF INTERRUPT MASK TO 8259 - 2
                                                        mov
                                                                  al, EOI
443 00005C33 E6A0
                                           <1>
                                                                  INTB00, al
                                                                                             ; TO 8259 - 2
                                                         out
                                                                  INTA00,
444 00005C35 E620
                                           <1>
                                                                                             ; TO 8259 - 1
                                                                                   al
                                                        out
445 00005C37 5F
                                           <1>
                                                      pop
                                                                  edi
                                                                                            ; RESTORE REGISTERS
                                                       pop
446 00005C38 58
                                            <1>
                                                                  eax
447 00005C39 1F
                                            <1>
                                                         pop
                                                                  ds
                                                                                             ; END OF INTERRUPT
448 00005C3A CF
                                            <1>
                                                         iretd
449
                                            <1>
450
                                            <1>
                                            <1>
                                                         ; 29/05/2016 - TRDOS 386 (TRDOS v2.0)
                                                         ; 22/08/2014 (Retro UNIX 386 v1)
452
                                            <1>
453
                                            <1>
                                                          ; IBM PC/AT BIOS source code ---- 10/06/85 (bios2.asm)
                                                                                          ; WAIT TILL UPDATE NOT IN PROGRESS
454
                                            <1> UPD IPR:
455 00005C3B 51
                                            <1>
                                                         push ecx
                                            <1>
457
                                            <1>
                                                         ; 29/05/2016
                                                          mov ecx, ((1984+244)*4)/2; AWARD BIOS 1999, ATIME.ASM
458 00005C3C B968110000
                                            <1>
                                            <1>
                                                                                            ; 'WAITCPU CK UD STAT'
459
```

```
; (244Us + 1984Us)
460
                                <1>
                                                                  ; (assume each read takes
                                <1>
462
                                                                  ; 2 microseconds).
463
                                <1>
                                         ;mov ecx, 65535
                                <1>
                                                                  ; SET TIMEOUT LOOP COUNT (= 800)
464
                                               ;mov cx, 800
                                <1> UPD 10:
465
                              ; ADDRESS STATUS REGISTER A

(1) cli ; NO TIMER INTERRUPTS DURING UPDATES

(1) call CMOS_READ ; READ UPDATE IN PROCESS FLAG

(1) test al, 80h ; IF UIP BIT IS ON ( CANNOT READ

(1) jz short UPD_90 ; EXIT WITH CY= 0 IF CAN READ CLOCK N

(1) sti ; ALLOW INTERRUPTS WHILE WAITING

(1) loop UPD_10 ; LOOP TILL READY OR TIMEOUT

(1) xor eax, eax ; CLEAR RESULTS IF ERROR
                               <1> mov al, CMOS_REG_A
466 00005C41 B00A
                                                                        ; ADDRESS STATUS REGISTER A
467 00005C43 FA
468 00005C44 E80D000000
469 00005C49 A880
                                                                  ; IF UIP BIT IS ON ( CANNOT READ TIME )
                                               short UPD_90 ; EXIT WITH CY= 0 IF CAN READ CLOCK NOW : ALLOW INTERRUPTS WHILE WAITING
470 00005C4B 7406
 471 00005C4D FB
472 00005C4E E2F1
                                              eax, eax
473 00005C50 31C0
474
                               <1>
                                               ; xor ax, ax
                               <1> stc
475 00005C52 F9
                                                                  ; SET CARRY FOR ERROR
                                <1> UPD 90:
476
                                               ecx
477 00005C53 59
                                                                  ; RESTORE CALLERS REGISTER
                                <1> pop
 478 00005C54 FA
                                                                  ; INTERRUPTS OFF DURING SET
                                <1>
                                         cli
479 00005C55 C3
                                <1>
                                                                  ; RETURN WITH CY FLAG SET
                                         retn
480
                                <1>
481
                                <1>
                                    ; 29/05/2016 - TRDOS 386 (TRDOS v2.0)
; 22/08/2014 (Retro UNIX 386 v1)
; IBM PC/AT BIOS source code ---- 10/06/85 (test4.asm)
482
                                <1>
483
                                <1>
484
                                <1>
485
                                <1>
                                <1> ;--- CMOS READ -------
486
                                <1>; READ BYTE FROM CMOS_SYSTEM CLOCK CONFIGURATION TABLE
487
488
                                489
490
                                               BITS 6-0 = ADDRESS OF TABLE LOCATION TO READ :
491
                                <1>;
                                <1>;
492
493
                                <1>; OUTPUT: (AL) VALUE AT LOCATION (AL) MOVED INTO (AL). IF BIT 7 OF (AL) WAS:
                                494
                                               NORMAL INTERRUPTS ARE DISABLED TO PROTECT CMOS DATA INTEGRITY. :
495
                                <1>;
                                               THE CMOS ADDRESS REGISTER IS POINTED TO A DEFAULT VALUE AND :
                                               THE INTERRUPT FLAG RESTORED TO THE ENTRY STATE ON RETURN.
497
                                <1>;
498
                                               ONLY THE (AL) REGISTER AND THE NMI STATE IS CHANGED.
                                <1>;
                                <1> ;------
499
500
                                <1>
                                <1> CMOS READ:
                               ; SAVE INTERRUPT ENABLE STATUS AND FLAGS
502 00005C56 9C
 503 00005C57 D0C0
 504 00005C59 F9
505 00005C5A D0D8
                                                                  ; HIGH BIT ON TO DISABLE NMI - OLD IN CY
506 00005C5C FA
507 00005C5D E670
508
509
                                    ; NEWIODELAY; AWARD BIOS 1999, ATIME.ASM; in al, CMOS_DATA; READ THE REQUESTED CMOS LOCATION push ax; SAVE (AH) REGISTER VALUE AND CMOS BYTE; 15/03/2015; IBM PC/XT Model 286 BIOS source code; ----- 10/06/85 (test/d acm)
                               <1>
510 00005C5F E6EB
511
                                <1>
512 00005C61 E471
                                <1>
513 00005C63 6650
                                <1>
                                <1>
514
                                               ; ---- 10/06/85 (test4.asm)
515
                                <1>
                               516 00005C65 B01E
                                               al, CMOS SHUT DOWN*2 ; GET ADDRESS OF DEFAULT LOCATION
                                         ;mov al, CMOS_REG_D*2 ; GET ADDRESS OF DEFAULT LOCATION
517
                                              al, 1 ; PUT ORIGINAL NMI MASK BIT INTO ADDRESS CMOS_PORT, al ; SET DEFAULT TO READ ONLY REGISTER ax ; RESTORE (AH) AND (AL), CMOS BYTE
518 00005C67 D0D8
                                <1>
                                         rcr
519 00005C69 E670
                               <1>
                                         out
520 00005C6B 6658
                                <1>
                                         pop
 521 00005C6D 9D
                                        popf
                                <1>
                                                                  ; RETURN WITH FLAGS RESTORED
522 00005C6E C3
                                <1>
                                         retn
523
                                <1>
524
                                <1>; 17/01/2016 (TRDOS 386 = TRDOS v2.0)
525
                                <1>
                                526
                                <1>; WRITE BYTE TO CMOS SYSTEM CLOCK CONFIGURATION TABLE
527
528
                                529
530
                                               BITS 6-0 = ADDRESS OF TABLE LOCATION TO WRITE :
531
                                <1>;
                                        (AH) = NEW VALUE TO BE PLACED IN THE ADDRESSED TABLE LOCATION
532
                                <1>;
533
534
                                <1>; OUTPUT: VALUE IN (AH) PLACED IN LOCATION (AL) WITH NMI LEFT DISABLED
                                <1> ;
                                               IF BIT 7 OF (AL) IS ON, DURING THE CMOS UPDATE BOTH NMI AND
535
536
                                <1>;
                                               NORMAL INTERRUPTS ARE DISABLED TO PROTECT CMOS DATA INTEGRITY. :
                                               THE CMOS ADDRESS REGISTER IS POINTED TO A DEFAULT VALUE AND :
537
                                <1>;
538
                                               THE INTERRUPT FLAG RESTORED TO THE ENTRY STATE ON RETURN.
                                              ONLY THE CMOS LOCATION AND THE NMI STATE IS CHANGED.
539
                                <1> ;------
540
                                <1>
                                                                  ; WRITE (AH) TO LOCATION (AL)
                                <1> CMOS_WRITE:
543 00005C6F 9C
                                         pushf
                                                                 ; SAVE INTERRUPT ENABLE STATUS AND FLAGS
                                <1>
                                                                ; SAVE WORK REGISTER VALUES
544 00005C70 6650
                                <1>
                                         push ax
545 00005C72 D0C0
                               <1>
                                         rol
                                               al, 1
                                                                 ; MOVE NMI BIT TO LOW POSITION
                                                                 ; FORCE NMI BIT ON IN CARRY FLAG
546 00005C74 F9
                               <1>
                                        stc
                                                                ; HIGH BIT ON TO DISABLE NMI - OLD IN CY
547 00005C75 D0D8
                               <1>
                                               al, 1
                                        rcr
                                                                 ; DISABLE INTERRUPTS
548 00005C77 FA
                               <1>
                                         cli
                                               CMOS PORT, al
                                                                 ; ADDRESS LOCATION AND DISABLE NMI
549 00005C78 E670
                               <1>
                                         out
                                                              ; GET THE DATA BYTE TO WRITE
550 00005C7A 88E0
                               <1>
                                         mov
                                               al, ah
                                               CMOS DATA, al
 551 00005C7C E671
                               <1>
                                         out
                                                                  ; PLACE IN REQUESTED CMOS LOCATION
                                                al, CMOS_SHUT_DOWN*2 ; GET ADDRESS OF DEFAULT LOCATION
552 00005C7E B01E
                               <1>
                                         mov
                               <1>
                                               al, CMOS_REG_D*2 ; GET ADDRESS OF DEFAULT LOCATION
553
                                         ;mov
                                                                ; PUT ORIGINAL NMI MASK BIT INTO ADDRESS
554 00005C80 D0D8
                               <1>
                                         rcr
                                               al, 1
                                                                 ; SET DEFAULT TO READ ONLY REGISTER
555 00005C82 E670
                               <1>
                                         out
                                                CMOS_PORT, al
                                                                ; I/O DELAY
556 00005C84 90
                               <1>
                                         nop
                                                               ; OPEN STANDBY LATCH
557 00005C85 E471
                               <1>
                                                al, CMOS DATA
                                         in
558 00005C87 6658
                               <1>
                                         pop
                                                                  ; RESTORE WORK REGISTERS
559 00005C89 9D
                               <1>
                                         popf
560 00005C8A C3
                               <1>
                                         retn
561
                                <1>
                                <1>; /// End Of TIMER FUNCTIONS ///
562
2161
2162 00005C8B 90<rept>
```

```
2164
                                    gdt: ; Global Descriptor Table
                                         ; (30/07/2015, conforming cs)
2165
2166
                                         ; (26/03/2015)
2167
                                         ; (24/03/2015, tss)
                                         ; (19/03/2015)
2168
2169
                                         ; (29/12/2013)
2170
2171 00005C90 0000000000000000
                                         dw 0, 0, 0, 0
                                                        ; NULL descriptor
                                         ; 18/08/2014
2172
                                                     ; 8h kernel code segment, base = 00000000h
2173
                                         ;dw 0ffffh, 0, 9E00h, 00Cfh ; KCODE ; 30/12/2016
2174
                                         dw Offffh, 0, 9A00h, 00Cfh; KCODE
2175 00005C98 FFFF0000009ACF00
                                                     ; 10h kernel data segment, base = 00000000h
2177 00005CA0 FFFF00000092CF00
                                         dw OFFFFh, 0, 9200h, 00CFh; KDATA
2178
                                                      ; 1Bh user code segment, base address = 400000h ; CORE
                                         ;dw OFBFFh, 0, OFE40h, OOCFh ; UCODE ; 30/12/2016 dw OFBFFh, 0, OFA40h, OOCFh ; UCODE
2180 00005CA8 FFFB000040FACF00
2181
                                                     ; 23h user data segment, base address = 400000h ; CORE
2182 00005CB0 FFFB000040F2CF00
                                         dw OFBFFh, 0, OF240h, OOCFh ; UDATA
2183
                                                     ; Task State Segment
2184 00005CB8 6700
                                         dw \ 0067h; Limit = 103; (104-1, tss size = 104 byte,
2185
                                                            ; no IO permission in ring 3)
                                    gdt_tss0:
                                        dw 0 ; TSS base address, bits 0-15
2187 00005CBA 0000
2188
                                    gdt_tss1:
2189 00005CBC 00
                                         db 0 ; TSS base address, bits 16-23
                                                   ; 49h
2190
2191 00005CBD E9
                                         db 11101001b; E9h => P=1/DPL=11/0/1/0/B/1 --> B = Task is busy (1)
2192 00005CBE 00
                                         db 0 ; G/0/0/AVL/LIMIT=0000 ; (Limit bits 16-19 = 0000) (G=0, 1 byte)
2193
                                    gdt_tss2:
2194 00005CBF 00
                                         db 0 ; TSS base address, bits 24-31
2195
2196
                                         ;; 9Eh = 1001 1110b (GDT byte 5) P=1/DPL=00/1/TYPE=1110,
2197
                                                                  ;; Type= 1 (code)/C=1/R=1/A=0
2198
                                                ; P= Present, DPL=0=ring 0, 1= user (0= system)
2199
                                                ; 1= Code C= Conforming, R= Readable, A = Accessed
2200
2201
2202
                                         ;; 9Ah = 1001 1010b (GDT byte 5) P=1/DPL=00/1/TYPE=1010,
2203
                                                                  ;; Type= 1 (code)/C=0/R=1/A=0
                                               ; P= Present, DPL=0=ring 0, 1= user (0= system)
2204
                                                ; 1= Code C= non-Conforming, R= Readable, A = Accessed
2205
2206
2207
                                         ;; 92h = 1001 0010b (GDT byte 5) P=1/DPL=00/1/TYPE=1010,
2208
                                                                ;; Type= 0 (data)/E=0/W=1/A=0
                                               ; P= Present, DPL=0=ring 0, 1= user (0= system)
2209
                                               ; 0= Data E= Expansion direction (1= down, 0= up)
2210
                                                ; W= Writeable, A= Accessed
2211
2212
2213
                                         ;; FEh = 1111 1110b (GDT byte 5) P=1/DPL=11/1/TYPE=1110,
2214
                                                                 ;; Type= 1 (code)/C=1/R=1/A=0
                                                ; P= Present, DPL=3=ring 3, 1= user (0= system)
2215
2216
                                               ; 1= Code C= Conforming, R= Readable, A = Accessed
2217
2218
                                         ;; FAh = 1111 1010b (GDT byte 5) P=1/DPL=11/1/TYPE=1010,
2219
                                                                  ;; Type= 1 (code)/C=0/R=1/A=0
                                                ; P= Present, DPL=3=ring 3, 1= user (0= system)
2220
2221
                                                ; 1= Code C= non-Conforming, R= Readable, A = Accessed
2222
2223
                                         ;; F2h = 1111 0010b (GDT byte 5) P=1/DPL=11/1/TYPE=0010,
2224
                                                                  ;; Type= 0 (data)/E=0/W=1/A=0
                                                ; P= Present, DPL=3=ring 3, 1= user (0= system)
2225
2226
                                                ; 0= Data E= Expansion direction (1= down, 0= up)
2227
2228
                                         ;; CFh = 1100 1111b (GDT byte 6) G=1/B=1/0/AVL=0, Limit=1111b (3)
2229
                                                ;; Limit = FFFFFh (=> FFFFFh+1= 100000h) // bits 0-15, 48-51 //
2230
2231
                                                      = 100000h * 1000h (G=1) = 4GB
2232
                                                ;; Limit = FFBFFh (=> FFBFFh+1= FFC00h) // bits 0-15, 48-51 //
2233
                                                      = FFC00h * 1000h (G=1) = 4GB - 4MB
2234
                                               ; G= Granularity (1= 4KB), B= Big (32 bit),
2235
                                                ; AVL= Available to programmers
2236
2237
                                    adtd:
2238 00005CC0 2F00
                                           dw gdt_end - gdt - 1    ; Limit (size)
2239 00005CC2 [905C0000]
                                           dd gdt
                                                                  ; Address of the GDT
2240
                                         ; 20/08/2014
2241
2242
2243 00005CC6 7F02
                                           dw idt_end - idt - 1   ; Limit (size)
                                                     ; Address of the IDT
                                           dd idt
2244 00005CC8 [D8550100]
2245
                                    ; 20/02/2017
2246
2247
                                    ;;; 11/03/2015
                                    %include 'diskdata.s'
                                                            ; DISK (BIOS) DATA (initialized)
2248
  1
  2
                                <1>; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.0 - diskdata.s
  3
                                <1>; ------
                                <1> ; Last Update: 24/01/2016
   4
  5
                                <1> ; -----
  6
                                <1> ; Beginning: 24/01/2016
  7
                                8
                                <1>; Assembler: NASM version 2.11 (trdos386.s)
                                <1> ; ------
  9
                                <1> ; Turkish Rational DOS
 10
 11
                                <1>; Operating System Project v2.0 by ERDOGAN TAN (Beginning: 04/01/2016)
 12
                                <1>;
 13
                                <1>; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
 14
                                <1>; diskdata.inc (11/03/2015)
                                <1> ;
 1.5
 16
                                <1>; Derived from 'IBM PC-XT-286' BIOS source code (1986)
                                17
 18
                                <1>
 19
                                <1>; Retro UNIX 386 v1 Kernel - DISKDATA.INC
```

```
<1>; Last Modification: 11/03/2015
 20
                            <1> ; (Initialized Disk Parameters Data section for 'DISKIO.INC')
 21
                            <1>;
 22
 23
                            <1>
 24
                            <1> ;------
                            <1>; 80286 INTERRUPT LOCATIONS : <1>; REFERENCED BY POST & BIOS :
 2.5
 26
 27
                            <1> ;-----
 28
                            <1>
 29 00005CCC [2F5D0000]
                           <1> DISK_POINTER: dd MD_TBL6
                                                                    ; Pointer to Diskette Parameter Table
 30
                            <1>
 31
                            <1>; IBM PC-XT Model 286 source code ORGS.ASM (06/10/85) - 14/12/2014
 32
                            33
                            <1> ; DISK BASE
                            <1>; THIS IS THE SET OF PARAMETERS REQUIRED FOR
 34
 35
                            <1>;
                                   DISKETTE OPERATION. THEY ARE POINTED AT BY THE
                                  DATA VARIABLE @DISK POINTER. TO MODIFY THE PARAMETERS,
 36
                            <1>; BUILD ANOTHER PARAMETER BLOCK AND POINT AT IT
 37
 38
                            <1> ;-----
 39
                            <1>
 40
                            <1> ;DISK_BASE:
                                   DB
                                        11011111B ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
2 ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
                            <1> ;
 41
                            <1>;
 42
                                    DB
                                         MOTOR_WAIT ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
                           <1> ;
 43
                                 DB MOTOR_WAIT ; WAIT TIME AFTER OPERATION TILL IDB 2 ; 512 BYTES/SECTOR ; DB 15 ; EOT (LAST SECTOR ON TRACK) db 18 ; (EOT for 1.44MB diskette) DB 01BH ; GAP LENGTH DB 0FFH ; DTL ; DB 054H ; GAP LENGTH FOR FORMAT db 06ch ; (for 1.44MB dsikette) DB 0F6H ; FILL BYTE FOR FORMAT DB 15 ; HEAD SETTLE TIME (MILLISECONDS) DB 8 ; MOTOR START TIME (1/8 SECONDS)
                            <1>;
 44
 45
                            <1>;
 46
                            <1>;
                           <1>;
 47
                           <1>;
 48
 49
                            <1>;
 50
                            <1>;
 51
                            <1>;
 52
                            <1>;
 53
                            <1>;
 54
                            <1>
 55
                            <1> ;-----
                            <1>; ROM BIOS DATA AREAS :
 56
 57
                            <1> ;------
 58
                            <1>
                            <1> ; DATA
                                         SEGMENT AT 40H
 59
                                                              ; ADDRESS= 0040:0000
 60
                            <1>
                            <1> ;@EQUIP_FLAG DW ? ; INSTALLED HARDWARE FLAGS
 61
 62
                            <1>
                            <1> ;-----
 63
                            <1>; DISKETTE DATA AREAS :
 64
 65
                            <1> ;-----
 66
                           67
 68
 69
 70
 71
                                                        ; BIT 7 = CURRENT OPERATION IS A WRITE
 72
                            <1>;
                            <1>; @MOTOR COUNT DB ?
 73
                                                          ; TIME OUT COUNTER FOR MOTOR(S) TURN OFF
 74
                            <1> ;@DSKETTE_STATUS DB? ; RETURN CODE STATUS BYTE
                                                         ; CMD_BLOCK IN STACK FOR DISK OPERATION
 75
                            <1>;
                            <1>;@NEC_STATUS DB 7 DUP(?)
 76
                                                         ; STATUS BYTES FROM DISKETTE OPERATION
 77
                            <1>
 78
                            <1> ;-----
                            <1>; POST AND BIOS WORK DATA AREA :
 79
 80
                            <1> ;-----
 81
                            <1>
                            <1> ;@INTR_FLAG DB ? ; FLAG INDICATING AN INTERRUPT HAPPENED
 82
 83
                            <1>
                            <1> ;------
 84
                            <1> ; TIMER DATA AREA
 85
                            <1> ;-----
 86
 87
                            <1>
 88
                            <1>; 17/12/2014 (IRQ 0 - INT 08H)
                           89
 90
 91
 92
                            <1>
                            <1> ;-----
 93
                            <1>; ADDITIONAL MEDIA DATA
 94
                            <1> ;------
 95
 96
                                                     ; LAST DISKETTE DATA RATE SELECTED ; DRIVE 0 MEDIA STATE ; DRIVE 1 MEDIA STATE
                                            ?
                            <1> ;@LASTRATE DB
 97
 98
                            <1> ;@DSK_STATE DB
                                               ?
                            <1>; DB ?
 99
                                            ?
                                                        ; DRIVE O OPERATION START STATE
100
                            <1>;
                                         DB
                                                       ; DRIVE 1 OPERATION START STATE
101
                            <1>;
                                         DB
                                        DB
                                                     ; DRIVE 0 PRESENT CYLINDER
; DRIVE 1 PRESENT CYLINDER
                            <1> ;@DSK TRK
102
103
                            <1> ;
104
                            <1>
                                         ENDS
                                                          ; END OF BIOS DATA SEGMENT
105
                            <1> ; DATA
106
107
                            <1> :-----
                            <1> ; DRIVE TYPE TABLE
108
                            <1> ;-----
109
                            <1>
                                        ; 16/02/2015 (unix386.s, 32 bit modifications)
110
111
                            <1> DR TYPE:
112 00005CD0 01
                                         DB 01
                                                         ;DRIVE TYPE, MEDIA TABLE
                           <1>
                                         ;DW MD_TBL1
113
                           <1>
114 00005CD1 [EE5C0000]
                           <1>
                                         dd MD_TBL1
115 00005CD5 82
                           <1>
                                         DB
                                              02+BIT7ON
                          117 00005CD6 [FB5C0000]
118 00005CDA 02
119
                                        dd MD_TBL3
DB 03
120 00005CDB [085D0000]
                           <1>
                                         DB 03; DW MD_TBL4
121 00005CDF 03
                           <1>
                           <1>
123 00005CE0 [155D0000]
                                         dd MD TBL4
                           <1>
124 00005CE4 84
                           <1>
                                         DB 04+BIT7ON
```

```
; DW MD_TBL5

<1> dd MD_TBL5

<1> DB 04

<1> ; DW MD_TBL6

<1> dd MD_TBL6

<1> dd MD_TBL6

<1> DR_TYPE_E equ $ ; END OI

<1> ; DR_CNT EQU (DR_TYPE_E-DR_TYPE)/3

<1> DR_CNT equ (DR_TYPE_E-DR_TYPE)/5

<1> :-----
                                                                                                                      ;DW MD_TBL5
125
126 00005CE5 [225D0000]
127 00005CE9 04
128
129 00005CEA [2F5D0000]
130
                                                                                                                                                                               ; END OF TABLE
131
132
133
                                                                           <1> ;-----
                                                                            <1>; MEDIA/DRIVE PARAMETER TABLES
134
                                                                            <1> ;------
135
136
137
                                                                            <1> ; 360 KB MEDIA IN 360 KB DRIVE
                                                                           <1> ;------
138
139
                                                                           <1> MD_TBL1:
                                                                                                 DB 11011111B ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
DB 2 ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
DB MOTOR_WAIT ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
                                                                           <1>
<1>
140 00005CEE DF
                                                                                                DB
141 00005CEF 02
                                                                        <1>
142 00005CF0 25
143 00005CF1 02
144 00005CF2 09
145 00005CF3 2A
146 00005CF4 FF
147 00005CF5 50
148 00005CF6 F6
149 00005CF7 OF
150 00005CF8 08
151 00005CF9 27
152 00005CFA 80
                                                                           153
154
155
                                                                           <1> ;------
                                                                       <1> MD_TBL2:
156
                                                                                                               11011111B ; SRT=D, HD UNLOAD=OF - 1ST SPECIFY BYTE
157 00005CFB DF
                                                                                                 DB 2 ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
DB MOTOR_WAIT ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
158 00005CFC 02
159 00005CFD 25
160 00005CFE 02
161 00005CFF 09
162 00005D00 2A
163 00005D01 FF
164 00005D02 50
165 00005D03 F6
166 00005D04 OF
167 00005D05 08
168 00005D06 27
169 00005D07 40
170
                                                                           <1>; 1.2 MB MEDIA IN 1.2 MB DRIVE
171
                                                                           <1> ;-----
172
                                                                         173
                                                                           <1> MD_TBL3:
                                                                          OB 11011111B ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE

1> DB 2 ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE

1> DB MOTOR_WAIT ; WAIT TIME AFTER OPERATION TILL MOTOR OFF

1> DB 2 ; 512 BYTES/SECTOR

1> DB 15 ; EOT (LAST SECTOR ON TRACK)

1> DB 01BH ; GAP LENGTH

1> DB 054H ; GAP LENGTH FOR FORMAT

1> DB 056H ; FILL BYTE FOR FORMAT

1> DB 15 ; HEAD SETTLE TIME (MILLISECONDS)

1> DB 8 ; MOTOR START TIME (1/8 SECONDS)

1> DB 79 ; MAX. TRACK NUMBER

1> ; 720 KB MEDIA IN 720 KB DRIVE :
174 00005D08 DF
175 00005D09 02
176 00005D0A 25
177 00005D0B 02
178 00005D0C OF
179 00005D0D 1B
180 00005D0E FF
181 00005D0F 54
182 00005D10 F6
183 00005D11 OF
184 00005D12 08
185 00005D13 4F
186 00005D14 00
187
                                                                           <1>; 720 KB MEDIA IN 720 KB DRIVE
188
                                                                           <1> ;-----
189
190
                                                                            <1> MD_TBL4:
191 00005D15 DF
                                                                                                               11011111B ; SRT=D, HD UNLOAD=OF - 1ST SPECIFY BYTE
                                                                           <1> DB
                                                                                                               2 ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
MOTOR_WAIT ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
192 00005D16 02
                                                                           <1>
                                                                                                 DB
                                                                          <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1> DB <1 DB
193 00005D17 25
                                                                                                          MOTOR_WAIT ; WAIT TIME AFTER OPERATION TILL I

2 ; 512 BYTES/SECTOR

09 ; EOT (LAST SECTOR ON TRACK)

02AH ; GAP LENGTH

0FFH ; DTL

050H ; GAP LENGTH FOR FORMAT

0F6H ; FILL BYTE FOR FORMAT

15 ; HEAD SETTLE TIME (MILLISECONDS)

8 ; MOTOR START TIME (1/8 SECONDS)

79 ; MAX. TRACK NUMBER

RATE_250 ; DATA TRANSFER RATE
194 00005D18 02
                                                                          195 00005D19 09
196 00005D1A 2A
197 00005D1B FF
198 00005D1C 50
199 00005D1D F6
200 00005D1E 0F
201 00005D1F 08
202 00005D20 4F
203 00005D21 80
                                                                           <1>
                                                                                              DB
                                                                           <1> ;------
204
                                                                           <1>; 720 KB MEDIA IN 1.44 MB DRIVE
205
                                                                            <1> ;----
206
                                                                           <1> MD_TBL5:
208 00005D22 DF
                                                                           <1>
                                                                                                                11011111B
                                                                                                                                          ; SRT=D, HD UNLOAD=OF - 1ST SPECIFY BYTE
                                                                       2 ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE MOTOR_WAIT ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
209 00005D23 02
                                                                           <1>
                                                                                                  DB
210 00005D24 25
                                                                                                              y wall lime after operation of the contract of
211 00005D25 02
212 00005D26 09
213 00005D27 2A
214 00005D28 FF
215 00005D29 50
                                                                                                                                         ; FILL BYTE FOR FORMAT ; HEAD SETTLE TIME (MILLISECONDS)
216 00005D2A F6
217 00005D2B OF
218 00005D2C 08
                                                                                                                                          ; MOTOR START TIME (1/8 SECONDS)
                                                                          219 00005D2D 4F
220 00005D2E 80
221
                                                                           222
223
224
                                                                           <1> MD_TBL6:
225 00005D2F AF
                                                                                                               10101111B ; SRT=A, HD UNLOAD=OF - 1ST SPECIFY BYTE
                                                                         <1> DB
                                                                                                               2 ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE MOTOR_WAIT ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
                                                                       <1>
<1>
<1>
<1>
226 00005D30 02
                                                                                                 DB
                                                                                                 DB
227 00005D31 25
                                                                                                 DB 2 ; 512 BYTES/SECTOR
DB 18 ; EOT (LAST SECTOR ON TRACK)
228 00005D32 02
229 00005D33 12
                                                                                                DB
```

```
230 00005D34 1B
231 00005D35 FF
232 00005D36 6C
233 00005D37 F6
                                                            ; HEAD SETTLE TIME (MILLISECONDS)
234 00005D38 OF
235 00005D39 08
236 00005D3A 4F
                                                 RATE_500
                                                            ; DATA TRANSFER RATE
237 00005D3B 00
                                 <1>
                                         DB
238
                                 <1>
239
                                 <1>
240
                                 <1> ; << diskette.inc >>
                                  241
242
                                 <1> ;
                                 <1> ;-----
243
                                  <1>; ROM BIOS DATA AREAS :
244
245
                                 <1> ;-----
246
                                               SEGMENT AT 40H
                                 <1> ; DATA
                                                                          ; ADDRESS= 0040:0000
247
248
                                  <1>
249
                                 <1>;-
                                 <1>; FIXED DISK DATA AREAS :
250
251
                                 <1>
252
                                 253
254
255
256
257
                                 <1>
258
                                 <1> ;-
                                 <1>; ADDITIONAL MEDIA DATA :
259
260
                                 <1> ;-----
                                <1>
<1> ;@LASTRATE DB ? ; LAST DISKETTE DATA RATE SELECTED
<1> ;HF_STATUS DB 0 ; STATUS REGISTER
<1> ;HF_ERROR DB 0 ; ERROR REGISTER
<1> ;HF_INT_FLAG DB 0 ; FIXED DISK INTERRUPT FLAG
<1> ;HF_CNTRL DB 0 ; COMBO FIXED DISK/DISKETTE CARD BIT 0=1
<1> ;@DSK_STATE DB ? ; DRIVE 0 MEDIA STATE
<1> ; DB ? ; DRIVE 1 MEDIA STATE
<1> ; DB ? ; DRIVE 0 OPERATION START STATE
<1> ; DRIVE 1 OPERATION START STATE
<1> ;@DSK_TRK DB ? ; DRIVE 1 OPERATION START STATE
<1> ;@DSK_TRK DB ? ; DRIVE 0 PRESENT CYLINDER
<1> ; DRIVE 1 PRESENT CYLINDER
<1> ; DRIVE 1 PRESENT CYLINDER
261
                                  <1>
262
263
264
265
266
267
268
269
270
271
272
273
                                 <1>
274
                                 <1> ; DATA ENDS
                                                                     ; END OF BIOS DATA SEGMENT
275
                                 <1>;
                                 277
                                 <1>
                                 <1> ERR TBL:
278
                                279 00005D3C E0
                                                 BAD ADDR MARK, BAD SEEK, BAD CMD, UNDEF ERR
 280 00005D3D 024001BB
                                 <1>
 281 00005D41 04BB100A
                                 <1>
                                                 RECORD_NOT_FND, UNDEF_ERR, BAD_ECC, BAD_SECTOR
                                 <1>
282
283
                                 <1>; 17/12/2014 (mov ax, [cfd])
 284
                                 <1> ; 11/12/2014
                                 <1> cfd: db 0
285 00005D45 00
                                                                ; current floppy drive (for GET_PARM)
; instead of 'DISK_POINTER'
                                 <1> ; 17/12/2014
286
                                 <1> pfd: db 1
287 00005D46 01
                                                                   ; previous floppy drive (for GET_PARM)
288
                                 <1>
                                                                     ; (initial value of 'pfd
                                                                     ; must be different then 'cfd' value
289
                                 <1>
290
                                 <1>
                                                                     ; to force updating/initializing
291
                                 <1>
                                                                     ; current drive parameters)
292 00005D47 90
                                 <1> align 2
293
                                 <1>
                                 <1> HF_PORT: dw 1F0h ; Default = 1F0h
 294 00005D48 F001
295
                                 <1>
                                                             ; (170h)
                                 <1> HF_REG_PORT: dw 3F6h ; HF_PORT + 206h
296 00005D4A F603
297
                                 <1>
298
                                 <1> ; 05/01/2015
                                 <1> hf_m_s: db 0 ; (0 = Master, 1 = Slave)
299 00005D4C 00
300
                                 <1>
                                  301
2249
                                     Align 2
2250 00005D4D 90
2251
                                     ; 04/11/2014 (Retro UNIX 386 v1)
2252
2253 00005D4E 0000
                                     mem_1m_1k: dw 0 ; Number of contiguous KB between
                                     ; 1 and 16 MB, max. 3C00h = 15 MB. mem_16m_64k: dw 0 ; Number of contiguous 64 KB blocks
2254
2255 00005D50 0000
2256
                                                   ; between 16 MB and 4 GB.
2257
                                      ; 12/11/2014 (Retro UNIX 386 v1)
2258
2259 00005D52 00
                                     boot drv:
                                                  db 0 ; boot drive number (physical)
2260
                                      ; 24/11/2014
2261 00005D53 00
                                      drv: db 0
2262 00005D54 00
                                                 db 0 ; last hdd
                                      last drv:
2263 00005D55 00
                                                  db 0 ; number of hard disk drives
                                     hdc:
2264
                                                      ; (present/detected)
2265
                                     ; 24/11/2014 (Retro UNIX 386 v1)
2266
2267
                                      ; Physical drive type & flags
2268 00005D56 00
                                                  db 0 ; floppy drive type
                                      fd0 type:
2269 00005D57 00
                                                  db 0 ; 4 = 1.44 Mb, 80 track, 3.5" (18 spt)
                                      fd1_type:
                                                      ; 6 = 2.88 \text{ Mb}, 80 \text{ track}, 3.5" (36 spt)
2270
                                                       ; 3 = 720 Kb, 80 track, 3.5" (9 spt); 2 = 1.2 Mb, 80 track, 5.25" (15 spt)
2271
2272
2273
                                                       ; 1 = 360 \text{ Kb}, 40 \text{ track}, 5.25" (9 \text{ spt})
2274 00005D58 00
                                                  db 0 ; EDD status for hd0 (bit 7 = present flag)
                                     hd0_type:
2275 00005D59 00
                                     hd1_type:
                                                  db 0 ; EDD status for hd1 (bit 7 = present flag)
                                     hd2_type:
                                                  db 0 ; EDD status for hd2 (bit 7 = present flag)
2276 00005D5A 00
2277 00005D5B 00
                                                  db 0 ; EDD status for hd3 (bit 7 = present flag)
                                     hd3_type:
2278
                                                      ; bit 0 - Fixed disk access subset supported
                                                       ; bit 1 - Drive locking and ejecting
2279
2280
                                                       ; bit 2 - Enhanced disk drive support
2281
                                                       ; bit 3 = Reserved (64 bit EDD support)
```

```
; (If bit 0 is '1' Retro UNIX 386 v1
2282
2283
                                                    ; will interpret it as 'LBA ready'!)
2284
2285
                                   ; 11/03/2015 - 10/07/2015
2286 00005D5C 00000000000000000000
                                   drv.cylinders: dw 0,0,0,0,0,0,0
2286 00005D65 0000000000
2287 00005D6A 00000000000000000000
                                   drv.heads: dw 0,0,0,0,0,0
2287 00005D73 0000000000
2288 00005D78 00000000000000000000
                                   drv.spt:
                                               dw 0,0,0,0,0,0,0
2288 00005D81 0000000000
2289 00005D86 00000000000000000000
                                   drv.size:
                                                dd 0,0,0,0,0,0,0
2289 00005D8F 000000000000000000000
2289 00005D98 00000000000000000000
2289 00005DA1 00
                                   drv.status: db 0,0,0,0,0,0,0 drv.error: db 0,0,0,0,0,0,0,0
2290 00005DA2 0000000000000
2291 00005DA9 00000000000000
2293
                                   Align 2
2294
2295
                                   ::: 11/03/2015
                                   %include 'kybdata.s' ; KEYBOARD (BIOS) DATA
2296
                                2
                                <1> ; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.0 - kybdata.s
  4
                                <1>; Last Update: 17/01/2016
                                5
                                <1> ; Beginning: 17/01/2016
                                <1>; ------
  7
  8
                                <1>; Assembler: NASM version 2.11 (trdos386.s)
  9
 10
                                <1>; Turkish Rational DOS
 11
                                <1>; Operating System Project v2.0 by ERDOGAN TAN (Beginning: 04/01/2016)
 12
 13
                                <1>; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
 14
                                <1>; kybdata.inc (11/03/2015)
 15
                                <1>;
                                <1>; Derived from 'IBM PC-XT-286' BIOS source code (1986)
 16
                                17
 18
                                <1>
 19
                                <1>; Retro UNIX 386 v1 Kernel - KYBDATA.INC
 20
                                <1> ; Last Modification: 11/03/2015
                                         (Data Section for 'KEYBOARD.INC')
 22
                                <1> ;
 23
                                <1> ; /////// KEYBOARD DATA //////////
 24
                                <1>
 25
                                <1>; 05/12/2014
 26
                                <1>; 04/12/2014 (derived from pc-xt-286 bios source code -1986-)
 27
                                <1>; 03/06/86 KEYBOARD BIOS
 28
                                <1>
 29
                                <1>; KEY IDENTIFICATION SCAN TABLES
 30
 31
 32
                               <1>
 33
                               <1> ;----
                                             TABLES FOR ALT CASE -----
 34
                               <1> ;----
                                              ALT-INPUT-TABLE
 35 00005DB0 524F50514B
                               <1> K30: db
                                              82,79,80,81,75
                               <1> db 76,77,71,72,73
 36 00005DB5 4C4D474849
                                                                       ; 10 NUMBER ON KEYPAD
                               <1> ;----
                                              SUPER-SHIFT-TABLE
 37
                              db 16,17,18,19,20,21
<1> db 22,23,24,25,30,31
<1> db 32,33,34,35,36,37
<1> db 38,44,45,46,47,48
<1> db 49,50
 38 00005DBA 101112131415
                                               16,17,18,19,20,21 ; A-Z TYPEWRITER CHARS
 39 00005DC0 161718191E1F
 40 00005DC6 202122232425
  41 00005DCC 262C2D2E2F30
 42 00005DD2 3132
 43
                               <1>
 44
                               <1> ;----
                                               TABLE OF SHIFT KEYS AND MASK VALUES
                               <1> ;----
 45
                                              KEY_TABLE
                               <1> _K6: db
 46 00005DD4 52
                                               INS KEY
                                                                         ; INSERT KEY
                               CAPS_KEY, NUM_KEY, SCROLL_KEY, ALT_KEY, CTL_KEY
 47 00005DD5 3A4546381D
                               <1> db LEFT_KEY,RIGHT_KEY <1> _K6L equ $-_K6
 48 00005DDA 2A36
 49
                               <1>
 50
                               <1>;----
                                              MASK TABLE
                               <1>_K7: db INS_SHIFT
 52 00005DDC 80
                                                                        ; INSERT MODE SHIFT
                               <1> db
                                              CAPS_SHIFT, NUM_SHIFT, SCROLL_SHIFT, ALT_SHIFT, CTL_SHIFT
  53 00005DDD 4020100804
 54 00005DE2 0201
                                      db
                               <1>
                                              LEFT SHIFT, RIGHT SHIFT
                               <1>
 55
                               <1> ;----
                                              TABLES FOR CTRL CASE
                                                                             ;---- CHARACTERS -----
                               <1> _K8: db
                                              27,-1,0,-1,-1, ; Esc, 1, 2, 3, 4, 5
30,-1,-1,-1,-1,31 ; 6, 7, 8, 9, 0, -
-1,127,-1,17,23,5 ; =, Bksp, Tab, Q, W, E
 57 00005DE4 1BFF00FFFFF
                                    . ab
db
  58 00005DEA 1EFFFFFFFF1F
                               <1>
  59 00005DF0 FF7FFF111705
                               <1>
                               <1> db
                                               18,20,25,21,9,15 ; R, T, Y, U, I, O
  60 00005DF6 12141915090F
                                               16,27,29,10,-1,1 ; P, [, ], Enter, Ctrl, A
19,4,6,7,8,10 ; S, D, F, G, H, J
  61 00005DFC 101B1D0AFF01
                               <1>
                                         db
  62 00005E02 13040607080A
                              <1>
                                         db
  63 00005E08 OBOCFFFFFFF
                               <1>
                                               11,12,-1,-1,-1 ; K, L, :, ', `, LShift
  64 00005E0E 1C1A18031602
                                                                  ; Bkslash, Z, X, C, V, B
                               <1>
                                               28,26,24,3,22,2
                                         db
                                               14,13,-1,-1,-1 ; N, M, ,, ., /, RShift
  65 00005E14 0E0DFFFFFFF
                               <1>
                                         db
                                               150,-1,'',-1
  66 00005E1A 96FF20FF
                               <1>
                                                                 ; *, ALT, Spc, CL
                                                                 ;---- FUNCTIONS -----
                               <1>
                                       ;
db
 68 00005E1E 5E5F60616263
                               <1>
                                               94,95,96,97,98,99 ; F1 - F6
                                      db
                                               100,101,102,103,-1,-1 ; F7 - F10, NL, SL
119,141,132,142,115,143 ; Home, Up, PgUp, -, Left, Pad5
 69 00005E24 64656667FFFF
                               <1>
 70 00005E2A 778D848E738F
                               <1>
                                      db
                                               116,144,117,145,118,146; Right, +, End, Down, PgDn, Ins
  71 00005E30 749075917692
                               <1>
                                        db
 72 00005E36 93FFFFFF898A
                               <1>
                                                                      ; Del, SysReq, Undef, WT, F11, F12
                                        db
                                               147,-1,-1,-1,137,138
                               <1>
 73
                               <1> ;----
                                               TABLES FOR LOWER CASE -----
 75 00005E3C 1B3132333435363738- <1> K10: db
                                               27, '1234567890-=',8,9
  75 00005E45 39302D3D0809
 76 00005E4B 71776572747975696F- <1>
                                         db
                                               'qwertyuiop[]',13,-1,'asdfghjkl;',39
  76 00005E54 705B5D0DFF61736466- <1>
 76 00005E5D 67686A6B6C3B27 <1>
 77 00005E64 60FF5C7A786376626E- <1>
                                               96,-1,92,'zxcvbnm,./',-1,'*',-1,' ',-1
                                        db
 77 00005E6D 6D2C2E2FFF2AFF20FF <1>
                               <1> ;----
                                              LC TABLE SCAN
 79 00005E76 3B3C3D3E3F
                               <1> db
                                              59,60,61,62,63
                                                                      ; BASE STATE OF F1 - F10
 80 00005E7B 4041424344
                               <1>
                                         db
                                              64,65,66,67,68
```

```
db
 81 00005E80 FFFF
                                        -1,-1
                                                       ; NL, SL
                           <1>
                           <1>
                           <1> ;----
                                        KEYPAD TABLE
 83
 71,72,73,-1,75,-1 ; BASE STATE OF KEYPAD KEYS
                                        77,-1,79,80,81,82,83
 86 00005E8F FFFF5C8586
                           <1>
                                   db
                                        -1,-1,92,133,134 ; SysRq, Undef, WT, F11, F12
                          <1>
                          <1> ;----
                                         TABLES FOR UPPER CASE -----
 88
 89 00005E94 1B21402324255E262A- <1> K11: db
                                         27,'!@#$%',94,'&*()_+',8,0
 89 00005E9D 28295F2B0800 <1>
 90 00005EA3 51574552545955494F- <1>
                                        'QWERTYUIOP{}',13,-1,'ASDFGHJKL:"'
 90 00005EAC 507B7D0DFF41534446- <1>
 90 00005EB5 47484A4B4C3A22 <1>
 91 00005EBC 7EFF7C5A584356424E- <1>
                                 db
                                        126,-1,'|ZXCVBNM<>?',-1,'*',-1,' ',-1
 91 00005EC5 4D3C3E3FFF2AFF20FF <1>
                           <1> ;----
                                        UC TABLE SCAN
                          <1> K12: db
                                                             ; SHIFTED STATE OF F1 - F10
 93 00005ECE 5455565758
                                        84,85,86,87,88
 94 00005ED3 595A5B5C5D
                           <1> db
                                        89,90,91,92,93
 95 00005ED8 FFFF
                                                        ; NL, SL
                           <1>
                                         -1,-1
                           <1>
 96
 97
                           <1> ;----
                                        NUM STATE TABLE
 98 00005EDA 3738392D3435362B31- <1> K14: db
                                         '789-456+1230.'
                                                             ; NUMLOCK STATE OF KEYPAD KEYS
 98 00005EE3 3233302E
                     <1>
 99
                           <1>
                                        -1,-1,124,135,136 ; SysRq, Undef, WT, F11, F12
100 00005EE7 FFFF7C8788
                                  db
                           <1>
101
                           <1>
102
                           <1> ; 26/08/2014
                           <1>; Retro UNIX 8086 v1 - UNIX.ASM (03/03/2014)
103
                           <1> ; Derived from IBM "pc-at"
104
105
                           <1>; rombios source code (06/10/1985)
106
                           <1> ; 'dseg.inc'
107
                           <1>
108
                           <1> ;-----;
109
                           <1>; SYSTEM DATA AREA ;
                           <1> ;-----
110
                           <1> BIOS_BREAK db 0 ; BIT 7=1 IF BREAK KEY HAS BEEN PRESSED
111 00005EEC 00
112
                           <1>
113
                           <1> ;------
                           <1>; KEYBOARD DATA AREAS ;
114
115
                           <1> ;-----
116
                           <1>
                           117 00005EED 00
                                                              ; KEYBOARD SHIFT STATE AND STATUS FLAGS
118 00005EEE 00
119 00005EEF 00
120 00005EF0 00
                           121 00005EF1 00
122 00005EF2 [025F0000]
                           123 00005EF6 [225F0000]
124 00005EFA [025F0000]
125 00005EFE [025F0000]
126
                           <1>; ----- HEAD = TAIL INDICATES THAT THE BUFFER IS EMPTY
127 00005F02 0000<rept>
                           <1> KB BUFFER
                                        times 16 dw 0
                                                              ; ROOM FOR 16 SCAN CODE ENTRIES
128
                           <1>
129
                           <1>; /// End Of KEYBOARD DATA ///
2297
                               %include 'vidata.s'; VIDEO (BIOS) DATA
                           1
                           <1>; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.0 - vidata.s
  3
                           <1> ; Last Update: 31/07/2016
                           <1> ; -----
  5
  6
                           <1> ; Beginning: 16/01/2016
                           <1> ; -----
  8
                           <1>; Assembler: NASM version 2.11 (trdos386.s)
  9
 10
                           <1>; Turkish Rational DOS
 11
                           <1>; Operating System Project v2.0 by ERDOGAN TAN (Beginning: 04/01/2016)
 12
                           <1> ; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
 13
                           <1>; vidata.inc (11/03/2015)
 14
 15
                           <1>;
 16
                           <1> ; Derived from 'IBM PC-AT' BIOS source code (1985)
                            17
 18
                           <1>
 19
                           <1>; Retro UNIX 386 v1 Kernel - VIDATA.S
                           <1> ; Last Modification: 11/03/2015
 20
                                            (Data section for 'VIDEO.INC')
 21
                           <1>;
 22
                           <1>;
                           <1> ; /////// VIDEO DATA //////////
 23
 24
 25
 26
                           <1>; VIDEO DISPLAY DATA AREA
 28 00005F22 03
                                                    ; CURRENT DISPLAY MODE (TYPE)
                                              3
                           <1> CRT MODE: db
 29 00005F23 29
                                                        ; CURRENT SETTING OF THE 3X8 REGISTER
                           <1> CRT_MODE_SET:
                                                    29h
 30
                           <1>
                                                    ; (29h default setting for video mode 3)
 31
                           <1>
                                                    ; Mode Select register Bits
 32
                           <1>
                                                    ; BIT 0 - 80x25 (1), 40x25 (0)
                                                       BIT 1 - ALPHA (0), 320x200 GRAPHICS (1)
 33
                           <1>
 34
                           <1>
                                                       BIT 2 - COLOR (0), BW (1)
                                                       BIT 3 - Video Sig. ENABLE (1), DISABLE (0)
 35
                           <1>
 36
                           <1>
                                                       BIT 4 - 640x200 B&W Graphics Mode (1)
                                                       BIT 5 - ALPHA mode BLINKING (1)
BIT 6, 7 - Not Used
 37
                            <1>
 38
                           <1>
 39
                           <1>
                           <1> ; Mode 0 - 2Ch = 101100b ; 40x25 text, 16 gray colors
 40
                           <1> ; Mode 1 - 28h = 101000b ; 40x25 text, 16 fore colors, 8 back colors
 41
                           <1>; Mode 2 - 2Dh = 101101b ; 80x25 text, 16 gray colors
 42
                           <1>; Mode 3 - 29h = 101001b; 80x25 text, 16 fore color, 8 back color
 43
                           <1>; Mode 4 - 2Ah = 101010b ; 320x200 graphics, 4 colors
 44
                           <1>; Mode 5 - 2Eh = 101110b ; 320x200 graphics, 4 gray colors
 45
                           <1> ; Mode 6 - 1Eh = 011110b ; 640x200 graphics, 2 colors
 46
 47
                            <1>; Mode 7 - 29h = 101001b ; 80x25 text, black & white colors
 48
                           <1> ; Mode & 37h = Video signal OFF
 49
                           <1>
 50
                           <1> ; 24/06/2016
```

```
<1> CRT COLS: db
 51 00005F24 50
                                                       80
                                                             ; Number of columns
                                 <1>
                                <1>; 01/07/2016
 53
 54 00005F25 00
                                <1> CRT_PALETTE: db
                                                       0
                                                              ; Current palette setting
 55
                                <1>
 56
                                <1>; 03/07/2016
 57 00005F26 10
                                <1> CHAR HEIGHT: db
                                                              ; Default character height
                                                       16
                                <1> VGA_VIDEO_CTL:
                                                             60h ; ROM BIOS DATA AREA Offset 87h
0F9h ; Feature Bit Switches (the basic screen)
 58 00005F27 60
                                                       db
 59 00005F28 F9
                                <1> VGA_SWITCHES:
                                                       db
                                <1> VGA_MODESET_CTL: db051h ; Basic mode set options (VGA video flags)
 60 00005F29 51
                                                              ; ROM BIOS DATA AREA Offset 89h
 61
                                <1>
 62
                                 <1>
                                                              ; Bit 7, 4 : Mode
                                                                      01 : 400-line mode
 63
                                <1>
 64
                                 <1>
                                                              ; Bit 6 : Display switch enabled = 1
                                                              ; Bit 5
                                                                         : Reserved = 0
: Default palette loading
 65
                                 <1>
 66
                                 <1>
                                                              ; Bit 3
 67
                                 <1>
                                                                      disabled = 0
                                                              ; Bit 2 : Color monitor = 0
 68
                                 <1>
 69
                                 <1>
                                                              ; Bit 1 = Gray scale summing
 70
                                 <1>
                                                                      disabled = 0
 71
                                 <1>
                                                              ; Bit 0 = VGA active = 1
                                                       25
 72 00005F2A 19
                                 <1> VGA_ROWS:
                                                db
 73
                                 <1>
 74
                                 <1> ; 16/01/2016
 75
                                 <1> chr_attrib: ; Character color/attributes for viode pages (0 to 7)
 76 00005F2B 0707070707070707
                                 <1> db 07h, 07h, 07h, 07h, 07h, 07h, 07h, 07h
                                 <1>; 30/01/2016
                                <1> vmode:
 78
 79 00005F33 0303030303030303
                                <1>
                                                 3,3,3,3,3,3,3; video modes for pseudo screens
                                <1>
 80
 81
                                 <1> CURSOR_MODE: ; cursor start (ch) = 14, cursor end (cl) = 15
 82 00005F3B 0F0E
                                              15, 14 ; 07/07/2016 - TRDOS 386 (TRDOS v2.0)
                                 <1>
                                        db
                                <1>
 83
 84
                                 <1> ;align 4
 85
                                 <1> ; VGA_BASE: ; 26/07/2016
                                                 OB8000h ; (Mode < ODh) or OA0000h (mode >= ODh)
 86
                                 <1>;
                                         dd
 87
                                 <1>
 88 00005F3D 90
                                 <1> align 2
 89
                                 <1>
                                <1> vga_modes:
 90
                                      ; 25/07/2016
 91
                                 <1>
                                92
 93
 95 00005F3E 0302010007040506
                                <1> vga_g_modes: ; 31/07/2016
<1> db 13h, 0F0h, 12h, 6Ah, 0Dh, 0Eh, 10h, 11h
 96
 97 00005F46 13F0126A0D0E1011
 98
                                 <1> vga_mode_count equ $ - vga_modes
 99
                                 <1> vga_g_mode_count equ $ - vga_g_modes
100
                                 <1>
101
                                 <1> vga_mode_tbl_ptr:
102
                                 <1>
                                          ; 25/07/2016
103 00005F4E [AE5F0000]
                                <1>
                                          dd
                                                vga_mode_03h
104 00005F52 [AE5F0000]
                                <1>
                                                 vga_mode_03h; mode 02h -> mode 03h
105 00005F56 [EE5F0000]
                                <1>
                                                vga_mode_01h
                                          dd
106 00005F5A [EE5F0000]
                                <1>
                                          dd
                                                 vga_mode_01h ; mode 00h -> mode 01h
                                                vga mode 07h
                                <1>
                                          ;dd
                                                 vga_mode_03h ; mode 07h -> mode 03h
vga_mode_04h
108 00005F5E [AE5F0000]
                                <1>
                                          dd
109 00005F62 [2E600000]
                                <1>
                                          dd
                                                 vga_mode_04h; mode 05h -> mode 04h
110 00005F66 [2E600000]
                                <1>
                                          dd
111 00005F6A [6E600000]
                                <1>
                                          dd
                                                 vga_mode_06h
112 00005F6E [AE600000]
                                <1>
                                          dd
                                                 vga_mode_13h
                                                 vga_mode_F0h
113 00005F72 [EE600000]
                                <1>
                                          dd
114 00005F76 [2E610000]
                                <1>
                                                 vga_mode_12h
115 00005F7A [6E610000]
                                <1>
                                          dd
                                                 vga_mode_6Ah
116 00005F7E [AE610000]
                                <1>
                                          dd
                                                 vga_mode_0Dh
117 00005F82 [EE610000]
                                <1>
                                          dd
                                                 vga mode 0Eh
118 00005F86 [2E620000]
                                <1>
                                          dd
                                                 vga_mode_10h
119 00005F8A [6E620000]
                                <1>
                                          dd
                                                 vga_mode_11h
120
                                <1>
                                 <1> vga memmodel:
121
                                        ; 25/07/2016
122
                                 <1>
                                          ; 07/07/2016
123
                                 <1>
124
                                 <1>
                                          CTEXT equ 0
                                 <1>
125
                                          ;MTEXT equ 1
                                          MTEXT equ 0 ; mode 07h \rightarrow mode 03h
126
                                 <1>
127
                                 <1>
                                          CGA equ 2
128
                                 <1>
                                          LINEAR8 equ 5
                                          PLANAR4
129
                                 <1>
                                                       equ 4
                                                       equ 3
130
                                 <1>
                                          PLANAR1
131 00005F8E 0000000000020202
                                <1>
                                          db CTEXT, CTEXT, CTEXT, MTEXT, CGA, CGA, CGA
                                 <1> vga_g_memmodel: ; 31/07/2016
133 00005F96 0504040404040403
                                 <1>
                                        db
                                                LINEAR8, PLANAR4, PLANAR4, PLANAR4, PLANAR4, PLANAR4, PLANAR4, PLANAR1
                                 <1> ;vga_pixbits:
                                 <1>; 25/07/2016
135
                                          ; 08/07/2016
136
                                 <1>;
137
                                 <1>;
                                          db
                                                4, 4, 4, 4, 4, 2, 2, 1, 8, 4, 4, 4, 4, 4, 4, 1
138
                                 <1> vga_dac_s:
139 00005F9E 020202020001010103- <1>
                                          db
                                                 2, 2, 2, 2, 0, 1, 1, 1, 3, 3, 2, 2, 1, 1, 2, 2
139 00005FA7 03020201010202
                                 <1>
140
                                 <1>
                                 <1> vga_params:
141
                                         ; 25/07/2016
142
                                <1>
143
                                <1>
                                          ; 19/07/2016
144
                                 <1>
                                          ; 03/07/2016
                                          ; derived from 'Plex86/Bochs VGABios' source code
145
                                 <1>
                                          ; vgabios-0.7a (2011)
146
                                 <1>
                                          ; by the LGPL VGABios Developers Team (2001-2008)
147
                                 <1>
148
                                 <1>
                                          ; 'vgatables.h'
                                          ; Oracle VirtualBox 5.0.24 VGABios Source Code
149
                                <1>
                                          ; ('vgabios.c', 'vgatables.h', 'vgafonts.h', 'vgarom.asm')
150
                                <1>
151
                                 <1>
152
                                <1> vga_mode_03h: ; mode 03h, 80*25 text, CGA colors
153 00005FAE 5018100010
                                <1>
                                          db 80, 24, 16, 00h, 10h; tw, th-1, ch, slength (5)
154 00005FB3 00030002
                                <1>
                                          db
                                                 00h, 03h, 00h, 02h; sequ regs (4)
```

```
67h ; misc reg (1)
155 00005FB7 67
                                 <1>
                                          db
156 00005FB8 5F4F50825581BF1F
                                <1>
                                          db
                                                 5Fh, 4Fh, 50h, 82h, 55h, 81h, 0BFh, 1Fh
157 00005FC0 004F
                                          db
                                                 00h, 4Fh
                                 <1>
                                 <1> vga_p_cm_pos equ $ - vga_mode_03h
159 00005FC2 0D0E00000000
                                          db
                                                 ODh, OEh, OOh, OOh, OOh, OOh
                                 <1>
160 00005FC8 9C8E8F281F96B9A3
                                 <1>
                                           db
                                                 9Ch, 8Eh, 8Fh, 28h, 1Fh, 96h, 0B9h, 0A3h
161 00005FD0 FF
                                 <1>
                                                 OFFh ; crtc regs (25)
162 00005FD1 0001020304051407
                                                 00h, 01h, 02h, 03h, 04h, 05h, 14h, 07h
                                 <1>
                                          db
163 00005FD9 38393A3B3C3D3E3F
                                 <1>
                                                 38h, 39h, 3Ah, 3Bh, 3Ch, 3Dh, 3Eh, 3Fh
                                          db
164 00005FE1 0C000F08
                                                 OCh, OOh, OFh, O8h ; actl regs (20)
                                 <1>
                                          db
                                                 00h, 00h, 00h, 00h, 00h, 10h, 0Eh, 0Fh, 0Ffh; grdc regs (9)
165 00005FE5 000000000100E0FFF <1>
                                          db
                                 <1> vga mode 01h: ; mode 01h, 40*25 text, CGA colors
                                                40, 24, 16, 00h, 08h; tw, th-1, ch, slength
167 00005FEE 2818100008
                                 <1>
                                          db
168 00005FF3 08030002
                                                 08h, 03h, 00h, 02h ; sequ regs
                                <1>
                                                 67h ; misc reg
2Dh, 27h, 28h, 90h, 2Bh, 0A0h, 0BFh, 1Fh
169 00005FF7 67
                                 <1>
                                          db
170 00005FF8 2D2728902BA0BF1F
                                 <1>
                                          db
171 00006000 004F0D0E00000000
                                                 00h, 4Fh, 0Dh, 0Eh, 00h, 00h, 00h, 00h
                                 <1>
                                          db
172 00006008 9C8E8F141F96B9A3
                                                 9Ch, 8Eh, 8Fh, 14h, 1Fh, 96h, 0B9h, 0A3h
                                 <1>
                                          db
173 00006010 FF
                                 <1>
                                          db
                                                 OFFh ; crtc_regs
174 00006011 0001020304051407
                                                 00h, 01h, 02h, 03h, 04h, 05h, 14h, 07h
                                 <1>
                                          db
175 00006019 38393A3B3C3D3E3F
                                 <1>
                                                 38h, 39h, 3Ah, 3Bh, 3Ch, 3Dh, 3Eh, 3Fh
176 00006021 0C000F08
                                 <1>
                                          db
                                                 OCh, OOh, OFh, O8h ; actl regs
177 00006025 000000000100E0FFF
                                                 00h, 00h, 00h, 00h, 00h, 10h, 0Eh, 0Fh, 0FFh; grdc regs
                               <1>
                                          db
                                 179
                                 <1>;
                                                 80, 24, 16, 00h, 10h ; tw, th-1, ch, slength
                                          db
180
                                 <1>;
                                           db
                                                 00h, 03h, 00h, 02h; sequ regs
181
                                 <1>;
                                                 66h ; misc reg
                                                 5Fh, 4Fh, 50h, 82h, 55h, 81h, 0BFh, 1Fh
182
                                 <1>;
                                           db
                                 <1>;
                                                 00h, 4Fh, 0Dh, 0Eh, 00h, 00h, 00h, 00h
183
                                                 9Ch, 8Eh, 8Fh, 28h, 0Fh, 96h, 0B9h, 0A3h
                                 <1>;
184
                                          db
185
                                 <1>;
                                           db
                                                 OFFh ; crtc regs
                                                 00h, 08h, 08h, 08h, 08h, 08h, 08h, 08h
186
                                 <1>;
                                           db
                                 <1>;
                                                 10h, 18h, 18h, 18h, 18h, 18h, 18h
187
                                           db
188
                                 <1>;
                                                 OEh, OOh, OFh, O8h ; actl regs
                                                 00h, 00h, 00h, 00h, 00h, 10h, 0Ah, 0Fh, 0FFh; grdc regs
                                 <1>;
189
                                          db
190
                                 <1> vga_mode_04h:
                                                      ; 320*200 graphics, 4 colors, CGA
                                                40, 24, 8, 00h, 08h ; tw, th-1, ch, slength 09h, 03h, 00h, 02h ; sequ regs
191 0000602E 2818080008
                                 <1>
192 00006033 09030002
                                          db
                                 <1>
193 00006037 63
                                                 63h ; misc reg
                                 <1>
                                           db
194 00006038 2D2728902B80BF1F
                                                 2Dh, 27h, 28h, 90h, 2Bh, 80h, 0BFh, 1Fh
                                 <1>
                                          db
195 00006040 00C100000000000
                                                 00h, 0Clh, 00h, 00h, 00h, 00h, 00h
                                 <1>
                                                 9Ch, 8Eh, 8Fh, 14h, 00h, 96h, 0B9h, 0A2h
196 00006048 9C8E8F140096B9A2
                                 <1>
                                          db
197 00006050 FF
                                                 OFFh ; crtc regs
                                 <1>
                                          db
198 00006051 0013151702040607
                                 <1>
                                                 00h, 13h, 15h, 17h, 02h, 04h, 06h, 07h
                                                 10h, 11h, 12h, 13h, 14h, 15h, 16h, 17h
199 00006059 1011121314151617
                                 <1>
                                          db
200 00006061 01000300
                                                 01h, 00h, 03h, 00h; actl regs
                                 <1>
                                           db
                                                 00h, 00h, 00h, 00h, 00h, 30h, 0Fh, 0Fh, 0FFh; grdc regs
201 00006065 000000000300F0FFF <1>
                                          db
                                                 : ; 640*200 graphics, 2 colors, CGA
80, 24, 8, 00h, 10h ; tw, th-1, ch, slength
                                 <1> vga_mode_06h:
203 0000606E 5018080010
                                 <1>
                                           db
204 00006073 01010006
                                                 01h, 01h, 00h, 06h; sequ regs
                                 <1>
                                           db
205 00006077 63
                                 <1>
                                           db
                                                 63h ; misc reg
206 00006078 5F4F50825480BF1F
                                 <1>
                                          db
                                                 5Fh, 4Fh, 50h, 82h, 54h, 80h, 0BFh, 1Fh
207 00006080 00C100000000000
                                                 00h, 0Clh, 00h, 00h, 00h, 00h, 00h
                                 <1>
                                          db
208 00006088 9C8E8F280096B9C2
                                 <1>
                                                 9Ch, 8Eh, 8Fh, 28h, 00h, 96h, 0B9h, 0C2h
209 00006090 FF
                                 <1>
                                          db
                                                 OFFh ; crtc regs
210 00006091 00171717171717
                                 <1>
                                          db
                                                 00h, 17h, 17h, 17h, 17h, 17h, 17h, 17h
211 00006099 17171717171717
                                                 17h, 17h, 17h, 17h, 17h, 17h, 17h, 17h
                                 <1>
                                          db
212 000060A1 01000100
                                          db
                                                 01h, 00h, 01, 00h ; actl regs
                                 <1>
213 000060A5 0000000000000D0FFF
                                <1>
                                                 00h, 00h, 00h, 00h, 00h, 00h, 0Fh, 0Fh; grdc regs
                                          db
                                 <1> vga_mode_13h: ; mode 13h, 300*200, 256 colors, linear
215 000060AE 2818080000
                                 <1>
                                          db
                                                 40, 24, 8, 0, 0 ; tw, th-1, ch, slength (5)
                                                 01h, 0Fh, 00h, 0Eh; sequ regs (4)
216 000060B3 010F000E
                                 <1>
                                          db
217 000060B7 63
                                 <1>
                                          db
                                                 63h ; misc reg (1)
218 000060B8 5F4F50825480BF1F
                                 <1>
                                                 5Fh, 4Fh, 50h, 82h, 54h, 80h, 0BFh, 1Fh
219 000060C0 004100000000000
                                                 00h, 41h, 00h, 00h, 00h, 00h, 00h
                                 <1>
                                          db
                                                 9Ch, 8Eh, 8Fh, 28h, 40h, 96h, 0B9h, 0A3h
220 000060C8 9C8E8F284096B9A3
                                 <1>
                                          db
221 000060D0 FF
                                 <1>
                                          db
                                                 OFFh ; crtc regs (25)
222 000060D1 0001020304050607
                                          db
                                                 00h, 01h, 02h, 03h, 04h, 05h, 06h, 07h
                                 <1>
223 000060D9 08090A0B0C0D0E0F
                                 <1>
                                                 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh
                                          db
                                                 41h, 00h, 0Fh, 00h ; actl regs (20)
224 000060E1 41000F00
                                 <1>
                                          db
225 000060E5 00000000040050FFF <1>
                                          db
                                                 00h, 00h, 00h, 00h, 00h, 40h, 05h, 0Fh, 0FFh; grdc regs (9)
                                 <1> vga_mode_setl equ $ - vga_mode_13h ; = 64
226
                                 <1> vga_mode_F0h: ; mode X ; 320*240, 256 colors, planar
227
                                                 40, 24, 8, 0, 0
228 000060EE 2818080000
                                <1>
                                                                   ; tw, th-1, ch, slength
229 000060F3 010F0006
                                 <1>
                                                 01h, 0Fh, 00h, 06h; sequ regs
                                           db
230 000060F7 E3
                                                 OE3h ; misc reg
                                 <1>
                                           db
231 000060F8 5F4F508254800D3E
                                 <1>
                                                 5Fh, 4Fh, 50h, 82h, 54h, 80h, 0Dh, 3Eh
                                          db
232 00006100 004100000000000
                                 <1>
                                          db
                                                 00h, 41h, 00h, 00h, 00h, 00h, 00h
233 00006108 EAACDF2800E706E3
                                                 OEAh, OACh, ODFh, 28h, OOh, OE7h, O6h, OE3h
                                 <1>
                                           db
234 00006110 FF
                                                 OFFh ; crtc regs (25)
                                 <1>
                                          db
235 00006111 0001020304050607
                                                 00h, 01h, 02h, 03h, 04h, 05h, 06h, 07h
                                 <1>
                                           db
                                                 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh
41h, 00h, 0Fh, 00h ; actl regs
236 00006119 08090A0B0C0D0E0F
237 00006121 41000F00
                                           db
238 00006125 00000000040050FFF <1>
                                                 00h, 00h, 00h, 00h, 00h, 40h, 05h, 0Fh, 0FFh; grdc regs
                                 <1> vga_mode_12h: ; mode 12h, 640*480, 16 colors, planar
239
240 0000612E 501D100000
                                 <1>
                                           db
                                                 80, 29, 16, 0, 0; tw, th-1, ch, slength
241 00006133 010F0006
                                 <1>
                                                 01h, 0Fh, 00h, 06h; sequ regs
                                           db
242 00006137 E3
                                 <1>
                                          db
                                                 OE3h ; misc reg
243 00006138 5F4F508254800B3E
                                 <1>
                                                 5Fh, 4Fh, 50h, 82h, 54h, 80h, 0Bh, 3Eh
                                          db
244 00006140 0040000000000000
                                                 00h, 40h, 00h, 00h, 00h, 00h, 00h
                                 <1>
                                          db
245 00006148 EA8CDF2800E704E3
                                 <1>
                                          db
                                                 OEAh, 8Ch, ODFh, 28h, OOh, OE7h, O4h, OE3h
246 00006150 FF
                                                 OFFh ; crtc regs
                                 <1>
                                          db
                                                 00h, 01h, 02h, 03h, 04h, 05h, 14h, 07h
247 00006151 0001020304051407
                                 <1>
                                          db
248 00006159 38393A3B3C3D3E3F
                                 <1>
                                                 38h, 39h, 3Ah, 3Bh, 3Ch, 3Dh, 3Eh, 3Fh
249 00006161 01000F00
                                 <1>
                                          db
                                                 01h, 00h, 0Fh, 00h ; actl regs
250 00006165 00000000000050FFF <1>
                                                 00h, 00h, 00h, 00h, 00h, 05h, 0Fh, 0FFh; grdc regs
                                           db
                                 <1> vga mode 6Ah: ; mode 6Ah, 800*600, 16 colors, planar
                                                 100, 36, 16, 0, 0 ; tw, th-1, ch, slength
252 0000616E 6424100000
                                 <1>
                                           db
253 00006173 010F0006
                                 <1>
                                           db
                                                 01h, 0Fh, 00h, 06h; sequ regs
254 00006177 E3
                                                 OE3h ; misc req
                                 <1>
                                          db
                                                 7Fh, 63h, 63h, 83h, 6Bh, 1Bh, 72h, 0F0h
255 00006178 7F6363836B1B72F0
                                 <1>
                                           db
256 00006180 0060000000000000
                                 <1>
                                          db
                                                 00h, 60h, 00h, 00h, 00h, 00h, 00h
257 00006188 598D5732005773E3
                                                 59h, 8Dh, 57h, 32h, 00h, 57h, 73h, 0E3h
                                 <1>
                                          db
258 00006190 FF
                                 <1>
                                                 OFFh ; crtc regs
259 00006191 0001020304051407
                                 <1>
                                          db
                                                 00h, 01h, 02h, 03h, 04h, 05h, 14h, 07h
```

```
260 00006199 38393A3B3C3D3E3F
                                             38h, 39h, 3Ah, 3Bh, 3Ch, 3Dh, 3Eh, 3Fh
                             <1>
                                       db
 261 000061A1 01000F00
                              <1>
                                       db
                                             01h, 00h, 0Fh, 00h ; actl regs
                                            00h, 00h, 00h, 00h, 00h, 05h, 0Fh, 0FFh; grdc regs
262 000061A5 00000000000050FFF <1>
                                      db
                              <1> vga_mode_0Dh: ; mode 0Dh, 320*200, 16 colors, planar
264 000061AE 2818080020
                                            40, 24, 8, 0, 20h; tw, th-1, ch, slength
                              <1>
                                       db
265 000061B3 090F0006
                              <1>
                                       db
                                             09h, 0Fh, 00h, 06h; sequ regs
266 000061B7 63
                              <1>
                                      db
                                            63h ; misc reg
267 000061B8 2D2728902B80BF1F
                                   db
db
db
                                             2Dh, 27h, 28h, 90h, 2Bh, 80h, 0BFh, 1Fh
                             <1>
268 000061C0 00C0000000000000
                             <1>
                                             00h, 0C0h, 00h, 00h, 00h, 00h, 00h
                                             9Ch, 8Eh, 8Fh, 14h, 00h, 96h, 0B9h, 0E3h
269 000061C8 9C8E8F140096B9E3
                              <1>
270 000061D0 FF
                              <1>
                                   db
                                             OFFh ; crtc regs
271 000061D1 0001020304050607
                                             00h, 01h, 02h, 03h, 04h, 05h, 06h, 07h
                              <1>
                                      db
272 000061D9 1011121314151617
                                             10h, 11h, 12h, 13h, 14h, 15h, 16h, 17h
                             <1>
                                      db
273 000061E1 01000F00
                             <1>
                                             01h, 00h, 0Fh, 00h ; actl regs
274 000061E5 00000000000050FFF <1>
                                             00h, 00h, 00h, 00h, 00h, 05h, 0Fh, 0FFh; grdc regs
                                     db
275
                              <1> vga_mode_0Eh: ; mode 0Eh, 640*200, 16 colors, planar
                                           80, 24, 8, 0, 40h; tw, th-1, ch, slength
276 000061EE 5018080040
                              <1> db
                                             01h, 0Fh, 00h, 06h; sequ regs
277 000061F3 010F0006
                              <1>
                                      db
278 000061F7 63
                              <1>
                                             63h ; misc reg
                                      db
279 000061F8 5F4F50825480BF1F
                                             5Fh, 4Fh, 50h, 82h, 54h, 80h, 0BFh, 1Fh
                             <1>
                                      db
                                   db
db
db
                                             00h, 0C0h, 00h, 00h, 00h, 00h, 00h
280 00006200 00C0000000000000
                             <1>
                                             9Ch, 8Eh, 8Fh, 28h, 00h, 96h, 0B9h, 0E3h
281 00006208 9C8E8F280096B9E3
                              <1>
282 00006210 FF
                              <1>
                                             OFFh ; crtc regs
283 00006211 0001020304050607
                              <1>
                                             00h, 01h, 02h, 03h, 04h, 05h, 06h, 07h
284 00006219 1011121314151617
                                             10h, 11h, 12h, 13h, 14h, 15h, 16h, 17h
                             <1>
                                      db
285 00006221 01000F00
                              <1>
                                       db
                                             01h, 00h, 0Fh, 00h ; actl regs
286 00006225 000000000000050FFF <1>
                                     db
                                             00h, 00h, 00h, 00h, 00h, 05h, 0Fh, 0FFh; grdc regs
                              <1> vga_mode_10h: ; mode 10h, 640*350, 16 colors, planar
287
                                   db
288 0000622E 50180E0080
                                             80, 24, 14, 0, 80h; tw, th-1, ch, slength
                              <1>
289 00006233 010F0006
                                             01h, 0Fh, 00h, 06h; sequ regs
                              <1>
                                       db
                                    db
290 00006237 A3
                              <1>
                                             OA3h ; misc reg
                                     db
db
291 00006238 5F4F50825480BF1F
                                             5Fh, 4Fh, 50h, 82h, 54h, 80h, 0BFh, 1Fh
                              <1>
                                             00h, 40h, 00h, 00h, 00h, 00h, 00h
292 00006240 004000000000000
                             <1>
293 00006248 83855D280F63BAE3
                             <1>
                                     db
                                             83h, 85h, 5Dh, 28h, 0Fh, 63h, 0BAh, 0E3h
                                     db
294 00006250 FF
                              <1>
                                             OFFh ; crtc regs
                                             00h, 01h, 02h, 03h, 04h, 05h, 14h, 07h
295 00006251 0001020304051407
                              <1>
                                      db
296 00006259 38393A3B3C3D3E3F
                                     db
                             <1>
                                             38h, 39h, 3Ah, 3Bh, 3Ch, 3Dh, 3Eh, 3Fh
                                     db
                                             01h, 00h, 0Fh, 00h ; actl regs
297 00006261 01000F00
                              <1>
298 00006265 00000000000050FFF <1>
                                       db
                                             00h, 00h, 00h, 00h, 00h, 05h, 0Fh, 0FFh; grdc regs
                             <1> vga_mode_11h: ; mode 11h, 640*480, mono color, planar
                                   db
300 0000626E 501D100000
                              <1>
                                            80, 29, 16, 0, 0 ; tw, th-1, ch, slength
 301 00006273 010F0006
                              <1>
                                       db
                                             01h, 0Fh, 00h, 06h; sequ regs
                                  db
db
db
dh
302 00006277 E3
                                            OE3h ; misc reg
                              <1>
 303 00006278 5F4F508254800B3E
                              <1>
                                            5Fh, 4Fh, 50h, 82h, 54h, 80h, 0Bh, 3Eh
 304 00006280 004000000000000
                              <1>
                                             00h, 40h, 00h, 00h, 00h, 00h, 00h
305 00006288 EA8CDF2800E704E3
                              <1>
                                             OEAh, 8Ch, 0DFh, 28h, 00h, 0E7h, 04h, 0E3h
                                     db
306 00006290 FF
                              <1>
                                             OFFh ; crtc regs
 307 00006291 003F003F003F003F
                                     db
                                             00h, 3Fh, 00h, 3Fh, 00h, 3Fh, 00h, 3Fh
                              <1>
308 00006299 003F003F003F003F
                              <1>
                                      db
                                             00h, 3Fh, 00h, 3Fh, 00h, 3Fh, 00h, 3Fh
309 000062A1 01000F00
                                             01h, 00h, 0Fh, 00h ; actl regs
                              <1>
                                      db
310 000062A5 00000000000050FFF <1>
                                     db
                                             00h, 00h, 00h, 00h, 00h, 05h, 0Fh, 0FFh; grdc regs
311
                              <1> end_of_vga_params:
312
                              <1>
313
                              <1> ; /// End Of VIDEO DATA ///
2298
                                  ;%include 'diskdata.s' ; DISK (BIOS) DATA (initialized)
2299
2300
                                  Align 2
2301
2302
2303
                                  %include 'sysdefs.s'; 24/01/2015
                              1
  2
                              <1> ; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.0) - SYSTEM DEFINITIONS : sysdefs.s
                              <1> ; Last Update: 31/12/2017
  5
                              <1> ; -----
  6
                              <1> ; Beginning: 24/01/2016
  8
                              <1>; Assembler: NASM version 2.11 (trdos386.s)
                              <1> ; ------
  9
                              <1> ; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
 10
 11
                              <1>; sysdefs.inc (14/11/2015)
                              12
 13
                              <1>
 14
                              <1>; Retro UNIX 386 v1 Kernel - SYSDEFS.INC
                              <1> ; Last Modification: 14/11/2015
 15
 16
                              <1>;
 17
                              <1>; /////// RETRO UNIX 386 V1 SYSTEM DEFINITIONS //////////////
 18
                              <1>; (Modified from
 19
                                      Retro UNIX 8086 v1 system definitions in 'UNIX.ASM', 01/09/2014)
                              <1>; ((UNIX.ASM (RETRO UNIX 8086 V1 Kernel), 11/03/2013 - 01/09/2014))
 20
 21
                              <1>; UNIX.ASM (MASM 6.11) --> SYSDEFS.INC (NASM 2.11)
 23
 24
                              <1>; Derived from UNIX Operating System (v1.0 for PDP-11)
                              <1>; (Original) Source Code by Ken Thompson (1971-1972)
 25
                              <1>; <Bell Laboratories (17/3/1972)>
 26
 27
                              <1> ; <Preliminary Release of UNIX Implementation Document>
 28
                              <1>;
                              29
 30
                              <1>
                                             equ 16 ; number of processes
 31
                              <1> nproc
 32
                              <1> nfiles
                                             equ 50
                                             8 ; 8+1 -> 8 (10/05/2013)
 33
                              <1> ntty equ
 34
                              <1> nbuf equ
                                             4 ; 6 ;; 21/08/2015 - 'namei' buffer problem when nbuf > 4
                                             ; NOTE: If fd0 super block buffer addres is beyond of the 1st
 35
                              <1>
 36
                              <1>
                                             ; 32K, DMA r/w routine or someting else causes a jump to
                                             ; kernel panic routine (in 'alloc' routine, in u5.s)
 37
                              <1>
 38
                              <1>
                                             ; because of invalid buffer content (r/w error).
                                             ; When all buffers are set before the end of the 1st 32k,
 39
                              <1>
                                             ; there is no problem!? (14/11/2015)
 40
                              <1>
 41
                              <1>
 42
                              <1> ;csgmnt
                                                   2000h ; 26/05/2013 (segment of process 1)
                                             equ
                              <1> ;core equ
 43
                                             0
                                                      ; 19/04/2013
                                                 32768 - 64 ; 04/06/2013 (24/05/2013)
 44
                                             equ
 45
                              <1>
                                     ; (if total size of argument list and arguments is 128 bytes)
```

```
; maximum executable file size = 32768 - (64 + 40 + 128 - 6) = 32530 bytes
 47
                                            ; maximum stack size = 40 bytes (+6 bytes for 'IRET' at 32570)
                                            ; initial value of user's stack pointer = 32768-64-128-2 = 32574
 48
                                  <1>
 49
                                  <1>
                                                  (sp=32768-args_space-2 at the beginning of execution)
                                            ; argument list offset = 32768-64-128 = 32576 (if it is 128 bytes); 'u' structure offset (for the '/core' dump file) = 32704
 50
                                  <1>
 51
                                  <1>
                                            ; '/core' dump file size = 32768 bytes
 52
 53
                                  <1>
 54
                                  <1>; 08/03/2014
                                                        6C0h ; 256*16 bytes (swap data segment size for 16 processes)
 55
                                  <1> ;sdseamnt equ
 56
                                  <1> ; 19/04/2013 Retro UNIX 8086 v1 feaure only !
 57
                                  <1> ;;sdsegmnt equ
                                                       740h ; swap data segment (for user structures and registers)
 58
                                  <1>
 59
                                  <1>; 30/08/2013
                                  <1> time_count equ 4 ; 10 --> 4 01/02/2014
 60
 61
                                  <1>
                                  <1> ; 05/02/2014
 62
 63
                                  <1> ; process status
 64
                                  <1> ;SFREE
                                                   equ 0
 65
                                  <1> ;SRUN equ 1
 66
                                  <1> ; SWAIT
                                                   equ 2
 67
                                  <1> ;SZOMB
                                                   equ 3
 68
                                  <1> ;SSLEEP
                                                   equ 4 ; Retro UNIX 8086 V1 extension (for sleep and wakeup)
 69
                                  <1>
 70
                                  <1>; 09/03/2015
 71
                                   <1> userdata equ 80000h ; user structure data address for current user ; temporary
 72
                                  <1> swap queue equ 90000h - 2000h; swap queue address; temporary
 73
                                  <1> swap_alloc_table equ 0D0000h ; swap allocation table address ; temporary
 74
 75
                                  <1> ; 17/09/2015
 76
                                  <1> ESPACE equ 48 ; [u.usp] (at 'sysent') - [u.sp] value for error return
 77
                                  <1>
 78
                                  <1> ; 31/12/2017
 79
                                  <1> ; 19/02/2017
                                  <1>; 15/10/2016
 80
                                  <1>; 20/05/2016
 81
                                  <1> ; 19/05/2016
 82
                                  <1> ; 18/05/2016
 83
 84
                                  <1>; 29/04/2016
 85
                                  <1>; TRDOS 386 (TRDOS v2.0) system calls - temporary List
                                  <1>; 14/07/2013 - 21/09/2015 (Retro UNIX 8086 & 386 system calls)
 86
                                  <1> ver equ 0 ; Get TRDOS version (v2.0) <1> exit equ 1
 87
 88
                                  <1> _fork
 89
                                                   equ 2
                                  <1> _read <1> _write
 90
                                                   equ 3
 91
                                                   equ 4
                                  <1> _open equ 5
 92
                                  <1> _close <1> _wait
 93
                                                   equ 6
 94
                                                   equ 7
                                  <1> _creat
 95
                                                   equ 8
 96
                                  <1> _rename
                                                   equ 9 ; TRDOS 386, Rename File (31/12/2017)
                                  equ 10 ; TRDOS 386, Delete File (29/12/2017)
 97
 98
 99
                                  <1> _chdir
                                                   equ 12
                                  <1> _time
<1> _mkdir
100
                                                   equ 13 ; TRDOS 386, Get Sys Date&Time (30/12/2017)
101
                                                   equ 14
                                  <1> _chmod
102
                                                   equ 15; TRDOS 386, Change Attributes (30/12/2017)
                                  <1> _rmdir
<1> _break
                                                   equ 16; TRDOS 386, Remove Directory (29/12/2017)
103
104
                                                   equ 17
                                  <1> _drive
                                                   equ 18 ; TRDOS 386, Get/Set Current Drv (30/12/2017)
105
106
                                  <1> _seek equ 19
                                  107
108
109
                                                  equ 22; TRDOS 386, Change Cmd Prompt (31/12/2017)
                                  <1> _path equ 23 ; TRDOS 386, Get/Set Run Path (31/12/2017)
<1> _env equ 24 ; TRDOS 386, Get/Set Env Vars (31/12/2017)
110
111
                                                  equ 25 ; TRDOS 386, Set Sys Date&Time (30/12/2017)
                                  <1> stime
112
                                  <1> _quit equ 26 <1> _intr equ 27
113
114
                                  <1> _dir equ 28 ; TRDOS 386, Get Curr Drive&Dir (30/12/2017)
115
116
                                  <1> _emt equ 29
                                  117
118
119
                                  <1> _audio
                                               equ 32 ; TRDOS 386 Video Functions (16/05/2016)
                                                   equ 33; TRDOS 386 Timer Functions (18/05/2016)
                                  <1> _timer <1> _sleep
120
121
                                                   equ 34 ; Retro UNIX 8086 v1 feature only !
                                  <1> _msg equ 35 ; Retro UNIX 386 v1 feature only !
122
                                  <1> _geterr equ 36 ; Retro UNIX 386 v1 feature only !
123
124
                                       fpsave equ 37; TRDOS 386 FPU state option (28/02/2017)
                                  <1> _pri equ 38 ; change priority - TRDOS 386 (20/05/2016)
125
                                  <1> _rele equ 39 ; TRDOS 386 (19/05/2016)
126
                                  <1> _fff equ 40 ; Find First File - TRDOS 386 (15/10/2016)
<1> _fnf equ 41 ; Find Next File - TRDOS 386 (15/10/2016)
127
128
129
                                                   equ 42 ; Allocate memory - TRDOS 386 (19/02/2017)
                                                   ; TRDOS 386 (19/02/2017) DMA buff fuctions
130
                                  <1>
131
                                  <1> _dalloc equ 43 ; Deallocate mem - TRDOS 386 (19/02/2017)
                                  <1> calbac equ 44 ; Set IRQ callback - TRDOS 386 (20/02/2017)
132
133
                                  <1> dma equ 45; DMA service - TRDOS 386 (20/08/2017)
134
                                  <1>
                                  <1> %macro sys 1-4
135
                                          ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
136
                                  <1>
                                          ; 03/09/2015
137
                                  <1>
138
                                  <1>
                                          ; 13/04/2015
                                          ; Retro UNIX 386 v1 system call.
139
                                  <1>
140
                                  <1>
                                          %if %0 >= 2
                                              mov ebx, %2
141
                                  <1>
142
                                  <1>
                                               %if %0 >= 3
                                                  mov ecx, %3
143
                                  <1>
                                                   %if %0 = 4
144
                                  <1>
145
                                  <1>
                                                    mov edx, %4
146
                                  <1>
                                                   %endif
147
                                  <1>
                                               %endif
148
                                  <1>
                                          %endif
149
                                  <1>
                                          mov eax, %1
                                          ;int 30h
150
                                  <1>
```

```
int 40h; TRDOS 386 (TRDOS v2.0)
151
152
                                 <1> %endmacro
153
                                 <1>
154
                                 <1> ; TRDOS 386 system calls, interrupt number
155
                                 <1> ; 25/12/2016
156
                                 <1> SYSCALL_INT_NUM equ '40'; '40h'
157
                                 <1> ; 13/05/2015 - ERROR CODES
158
159
                                 <1> ERR_FILE_NOT_OPEN equ 10 ; 'file not open !' error
                                 160
161
                                 <1> ; 14/05/2015
162
                                 <1> ERR DIR ACCESS
                                                        equ 11 ; 'permission denied !' error
                                 <1> ERR_FILE_NOT_FOUND equ 12 ; 'file not found !' error
163
                                 <1> ERR_TOO_MANY_FILES equ 13 ; 'too many open files !' error
164
                                                        equ 14 ; 'directory already exists !' error
165
                                 <1> ERR DIR EXISTS
166
                                 <1> ; 16/05/2015
                                 <1> ERR DRV NOT RDY
167
                                                        equ 15 ; 'drive not ready !' error
                                 <1> ; 18/05/2015
168
169
                                 <1> ERR_DEV_NOT_RDY
                                                        equ 15 ; 'device not ready !' error
                                                        equ 11 ; 'permission denied !' error
                                 <1> ERR DEV ACCESS
170
                                 <1> ERR_DEV_NOT_OPEN equ 10 ; 'device not open !' error
171
                                 <1>; 07/06/2015
172
                                 <1> ERR_FILE_EOF
                                                     equ 16 ; 'end of file !' error
173
174
                                 <1> ERR_DEV_VOL_SIZE equ 16; 'out of volume!' error
                                 <1>; 09/06/2015
175
176
                                 <1> ERR_DRV_READ
                                                     equ 17 ; 'disk read error !'
177
                                 <1> ERR DRV WRITE
                                                           equ 18 ; 'disk write error !'
                                 <1>; 16/06/2015
178
179
                                 <1> ERR_NOT_DIR
                                                     equ 19 ; 'not a (valid) directory !' error
                                 <1> ERR FILE SIZE
                                                          equ 20 ; 'file size error !'
180
181
                                 <1> ; 22/06/2015
182
                                 <1> ERR NOT SUPERUSER equ 11 ; 'permission denied !' error
                                 <1> ERR NOT OWNER
                                                        equ 11 ; 'permission denied !' error
183
                                                        equ 11 ; 'permission denied !' error
184
                                 <1> ERR NOT FILE
185
                                 <1>; 23/06/2015
186
                                 <1> ERR_FILE_EXISTS
                                                        equ 14 ; 'file already exists !' error
                                 <1> ERR DRV NOT SAME equ 21 ; 'not same drive !' error
187
                                 <1> ERR_DIR_NOT_FOUND equ 12 ; 'directory not found !' error
188
189
                                 <1> ERR_NOT_EXECUTABLE equ 22 ; 'not executable file !' error
                                 <1>; 27/06/2015
190
191
                                 <1> ERR_INV_PARAMETER equ 23 ; 'invalid parameter !' error
192
                                 <1> ERR_INV_DEV_NAME         equ 24 ; 'invalid device name !' error
                                 <1>; 29/06/2015
193
194
                                 <1> ERR_TIME_OUT
                                                    equ 25 ; 'time out !' error
195
                                 <1> ERR_DEV_NOT_RESP equ 25 ; 'device not responding !' error
196
                                 <1> ; 10/10/2016
                                 <1> ERR_INV_FILE_NAME equ 26 ; 'invalid file name !' error
197
                                                           equ 23 ; 'invalid flags !' error
198
                                 <1> ERR_INV_FLAGS
                                 <1>; For code compatibility with previous version of TRDOS (2011)
199
                                 <1>; (Temporary error codes for current TRDOS 386 -2016- version)
200
201
                                 <1> ERR_NO_MORE_FILES equ 12 ; 'no more files !' error
                                 <1> ERR_PATH_NOT_FOUND equ 3 ; 'path not found !' error
202
                                                          ; 'dir not found !'; TRDOS 8086
203
                                 <1>
204
                                 <1> ERR_NOT_FOUND:
                                                           equ 2 ; 'file not found !' ; TRDOS 8086
205
                                 <1> ERR_DISK_SPACE
                                                           equ 39; 'out of volume!' TRDOS 8086
206
                                 <1>
                                                           ; 'insufficient disk space !' ; 27h
                                                          equ 30 ; 'disk write protected !' ; 16/10/2016
207
                                 <1> ERR DISK WRITE
                                 <1> ERR_ACCESS_DENIED equ 5 ; 'access denied !' ; TRDOS 8086
208
209
                                 <1> ; 28/02/2017
210
                                 <1> ERR PERM DENIED
                                                           equ 11 ; 'permission denied !' error
211
                                 <1> ; 18/05/2016
212
                                 <1> ERR MISC
                                                     equ 27 ; miscellaneous/other errors
213
                                 <1>; 1\overline{5}/10/2016
214
                                 <1> ; TRDOS 8086 -> TRDOS 386 (0Bh -> 28)
215
                                 <1> ERR_INV_FORMAT
                                                      equ 28 ; 'invalid format !' error
216
                                 <1>; TRDOS 8086 -> TRDOS 386 (0Dh -> 29)
                                 <1> ERR INV DATA equ 29 ; 'invalid data !' error
217
                                 <1>; \overline{\text{TRDOS}}8086 -> \overline{\text{TRDOS}}386 (0Eh -> 20)
218
219
                                 <1> ERR_ZERO_LENGTH
                                                          equ 20 ; 'zero length !' error
220
                                 <1>; TRDOS 8086 -> TRDOS 386 (15h -> 17, 1Dh -> 18, 1Eh -> 17)
221
                                 <1> ERR DRV NR READ
                                                          equ 17 ; 'drive not ready or read error !'
                                 <1> ERR_DRV_NR_WRITE equ 18 ; 'drive not ready or write error !'
<1> ; 15/10/2016
222
223
224
                                 <1> ERR_INV_PATH_NAME equ 19 ; 'bad path name !' error
                                 <1> ERR_BAD_CMD_ARG
<1> ERR_INV_FNUMBER
                                                           equ 1; 'bad command argument!'; TRDOS 8086
225
                                                           equ 1; 'invalid function number !'; TRDOS 8086
226
                                 <1> ERR_BIG_FILE equ 8; 'big file & out of memory !; TRDOS 8086
227
                                 <1> ERR_BIG_DATA equ 8 ; 'big data & out of memory ! ; TRDOS 8086
228
                                                     equ 35 ; 'cluster not available !' ; TRDOS 8086
229
                                  <1> ERR CLUSTER
                                 <1> ERR_OUT_OF_MEMORY equ 4; 'out of memory!'
230
                                                          ; 'insufficient memory !'
231
                                 <1>
                                  <1> ERR_P_VIOLATION
                                                           equ 6 ; 'protection violation !'
232
233
                                 <1> ERR PAGE FAULT
                                                            equ 224 ; 'page fault !' ; 0E0h
234
                                 <1> ERR_SWP_DISK_READ
                                                            equ 40
235
                                 <1> ERR_SWP_DISK_NOT_PRESENT
                                                                equ 41
236
                                 <1> ERR_SWP_SECTOR_NOT_PRESENT equ 42
                                 <1> ERR SWP NO FREE SPACE
237
                                                                 equ 43
                                 <1> ERR_SWP_DISK_WRITE
238
                                                                 equ 44
                                 <1> ERR SWP NO PAGE TO SWAP
239
                                                                 equ 45
                                 <1>; 1\overline{0}/04\overline{/}20\overline{1}7
240
                                 <1> ERR BUFFER
                                                     equ 46 ; 'buffer error !'
241
                                 <1>; 28/08/2017 (20/08/2017)
242
                                 <1> ERR_DMA
                                                           equ -1 ; DMA buffer (allocation/misc.) error!
243
244
                                 <1>
245
                                 <1>; 26/08/2015
246
                                 <1> ; 24/07/2015
                                 <1>; 24/06/2015
247
248
                                 <1> MAX ARG LEN
                                                     equ 256; max. length of sys exec arguments
249
                                  <1>; 01/07/2015
                                 <1> MAX_MSG_LEN
250
                                                     equ 255; max. msg length for 'sysmsg'
2.51
                                 <1>;
252
                                 <1>; 06/10/2016
                                 <1> OPENFILES
253
                                                     equ 10 ; max. number of open files (system)
254
                                 <1> ; 07/10/2016
255
                                 <1> : NUMOFDEVICES
                                                           equ 20 ; max. num of available devices (sys)
```

```
2304
                                  %include 'trdosk0.s'; 04/01/2016
  1
  2
                               <1>; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.0) - DEFINITIONS : trdosk0.s
  3
                               <1> ; ------
                               <1> ; Last Update: 29/02/2016
  4
                               <1> ; ------
                               <1> ; Beginning: 04/01/2016
  6
  7
                               8
                               <1>; Assembler: NASM version 2.11 (trdos386.s)
  9
                               <1>; Derived from TRDOS Operating System v1.0 (8086) source code by Erdogan Tan
 10
 11
                               <1>; TRDOS2.ASM (09/11/2011)
                               12
 13
                               <1>; TRDOS2.ASM (c) 2004-2011 Erdogan TAN [ 17/01/2004 ] Last Update: 09/11/2011
 14
                               <1>;
                               <1> ; Masterboot / Partition Table at Beginning+1BEh
 15
                               <1> ptBootable
 16
                                               equ 0
                               <1> ptBeginHead
 17
                                                 equ 1
                               <1> ptBeginSector equ 2
 18
                               <1> ptBeginCylinder equ 3
 19
                               <1> ptFileSystemID equ 4
 20
                              <1> ptEndHead
 21
                                                 equ 5
 22
                               <1> ptEndSector
                                                 equ 6
                               <1> ptEndCylinder
 23
                                                 equ 7
 24
                               <1> ptStartSector
                                                 equ 8
 25
                               <1> ptSectors
                                                 equ 12
 26
                              <1>
 27
                               <1> ; Boot Sector Parameters at 7C00h
                              <1> DataAreal equ -4
 28
 29
                               <1> DataArea2
                                               equ -2
 30
                               <1> BootStart
                                              equ 0h
                                               equ 03h
                              <1> OemName
 31
 32
                              <1> BytesPerSec
                                              equ OBh
                              <1> SecPerClust
                                              equ ODh
 33
 34
                              <1> ResSectors
                                               equ OEh
 35
                              <1> FATs
                                               equ 10h
                              <1> RootDirEnts
 36
                                              equ 11h
                              <1> Sectors
                                               equ 13h
 37
                                               equ 15h
 38
                              <1> Media
                              <1> FATSecs
 39
                                               equ 16h
 40
                               <1> SecPerTrack equ 18h
                                               equ 1Ah
                              <1> Heads
 41
                                               equ 1Ch
 42
                              <1> Hidden1
                              <1> Hidden2
                                               equ 1Eh
 43
 44
                               <1> HugeSec1
                                               equ 20h
                              <1> HugeSec2
 45
                                               equ 22h
                                              equ 24h
 46
                              <1> DriveNumber
 47
                               <1> Reserved1
                                               equ 25h
 48
                              <1> bootsignature equ 26h
 49
                               <1> VolumeID
                                               equ 27h
 50
                               <1> VolumeLabel
                                              equ 2Bh
                              <1> FileSysType
                                              equ 36h
 51
                                               equ 3Eh
 52
                               <1> Reserved2
                                                                              ; Starting cluster of P2000
 53
                               <1>
 54
                               <1> ; FAT32 BPB Structure
                               <1> FAT32 FAT Size equ 36
 55
 56
                               <1> FAT32_RootFClust equ 44
 57
                               <1> FAT32_FSInfoSec equ 48
                              <1> FAT32 DrvNum equ 64
 58
 59
                               <1> FAT32_BootSig equ 66
                              <1> FAT32_VolID equ 67
<1> FAT32_VolLab equ 71
 60
 61
                               <1> FAT32_FilSysType equ 82
 62
 63
                               <1>
 64
                               <1> ; BIOS Disk Parameters
 65
                               <1> DPDiskNumber equ 0h
                              <1> DPDType equ 1h
 66
 67
                               <1> DPReturn
                                              equ 2h
 68
                              <1> DPHeads
                                              equ 3h
                               <1> DPCylinders equ 4h
 69
 70
                               <1> DPSecPerTrack equ 6h
                              <1> DPDisks
 71
                                              equ 7h
                                               equ 8h
 72
                               <1> DPTableOff
 73
                               <1> DPTableSeg
                                              equ 0Ah
                              <1> DPNumOfSecs equ 0Ch
 74
 75
                               <1>
 76
                               <1>; BIOS INT 13h Extensions (LBA extensions)
 77
                               <1> ; Just After DP Data (DPDiskNumber+)
 78
                               <1> DAP_PacketSize equ 10h ; If extensions present, this byte will be >=10h
                               <1> DAP_Reserved1 equ 11h ; Reserved Byte
 79
                               <1> DAP_NumOfBlocks equ 12h ; Value of this byte must be 0 to 127
                               <1> DAP_Reserved2 equ 13h ; Reserved Byte
 81
 82
                               <1> DAP Destination equ 14h ; Address of Transfer Buffer as SEGMENT:OFFSET
                               <1> DAP_LBA_Address equ 18h ; LBA=(C1*H0+H1)*S0+S1-1
 83
 84
                               <1>
                                                        ; C1= Selected Cylinder Number
 85
                               <1>
                                                        ; H0= Number Of Heads (Maximum Head Number + 1)
 86
                              <1>
                                                        ; H1= Selected Head Number
 87
                               <1>
                                                        ; S0= Maximum Sector Number
                                                        ; S1= Selected Sector Number
                              <1>
 88
 89
                              <1>
                                                        ; QUAD WORD
 90
                               <1>; DAP_Flat_Destination equ 20h; 64 bit address, if value in 4h is FFFF:FFFFh
 91
                              <1>
                                                            ; QUAD WORD (Also, value in 0h must be 18h)
                                                            ; TR-DOS will not use 64 bit Flat Address
 92
                              <1>
 93
                               <1>
                              <1> ; INT 13h Function 48h "Get Enhanced Disk Drive Parameters"
 94
 95
                               <1> ; Just After DP Data (DPDiskNumber+)
 96
                               <1> GetDParams_48h equ 20h ; Word. Data Length, must be 26 (1Ah) for short data.
 97
                               <1> GDP_48h_InfoFlag equ 22h ; Word
                              <1>; Bit 1 = 1 -> The geometry returned in bytes 4-15 is valid.
 98
 99
                               <1> GDP_48h_NumOfPCyls equ 24h; Double Word. Number physical cylinders.
100
                               <1> GDP 48h NumOfPHeads equ 28h; Double Word. Number of physical heads.
101
                              <1> GDP_48h_NumOfPSpT equ 2Ch ; Double word. Num of physical sectors per track.
102
                               <1> GDP_48h_LBA_Sectors equ 30h ; 8 bytes. Number of physical/LBA sectors.
103
                               <1> GDP_48h_BytesPerSec equ 38h; Word. Number of bytes in a sector.
```

```
105
                                  <1> ; TR-DOS Standalone Program Extensions to the DiskParams Block
                                  <1> ; Just After DP Data (DPDiskNumber+)
106
107
                                  <1> TRDP_CurrentSector equ 3Ah ; DX:AX (LBA)
108
                                  <1> TRDP_SectorCount equ 3Eh ; CX (or Counter)
109
                                  <1>
110
                                  <1>
                                  <1> ; DOS Logical Disks
111
112
                                  <1> LD_Name equ 0
                                  <1> LD DiskType equ 1
113
114
                                  <1> LD_PhyDrvNo equ 2
115
                                  <1> LD_FATType equ 3
                                  <1> LD_FSType equ 4
116
                                  <1> LD LBAYes equ 5
117
                                  <1> LD BPB equ 6
118
                                  <1> LD FATBegin equ 96
119
                                  <1> LD ROOTBegin equ 100
120
121
                                  <1> LD_DATABegin equ 104
122
                                  <1> LD_StartSector equ 108
123
                                  <1> LD TotalSectors equ 112
124
                                  <1> LD_FreeSectors equ 116
125
                                  <1> LD_Clusters equ 120
                                  <1> LD_PartitionEntry equ 124
126
127
                                  <1> LD_DParamEntry equ 125
128
                                  <1> LD_MediaChanged equ 126
129
                                  <1> LD_CDirLevel equ 127
130
                                  <1> LD_CurrentDirectory equ 128
131
                                  <1>
132
                                  <1> ; Singlix FS Extensions to DOS Logical Disks
                                  <1> ; 03/01/2010 (LD_BPB compatibility for CHS r/w)
133
134
                                  <1>
135
                                  <1> LD FS Name equ 0
                                  <1> LD_FS_DiskType equ 1
136
137
                                  <1> LD_FS_PhyDrvNo equ 2
138
                                  <1> LD_FS_FATType equ 3
                                  <1> LD FS FSType equ 4
139
                                  <1> LD FS LBAYes equ 5
140
141
                                  <1> LD_FS_BPB equ 6
142
                                  <1> LD_FS_MediaAttrib equ 6
                                  <1> LD_FS_VersionMajor equ 7
143
144
                                  <1> LD_FS_RootDirD equ 8
145
                                  <1> LD FS MATLocation equ 12
                                  <1> LD FS Reserved1 equ 16 ;1 reserved byte
146
147
                                  <1> LD_FS_BytesPerSec equ 17 ; LD_BPB + OBh
148
                                  <1> LD_FS_Reserved2 equ 19 ;2 reserved byte
149
                                  <1> LD_FS_DATLocation equ 20
                                  <1> LD FS DATSectors equ 24
150
                                  <1> LD FS Reserved3 equ 28 ;3 reserved word
151
152
                                  <1> LD_FS_SecPerTrack equ 30 ; LD_BPB + 18h
                                  <1> LD FS NumHeads equ 32
                                                              ; LD_BPB + 1Ah
153
                                  <1> LD FS UnDelDirD equ 34
154
155
                                  <1> LD FS Reserved4 equ 38 ;4 reserved word
                                  <1> LD_FS_VolumeSerial equ 40
156
157
                                  <1> LD_FS_VolumeName equ 44
158
                                  <1> LD_FS_BeginSector equ 108
159
                                  <1> LD_FS_VolumeSize equ 112
                                  <1> LD FS FreeSectors equ 116
160
                                  <1> LD_FS_FirstFreeSector equ 120
161
162
                                  <1> LD_FS_PartitionEntry equ 124
                                  <1> LD FS DParamEntry equ 125
163
164
                                  <1> LD_FS_MediaChanged equ 126
                                  <1> LD FS CDirLevel equ 127
165
                                  <1> LD_FS_CDIR_Converted equ 128
166
167
                                  <1>
                                  <1> ; Valid FAT Types
168
                                  <1> FS_FAT12 equ 1
169
170
                                  <1> FS FAT16 CHS equ 2
171
                                  <1> FS_FAT32_CHS equ 3
172
                                  <1> FS_FAT16_LBA equ 4
                                  <1> FS_FAT32_LBA equ 5
173
174
                                  <1>
175
                                  <1> ; Cursor Location
176
                                  <1> CCCpointer equ 0450h
                                                             ; BIOS data, current cursor column
                                  <1> ; FAT Clusters EOC sign
177
178
                                  <1> FAT12EOC equ 0FFFh
179
                                  <1> FAT16EOC equ 0FFFFh
180
                                  <1> ;FAT32EOC equ OFFFFFFFh ; It is not direct usable for 8086 code
181
                                  <1> ; BAD Cluster
                                  <1> FAT12BADC equ 0FF7h
182
                                  <1> FAT16BADC equ 0FFF7h
183
184
                                  <1> ;FAT32BADC equ 0FFFFFF7h ; It is not direct usable for 8086 code
185
                                  <1>; MS-DOS FAT16 FS (Maximum Possible) Last Cluster Number= 0FFF6h
                                  <1>
186
187
                                  <1> ; TRFS
188
                                  <1>
189
                                  <1> bs_FS_JmpBoot equ 0 ; jmp short bsBootCode
190
                                                     ; db 0EBh, db 3Fh, db 90h
191
                                  <1> bs_FS_Identifier equ 3 ; db 'FS', db 0
192
                                  <1> bs_FS_BytesPerSec equ 6 ; dw 512
                                  <1> bs_FS_MediaAttrib equ 8 ; db 3
193
194
                                  <1> bs_FS_PartitionID equ 9 ; db 0A1h
195
                                  <1> bs_FS_VersionMaj equ 10 ; db 01h
                                  <1> bs_FS_VersionMin equ 11 ; db 0
196
                                  <1> bs FS BeginSector equ 12 ; dd 0
197
                                  <1> bs_FS_VolumeSize equ 16 ; dd 2880
198
199
                                  <1> bs_FS_StartupFD equ 20 ; dd 0
                                  <1> bs FS MATLocation equ 24; dd 1
200
201
                                  <1> bs_FS_RootDirD equ 28 ; dd 8
202
                                  <1> bs_FS_SystemConfFD equ 32 ; dd 0
                                  <1> bs FS SwapFD equ 36; dd 0
203
204
                                  <1> bs_FS_UnDelDirD equ 40 ; dd 0
205
                                  <1> bs FS DriveNumber equ 44 ; db 0
206
                                  <1> bs_FS_LBA_Ready equ 45; db 0
207
                                  <1> bs_FS_MagicWord equ 46
208
                                  <1> bs_FS_SecPerTrack equ 46 ; db 0A1h
```

```
210
                                <1> bs FS OperationSys equ 48 ; db "TR-SINGLIX v1.0b"
                                <1> bs_FS_Terminator equ 64; db 0
211
212
                                <1> bs_FS_BootCode equ 65
213
                                <1>
214
                                <1> FS_MAT_DATLocation equ 12
                                <1> FS MAT DATScount equ 16
215
                                <1> FS_MAT_FreeSectors equ 20
216
217
                                <1> FS_MAT_FirstFreeSector equ 24
                                <1> FS RDT VolumeSerialNo equ 28
218
219
                                <1> FS_RDT_VolumeName equ 64
220
                                <1>
                                <1> ; FAT12 + FAT16 + FAT32
221
222
                                <1> BS_JmpBoot equ 0
223
                                <1> BS OEMName equ 3
224
                                <1> BPB_BytsPerSec equ 11
                                <1> BPB SecPerClust equ 13
225
                                <1> BPB_RsvdSecCnt equ 14
226
                                <1> BPB NumFATs equ 16
227
228
                                <1> BPB RootEntCnt equ 17
229
                                <1> BPB_TotalSec16 equ 19
                                <1> BPB Media equ 21
230
                                <1> BPB_FATSz16 equ 22
231
232
                                <1> BPB_SecPerTrk equ 24
233
                                <1> BPB_NumHeads equ 26
234
                                <1> BPB_HiddSec equ 28
235
                                <1> BPB_TotalSec32 equ 32
236
                                <1>
237
                                <1>; FAT12 and FAT16 only
                                <1> BS_DrvNum equ 36
238
239
                                <1> BS_Reserved1 equ 37
240
                                <1> BS BootSig equ 38
                                <1> BS_VolID equ 39
241
242
                                <1> BS VolLab equ 43
                                <1> BS FilSysType equ 54 ; 8 bytes
243
244
                                <1> BS_BootCode equ 62
245
                                <1>
                                <1> ; FAT32 only
246
                                <1> BPB FATSz32 equ 36 ; FAT32, 4 bytes
247
248
                                <1> BPB ExtFlags equ 40 ; FAT32, 2 bytes
                                <1> BPB_FSVer equ 42 ; FAT32, 2 bytes
249
                                <1> BPB RootClus equ 44 ; FAT32, 4 bytes
250
                                <1> BPB_FSInfo equ 48 ; FAT 32, 2 bytes
251
                                <1> BPB BkBootSec equ 50 ; FAT32, 2 bytes
252
253
                                <1> BPB_Reserved equ 52 ; FAT32, 12 bytes
                                <1> BS_FAT32_DrvNum equ 64 ; FAT32, 1 byte
254
                                <1> BS FAT32 Reserved1 equ 65 ; FAT32, 1 byte
255
                                <1> BS_FAT32_BootSig equ 66 ; FAT32, 1 byte
256
                                <1> BS_FAT32_VolID equ 67 ; FAT32, 4 bytes
<1> BS_FAT32_VolLab equ 71 ; FAT32, 11 bytes
257
258
259
                                <1> BS_FAT32_FilSysType equ 82 ; FAT32, 8 bytes
260
                                <1> BS_FAT32_BootCode equ 90
261
                                <1>
262
                                <1>; 29/02/2016
263
                                <1> ; (FAT32 Free Cluster Count & First Free Cluster values)
264
                                <1> ;[BPB_Reserved] = Free Cluster Count (offset 52)
                                <1> ;[BPB_Reserved+4] = First Free Cluster (offset 56)
265
266
                                <1>
267
                                <1> BS_Validation equ 510
268
                                <1>
269
                                <1> ; 15/02/2016
270
                                <1>; FILE.ASM - 09/10/2011
                                <1> ; Directory Entry Structure
271
272
                                <1>; 29/10/2009 (According to Microsoft FAT32 File System Specification)
273
                                <1> DirEntry_Name equ 0
274
                                <1> DirEntry_Attr equ 11
                                <1> DirEntry_NTRes equ 12
275
                                <1> DirEntry_CrtTimeTenth equ 13
<1> DirEntry_CrtTime equ 14
<1> DirEntry_CrtDate equ 16
276
277
278
                                <1> DirEntry LastAccDate equ 18
279
                                <1> DirEntry_FstClusHI equ 20
<1> DirEntry_WrtTime equ 22
280
281
282
                                <1> DirEntry_WrtDate equ 24
283
                                <1> DirEntry_FstClusLO equ 26
284
                                <1> DirEntry_FileSize equ 28
                                    %include 'trdosk1.s'; 04/01/2016
2305
                                1
  2
                                <1>; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.1) - SYS INIT : trdosk1.s
  3
                                <1> ; ------
                                <1> ; Last Update: 20/01/2018
  4
                                 <1>; ------
  6
                                <1>; Beginning: 04/01/2016
  7
  8
                                <1>; Assembler: NASM version 2.11 (trdos386.s)
                                <1> ; ------
  9
 10
                                <1>; Derived from TRDOS Operating System v1.0 (8086) source code by Erdogan Tan
 11
                                <1>; TRDOS2.ASM (09/11/2011)
                                12
                                <1>; TRDOS2.ASM (c) 2004-2011 Erdogan TAN [ 17/01/2004 ] Last Update: 09/11/2011
 13
                                <1> ;
 14
 15
                                <1>
                                <1> sys_init:
 16
                                         ; 20/01/2018 (v2.0.1)
 17
                                <1>
 18
                                <1>
                                         ; 23/01/2017
                                                       (v2.0.0)
                                          ; 07/05/2016
 19
                                <1>
 20
                                <1>
                                         ; 02/05/2016
 21
                                <1>
                                         ; 24/04/2016
                                         ; 14/04/2016
 22
                                <1>
 23
                                         ; 13/04/2016
                                <1>
 2.4
                                <1>
                                         ; 30/03/2016
 25
                                <1>
                                         ; 24/01/2016
 26
                                <1>
                                         ; 06/01/2016
 27
                                <1>
                                         ; 04/01/2016
 28
                                <1>
```

<1> bs FS Heads equ 47; db 01h

209

```
; 23/01/2017 - reset timer frequency (to 18.2Hz)
 30 000062AE B036
                               <1>
                                        mov
                                              al, 00110110b ; 36h
 31 000062B0 E643
                              <1>
                                        out
                                              43h, al
 32 000062B2 31C0
                              <1>
                                         xor eax, eax ; sub
                                                                 al, al ; 0
 33 000062B4 E640
                              <1>
                                        out
                                              40h, al ; LB
                              <1>
 34 000062B6 E640
                                        out
                                              40h, al ; HB
                               <1>
                                        ; 30/03/2016
 36
                               <1>
 37
                               <1>
                                        ; Clear Logical DOS Disk Description Tables Area
 38
                               <1>
                                        ;xor eax, eax
 39 000062B8 BF00010900
                                        mov edi, Logical_DOSDisks
                               <1>
 40 000062BD B980060000
                                              ecx, 6656/4; 26*256 = 6656 bytes
                               <1>
                                        mov
                                        rep stosd; 1664 times 4 bytes
 41 000062C2 F3AB
                               <1>
                               <1>
 43 000062C4 B83F3A2F00
                               <1>
                                        mov eax, '?:/'
 44 000062C9 A3[87590100]
                               <1>
                                        mov
                                              [Current_Dir_Drv], eax
                                        ; Logical DRV INIT (only for hard disks)
                               <1>
 46
 47 000062CE E813040000
                               <1>
                                         call ldrv_init ; trdosk2.s
                               <1>
 48
 49
                               <1>
                                        ; When floppy_drv_init call is disabled
                                        ; media changed sign is needed
                               <1>
 51
                               <1>
                                        ; for proper drive initialization
                               <1>
                                        mov esi, Logical_DOSDisks
 53 000062D3 BE00010900
                               <1>
 54 000062D8 B001
                               <1>
                                        mov
                                               al, 1 ; Initialization sign (invalid_fd_parameter)
 55 000062DA 83C67E
                              <1>
                                        add esi, LD MediaChanged; Media Change Status = 1 (init needed)
                              <1>
 56 000062DD 8806
                                        mov
                                              [esi], al ; A:
 57 000062DF 81C600010000
                              <1>
                                              esi, 100h
                                        add
                                       mov [esi], al; B:
 58 000062E5 8806
                              <1>
 59
                               <1>
                              <1> _current_drive_bootdisk:
<1> mov dl, [boot dr
 61 000062E7 8A15[525D0000]
                                     mov dl, [boot_drv] ; physical drive number
 62 000062ED 80FAFF
                              <1>
                                         cmp dl, OFFh
 63 000062F0 740A
                              <1>
                                      je short _last_dos_diskno_check
                              <1> _boot_drive_check:
                             <1> cmp dl, 80h
 65 000062F2 80FA80
 66 000062F5 7218
                              <1>
                                        jb
                                              short _current_drive_a
 67 000062F7 80EA7E
                              <1>
                                         sub
                                              dl, 7Eh; C = 2, D = 3
 68 000062FA EB13
                                        jmp short _current_drive_a
                              <1>
                              <1>
                           <1> _last_dos_diskno_check:
<1> _ mov _ dl. [T.ast_f]
                                    mov dl, [Last_DOS_DiskNo]
 71 000062FC 8A15[610D0100]
 72 00006302 80FA02
                                         cmp dl, 2
                              <1>
                                        ja
                                              short _current_drive_c
 73 00006305 7706
                              <1>
                              <1><1><1>
 74 00006307 7406
                                        jе
                                               short _current_drive_a
                                        xor dl, dl; A:
 75 00006309 30D2
 76 0000630B EB02
                              <1>
                                       jmp short _current_drive_a
 77
                               <1>
                              <1> _current_drive_c:
 79 0000630D B202
                                      mov dl, 2 ; C:
                               <1>
                               <1>
 81
                               <1> _current_drive_a:
                                    mov [\overline{drv}], dl
 82 0000630F 8815[535D0000]
                              <1>
 83 00006315 BE[630D0100]
                               <1>
                                        mov esi, msg_CRLF_temp
 84 0000631A E8AE000000
                               <1>
                                        call print_msg
                               <1>
                                      mov dl, [drv]
 86 0000631F 8A15[535D0000]
                              <1>
 87
                               <1> _default_drive_c:
                              <1>
                                    call change_current_drive
 88 00006325 E81A0C0000
 89 0000632A 731C
                              <1>
                                        jnc short _start_mainprog
                               <1>
                              <1> _drv_not_ready_error:
 92 0000632C BE[1E100100]
                              <1> mov esi, msgl_drv_not_ready
 93 00006331 E897000000
                               <1>
                                        call print_msg
                                        ;jmp_end_of_mainprog
 94
                               <1>
                               <1>
                                        ; 20/01/2018
 96
                               <1>
 97 00006336 B202
                               <1>
                                        mov dl, 2
 98 00006338 3815[535D0000]
                              <1>
                                              [drv], dl
                                        cmp
 99 0000633E 736B
                               <1>
                                        jnb short _end_of_mainprog
100 00006340 8815[535D0000]
                               <1>
                                        mov
                                              [drv], dl
101 00006346 EBDD
                               <1>
                                              short default drive c
                                        jmp
102
                               <1>
103
                               <1> _start_mainprog:
                               <1>
                                     ; 07/01/2017
104
105
                               <1>
                                        ; 07/05/2016
                                       ; 02/05/2016
106
                               <1>
                                        ; 24/04/2016
107
                               <1>
                               <1>
                                        ; Retro UNIX 386 v1, 'sys_init' (u0.s)
108
109
                               <1>
                                        ; 23/06/2015
110
                               <1>
                                        ; 02/05/2016
111
                               <1>
                               <1>
                                         ; 24/04/2016
112
113 00006348 66B80100
                               <1>
                                        mov
                                              ax, 1
114 0000634C A2[B3030300]
                               <1>
                                        mov
                                               [u.uno], al
115 00006351 66A3[4E030300]
                               <1>
                                               [mpid], ax
                                        mov
116 00006357 66A3[20000300]
                               <1>
                                        mov
                                              [p.pid], ax
117 0000635D A2[B0000300]
                               <1>
                                        mov
                                               [p.stat], al
118 00006362 C605[A8030300]04
                                              byte [u.quant], time_count ; 07/01/2017
                               <1>
                                        mov
119
                               <1>
120 00006369 A1[C0580100]
                               <1>
                                        mov
                                               eax, [k_page_dir]
121 0000636E A3[B8030300]
                               <1>
                                        mov
                                               [u.pgdir], eax; reset
                               <1>
123 00006373 E85FE8FFFF
                               <1>
                                        call
                                              allocate_page
124 00006378 0F82A3000000
                               <1>
                                         jс
                                               panic
                                              [u.upage], eax; user structure page
125 0000637E A3[B4030300]
                               <1>
                                        mov
126 00006383 A3[C0000300]
                               <1>
                                         mov
                                               [p.upage], eax
                                         call clear_page
127 00006388 E8C4E8FFFF
                               <1>
                               <1>
128
                                         ; 24/08/2015
129
                               <1>
130 0000638D FE0D[5B030300]
                               <1>
                                         dec byte [sysflg] ; FFh = ready for system call
131
                               <1>
                                                           ; 0 = executing a system call
132
                               <1>
                                         ; 13/04/2016
133
                               <1>
                                         ; Clear Environment Variables Page/Area
```

```
134 00006393 BF00300900
                                <1>
                                         mov
                                                edi, Env_Page ; 93000h
135 00006398 B980000000
                                <1>
                                         mov
                                                ecx, Env_Page_Size / 4
                                                                        ; 512/4 (4096/4)
136 0000639D 31C0
                                <1>
                                                eax, eax
                                          xor
137 0000639F F3AB
                                <1>
                                         rep
                                                stosd
138
                                <1>
                                         ; 14/04/2016
                                <1>
140 000063A1 E8F5340000
                                <1>
                                         call mainprog_startup_configuration
141
                                <1>
                                <1>
142 000063A6 E8DA0C0000
                                          call
                                                  dos_prompt
143
                                <1>
                               <1> _end_of_mainprog:
<1> _mov e:
144
145 000063AB BE[630D0100]
                                         mov esi, msg_CRLF_temp
                                          call print_msg
146 000063B0 E818000000
                               <1>
                                         mov esi, mainprog_Version call print_msg
147 000063B5 BE[690D0100]
                               <1>
148 000063BA E80E000000
                               <1>
                                         ; 24/01/2016
                               <1>
150 000063BF 28E4
                                <1>
                                       sub ah, ah
151 000063C1 E859A8FFFF
                                <1>
                                          call
                                               int16h ; call getch
152 000063C6 E939ADFFFF
                                <1>
                                         jmp cpu_reset
153
                                <1>
                                <1> infinitiveloop: jmp short infinitiveloop
154 000063CB EBFE
                                <1>
155
156
                                <1> print_msg:
                                       ; 13/05/2016
157
                                <1>
158
                                <1>
                                         ; 04/01/2016
159
                                <1>
                                         ; 01/07/2015
                                        ; 13/03/2015 (Retro UNIX 386 v1)
160
                                <1>
                                <1>
                                         ; 07/03/2014 (Retro UNIX 8086 v1)
161
                                <1>
                                         ; (Modified registers: EAX, EBX, ECX, EDX, ESI, EDI)
162
163
                                <1>
164 000063CD 8A3D[EE580100]
                                                bh, [ACTIVE PAGE] ; 04/01/2016 (ptty)
                                <1>
                                         mov
                                <1>
                                         ;mov bl, 07h; Black background, light gray forecolor
165
166
                                <1>
167 000063D3 AC
                                <1>
                                         lodsb
                                <1> pmsg1:
168
169 000063D4 56
                                         push esi
                                <1>
                                         ;mov bh, [ACTIVE_PAGE] ; 04/01/2016 (ptty)
                                <1>
170
                                                bl, 07h; Black background, light gray forecolor
171 000063D5 B307
                                <1>
172 000063D7 E8DFB8FFFF
                                         call _write_tty
                               <1>
173 000063DC 5E
                               <1>
                                         pop esi
174 000063DD AC
                                <1>
                                         lodsb
175 000063DE 20C0
                               <1>
                                         and al, al
176 000063E0 75F2
                               <1>
                                               short pmsg1
                                          jnz
177 000063E2 C3
                                <1>
                                         retn
178
                                <1>
179
                                <1> clear_screen:
                                      ; 13/05/2016
180
                                <1>
181
                                <1>
                                         ; 30/01/2016
                                         ; 24/01/2016
182
                                <1>
183
                                <1>
                                         ; 04/01/2016
184 000063E3 0FB61D[EE580100]
                                <1>
                                         movzx ebx, byte [ACTIVE PAGE]; video page number (0 to 7)
185 000063EA 8AA3[335F0000]
                                <1>
                                         mov ah, [ebx+vmode]; default = 03h (80x25 text)
186 000063F0 80FC04
                                <1>
                                               ah, 4
187 000063F3 7205
                                <1>
                                          jb
                                                short cls1
188 000063F5 80FC07
                                <1>
                                          cmp
                                                ah, 7
                                                short vga_clear
189 000063F8 7526
                                <1>
                                         jne
                                <1> cls1:
190
191
                                <1>
                                                bh, bl
                                         ;mov bl, 7
192
                                <1>
193 000063FA 3A25[225F0000]
                                <1>
                                          cmp
                                                ah, [CRT_MODE] ; current video mode ?
194
                                <1>
                                                short cls2 ; yes (current video mode = 3)
                                         ;je
195
                                <1>
                                         ;;call set_mode_3 ; set video mode to 3 (& clear screen)
196
                                <1>
197
                                <1>
                                         ;jmp set_mode_3
198 00006400 0F85BFB8FFFF
                                <1>
                                               set_mode_3
                                          jne
199
                                <1> cls2:
200 00006406 88DF
                                                bh, bl ; video page (0 to 7)
                                <1>
                                         mov
201 00006408 B307
                                <1>
                                                bl, 07h; attribute to be used on blanked line
                                          mov
202 0000640A 28C0
                               <1>
                                               al, al ; 0 = entire window
                                         sub
203 0000640C 6631C9
                                <1>
                                         xor
                                               CX, CX
204 0000640F 66BA4F18
                                <1>
                                         mov
                                                dx, 184Fh
205 00006413 E8FBB5FFFF
                                <1>
                                         call _scroll_up ; 24/01/2016
206
                                <1>
207
                                <1>
                                         ;mov bh, [ACTIVE_PAGE] ; video page number (0 to 7)
208 00006418 6631D2
                                <1>
                                          xor
                                                dx, dx
209 0000641B E831B9FFFF
                                <1>
                                          call _set_cpos ; 24/01/2016
210
                                <1>
                                         ;retn
211
                                <1> vga_clear:
212 00006420 C3
                                <1>
                                         retn
213
                                <1>
214
                                <1> panic:
                                         ; 13/05/2016 (TRDOS 386 = TRDOS v2)
215
                                <1>
216
                                <1>
                                         ; 13/03/2015 (Retro UNIX 386 v1)
                                <1>
                                         ; 07/03/2014 (Retro UNIX 8086 v1)
217
218 00006421 BE[021A0100]
                                          mov esi, panic msg
                                <1>
219 00006426 E8A2FFFFFF
                                         call print_msg
                                <1>
                                <1> key_to_reboot:
220
                                           ; 24/01/2016
221
                                <1>
222 0000642B 28E4
                               <1>
                                           sub ah. ah
223 0000642D E8EDA7FFFF
                               <1>
                                           call
                                                  int16h; call getch
                                          ; wait for a character from the current tty
224
                                <1>
225
                                <1>
226 00006432 B00A
                                <1>
                                               al, OAh
                                               bh, [ptty] ; [ACTIVE_PAGE]
227 00006434 8A3D[EE580100]
                                <1>
                                         mov
                                               bl, 07h; Black background,
228 0000643A B307
                                <1>
                                         mov
                                                     ; light gray forecolor
                                <1>
230 0000643C E87AB8FFFF
                                               write_tty
                                <1>
                                          call
231 00006441 E9BEACFFFF
                                <1>
                                         jmp
                                               cpu_reset
                                <1>
233
                                <1> ctrlbrk:
234
                                <1>
                                       ; 12/11/2015
235
                                <1>
                                         ; 13/03/2015 (Retro UNIX 386 v1)
236
                                <1>
                                         ; 06/12/2013 (Retro UNIX 8086 v1)
237
                                <1>
```

```
238
                                <1>
                                         ; INT 1Bh (control+break) handler
239
                                <1>
240
                                                ; Retro Unix 8086 v1 feature only!
                                <1>
241
                                <1>
242 00006446 66833D[AA030300]00 <1>
                                               word [u.intr], 0
                                          cmp
243 0000644E 7645
                                <1>
                                          jna
                                               short cbrk4
                                <1> cbrk0:
                                      ; 12/11/2015
245
                                <1>
246
                                <1>
                                         ; 06/12/2013
247 00006450 66833D[AC030300]00 <1>
                                         cmp word [u.quit], 0
248 00006458 743B
                                <1>
                                         jz
                                              short cbrk4
249
                                <1>
                                         ; 20/09/2013
250
                               <1>
                                         push ax
251 0000645A 6650
                               <1>
252 0000645C A0[EE580100]
                               <1>
                                         mov al, [ptty]
253
                                <1>
                                         ; 12/11/2015
                                <1>
255
                                <1>
                                         ; ctrl+break (EOT, CTRL+D) from serial port
256
                                <1>
257
                                <1>
                                         ; or ctrl+break from console (pseudo) tty
258
                                <1>
                                         ; (!redirection!)
                                <1>
260 00006461 3C08
                                <1>
                                          cmp al, 8; serial port tty nums > 7
                                         jb
261 00006463 7211
                                <1>
                                                   short cbrk1 ; console (pseudo) tty
                                <1>
262
                                         ;
263
                                <1>
                                         ; Serial port interrupt handler sets [ptty]
264
                                <1>
                                         ; to the port's tty number (as temporary).
                                <1>
265
266
                                <1>
                                         ; If active process is using a stdin or
267
                                <1>
                                         ; stdout redirection (by the shell),
268
                                <1>
                                          ; console tty keyboard must be available
                                         ; to terminate running process,
269
                                <1>
270
                                <1>
                                         ; in order to prevent a deadlock.
271
                                <1>
                                         push edx
272 00006465 52
                                <1>
273 00006466 0FB615[B3030300]
                                <1>
                                          movzx edx, byte [u.uno]
274 0000646D 3A82[7F000300]
                                <1>
                                          cmp al, [edx+p.ttyc-1]; console tty (rw)
275 00006473 5A
                                <1>
                                          pop
                                               edx
276 00006474 7412
                                <1>
                                                short cbrk2
                                          jе
                                <1> cbrk1:
278 00006476 FEC0
                                <1>
                                         inc al ; [u.ttyp] : 1 based tty number
                                         ; 06/12/2013
279
                                <1>
280 00006478 3A05[94030300]
                                <1>
                                          cmp al, [u.ttyp] ; recent open tty (r)
                                          je short cbrk2
281 0000647E 7408
                                <1>
282 00006480 3A05[95030300]
                                <1>
                                          cmp al, [u.ttyp+1]; recent open tty (w)
283 00006486 750B
                                <1>
                                          jne short cbrk3
                                <1> cbrk2:
                                <1> ;; 06/12/2013
285
286
                                <1>
                                         ;mov ax, [u.quit]
                                         ; and ax, ax
287
                                <1>
288
                                <1>
                                         jz short cbrk3;
289
                                <1>
                                         ;
290 00006488 6631C0
                                         xor ax, ax; 0 dec ax
                                <1>
291 0000648B 6648
                                <1>
                                <1>
                                         ; OFFFFh = 'ctrl+brk' keystroke
293 0000648D 66A3[AC030300]
                                <1>
                                         mov [u.quit], ax
                                <1> cbrk3:
295 00006493 6658
                                <1> pop
                                                ax
296
                                <1> cbrk4:
297 00006495 C3
                                <1>
                                        retn
298
                                <1>
299
                                <1>
                                <1> ; 31/12/2017
300
301
                                <1> ; TRDOS 386 - 30/12/2017
302
                                <1> %define get_rtc_date RTC_40
303
                                <1> %define get_rtc_time RTC_20
                                <1> %define set_rtc_date RTC_50
304
                                <1> %define set_rtc_time RTC_30
305
306
                                <1> get_rtc_date_time:
307
                                <1>; Retro UNIX 8086 v1 - UNIX.ASM (01/09/2014)
308
                                <1> ;epoch:
                                      ; 30/12/2017 (TRDOS 386 = TRDOS v2.0)
309
                                <1>
                                         ; 15/03/2015 (Retro UNIX 386 v1 - 32 bit version)
310
                                <1>
                                       ; 09/04/2013 (Retro UNIX 8086 v1 - UNIX.ASM)
311
                                <1>
                                        ; 'epoch' procedure prototype:
312
                                <1>
                                                           UNIXCOPY.ASM, 10/03/2013
313
                                <1>
314
                                <1>
                                         ; 14/11/2012
                                        ; unixboot.asm (boot file configuration)
315
                                <1>
316
                                <1>
                                         ; version of "epoch" procedure in "unixproc.asm"
317
                                         ; 21/7/2012
                                <1>
                                         ; 15/7/2012
318
                                <1>
319
                                <1>
                                          ; 14/7/2012
                                          ; Erdogan Tan - RETRO UNIX v0.1
320
                                <1>
321
                                <1>
                                          ; compute current date and time as UNIX Epoch/Time
322
                                <1>
                                          ; UNIX Epoch: seconds since 1/1/1970 00:00:00
323
                                <1>
324
                                <1>
                                           ; ((Modified registers: EAX, EDX, ECX, EBX))
325
                                <1>
                                <1>
326
327 00006496 E8B0F5FFFF
                                <1>
                                          call get_rtc_time
                                                                   ; Return Current Time
328 0000649B 86E9
                                <1>
                                            xchg
                                                    ch,cl
329 0000649D 66890D[B4550100]
                                <1>
                                            mov [hour], cx
330 000064A4 86F2
                                            xchg dh,dl
                                <1>
331 000064A6 668915[B8550100]
                                <1>
                                            mov [second], dx
332
                                <1>
333 000064AD E80AF6FFFF
                                <1>
                                           call
                                                      get_rtc_date
                                                                         ; Return Current Date
334 000064B2 86E9
                                <1>
                                                      ch,cl
                                            xchq
335 000064B4 66890D[AE550100]
                                <1>
                                            mov [year], cx
336 000064BB 86F2
                                <1>
                                            xchg
                                                  dh,dl
337 000064BD 668915[B0550100]
                                <1>
                                           mov [month], dx
338
                                <1>
339 000064C4 66B93030
                                <1>
                                               cx, 3030h
                                         mov
340
                                <1>
                                          mov al, [hour]; Hour
341 000064C8 A0[B4550100]
                                <1>
                                <1>
                                                ; AL <= BCD number)
342
```

```
343 000064CD D410
                                           db 0D4h,10h
                                                                  ; Undocumented inst. AAM
                                <1>
                                <1>
                                                                   ; AH = AL / 10h
                                                                   ; AL = AL MOD 10h
345
                                <1>
346 000064CF D50A
                                <1>
                                           aad ; AX= AH*10+AL
347 000064D1 A2[B4550100]
                                <1>
                                         mov [hour], al
348 000064D6 A0[B5550100]
                                <1>
                                         mov
                                               al, [hour+1]; Minute
                                                ; AL <= BCD number)
                                <1>
                                                                   ; Undocumented inst. AAM
350 000064DB D410
                                <1>
                                          db 0D4h,10h
351
                                <1>
                                                                   ; AH = AL / 10h
                                                                   ; AL = AL MOD 10h
352
                                <1>
353 000064DD D50A
                                <1>
                                          aad ; AX=AH*10+AL
354 000064DF A2[B6550100]
                                <1>
                                         mov [minute], al
355 000064E4 A0[B8550100]
                                <1>
                                               al, [second] ; Second
                                         mov
                                <1>
                                                ; AL <= BCD number)
                                                                   ; Undocumented inst. AAM
357 000064E9 D410
                                <1>
                                          db 0D4h,10h
358
                                <1>
                                                                   ; AH = AL / 10h
                                                                   ; AL = AL MOD 10h
                                <1>
360 000064EB D50A
                                           aad ; AX = AH * 10 + AL
                                <1>
361 000064ED A2[B8550100]
                                <1>
                                         mov [second], al
362 000064F2 66A1[AE550100]
                                <1>
                                         mov
                                               ax, [year] ; Year (century)
                                         push
363 000064F8 6650
                                <1>
                                                    ax
                                             ; AL <= BCD number)
                                <1>
365 000064FA D410
                                           db 0D4h,10h
                                <1>
                                                                  ; Undocumented inst. AAM
                                <1>
                                                                   ; AH = AL / 10h
                                                                   ; AL = AL MOD 10h
367
                                <1>
368 000064FC D50A
                                <1>
                                           aad ; AX = AH*10+AL
369 000064FE B464
                                <1>
                                         mov ah, 100
370 00006500 F6E4
                                <1>
                                         mul
                                                ah
371 00006502 66A3[AE550100]
                                <1>
                                         mov
                                                [year], ax
372 00006508 6658
                                <1>
                                         pop
                                               ax
373 0000650A 88E0
                                <1>
                                                al, ah
                                <1>
                                                ; AL <= BCD number)
375 0000650C D410
                                <1>
                                                            ; Undocumented inst. AAM
                                              0D4h,10h
376
                                <1>
                                                                   ; AH = AL / 10h
                                                                   ; AL = AL MOD 10h
377
                                <1>
378 0000650E D50A
                                <1>
                                          aad ; AX=AH*10+AL
379 00006510 660105[AE550100]
                                <1>
                                         add [year], ax
380 00006517 A0[B0550100]
                                <1>
                                         mov al, [month]; Month
                                                ; AL <= BCD number)
381
                                <1>
                                           db 0D4h,10h
                                                            ; Undocumented inst. AAM
382 0000651C D410
                                <1>
383
                                <1>
                                                                   ; AH = AL / 10h
                                                                   ; AL = AL MOD 10h
384
                                <1>
385 0000651E D50A
                                           aad ; AX = AH * 10 + AL
                                <1>
386 00006520 A2[B0550100]
                                <1>
                                         mov [month], al
387 00006525 A0[B1550100]
                                <1>
                                         mov al, [month+1]
                                                                         ; Day
388
                                <1>
                                               ; AL <= BCD number)
389 0000652A D410
                                <1>
                                           db 0D4h,10h
                                                                 ; Undocumented inst. AAM
                                                                   ; AH = AL / 10h
                                <1>
390
391
                                <1>
                                                                   ; AL = AL MOD 10h
                                           aad ; AX= AH*10+AL
392 0000652C D50A
                                <1>
393 0000652E A2[B2550100]
                                <1>
                                           mov [day], al
394
                                <1>
395 00006533 C3
                                         retn ; 30/12/2017
                                <1>
396
                                <1>
397
                                <1> epoch:
398 00006534 E85DFFFFFF
                                <1>
                                         call get_rtc_date_time ; TRDOS 386 - 30/12/2017
399
                                <1>
400
                                <1> convert_to_epoch:
401
                                <1>
                                        ; 31/12/2017 (TRDOS 386 = TRDOS v2.0)
                                         ; 15/03/2015 (Retro UNIX 386 v1 - 32 bit modification)
402
                                <1>
403
                                <1>
                                         ; 09/04/2013 (Retro UNIX 8086 v1)
404
                                <1>
                                <1>
                                         ; ((Modified registers: EAX, EDX, EBX))
405
                                <1>
406
                                         ; Derived from DALLAS Semiconductor
407
                                <1>
408
                                <1>
                                         ; Application Note 31 (DS1602/DS1603)
                                <1>
                                         ; 6 May 1998
410 00006539 29C0
                                <1>
                                         sub
                                               eax, eax
411 0000653B 66A1[AE550100]
                                <1>
                                                ax, [year]
                                               ax, 1970
412 00006541 662DB207
                                <1>
                                         sub
413 00006545 BA6D010000
                                <1>
                                         mov
                                                edx, 365
414 0000654A F7E2
                                <1>
                                         mul
                                                edx
415 0000654C 31DB
                                <1>
                                         xor
                                                ebx, ebx
416 0000654E 8A1D[B0550100]
                                <1>
                                         mov
                                                bl, [month]
417 00006554 FECB
                                <1>
                                         dec
                                                bl
418 00006556 D0E3
                                <1>
                                         shl
                                               bl, 1
                                <1>
                                              edx, edx
                                         ;sub
420 00006558 668B93[BA550100]
                                               dx, [EBX+DMonth]
                                <1>
                                         mov
421 0000655F 8A1D[B2550100]
                                <1>
                                          mov
                                                  bl, [day]
422 00006565 FECB
                                <1>
                                         dec bl
423 00006567 01D0
                                <1>
                                         add
                                                eax, edx
424 00006569 01D8
                                <1>
                                         add
                                                eax, ebx
                                                     ; EAX = days since 1/1/1970
                               <1>
426 0000656B 668B15[AE550100]
                                <1>
                                                dx, [year]
                                               dx, 1969
427 00006572 6681EAB107
                                <1>
                                         sub
428 00006577 66D1EA
                                <1>
                                         shr
                                               dx, 1
429 0000657A 66D1EA
                               <1>
                                        shr
                                               dx, 1
                                                ; (year-1969)/4
430
                               <1>
                                       add
431 0000657D 01D0
                                <1>
                                               eax, edx
432
                                                ; + leap days since 1/1/1970
                                <1>
433 0000657F 803D[B0550100]02
                                                byte [month], 2 ; if past february
                                <1>
                                         cmp
434 00006586 7610
                                <1>
                                         jna
                                                short ctel
                                                dx, [year]
435 00006588 668B15[AE550100]
                               <1>
                                         mov
436 0000658F 6683E203
                               <1>
                                                dx, 3 ; year mod 4
437 00006593 7503
                                <1>
                                         jnz
                                               short ctel
                                                 ; and if leap year
438
                               <1>
                                                eax, 1
                                                         ; add this year's leap day (february 29)
439 00006595 83C001
                               <1>
                                         add
                               <1> cte1:
                                                            ; compute seconds since 1/1/1970
440
441 00006598 BA18000000
                               <1>
                                                edx, 24
442 0000659D F7E2
                               <1>
                                         mul
                                               edx
443 0000659F 8A15[B4550100]
                               <1>
                                         mov
                                                dl, [hour]
                                       add
444 000065A5 01D0
                               <1>
                                               eax, edx
                                                ; EAX = hours since 1/1/1970 \ 00:00:00
445
                               <1>
                               <1>
                                               ebx, 60
446
447 000065A7 B33C
                               <1>
                                         mov
                                               bl, 60
```

```
mul
448 000065A9 F7E3
                                  <1>
                                                   ebx
449 000065AB 8A15[B6550100]
                                  <1>
                                            mov
                                                   dl, [minute]
                                            add eax, edx
450 000065B1 01D0
                                  <1>
451
                                  <1>
                                                   ; EAX = minutes since 1/1/1970 00:00:00
452
                                  <1>
                                            ;mov ebx, 60
453 000065B3 F7E3
                                  <1>
                                            mul
                                                   ebx
454 000065B5 8A15[B8550100]
                                  <1>
                                           mov
                                                   dl, [second]
455 000065BB 01D0
                                            add
                                  <1>
                                                   eax, edx
456
                                  <1>
                                                    ; EAX -> seconds since 1/1/1970 00:00:00
457 000065BD C3
                                  <1>
                                             retn
458
                                   <1>
459
                                   <1> ;set date time:
460
                                   <1> convert_from_epoch:
                                        ; 31/12/2017 (v2.0.0)
461
                                   <1>
                                           ; 30/12/2017 (TRDOS 386 = TRDOS v2.0)
462
                                   <1>
                                          ; 15/03/2015 (Retro UNIX 386 v1 - 32 bit version)
; 20/06/2013 (Retro UNIX 3000 1)
463
                                   <1>
                                            ; 20/06/2013 (Retro UNIX 8086 v1)
464
                                          ; 'convert_from_epoch' procedure prototype:
                                   <1>
465
466
                                   <1>
                                                                UNIXCOPY.ASM, 10/03/2013
467
                                   <1>
468
                                   <1>
                                           ; ((Modified registers: EAX, EDX, ECX, EBX))
469
                                   <1>
                                   <1>
                                            ; Derived from DALLAS Semiconductor
470
471
                                   <1>
                                           ; Application Note 31 (DS1602/DS1603)
                                   <1>
                                            ; 6 May 1998
472
473
                                   <1>
474
                                   <1>
                                            ; INPUT:
475
                                            ; EAX = Unix (Epoch) Time
                                   <1>
                                   <1>
477 000065BE 31D2
                                                   edx, edx
                                  <1>
                                            xor
478 000065C0 B93C000000
                                  <1>
                                            mov
                                                   ecx, 60
479 000065C5 F7F1
                                   <1>
                                             div
                                                    ecx
                                                  [imin], eax ; whole minutes
480
                                  <1>
                                            ;mov
481
                                  <1>
                                                            ; since 1/1/1970
482 000065C7 668915[B8550100]
                                 <1>
                                                   [second], dx ; leftover seconds
                                            mov
                                                   edx, edx
483 000065CE 29D2
                                  <1>
                                            sub
484 000065D0 F7F1
                                  <1>
                                            div ecx
485
                                  <1>
                                            ;mov [ihrs], eax ; whole hours
                                                                ; since 1/1/1970
486
                                   <1>
487 000065D2 668915[B6550100] <1>
                                                    [minute], dx ; leftover minutes
                                            mov
488 000065D9 31D2
                                  <1>
                                          xor edx, edx
                                  <1>
                                            ;mov cx, 24
490 000065DB B118
                                  <1>
                                                   cl, 24
                                            mov
491 000065DD F7F1
                                  <1>
                                            div ecx
492
                                  <1>
                                            ;mov [iday], ax ; whole days
                                                               ; since 1/1/1970
493
                                   <1>
                                                   [hour], dx ; leftover hours
494 000065DF 668915[B4550100] <1>
                                        mov
add
                                            mov
495 000065E6 05DB020000
                                                    eax, 365+366; whole day since
                                  <1>
                                                                ; 1/1/1968
496
                                  <1>
497
                                  <1>
                                            ;mov [iday], ax
                                            push eax
498 000065EB 50
                                  <1>
499 000065EC 29D2
                                  <1>
                                            sub
                                                   edx, edx
500 000065EE B9B5050000
                                  <1>
                                             mov
                                                    ecx, (4*365)+1; 4 years = 1461 days
501 000065F3 F7F1
                                  <1>
                                            div
502 000065F5 59
                                  <1>
                                            pop
                                                   ecx
503
                                  <1>
                                                    [lday], ax ; count of quadyrs (4 years)
                                            ; mov
                                            push dx
504 000065F6 6652
                                  <1>
505
                                  <1>
                                            ;mov [qday], dx ; days since quadyr began
506 000065F8 6683FA3C
                                  <1>
                                                   dx, 31 + 29 ; if past feb 29 then
                                            cmp
                                                             ; add this quadyr's leap day
                                  <1>
507 000065FC F5
                                            cmc
                                 <1> adc eax, 0
                                                               ; to # of qadyrs (leap days)
508 000065FD 83D000
                                 510
                                            ;mov cx, [iday]
                                 <1>
<1>
511 00006600 91
                                            xchg ecx, eax ; ECX = lday, EAX = iday
                                            sub
                                                               ; iday - lday
512 00006601 29C8
                                                   eax, ecx
513 00006603 B96D010000
                                  <1>
                                            mov
                                                   ecx, 365
                                           xor edx, edx
514 00006608 31D2
                                  <1>
                                          ; EAX = iday-lday, EDX = 0
515
                                  <1>
516 0000660A F7F1
                                  <1>
                                            div ecx
                                            ;mov [iyrs], ax ; whole years since 1968
517
                                  <1>
518
                                  <1>
                                            ; jday = iday - (iyrs*365) - 1day
                                                                 ; days since 1/1 of current year
                                            ;mov [jday], dx
519
                                   <1>
                                             ;add eax, 1968
                                  <1>
520
521 0000660C 6605B007
                                  <1>
                                            add ax, 1968
                                                                 ; compute year
                                            mov [year], ax
522 00006610 66A3[AE550100]
                                  <1>
523 00006616 6689D1
                                  <1>
                                            mov
                                                   cx, dx
                                  <1>
                                            ;mov dx, [qday]
525 00006619 665A
                                  <1>
                                            pop
                                                   dx
526 0000661B 6681FA6D01
                                  <1>
                                                   dx, 365
                                                                      ; if qday \le 365 and qday \ge 60
                                             cmp
                                             ja
527 00006620 7709
                                                   short cfel ; jday = jday +1
                                  <1>
                                                                ; if past 2/29 and leap year then
528 00006622 6683FA3C
                                             cmp dx, 60
                                  <1>
529 00006626 F5
                                   <1>
                                                                ; add a leap day to the \# of whole
                                               cmc
                                             adc
530 00006627 6683D100
                                  <1>
                                                   cx, 0
                                                                 ; days since 1/1 of current year
                                  <1> cfe1:
                                  <1> ;mov [jday], cx
532
                                  <1>
                                                   bx, 12
                                                                 ; estimate month
533 0000662B 66BB0C00
                                             mov
534 0000662F 66BA6E01
                                 <1>
                                                   dx, 366
                                                                 ; mday, max. days since 1/1 is 365
                                            mov
                                 <1> and ax, 11b ; year mod 4 (and dx, 3)
<1> cfe2: ; Month calculation ; 0 to 11 (11 to 0)
535 00006633 6683E003
                                 <1>
                                 <1> cmp cx, dx ; mday = \# of days passed from 1/1
537 00006637 6639D1
                                                   short cfe3
538 0000663A 731D
                                 <1>
                                             jnb
                                  <1> dec <1> shl
539 0000663C 664B
                                                                ; month = month -1
                                                    bx
540 0000663E 66D1E3

      540
      0000663E
      66D1E3
      <1>
      shl

      541
      00006641
      668B93[BA550100]
      <1>
      mov

      542
      00006648
      66D1EB
      <1>
      shr

      543
      0000664B
      6683FB01
      <1>
      cmp

      544
      0000664F
      76E6
      <1>
      jna

      545
      00006651
      08C0
      <1>
      or

      546
      00006653
      75E2
      <1>
      jnz

      547
      00006655
      6642
      <1>
      inc

      548
      00006657
      EBDE
      <1>
      jmp

                                                   bx, 1
                                                   dx, [EBX+DMonth] ; # elapsed days at 1st of month
                                                            ; bx = month - 1 (0 \text{ to } 11)
                                                   bx, 1
                                                                 ; if month > 2 and year mod 4 = 0
                                                   bx, 1
                                                                ; then mday = mday + 1
                                                   short cfe2
                                                                 ; if past 2/29 and leap year then
                                                    al, al
                                                    short cfe2
                                                                 ; add leap day (to mday)
                                                                 ; mday = mday + 1
                                                   dx
                                                   short cfe2
548 00006657 EBDE
                                 <1>
                                           jmp
549
                                  <1> cfe3:
550 00006659 6643
                                  <1>
                                            inc
                                                    bx
                                                             ; \rightarrow bx = month, 1 to 12
                                                   [month], bx
551 0000665B 66891D[B0550100] <1>
552 00006662 6629D1
                                  <1>
                                                   cx, dx ; day = jday - mday + 1
                                             sub
```

```
553 00006665 6641
                                       inc cx
                               <1>
554 00006667 66890D[B2550100]
                               <1>
                                       mov [day], cx
                               <1>
556
                               <1>
                                        ; eax, ebx, ecx, edx is changed at return
557
                               <1>
                                        ; output ->
558
                               <1>
                                        ; [year], [month], [day], [hour], [minute], [second]
559
                               <1>
560 0000666E C3
                                        retn ; 31/12/2017 (TRDOS 386)
                               <1>
561
                               <1>
562
                               <1> set_rtc_date_time:
                                    ; 31/1\overline{2}/2017 (v2.0.0)
563
                               <1>
                                      ; 30/12/2017 (TRDOS 386)
564
                               <1>
                                       ; 15/03/2015 (Retro UNIX 386 v1 - 32 bit version)
565
                               <1>
                                      ; 20/06/2013 (Retro UNIX 8086 v1)
                               <1>
566
                                    call set_date_bcd
; Set real-time clock date
call set_rtc_date; RTC_50
567 0000666F E80F000000
                              <1>
568
                               <1>
569 00006674 E870F4FFFF
                              <1>
                                     ; Set real-time clock time
570
                               <1>
571 00006679 E832000000
                                       call set_time_bcd
jmp set_rtc_time; RTC_30
                               <1>
572 0000667E E9F7F3FFFF
                              <1>
573
                               <1>
                               <1> ; 31/12/2017
574
575
                              <1> set_date_bcd:
                              <1> mov al, [year+1]
576 00006683 A0[AF550100]
577 00006688 D40A
                                        aam ; ah = al / 10, al = al mod 10
                              <1>
                                             OD5h,10h ; Undocumented inst. AAD ; AL = AH * 10h + AL
578 0000668A D510
                              <1>
                                        db
                                    ; AL = AH ; mov ch, al; century (BCD) mov al, [year]
                              <1>
580 0000668C 88C5
                              <1>
581 0000668E A0[AE550100]
                              <1>
582 00006693 D40A
                                             ; ah = al / 10, al = al mod 10
                              <1>
                                        aam
583 00006695 D510
                              <1>
                                              0D5h,10h
                                                        ; Undocumented inst. AAD
                                                        ; AL = AH * 10h + AL
584
                              <1>
585 00006697 88C1
                                        mov cl, al ; year (BCD)
                              <1>
586 00006699 A0[B0550100]
                              <1>
                                       mov al, [month]
587 0000669E D40A
                               <1>
                                        aam ; ah = al / 10, al = al mod 10
                                              0D5h,10h
588 000066A0 D510
                              <1>
                                        db
                                                         ; Undocumented inst. AAD
                              <1>
                                                        ; AL = AH * 10h + AL
590 000066A2 88C6
                                             dh, al ; month (BCD)
                              <1>
                                        mov
 591 000066A4 A0[B2550100]
                              <1>
                                             al, [day]
                                        mov
592 000066A9 D40A
                              <1>
                                        aam ; ah = al / 10, al = al mod 10
                                        db 0D5h,10h ; Undocumented inst. AAD
593 000066AB D510
                              <1>
594
                              <1>
                                                        ; AL = AH * 10h + AL
                                        mov dh, al ; day (BCD)
595 000066AD 88C6
                              <1>
596 000066AF C3
                                      retn ; 30/12/2017
                               <1>
597
                               <1>
598
                               <1> ; 31/12/2017
599
                               <1> set_time_bcd:
                                      ; Read real-time clock time
600
                               <1>
601
                               <1>
                                        ; (get day light saving time bit status)
                                    cli
CALL UPD_IPR
602 000066B0 FA
                              <1>
                                       ; CHECK FOR UPDATE IN PROCESS; cf = 1 -> al = 0
603 000066B1 E885F5FFFF
                              <1>
604
                              <1>
605 000066B6 7207
                              <1>
                                        jc short stime1
                                     MOV AL, CMOS_REG_B ; ADDRESS ALARM REGISTER CALL CMOS_READ ; READ CURRENT VALUE OF DSE BIT
 606 000066B8 B00B
                              <1>
607 000066BA E897F5FFFF
                              <1>
608
                              <1> stime1:
609 000066BF FB
                              <1> sti
                                        AND AL,00000001B ; MASK FOR VALID DSE BIT MOV DL,AL ; SET [DL] TO ZERO FOR NO
610 000066C0 2401
                              <1>
611 000066C2 88C2
                              <1>
                                                                ; SET [DL] TO ZERO FOR NO DSE BIT
                                        ; DL = 1 or 0 (day light saving time)
612
                              <1>
613
                              <1>
614 000066C4 A0[B4550100]
                              <1>
                                             al, [hour]
                                        mov
                                             ; ah = al / 10, al = al mod 10
615 000066C9 D40A
                              <1>
                                        aam
616 000066CB D510
                              <1>
                                             OD5h,10h ; Undocumented inst. AAD
                               <1>
                                                        ; AL = AH * 10h + AL
617
                                        mov ch, al; hour (BCD)
618 000066CD 88C5
                               <1>
619 000066CF A0[B6550100]
                              <1>
                                        mov al, [minute]
620 000066D4 D40A
                              <1>
                                        aam ; ah = al / 10, al = al \mod 10
                                        db 0D5h,10h ; Undocumented inst. AAD
621 000066D6 D510
                              <1>
                                                         ; AL = AH * 10h + AL
622
                              <1>
                                       mov cl, al ; minute (BCD)
 623 000066D8 88C1
                              <1>
 624 000066DA A0[B8550100]
                                        mov al, [second]
                               <1>
625 000066DF D40A
                                        aam; ah = al / 10, al = al mod 10
                              <1>
                                        db 0D5h,10h ; Undocumented inst. AAD
626 000066E1 D510
                              <1>
                                                       ; AL = AH * 10h + AL
; second (BCD)
627
                               <1>
                                        mov dh, al
628 000066E3 88C6
                               <1>
                                        retn ; 30/12/2017
629 000066E5 C3
                               <1>
                                 %include 'trdosk2.s'; 04/01/2016
2306
                               1
  2
                               <1> ; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.2) - DRV INIT : trdosk2.s
  3
                               <1> ; ------
                               <1>; Last Update: 30/08/2020
  5
                               <1> ; Beginning: 04/01/2016
                               <1> ; ------
  7
                               <1>; Assembler: NASM version 2.14 (trdos386.s)
  8
  9
 10
                               <1>; Derived from TRDOS Operating System v1.0 (8086) source code by Erdogan Tan
 11
                               <1>; TRDOS2.ASM (09/11/2011)
                               12
                               <1>; DRV_INIT.ASM (c) 2009-2011 Erdogan TAN [26/09/2009] Last Update: 07/08/2011
 13
                               <1>;
 14
 15
                               <1>
                               <1> ldrv_init: ; Logical Drive Initialization
 16
                                      ; 30/08/2020
 17
                               <1>
 18
                               <1>
                                        ; 25/08/2020
 19
                               <1>
                                        ; 11/08/2020, 13/08/2020
 20
                               <1>
                                       ; 17/07/2020, 20/07/2020
                                        ; 14/07/2020, 15/07/2020
 21
                               <1>
                                        ; 30/01/2018
 22
                               <1>
 2.3
                               <1>
                                       ; 27/12/2017
 24
                               <1>
                                        ; 12/02/2016
                                        ; 06/01/2016
 25
                               <1>
                                              ('diskinit.inc', 'diskio.inc' integration)
                               <1>
 26
 27
                               <1>
                                        ; 04/01/2016 (TRDOS 386 = TRDOS v2.0)
```

```
; 07/08/2011
 28
                                <1>
 29
                                <1>
                                         ; 20/09/2009
                                         ; 2005
 30
                                <1>
 31
                                <1>
 32
                                <1>
                                         ; 15/07/2020
 33
                                <1>
                                         ;movzx ecx, byte [HF_NUM] ; number of fixed disks
                                <1>
                                         ;cmp cl, 1
                                         ;jnb short load_hd_partition_tables
 35
                                <1>
 36
                                <1>
 37 000066E6 A0[5C590100]
                               <1>
                                         mov
                                               al, [HF_NUM] ; number of fixed disks
 38 000066EB 20C0
                                <1>
                                         and al, al
 39 000066ED 7501
                                               short load hd partition tables
                                <1>
                                         jnz
 40
                                <1>
                                         ; no any hard disks
 41
                               <1>
 42 000066EF C3
                                <1>
                                         retn
 43
                                <1>
                                <1> load hd partition tables:
 45
                                      ;mov esi, [HDPM_TBL_VEC] ; primary master disk FDPT
                                <1>
 46
                               <1>
                                         ; 15/07/2020
 47 000066F0 BE[60590100]
                               <1>
                                         mov esi, HDPM_TBL_VEC
                                       mov edi, PTable_hd0
mov dl, 80h
 48 000066F5 BF[865D0100]
                               <1>
 49 000066FA B280
                                <1>
                                         ; 15/07/2020
                               <1>
 50
 51 000066FC A2[555D0000]
                               <1>
                                         mov [hdc], al
                               <1>
                                         ;xor ecx, ecx; 0
 52
 53
                                <1> load_next_hd_partition_table:
                               <1> ; 20/07/2020
 55 00006701 31C9
                               <1>
                                         xor ecx, ecx; 0
                                <1>
                                         ;push ecx
                                         push edi; *
 57 00006703 57
                               <1>
                                         ;push esi ; FDPT (+ DPTE) address
 58
                               <1>
                                         ; 15/07/2020
 59
                                <1>
 60 00006704 AD
                                         lodsd
                               <1>
 61 00006705 56
                               <1>
                                         push esi; **; next FDPT (+ DPTE) address ptr
                                <1>
 62
 63
                                <1>
                                         ;mov al, [esi+20] ; DPTE offset 4
                                         ; and al, 40h; LBA bit (bit 6)
 64
                                <1>
 65
                                <1>
                                         ;;shr al, 6
                                         ;mov [HD LBA yes], al
 66
                                <1>
 67
                                <1>
                                         ; 15/07/2020
 68
                                <1>
 69 00006706 8A4814
                                <1>
                                         mov cl, [eax+20]
                                              cl, 40h
 70 00006709 80E140
                                <1>
                                         and
 71 0000670C 880D[8A5E0100]
                                <1>
                                         mov [HD_LBA_yes], cl
                                <1>
 73 00006712 E844040000
                                <1>
                                         call load_masterboot
 74
                                <1>
                                         ;jc short pass_pt_this_hard_disk
 75
                                         ; 13/08/2020
                                <1>
 76 00006717 0F828A000000
                                <1>
                                         jс
                                               pass_pt_this_hard_disk
                                <1>
 78 0000671D BB[445D0100]
                                <1>
                                         mov ebx, PartitionTable
 79 00006722 89DE
                                <1>
                                         mov
                                               esi, ebx
                                         ;mov ecx, 16
 80
                                <1>
 81 00006724 B110
                               <1>
                                         mov cl, 16
 82 00006726 F3A5
                               <1>
                                         rep movsd
mov esi, ebx
 83 00006728 89DE
                               <1>
                                         ;mov byte [hdc], 4 ; 4 - partition index
                                <1>
                                         ; 15/07/2020
                                <1>
 86 0000672A C605[8B5E0100]04
                               <1>
                                         mov byte [PP_Counter], 4
                                <1> loc_validate_hdp_partition:
 87
 88
                                <1>
                                       ;cmp byte [esi+ptFileSystemID], 0
                                         ;jna short loc_validate_next_hdp_partition2
 89
                                <1>
                                         ; 13/08/2020
 90
                                <1>
 91 00006731 8A4604
                               <1>
                                         mov al, [esi+ptFileSystemID]
                                         and
 92 00006734 20C0
                               <1>
                                               al, al
 93 00006736 7457
                               <1>
                                               short loc_validate_next_hdp_partition2
                                         jг
                               <1>
 95 00006738 56
                                         push esi ; *** ; Masterboot partition table offset
                               <1>
                                         push edx ; **** ; dl = Physical drive number
 96 00006739 52
                               <1>
 97
                               <1>
 98
                               <1>
                                         ; 13/08/2020
                                         cmp al, 05h ; Extended partition CHS
 99 0000673A 3C05
                                <1>
100 0000673C 7404
                                                short loc_set_ep_counter
                               <1>
                                         jе
101 0000673E 3C0F
                               <1>
                                         cmp al, OFh ; Extended partition LBA
                                         jne short loc_validate_next_hdp_partition0
102 00006740 7511
                                <1>
103
                                <1>
104
                                <1>
                                         ;;inc byte [PP_Counter]
105
                                <1>
                                         ; 15/07/2020
                                         ;inc byte [EP Counter] ; disk has valid partition(s)
106
                                <1>
107
                                <1>
                                <1> loc_set_ep_counter:
108
109
                                <1>
                                      ; 13/08/2020
110 00006742 803D[8C5E0100]80
                              <1>
                                                byte [EP_Counter], 80h
                                         cmp
                                                short loc_validate_next_hdp_partition1
111 00006749 7342
                                <1>
                                         jnb
112
                                <1>
113 0000674B 8815[8C5E0100]
                                <1>
                                         mov
                                                byte [EP_Counter], dl ; disk drv has extd. part.
                               <1>
115 00006751 EB3A
                                                short loc_validate_next_hdp_partition1
                               <1>
                                         jmp
116
                               <1>
                               <1> loc_validate_next_hdp_partition0:
117
118 00006753 31FF
                               <1>
                                         xor edi, edi; 0
119
                               <1>
                                         ; Input -> ESI = PartitionTable offset
                                         ; DL = Hard disk drive number
120
                               <1>
121
                               <1>
                                         ; EDI = 0 -> Primary Partition
                                         ; EDI > 0 -> Extended Partition's Start Sector
122
                               <1>
123 00006755 E885010000
                                         call validate hd fat partition
                               <1>
                                         jnc short loc set valid hdp partition entry
124 0000675A 730E
                               <1>
125
                               <1>
126
                               <1>
                                         ;pop edx
127
                                         ;push edx
                               <1>
128 0000675C 8B1424
                               <1>
                                         mov
                                               edx, [esp] ; ****
129 0000675F 8B742404
                               <1>
                                               esi, [esp+4]; ***; 30/01/2018
                                         mov
                                         call validate_hd_fs_partition
130 00006763 E8D1020000
                               <1>
131 00006768 7223
                               <1>
                                         jc short loc_validate_next_hdp_partition1
                               <1> loc_set_valid_hdp_partition_entry:
132
```

```
mov cl, [Last_DOS_DiskNo] add cl, 'A'
133 0000676A 8A0D[610D0100]
                                <1>
134 00006770 80C141
                                 <1>
                                          ; ESI = Logical dos drive description table address
135
                                <1>
136 00006773 880E
                                <1>
                                          mov [esi+LD_Name], cl
                                <1>
                                          ; 15/07/2020
137
138 00006775 8A4602
                                <1>
                                           mov al, [esi+LD_PhyDrvNo] ; Physical drive number
                                          ;mov al, [esp] ; ****
139
                                <1>
140 00006778 2C7F
                                <1>
                                           sub al, 7Fh
141
                                 <1>
                                                 ; AL = 1 to 4
142 0000677A C0E002
                                           shl al, 2; AL = 4 to 16
                                 <1>
143
                                 <1>
144 0000677D 8A15[8B5E0100]
                                 <1>
                                                dl, [PP_Counter]
                                          mov
                                 <1>
145
146
                                 <1>
                                           ; sub al, [PP_Counter]
                                           sub al, dl; [PP_Counter]; 4 - partition index
147 00006783 28D0
                                 <1>
148
                                 <1>
                                 <1>
                                           ; AL = Partition entry/index, 0 based
                                           ; 0 \rightarrow hd 0, Partition Table offset = 0
150
                                 <1>
151
                                 <1>
                                           ; 15 \rightarrow hd 3, Partition Table offset = 3
152
                                 <1>
153
                                 <1>
                                          ;mov [esi+LD_PartitionEntry], al
                                 <1>
                                           ; 15/07/2020
155
                                 <1>
156 00006785 B404
                                 <1>
                                           mov ah, 4
157
                                 <1>
                                           ; sub ah, [PP_Counter]
158 00006787 28D4
                                 <1>
                                           sub ah, dl
159
                                 <1>
160
                                 <1>
                                           ; AH = Primary partition index, 0 to 3 ; pt entry
                                 <1>
                                                        (4 to 7 for logical disk partitions)
161
                                 <1>
162
163
                                 <1>
                                           ;mov [esi+LD_DParamEntry], ah
164 00006789 6689467C
                                           mov [esi+LD_PartitionEntry], ax
                                 <1>
                                 <1>
165
166
                                 <1> loc_validate_next_hdp_partition1:
                                          pop edx; **** ; dl = Physical drive number
pop esi; *** ; Masterboot partition table offset
167 0000678D 5A
                                <1>
168 0000678E 5E
                                <1>
169
                                 <1>
170
                                 <1> loc_validate_next_hdp_partition2:
                                        ; ESI = PartitionTable offset
171
                                 <1>
                                          ; DL = Hard/Fixed disk drive number
172
                                 <1>
173
                                 <1>
                                          ;dec byte [hdc] ; 4 - partition index
;jz short pass_pt_this_hard_disk
174
                                 <1>
175
                                 <1>
                                           ; 15/07/2020
                                 <1>
177 0000678F FE0D[8B5E0100]
                                 <1>
                                           dec byte [PP_Counter] ; 4 - partition index
                                                 short pass_pt_this_hard_disk
178 00006795 7410
                                 <1>
                                           jz
                                 <1>
180 00006797 83C610
                                          add esi, 16 ; 10h
                                 <1>
181 0000679A EB95
                                 <1>
                                          jmp
                                                 short loc_validate_hdp_partition
182
                                 <1>
                                 <1> loc_not_any_extd_partitions:
183
184
                                 <1>
                                         ; 15/07/2020
185 0000679C C3
                                <1>
                                           retn
                                <1>
186
                                <1> loc_next_hd_partition_table:
187
                                        inc dl
188 0000679D FEC2
                                <1>
                                <1>
                                           ;add esi, 32 ; next FDPT address
                                <1>
190
191 0000679F 83C740
                                 <1>
                                           add edi, 64; next partition table destination
192 000067A2 E95AFFFFFF
                                <1>
                                           jmp load_next_hd_partition_table
193
                                 <1>
194
                                 <1> pass_pt_this_hard_disk:
195
                                          ;pop esi ; FDPT (+ DPTE) address
                                 <1>
                                 <1>
                                           ; 15/07/2020
                                           pop esi ; ** ; next FDPT (+ DPTE) address ptr
197 000067A7 5E
                                 <1>
198 000067A8 5F
                                 <1>
                                                 edi ; * ; Ptable_hd?
                                           pop
                                           ;pop ecx
                                 <1>
200
                                           ;loop loc_next_hd_partition_table
                                 <1>
                                           dec byte [hdc]
jnz short loc_next_hd_partition_table
201 000067A9 FE0D[555D0000]
                                 <1>
202 000067AF 75EC
                                 <1>
203
                                 <1>
                                           ;cmp byte [PP_Counter], 1
;jnb short load_extended_dos_partitions
204
                                 <1>
205
                                 <1>
                                           ;; Empty partition table
206
                                 <1>
207
                                 <1>
                                           ;retn
208
                                 <1>
209
                                 <1>
                                          ; 11/08/2020
210
                                 <1>
                                          ; 17/07/2020
                                 <1> check extended partitions:
211
212
                                          ; 15/07/2020
                                 <1>
                                           ;cmp byte [EP_Counter], 0
213
                                 <1>
214
                                 <1>
                                           ;jna short loc_not_any_extd_partitions
                                           : 13/08/2020
                                <1>
216 000067B1 A0[8C5E0100]
                                 <1>
                                           mov al, [EP_Counter] ; 1st disk drv has extd partition
217 000067B6 08C0
                                 <1>
                                                 al, al ; 0 ?
                                           or
                                                 short loc not any extd partitions
218 000067B8 74E2
                                 <1>
                                           jz
                                <1>
                                 <1> load extended dos partitions:
220
                                           ;mov byte [hdc], 80h
221
                                 <1>
                                           ; 13/08/2020
222
                                <1>
223 000067BA A2[555D0000]
                                <1>
                                           mov byte [hdc], al; 1st disk drv has extd partition
224
                                 <1>
                                          ; 25/08/2020
225 000067BF 2C80
                                <1>
                                           sub al, 80h
226 000067C1 740E
                                <1>
                                                 short loc_set_ext_ptable_hd0
                                         jz
227 000067C3 C0E006
                                <1>
                                          shl al, 6; * 64
228 000067C6 0FB6F0
                                <1>
                                          movzx esi, al
229 000067C9 81C6[865D0100]
                                <1>
                                          add esi, PTable hd0
                                <1>
                                          jmp short next_hd_extd_partition
230 000067CF EB05
231
                                 <1>
                                          ; 25/08/2020
232
                                 <1>
233
                                 <1> loc_set_ext_ptable_hd0:
234 000067D1 BE[865D0100]
                                 <1>
                                          mov esi, PTable_hd0
235
                                 <1>
                                 <1> next hd extd partition:
236
237
                                         ; 17/07/2020
                                 <1>
```

```
;mov byte [EP Counter], 0 ; Reset for each physical disk
238
                                 <1>
239
                                 <1>
                                           ; 13/08/2020
240
                                 <1>
                                           ;mov byte [LD_Counter], 0 ; Reset logical drive index
241 000067D6 66C705[8C5E0100]00- <1>
                                                  word [EP_Counter], 0 ; Reset EP index and LD index
241 000067DE 00
                                 <1>
242
                                 <1>
243 000067DF 56
                                           push esi; ****; PTable hd? offset
                                 <1>
                                 <1>
244
245 000067E0 C605[8B5E0100]04
                                 <1>
                                                 byte [PP_Counter], 4
246
                                 <1>
                                                              ; set for each extd partition table
247
                                 <1>
                                           ;;mov ecx, 4
248
                                 <1>
                                           ;mov cl, 4
                                                dl, [hdc]
249 000067E7 8A15[555D0000]
                                 <1>
                                           mov
                                 <1> hd_check_fs_id_05h:
250
251 000067ED 8A4604
                                 <1>
                                           mov al, [esi+ptFileSystemID]
252 000067F0 3C05
                                 <1>
                                                 al, 05h; Is it an extended dos partition?
                                           cmp
                                                 short loc_set_ep_start_sector ; yes
253 000067F2 7411
                                 <1>
                                           jе
                                 <1> hd check_fs_id_0Fh:
254
255 000067F4 3C0F
                                 <1>
                                                 al, OFh; Is it an extended win4 (LBA mode) partition?
                                           cmp
256 000067F6 740D
                                 <1>
                                                 short loc_set_ep_start_sector ; yes
                                           jе
257
                                 <1>
                                 <1> continue_to_check_ep:
<1> ;add esi, 16
258
259
260
                                 <1>
                                           ;loop hd_check_fs_id_05h
                                           ; 15/07/2020
261
                                 <1>
262
                                 <1>
                                           ;dec cl
                                                 short continue_check_ep_next_disk
263
                                 <1>
                                           ; j z
264 000067F8 FE0D[8B5E0100]
                                                 byte [PP_Counter] ; 4 --> 0
                                 <1>
                                           dec
265 000067FE 742D
                                 <1>
                                           jг
                                                  short continue_check_ep_next_disk
266 00006800 83C610
                                 <1>
                                                 esi, 16
                                           add
267 00006803 EBE8
                                 <1>
                                                 short hd_check_fs_id_05h
                                           jmp
268
                                 <1>
                                 <1> loc_set_ep_start_sector:
269
270
                                 <1>
                                         ; dl = [hdc] ; Drive number
271
                                 <1>
                                           ; 15/07/2020
272 00006805 8B4E08
                                 <1>
                                           mov ecx, [esi+ptStartSector]
                                           ; 30/08/2020
                                 <1>
274 00006808 890D[8E5E0100]
                                           mov [MBR_EP_StartSector], ecx
                                 <1>
275
                                 <1>
                                           ; 20/07/2020
                                 <1> loc_validate_hde_partition_next:
276
277 0000680E 890D[925E0100]
                                 <1>
                                         mov [EP_StartSector], ecx ; Extended partition's start sector
278 00006814 BB[865B0100]
                                 <1>
                                           mov ebx, MasterBootBuff
279 00006819 803D[8A5E0100]01
                                           cmp byte [HD_LBA_yes], 1 ; LBA ready = Yes
                                 <1>
280 00006820 7227
                                 <1>
                                                  short loc_hd_load_ep_05h; cf = 1; 20/07/2020
                                           ; 11/08/2020
281
                                 <1>
282
                                 <1>
                                           ; (BugFix for extended partition type 05h beyond CHS limit)
                                           ; (Infact if extended partition starts at the beyond of CHS limit,
283
                                 <1>
284
                                 <1>
                                           ; it's partition ID must be 0Fh but they/somebodies had used 05h.)
285
                                 <1>
                                           ;cmp al, 05h
286
                                 <1>
                                           ;je short loc_hd_load_ep_05h
287
                                 <1> loc_hd_load_ep_0Fh:
288
                                 <1>
                                          ; 04/01/2016
289
                                 <1>
                                           ;push ecx
290
                                 <1>
                                           ; 15/07/2020
291
                                 <1>
                                           ;mov ecx, [esi+ptStartSector] ; sector number
292
                                 <1>
                                           ;mov ebx, MasterBootBuff ; buffer address
293
                                 <1>
                                           ; LBA read/write (with private LBA function)
                                 <1>
294
                                           ;((Retro UNIX 386 v1 - DISK I/O code by Erdogan Tan))
295
                                 <1>
                                           ; dl = physical drive number (0,1, 80h, 81h, 82h, 83h)
                                           ;mov ah, 1Bh ; LBA read
296
                                 <1>
297
                                 <1>
                                           ;mov al, 1 ; sector count
298 00006822 66B8011B
                                 <1>
                                           mov
                                                 ax, 1B01h
                                           call int13h
299 00006826 E802DAFFFF
                                 <1>
300
                                 <1>
                                           ;pop ecx
301
                                 <1>
                                           ;jnc short loc_hd_move_ep_table
302
                                 <1>
                                           ; 15/07/2020
                                           jnc short loc_validate_hde_partition
303 0000682B 732E
                                 <1>
304
                                 <1>
305
                                 <1> continue_check_ep_next_disk:
306
                                           ; 15/07/2020
                                 <1>
307
                                 <1>
                                           ;pop edi ; PTable_ep?
                                                 esi; ****; PTable hd?
308 0000682D 5E
                                 <1>
                                           pop
309 0000682E A0[5C590100]
                                                  al, [HF_NUM] ; number of hard disks
                                 <1>
                                           mov
310 00006833 047F
                                 <1>
                                                 al, 7Fh
                                 <1>
311 00006835 3805[555D0000]
                                           cmp
                                                 [hdc], al
312 0000683B 730B
                                 <1>
                                           jnb
                                                  short loc_validating_hd_partitions_ok
313 0000683D 83C640
                                 <1>
                                           add
                                                  esi, 64
314
                                 <1>
                                           ; 15/07/2020
                                 <1>
315
                                           ;add edi, 64
316 00006840 FE05[555D0000]
                                 <1>
                                                 byte [hdc]
                                           inc
317 00006846 EB8E
                                 <1>
                                                 short next_hd_extd_partition
                                 <1>
                                 <1> loc_validating hd partitions_ok:
319
320
                                          ; 15/07/<del>2</del>02<del>0</del>
321
                                           ;mov al, [Last_DOS_DiskNo]
                                 <1>
                                 <1> loc_drv_init_retn:
322
323 00006848 C3
                                 <1>
                                          retn
324
                                 <1>
                                 <1> loc hd load ep 05h:
325
                                          \frac{1}{20007}; 20/07/2020 ('diskio.s', int13h, cf = 1 -> bugfix)
326
                                 <1>
327
                                 <1>
                                           ;clc ; (Bug: int13h would not clear carry flag bit,
328
                                 <1>
                                                 ; even if there would not be an error)
                                                 ; ((Fix: now, int13h procedure clears carry flag
                                 <1>
329
330
                                 <1>
                                                  ; at the entrance of it.. 20/07/2020))
331
                                 <1>
                                           ; 15/07/2020
                                           ;push ecx
332
                                 <1>
333 00006849 8A7601
                                 <1>
                                           mov dh, [esi+ptBeginHead]
334 0000684C 668B4E02
                                 <1>
                                           mov cx, [esi+ptBeginSector]
335 00006850 66B80102
                                           mov ax, 0201h; Read 1 sector
                                 <1>
                                           ;mov ebx, MasterBootBuff
                                 <1>
337 00006854 E8D4D9FFFF
                                 <1>
                                           call int13h; 20/07/2020
338
                                 <1>
                                                        ; 'diskio.s' modification, 'clc'
339
                                 <1>
                                           ;pop ecx
                                           jс
                                                  short continue check ep next disk
340 00006859 72D2
                                 <1>
                                 <1>
                                           ; 15/07/2020
341
```

```
342
                                 <1>
                                           ;jmp short loc_validate_hde_partition
343
                                 <1>
344
                                 <1>
                                          ; 15/07/2020
345
                                 <1> ;loc_hd_move_ep_table:
                                        ;;pop edi
346
                                 <1>
347
                                 <1>
                                          ;;push edi ; PTable_ep?
                                         ;mov edi, [esp]
348
                                 <1>
                                          ;movesi, PartitionTable ; Extended
349
                                 <1>
350
                                 <1>
                                          ;mov ebx, esi
                                          ;;mov ecx, 16
351
                                 <1>
352
                                 <1>
                                        ;mov cl, 16
353
                                 <1>
                                                ;rep movsd
                                          ;mov esi, ebx
354
                                 <1>
355
                                 <1> ;loc_set_hde_sub_partition_count:
356
                                 <1>
                                        ;mov byte [PP_Counter], 4
357
                                 <1>
                                          ;mov byte [EP_Counter], 0
358
                                 <1>
                                 <1> loc_validate_hde_partition:
359
                                        ; 13/0<del>8</del>/20<del>2</del>0
360
                                 <1>
                                          ; 15/07/2020
361
                                 <1>
                                        ;mov byte [PP_Counter], 4
mov esi, PartitionTable ; (in MasterBootBuff)
362
                                 <1>
363 0000685B BE[445D0100]
                                 <1>
                                          ; 13/08/2020
364
                                 <1>
                                          ;jmp short get_minidisk_partition_entry
365
                                 <1>
366
                                 <1>
367
                                 <1> ;get_minidisk_partition_entry:
                                        ; 20/07/2020
368
                                 <1>;
                                          cmp byte [esi+ptFileSystemID], 0
                                 <1>;
369
370
                                 <1>;
                                          jа
                                                short loc_validate_minidisk_partition
371
                                 <1>;
                                          ; 13/08/2020
372
                                 <1>;
                                          jmp short continue_check_ep_next_disk
373
                                 <1>
374
                                 <1>;
                                          ; 11/08/2020
375
                                 <1> ;get_minidisk_partition_entry_next:
                                        ; 13/08/2020
                                 <1>;
376
                                          ; dec byte [PP_Counter]
377
                                 <1>;
                                          ;jz short continue_check_ep_next_disk
378
                                 <1>;
379
                                       ; 20/07/2020
                                 <1>;
380
                                 <1> ;;get_minidisk_partition_entry_next:
381
                                         ; 13/08/2020
                                 <1>;
                                          cmp esi, PartitionTable+64
382
                                 <1>;
                                                short continue check ep next disk
383
                                 <1>;
                                          jnb
                                 <1>;
384
385
                                 <1>;
                                          add esi, 16 ; 10h
                                 <1>;
386
                                          ;jmp short get_minidisk_partition_entry
387
                                 <1>
                                          ; 13/08/2020
388
                                 <1>
                                 <1> get_minidisk_partition_entry:
389
                                          ; 20/07/2020
390
                                <1>
391 00006860 807E0400
                                <1>
                                          cmp byte [esi+ptFileSystemID], 0
392 00006864 76C7
                                 <1>
                                          jna short continue_check_ep_next_disk ; 13/08/2020
393
                                 <1>
394
                                 <1> loc_validate_minidisk_partition:
                                         ; 13/0<del>8</del>/2020
395
                                 <1>
396
                                 <1>
                                          ; 20/07/2020
                                          ;push esi ; *** ; Extended partition table offset
397
                                 <1>
398
                                 <1>
                                          ; 13/08/2020
399
                                 <1>
400 00006866 FE05[8C5E0100]
                                 <1>
                                          inc byte [EP_Counter] ; current (sub partition) index
                                                                ; in current extended partition
401
                                 <1>
402
                                 <1>
403 0000686C BF[925E0100]
                                 <1>
                                          mov
                                                edi, EP_StartSector
404
                                 <1>
405
                                 <1>
                                          ; Input -> ESI = PartitionTable offset
406
                                          ; DL = Hard disk drive number
                                 <1>
407
                                 <1>
                                          ; EDI = Extended partition start sector pointer
408 00006871 E869000000
                                 <1>
                                          call validate_hd_fat_partition
409
                                 <1>
                                          ;pop ecx ; *
410 00006876 7308
                                 <1>
                                                short loc_set_valid_hde_partition_entry
                                          jnc
411
                                 <1>
                                                        ; jump down to deep !!!
412
                                 <1>
413
                                 <1>
                                          ;pop esi; ***; Extended partition table offset
                                           ; 13/08/2020
414
                                 <1>
415
                                 <1>
                                          ;mov esi, PartitionTable
                                 <1>
416
                                          ; 11/08/2020
417
                                 <1>
418
                                 <1>
                                          ; ESI = Extended partition table offset
419 00006878 8A15[555D0000]
                                 <1>
                                          mov dl, [hdc]
420
                                 <1>
421
                                          ;; DL = Hard disk drive number
                                 <1>
422
                                 <1>
                                           ; dec byte [PP_Counter]
                                          ;jz
                                                 short continue check ep next disk
423
                                 <1>
424
                                 <1>
                                          ;add esi, 16 ; 10h
425
                                 <1>
                                           ;mov dl, [hdc]
426
                                 <1>
                                          ;jmp short get_minidisk_partition_entry
427
                                 <1>
428
                                 <1>
                                          ; 11/08/2020
429
                                 <1>
                                          ;jmp short get_minidisk_partition_entry_next
430
                                 <1>
                                          ; 23/08/2020
431
                                 <1>
432 0000687E EB3D
                                 <1>
                                          jmp short validate_next_minidisk_partition_ok
433
                                 <1>
                                          ; 17/07/2020
                                 <1>
434
                                          ;; jumping down to deep levels !!!
435
                                 <1>
436
                                 <1>
                                          ; ((That is a pitty microsoft preferred ep table chain
437
                                 <1>
                                           ; instead of a single table as mbr partition table!?))
438
                                 <1>
                                 <1> loc_set_valid_hde_partition_entry:
439
440
                                 <1>
                                          ; 15/07/2020
                                          mov al, [hdc]; Hard disk drive number (>=80h)
441 00006880 A0[555D0000]
                                <1>
                                                 dl, al; mov dl, [hdc]
442 00006885 88C2
                                <1>
                                          mov
443 00006887 2C7F
                                <1>
                                          sub
                                                 al, 7Fh
                                <1>
                                                     ; 1 to 4
445 00006889 C0E002
                                <1>
                                          shl
                                                 al, 2 ; 4 to 16
446 0000688C 2A05[8B5E0100]
                                <1>
                                          sub
                                                al, [PP_Counter]; al - (4 - partition index)
```

```
; (disk number * 4) + partition index
447
                                 <1>
448
                                 <1>
                                 <1>
                                           ; AL = Partition entry/index, 0 based
449
450
                                 <1>
                                           ; 0 \rightarrow hd 0, Partition Table offset = 0
451
                                 <1>
                                             ; 15 \rightarrow hd 3, Partition Table offset = 3
452
                                 <1>
453
                                 <1>
                                           ;mov ah, 4 ; Logical dos partition (>= 4)
454
                                 <1>
                                           ; add ah, [EP_Counter]
455
                                 <1>
                                                  ; Logical disk partition index = 4 to 7
456
                                 <1>
                                                  ; (Primary disk partition index = 0 to 3)
457
                                 <1>
                                           ; 13/08/2020
458
                                 <1>
                                           mov ah, [LD_Counter] ; Logical drive index number
459 00006892 8A25[8D5E0100]
                                 <1>
460
                                 <1>
                                                             ; (in current extended partition)
461 00006898 80C404
                                           add
                                                 ah, 4 ; 4 to 7
                                 <1>
462
                                 <1>
                                           ; 15/07/2020
463
                                 <1>
                                           ; CX -> AX
464
                                 <1>
465
                                 <1>
                                           ;; 06/01/2016 (TRDOS v2.0)
466
                                 <1>
                                           ;; BUGFIX *
467
                                 <1>
                                           ;;mov [esi+LD_PartitionEntry], cl
468
                                 <1>
                                           ;;mov [esi+LD_DParamEntry], ch
                                           ;mov [esi+LD_PartitionEntry], cx
                                 <1>
469
470 0000689B 6689467C
                                 <1>
                                           mov
                                                 [esi+LD_PartitionEntry], ax
471
                                 <1>
472 0000689F 8A0D[610D0100]
                                 <1>
                                                  cl, [Last_DOS_DiskNo]
                                           mov
473 000068A5 80C141
                                 <1>
                                           add
                                                cl, 'A'
474 000068A8 880E
                                 <1>
                                           mov
                                                 [esi+LD_Name], cl
475
                                 <1>
                                           ; 17/07/2020
476
                                 <1>
477
                                 <1>
                                           ;cmp cl, 'Z'
478
                                 <1>
                                           ;jb
                                                 short logical_drive_count_ok_for_next
                                           ;pop esi ; ***
479
                                 <1>
                                           ;pop esi ; ****
480
                                 <1>
481
                                 <1>
                                           ;retn
482
                                 <1>
483
                                 <1> ;logical_drive_count_ok_for_next:
484
                                 <1>
485
                                 <1>
                                           ;; 15/07/2020
                                           ;inc byte [EP_Counter]
                                 <1>
486
                                           ; 13/08/2020
487
                                 <1>
488 000068AA FE05[8D5E0100]
                                 <1>
                                           inc byte [LD_Counter]
489
                                 <1>
490
                                 <1>
                                           ;mov dl, [hdc]
491
                                 <1>
492
                                 <1>
                                           ; 17/07/2020
                                           ;; Now,
493
                                 <1>
494
                                 <1>
                                           ;; we are swimming in deep of an extended partition !!!
495
                                 <1>
                                           ; (! sub or chained extended partition tables !)
                                 <1>
                                           ; ((Logical dos partitions in extended partition were called
496
497
                                 <1>
                                           ; as 'mini disk partition' in msdos 6.0 source code.))
498
                                 <1>
499
                                 <1> validate_next_minidisk_partition:
500
                                 <1>
                                          ; 13/08/2020
501
                                 <1>
                                           ;pop esi; ***; Extended partition table offset
502
                                 <1>
                                           ; 17/07/2020
503
                                 <1>
504
                                           ;cmp byte [EP_Counter], 4
                                 <1>
505
                                 <1>
                                           ; 13/08/2020
                                           cmp byte [LD_Counter], 4 ; maximum 4 logical disks
506 000068B0 803D[8D5E0100]04
                                 <1>
507
                                 <1>
                                                                    ; per extended partition
508 000068B7 0F8370FFFFFF
                                 <1>
                                           jnb
                                                 continue_check_ep_next_disk
509
                                 <1>
510
                                 <1> validate_next_minidisk_partition_ok:
                                 <1>
                                           ; 13/08/2020
511
512
                                 <1>
                                           ;dec byte [PP_Counter] ; 4 --> 0
513
                                 <1>
                                                 continue_check_ep_next_disk
                                           ;jz
514
                                 <1>
515
                                 <1>
                                           ;cmp esi, PartitionTable+64
516
                                 <1>
                                           ;jnb continue_check_ep_next_disk
517
                                 <1>
518
                                 <1>
                                           ;add esi, 16
                                           ; 13/08/2020
519
                                 <1>
                                           mov esi, PartitionTable+16
520 000068BD BE[545D0100]
                                 <1>
521
                                 <1>
                                           ; 20/07/2020
522
                                 <1>
523 000068C2 8A4604
                                 <1>
                                           mov al, [esi+ptFileSystemID]
524
                                 <1>
525
                                 <1>
                                           ; 20/07/2020
526 000068C5 3C05
                                           cmp al, 05h; Is it an extended dos partition?
                                 <1>
                                                  short loc_minidisk_next_ep_lba_chs ; 17/07/2020
527 000068C7 7408
                                 <1>
528 000068C9 3C0F
                                 <1>
                                                  al, OFh ; Is it an extended win4 (LBA mode) partition ?
                                           cmp
529 000068CB 0F855CFFFFF
                                 <1>
                                                  continue_check_ep_next_disk ; AL must be 0 here
                                           jne
                                 <1>
                                                                      ; (when it is not 05h or 0Fh)
531
                                                                      ; If AL is not ZERO -> EP Bug!
                                 <1>
532
                                 <1>
                                                                      ; (!Microsoft DOS convention!)
533
                                 <1> loc_minidisk_next_ep_lba_chs:
                                          ; 17/07/2020
534
                                 <1>
535 000068D1 8B4E08
                                 <1>
                                                 ecx, [esi+ptStartSector] ; relative start sector number
                                           ;add ecx, [EP_StartSector]
536
                                 <1>
                                           ; 30/08/2020
537
                                 <1>
538 000068D4 030D[8E5E0100]
                                 <1>
                                           add ecx, [MBR_EP_StartSector]
                                 <1>
539
                                           ; 20/07/2020
540 000068DA E92FFFFFF
                                 <1>
                                           jmp loc_validate_hde_partition_next
541
                                 <1>
542
                                 <1> validate_hd_fat_partition:
                                         ; 17/07/2020
543
                                 <1>
                                           ; 15/07/2020
544
                                 <1>
545
                                 <1>
                                                  (optimization)
                                           ; 14/07/2020
546
                                 <1>
547
                                 <1>
                                                 (fat16 -big- partition search bugfix)
548
                                 <1>
                                          ; 27/12/2017
                                           ; 12/02/2016
549
                                 <1>
550
                                 <1>
                                         ; 07/01/2016 (TRDOS 386 = TRDOS v2.0)
551
                                 <1>
                                           ; 07/08/2011
```

```
552
                                         ; 23/07/2011
                                 <1>
553
                                 <1>
                                          ; Input
                                          ; DL = Hard/Fixed Disk Drive Number
554
                                 <1>
555
                                 <1>
                                           ; ESI = PartitionTable offset
                                           ; EDI = Extend. Part. Start Sector Pointer
556
                                 <1>
557
                                 <1>
                                              EDI = 0 -> Primary Partition
                                          ; byte [Last_DOS_DiskNo]
558
                                 <1>
                                          ; Output
                                 <1>
559
560
                                 <1>
                                          ; cf=0 -> Validated
                                           ; ESI = Logical dos drv desc. table
561
                                 <1>
562
                                 <1>
                                           ; EBX = FAT boot sector buffer
563
                                 <1>
                                          ; byte [Last DOS DiskNo]
564
                                 <1>
                                           ; cf=1 -> Not a valid FAT partition
565
                                 <1>
                                          ; EAX, EDX, ECX, EDI -> changed
                                 <1>
566
567
                                 <1>
                                           ;mov esi, PartitionTable
568 000068DF 8A6604
                                 <1>
                                           mov
                                                 ah, [esi+ptFileSystemID]
569 000068E2 B002
                                 <1>
                                           mov al, 2; 27/12/2017
570 000068E4 80FC06
                                 <1>
                                                 ah, 06h; FAT16 CHS partition (>=32MB)
                                           ; 12/02/2016
571
                                <1>
572
                                 <1>
                                          ;;jb short loc_not_a_valid_fat_partition2
                                          ;jnb
573
                                 <1>
                                                short vhdp FAT16 32
                                           ; 14/07/2020 (BugFix)
574
                                 <1>
575 000068E7 7711
                                 <1>
                                                 short vhdp_FAT16_32
                                           jа
                                                 short loc_set_valid_hd_partition_params
576 000068E9 7425
                                 <1>
                                           jе
577
                                 <1>
                                 <1> vhdp_FAT12_16:
                                          ; 27/12/2017
579
                                <1>
580 000068EB FEC8
                                 <1>
                                           dec
                                                 al ; mov al, 1
581 000068ED 38C4
                                <1>
                                                 ah, al ; 1 ; FAT12 partition
                                           cmp
582 000068EF 741F
                                <1>
                                                 short loc_set_valid_hd_partition_params
583
                                 <1>
584 000068F1 FEC0
                                <1>
                                                 al; mov al, 2
                                          inc
585 000068F3 80FC04
                                <1>
                                                 ah, 04h; FAT16 CHS partition (< 32MB)
                                           cmp
586 000068F6 7418
                                 <1>
                                                 short loc_set_valid_hd_partition_params
587
                                 <1>
                                          ; 15/07/2020
588
                                 <1>
589
                                 <1>
                                          ; (ah = 05h, 02h or 03h)
                                 <1> loc not a valid fat partition1:
590
591 000068F8 F9
                                 <1>
                                          stc
592
                                 <1>
                                           ; cf=1
593 000068F9 C3
                                 <1>
                                           retn
594
                                 <1>
                                 <1> vhdp FAT16 32:
595
596
                                 <1>
                                          ; 15/07/2020
                                           ;mov al, 3
597
                                 <1>
                                                 al
598 000068FA FEC0
                                <1>
                                           inc
599 000068FC 80FC0C
                                <1>
                                                 ah, OCh ; FAT32 LBA partition
                                           cmp
600 000068FF 740F
                                <1>
                                           jе
                                                 short loc_set_valid_hd_partition_params
601 00006901 7706
                                <1>
                                                 short vhdp_check_FAT16_lba
                                           jа
602
                                 <1>
603
                                 <1> vhdp_check_FAT32_chs:
604 00006903 80FC0B
                                          cmp ah, OBh ; FAT32 CHS partition
                                 <1>
605 00006906 7408
                                 <1>
                                                 short loc_set_valid_hd_partition_params
                                 <1>
                                           ; jne short loc_not_a_valid_fat_partition1
606
607
                                 <1>
608
                                 <1>
                                 <1> loc_not_a_valid_fat_partition2:
609
610 00006908 C3
                                 <1>
611
                                 <1>
612
                                 <1> vhdp_check_FAT16_lba:
613 00006909 80FC0E
                                 <1>
                                                 ah, OEh ; FAT16 LBA partition
                                           cmp
614 0000690C 75EA
                                 <1>
                                                 short loc_not_a_valid_fat_partition1
615
                                 <1>
                                 <1>
616
                                           ;mov al, 2
617 0000690E FEC8
                                 <1>
                                           dec
                                                 al
618
                                 <1>
                                 <1> loc_set_valid_hd_partition_params:
619
620
                                 <1>
                                          ; 15/07/2020
621
                                 <1>
                                           ;inc byte [Last_DOS_DiskNo]; > 1
                                 <1>
623 00006910 31DB
                                 <1>
                                                 ebx, ebx
                                           xor
                                                 bh, [Last_DOS_DiskNo] ; * 256
624 00006912 8A3D[610D0100]
                                 <1>
                                           mov
625 00006918 FEC7
                                 <1>
                                                 bh ; 15/07/2020
626 0000691A 81C300010900
                                 <1>
                                           add
                                                 ebx, Logical_DOSDisks
627
                                 <1>
628 00006920 C6430102
                                 <1>
                                                 byte [ebx+LD_DiskType], 2
                                          mov
629 00006924 885302
                                 <1>
                                          mov
                                                 byte [ebx+LD_PhyDrvNo], dl
630
                                 <1>
                                                 byte [ebx+LD_FATType], al ; 2 or 3
                                                 byte [ebx+LD_FSType], ah; 06h, 0Eh, 0Bh, 0Ch
631
                                 <1>
                                           ; mov
632 00006927 66894303
                                 <1>
                                           mov
                                                 word [ebx+LD_FATType], ax
                                 <1>
634 0000692B 8B4E08
                                 <1>
                                           mov
                                                 ecx, [esi+ptStartSector]
                                 <1>
                                                 edi, edi
635 0000692E 09FF
                                 <1>
                                                 short pass_hd_FAT_ep_start_sector_adding
636 00006930 7402
                                           jΖ
                                 <1> loc add hd FAT ep start sector:
637
638
                                 <1>
                                          ; 17/07/2020
639 00006932 030F
                                 <1>
                                           add ecx, [edi]
                                 <1> pass_hd_FAT_ep_start_sector_adding:
<1> mov [ebx+LD_StartSector], ecx
640
641 00006934 894B6C
                                 <1> loc_hd_FAT_logical_drv_init:
642
643 00006937 89DD
                                 <1>
                                          mov
                                                ebp, ebx
                                                 dl, [ebx+LD_PhyDrvNo]
644
                                 <1>
                                           ;mov
645 00006939 A0[8A5E0100]
                                 <1>
                                                 al, [HD_LBA_yes] ; 07/01/2016
646 0000693E 884305
                                 <1>
                                                 [ebx+LD LBAYes], al
                                          mov
                                                 ebx, DOSBootSectorBuff; buffer address
647 00006941 BB[965E0100]
                                 <1>
                                           mov
648 00006946 08C0
                                 <1>
                                                 al, al
                                           or
                                                 short loc_hd_FAT_drv_init_load_bs_chs
649 00006948 740C
                                 <1>
                                           jz
650
                                 <1> loc_hd_FAT_drv_init_load_bs_lba:
651
                                 <1>
                                          ; DL = Physical drive number
652
                                 <1>
                                           ;mov ecx, [esi+ptStartSector] ; sector number
653
                                 <1>
                                          ;mov ebx, DOSBootSectorBuff; buffer address
654
                                 <1>
                                           ; LBA read/write (with private LBA function)
655
                                 <1>
                                          ;((Retro UNIX 386 v1 - DISK I/O code by Erdogan Tan))
                                 <1>
656
                                           ; dl = physical drive number (0,1, 80h, 81h, 82h, 83h)
```

```
<1> mov ah, 1Bh; LBA read
<1> mov al, 1; sector count
<1> call int13h
<1> jnc short loc_hd_drv_FAT_boot_validation
657 0000694A B41B
658 0000694C B001
659 0000694E E8DAD8FFFF
660 00006953 7313
                                 <1> loc_not_a_valid_fat_partition3:
<1> retn
661
662 00006955 C3

<1> retn
<1> loc_hd_FAT_drv_init_load_bs_chs:
<1> mov dh, [esi+ptBeginHead]
<1> mov cx, [esi+ptBeginSector]
<1> mov ax, 0201h; Read 1 sector
<1> ;mov ebx, DOSBootSectorBuff
<1> call int13h
<1> jc short loc_not_a_valid_fat_partition3
<1> loc_bd_dry_FAT_boot_validation:
663
664 00006956 8A7601
665 00006959 668B4E02
666 0000695D 66B80102
668 00006961 E8C7D8FFFF
669 00006966 72ED
670
                                  <1> loc hd drv FAT boot validation:
                                  <1> ;mov esi, DOSBootSectorBuff
<1> mov esi, ebx
671
                      <1>
672 00006968 89DE
673 0000696A 6681BEFE01000055AA <1> cmp word [esi+BS_Validation], 0AA55h 674 00006973 7512 <1> jne short loc_not_a_valid_fat_partition4 675 00006975 807E15F8 <1> cmp byte [esi+BPB_Media], 0F8h 676 00006979 750C <1> jne short loc_not_a_valid_fat_partition4
                          <1><1>
677
                                  <1>
                                        ; 27/12/2017
cmp byte [ebp+LD_FATType], 3
jne short loc_hd_FAT16_BPB
                                  <1>
679 0000697B 807D0303
                                  <1>
680 0000697F 7508
                                  <1>
681
                                  <1>
                                  <1> loc_hd_drv_FAT32_boot_validation:
682
                               <1>
683 00006981 807E4229
                                        cmp byte [esi+BS_FAT32_BootSig], 29h
684 00006985 7416
                                  <1>
                                                    short loc_hd_FAT32_BPB
                               <1> loc_not_a_valid_fat_partition4:
<1> stc
685
686
687 00006987 F9
688 00006988 C3
                                  <1>
                                              retn
                                  <1>
689
690
                                  <1> loc_hd_FAT16_BPB:
                                         cmp byte [esi+BS_BootSig], 29h
691 00006989 807E2629
                                  <1>
692 0000698D 75F8
                                  <1>
                                                    short loc_not_a_valid_fat_partition4
                                  <1>
694 0000698F 66837E1600
                                  <1>
                                                    word [esi+BPB_FATSz16], 0
                                             cmp
695 00006994 7607
                                  <1>
                                                    short loc_hd_big_FAT16_BPB
                                             jna
                                                    ecx, 32
696 00006996 B920000000
                                  <1>
                                             mov
                                                    short loc_hd_move_FAT_BPB
697 0000699B EB05
                                  <1>
                                            jmp
698
                                   <1>
                                   <1> loc_hd_FAT32_BPB:
699
700
                                   <1>
                                        ;cmp word [esi+BPB_FATSz16], 0
701
                                   <1>
                                             ;ja short loc_not_a_valid_fat_partition4
702
                                   <1> loc_hd_big_FAT16_BPB:
703 0000699D B92D000000
                                  <1>
                                            mov
                                                   ecx, 45
704
                                  <1>
705
                          <1> loc_hd_move_FAT_BPB:
706 000069A2 89EF
707
708 000069A4 57
709 000069A5 83C706
710 000069A8 F366A5
711 000069AB 5E
712 000069AC 0FB74614
713 000069B0 03466C
714 000069B3 894660
715 000069B6 807E0303
716 000069BA 7224
                          717
718 000069BC 8B462A
719 000069BF 0FB65E16
                                                           ebx, byte [esi+LD_BPB+BPB_NumFATs]
720 000069C3 F7E3
721 000069C5 034660
722
723 000069C8 894668
                                  <1>
724 000069CB 894664
                                           mov [esi+LD_ROOTBegin], eax
                                           ; If Root Directory Cluster <> 2 then
725
                                  <1>
726
                                  <1>
                                             ; change the beginning sector value
                                          ; of the root dir by adding sector offset.
mov eax, [esi+LD_BPB+BPB_RootClus]
727
                                  <1>
728 000069CE 8B4632
                                  <1>
729 000069D1 83E802
                                  <1>
                                             sub
                                                    eax, 2
                                           jz short short loc set 32bit FAT total sectors
730 000069D4 7435
                                  <1>
                                        ;movzx ebx, byte [esi+LD_BPB+BPB_SecPerClust]
mov bl, [esi+LD_BPB+BPB_SecPerClust]
mul ebx
731
                                  <1>
                                  <1>
732 000069D6 8A5E13
733 000069D9 F7E3
                                  <1>
                                  <1>
734 000069DB 014664
                                           add [esi+LD_ROOTBegin], eax
735 000069DE EB2B
                                  <1>
                                             jmp short loc_set_32bit_FAT_total_sectors
736
                                  <1>
                                   <1> loc_set_FAT16_RootDirLoc:
737
738 000069E0 0FB64616
                                   <1>
                                             movzx eax, byte [esi+LD_BPB+BPB_NumFATs]
                               <1>
739 000069E4 0FB7561C
                                              movzx edx, word [esi+LD_BPB+BPB_FATSz16]
740 000069E8 F7E2
                                  <1>
                                             add eax, [esi+LD_FATBegin]
mov [esi+LD_ROOTBegin], ea:
741 000069EA 034660
                                  <1>
                                  <1>
742 000069ED 894664
                                                    [esi+LD_ROOTBegin], eax
                                  <1> loc set FAT16 data begin:
744 000069F0 894668
                                  <1>
                                           mov [esi+LD_DATABegin], eax
745
                                   <1>
                                             ;mov eax, 20h ; Size of a directory entry
746
                                             ;;movzx
                                                          edx, word [esi+LD BPB+BPB RootEntCnt]
                                   <1>
                                            ;mov
                                                         dx, [esi+LD BPB+BPB RootEntCnt]
747
                                   <1>
                                              ; mul edx
748
                                   <1>
                                            ;;mov ecx, 511
749
                                   <1>
750
                                   <1>
                                            ;mov cx, 511
751
                                   <1>
                                           ;add eax, ecx
                                             ;inc ecx ; 512
752
                                   <1>
                                             ;div ecx
753
                                   <1>
                                             ; 14/07/2020
754
                                   <1>
                                             movzx eax, word [esi+LD BPB+BPB RootEntCnt]
755 000069F3 0FB74617
                                   <1>
756 000069F7 6683C00F
                                  <1>
                                             add ax, 15
                                  <1>
757 000069FB 66C1E804
                                           shr
                                                     ax, 4 ; / 16 ; (16 entries per sector)
                                           add [esi+LD_DATABegin], eax
                                  <1>
758 000069FF 014668
                                  <1>
                                             ;movzx eax, word [esi+LD_BPB+BPB_TotalSec16]
759
760 00006A02 668B4619
                                  <1>
                                             mov ax, [esi+LD_BPB+BPB_TotalSec16]
761 00006A06 6685C0
                                   <1>
                                             test ax, ax
```

```
762
                                           ;jz short loc set 32bit FAT total sectors
                                 <1>
763
                                 <1> ;loc set 16bit FAT total sectors:
                                           ;mov [esi+LD_TotalSectors], eax
764
                                 <1>
765
                                 <1>
                                           ;jmp short loc_set_hd_FAT_cluster_count
766
                                 <1>
                                           ; 14/07/2020
767 00006A09 7503
                                 <1>
                                           jnz short loc_set_hd_FAT_cluster_count
                                <1> loc_set_32bit_FAT_total_sectors:
                                <1>
                                       mov eax, [esi+LD_BPB+BPB_TotalSec32]; mov [esi+LD_TotalSectors], eax
769 00006A0B 8B4626
770
                                <1>
771
                                <1> loc set hd FAT cluster count:
772 00006A0E 894670
                                <1> mov [esi+LD_TotalSectors], eax ; 14/07/2020
773 00006A11 8B5668
                                <1>
                                           mov
                                                 edx, [esi+LD_DATABegin]
                                      sub eax, edx
xor edx, edx; 0
  movzx ecx, byte [esi+LD_BPB+BPB_SecPerClust]
  div ecx
mov [esi+LD_Clusters], eax
; Maximum Valid Cluster Number
; with 2
                                <1>
774 00006A14 2B566C
775 00006A17 29D0
                                <1>
776 00006A19 31D2
                                <1>
<1>
777 00006A1B 0FB64E13
778 00006A1F F7F1
                                <1>
779 00006A21 894678
                                <1>
780
                                 <1>
781
                                 <1>
                                           ; with 2 reserved clusters= EAX +2
782
                                 <1> loc_set_hd_FAT_fs_free_sectors:
                                           ;mov dword [esi+LD_FreeSectors], 0
call get_free_FAT_sectors
                                 <1>
784 00006A24 E855010000
                                 <1>
785 00006A29 720D
                                 <1>
                                           jс
                                                 short loc_validate_hd_FAT_partition_retn
786 00006A2B 894674
                                 <1>
                                          mov
                                                  [esi+LD_FreeSectors], eax
787 00006A2E C6467E06
                                 <1>
                                           mov
                                                  byte [esi+LD_MediaChanged], 6 ; Volume Name Reset
                                 <1>
789
                                           ; 15/07/2020
                                 <1>
790 00006A32 FE05[610D0100]
                                 <1>
                                                byte [Last_DOS_DiskNo] ; > 1
                                           inc
791
                                 <1>
792
                                 <1>
                                           ;mov cl, [Last_DOS_DiskNo]
793
                                 <1>
                                           ;add cl, 'A'
794
                                 <1>
                                           ;mov [esi+LD_FS_Name], cl
795
                                 <1>
796
                                 <1> loc_validate_hd_FAT_partition_retn:
797 00006A38 C3
                                 <1>
                                           retn
798
                                 <1>
                                 <1> validate_hd_fs_partition:
799
                                        ; \overline{0}3/\overline{0}2/\overline{2}018
800
                                 <1>
                                           ; 09/12/2017
                                 <1>
801
802
                                 <1>
                                         ; 13/02/2016
                                          ; 10/01/2016 (TRDOS 386 = TRDOS v2.0)
803
                                 <1>
                                           ; 29/01/2011
804
                                 <1>
805
                                 <1>
                                          ; 23/07/2011
806
                                 <1>
                                           ; Input
                                           ; DL = Hard/Fixed Disk Drive Number
807
                                 <1>
808
                                 <1>
                                           ; ESI = PartitionTable offset
                                 <1>
809
                                          ; byte [Last_DOS_DiskNo]
810
                                 <1>
                                           ; Output
                                           ; cf=0 -> Validated
                                 <1>
811
812
                                 <1>
                                           ; ESI = Logical dos drv desc. table
                                           ; EBX = Singlix FS boot sector buffer
; byte [Last_DOS_DiskNo]
813
                                 <1>
814
                                 <1>
815
                                 <1>
                                          ; cf=1 -> Not a valid 'Singlix FS' partition
816
                                 <1>
                                           ; EAX, EDX, ECX, EDI -> changed
817
                                 <1>
                                 <1>
                                           ;mov esi, PartitionTable
819 00006A39 8A6604
                                           mov ah, [esi+ptFileSystemID]
                                 <1>
820 00006A3C 80FCA1
                                 <1>
                                                 ah, OA1h; SINGLIX FS1 (trfs1) partition
                                           jne short loc_validate_hd_fs_partition_stc_retn
821 00006A3F 7549
                                 <1>
822
                                 <1> loc_set_valid_hd_fs_partition_params:
                                       inc byte [Last_DOS_DiskNo]; > 1
823 00006A41 FE05[610D0100]
                                 <1>
824 00006A47 30C0
                                 <1>
                                                 al, al ; mov al, 0
                                           xor
                                825
826 00006A49 29DB
827 00006A4B 8A3D[610D0100]
828 00006A51 81C300010900
829 00006A57 C6430102
                                                 byte [ebx+LD_DiskType], 2
830 00006A5B 885302
831
832
833 00006A5E 66894303
                                 <1>
                                           mov
                                                  [ebx+LD_FATType], ax
                                 <1>
834
                                           ;mov eax, [esi+ptStartSector]
                                           ;mov [ebx+LD StartSector], eax
835
                                 <1>
                                 <1> loc_hd_fs_logical_drv_init:
836
837 00006A62 89DD
                                 <1>
                                          mov ebp, ebx ; 10/01/2016
                                           ;mov dl, [ebx+LD PhyDrvNo]
                                 <1>
                                                 al, [HD_LBA_yes] ; 10/01/2016
839 00006A64 A0[8A5E0100]
                                 <1>
                                           mov
                                                  [ebx+LD LBAYes], al
840 00006A69 884305
                                 <1>
                                           mov
841 00006A6C 89DE
                                           mov
                                 <1>
                                                 esi, ebx
842 00006A6E BB[965E0100]
                                 <1>
                                           mov
                                                  ebx, DOSBootSectorBuff; buffer address
843 00006A73 08C0
                                                  al, al
                                 <1>
                                           or
844 00006A75 7515
                                 <1>
                                           jnz
                                                 short loc_hd_fs_drv_init_load_bs_lba
                                 <1> loc_hd_fs_drv_init_load_bs_chs:
845
846 00006A77 8A7601
                                 <1>
                                           mov dh, [esi+ptBeginHead]
847 00006A7A 668B4E02
                                                  cx, [esi+ptBeginSector]
                                 <1>
                                           mov
848 00006A7E 66B80102
                                 <1>
                                                 ax, 0201h; Read 1 sector
                                           mov
                                 <1>
                                           ;mov ebx, DOSBootSectorBuff
849
850 00006A82 E8A6D7FFFF
                                 <1>
                                           call int13h
851 00006A87 7311
                                 <1>
                                           jnc short loc hd drv fs boot validation
852
                                 <1> loc_validate_hd_fs_partition_err_retn:
853 00006A89 C3
                                 <1>
                                          retn
                                 <1> loc_validate_hd_fs_partition_stc_retn:
854
855 00006A8A F9
                                 <1>
856 00006A8B C3
                                 <1>
                                           retn
857
                                 <1> loc_hd_fs_drv_init_load_bs_lba:
                                           ; DL = Physical drive number
858
                                 <1>
859
                                 <1>
                                           ;mov esi, ebx
860 00006A8C 8B4E08
                                 <1>
                                                 ecx, [esi+ptStartSector] ; sector number
                                           ;mov ebx, DOSBootSectorBuff; buffer address
                                 <1>
861
862
                                 <1>
                                           ; LBA read/write (with private LBA function)
                                           ; ((Retro UNIX 386 v1 - DISK I/O code by Erdogan Tan))
863
                                 <1>
864
                                 <1>
                                           ; dl = physical drive number (0,1, 80h, 81h, 82h, 83h)
865 00006A8F B41B
                                 <1>
                                           mov ah, 1Bh; LBA read
866 00006A91 B001
                                 <1>
                                           mov al, 1; sector count
```

```
867 00006A93 E895D7FFFF
                              <1>
                                         call int13h
868 00006A98 72EF
                               <1>
                                         jс
                                              short loc validate hd fs partition err retn
                               <1> loc hd drv fs boot validation:
869
870
                               <1> ;mov esi, DOSBootSectorBuff
871 00006A9A 89DE
                               <1>
                                              esi, ebx ; Boot sector buffer
                                        mov
872 00006A9C 6681BEFE01000055AA <1>
                                        cmp
                                               word [esi+BS_Validation], 0AA55h
                                        jne short loc_validate_hd_fs_partition_stc_retn
                  <1>
873 00006AA5 75E3
                               <1>
875
                               <1>
                                        ;Singlix FS Extensions to TR-DOS (7/6/2009)
876 00006AA7 66817E034653 <1>
                                        cmp word [esi+bs FS Identifier], 'FS'; 03/02/2018
877 00006AAD 75DB
                              <1>
                                              short loc_validate_hd_fs_partition_stc_retn
                                        ;'Alh' check is not necessary
                               <1>
                                        ; if 'FS' check is passed as OK/Yes.
879
                              <1>
                           <1>
880 00006AAF 807E09A1
                                              byte [esi+bs_FS_PartitionID], 0A1h
881 00006AB3 75D5
                              <1>
                                              short loc_validate_hd_fs_partition_stc_retn
                                        jne
882
                               <1>
883 00006AB5 89EF
                                               edi, ebp ; 10/01/2016
                              <1>
                                        mov
884
                               <1>
                                        ;
885 00006AB7 8A462D
                               <1>
                                               al, byte [esi+bs_FS_LBA_Ready]
                                         mov
886 00006ABA 884705
                               <1>
                                               [edi+LD_FS_LBAYes], al
                                        mov
887
                               <1>
                                        ; 03/01/2010 CHS -> DOS FAT/BPB compatibility fix
                               <1>
889 00006ABD 8A4608
                               <1>
                                               al, [esi+bs_FS_MediaAttrib]
                                         mov
890 00006AC0 884706
                               <1>
                                               byte [edi+LD_FS_MediaAttrib], al
                               <1>
891
                                         ;
892 00006AC3 8A460A
                               <1>
                                         mov
                                               al, [esi+bs_FS_VersionMaj]
893 00006AC6 884707
                               <1>
                                               [edi+LD_FS_VersionMajor], al
                                        mov
894
                               <1>
                                        ;
895 00006AC9 668B4606
                               <1>
                                               ax, [esi+bs_FS_BytesPerSec]
                                        mov
896 00006ACD 66894711
                                               [edi+LD FS BytesPerSec], ax
                               <1>
                                        mov
897 00006AD1 8A462E
                               <1>
                                         mov
                                               al, [esi+bs_FS_SecPerTrack]
898 00006AD4 30E4
                               <1>
                                        xor
                                               ah, ah; 09/12/2017
899 00006AD6 6689471E
                                               [edi+LD_FS_SecPerTrack], ax
                              <1>
                                        mov
900 00006ADA 8A462F
                              <1>
                                               al, [esi+bs_FS_Heads]
901 00006ADD 66894720
                               <1>
                                               [edi+LD_FS_NumHeads], ax
                                        mov
                               <1>
903 00006AE1 8B4628
                              <1>
                                        mov
                                               eax, [esi+bs FS UnDelDirD]
                                               [edi+LD FS UnDelDirD], eax
904 00006AE4 894722
                              <1>
                                        mov
905 00006AE7 8B5618
                               <1>
                                               edx, [esi+bs_FS_MATLocation]
                                        mov
                                               [edi+LD FS MATLocation], edx
906 00006AEA 89570C
                              <1>
                                        mov
907 00006AED 8B461C
                              <1>
                                        mov
                                               eax, [esi+bs_FS_RootDirD]
908 00006AF0 894708
                              [edi+LD FS RootDirD], eax
                              <1>
                                        mov
909 00006AF3 8B460C
                                               eax, [esi+bs_FS_BeginSector]
910 00006AF6 89476C
                                               [edi+LD_FS_BeginSector], eax
911 00006AF9 8B4710
                                               eax, [edi+bs_FS_VolumeSize]
912 00006AFC 894770
                                               [edi+LD_FS_VolumeSize], eax
914 00006AFF 89D0
                                              eax, edx ; [edi+LD_FS_MATLocation]
915 00006B01 03476C
                                               eax, [edi+LD_FS_BeginSector]
916 00006B04 89FE
917
                              <1> mread_hd_fs_MAT_sector:
918
                              <1> ;mov ebx, DOSBootSectorBuff
                              <1>
919 00006B06 B901000000
                                        mov ecx, 1
                              <1> call disk_read
920 00006B0B E84C8D0000
                            921 00006B10 7248
922
923 00006B12 89DE
                              <1> use_hdfs_mat_sector_params:
924
                              <1> mov eax, [esi+FS_MAT_DATLocation]
<1> mov [edi+LD_FS_DATLocation], eax
925 00006B14 8B460C
926 00006B17 894714
                            <1>
                           927 00006B1A 8B4610
928 00006B1D 894718
929 00006B20 8B4614
930 00006B23 894774
931 00006B26 8B4618
932 00006B29 894778
933 00006B2C 8B4708
                                    add eax, [edi+LD_FS_BeginSector]
mov esi, edi
934 00006B2F 03476C
                              <1>
935 00006B32 89FE
                               <1>
                               <1> read_hd_fs_RDT_sector:
936
937 00006B34 BB[965E0100]
                              <1> mov ebx, DOSBootSectorBuff
                                        ;mov ecx, 1 mov cl, 1
938
                               <1>
939 00006B39 B101
                              <1>
                              <1> call disk_read
<1> jc short loc_validate_hd_fs_partition_retn
<1> : EDT will not be changed
940 00006B3B E81C8D0000
941 00006B40 7218
942
                              <1>
                                        ; EDI will not be changed
943 00006B42 89DE
                              <1>
                                       mov esi, ebx
944
                               <1> use_hdfs_RDT_sector_params:
                                    mov eax, [esi+FS_RDT_VolumeSerialNo]
mov [edi+LD_FS_VolumeSerial], eax
945 00006B44 8B461C
                               <1>
946 00006B47 894728
                               <1>
947 00006B4A 57
                               <1>
                                        push edi
                                       ;mov ecx, 16
948
                               <1>
949 00006B4B B110
                               <1>
                                               cl, 16
                                         mov
950 00006B4D 83C640
                               <1>
                                               esi, FS_RDT_VolumeName
                                               edi, LD_FS_VolumeName
                               <1>
951 00006B50 83C72C
                                         add
952 00006B53 F3A5
                               <1>
                                               movsd ; 64 bytes
                                         rep
953 00006B55 5E
                               <1>
                                              esi
                                         pop
954
                                               ; Volume Name Reset
                               <1>
955 00006B56 C6467E06
                               <1>
                                                  byte [esi+LD_FS_MediaChanged], 6
956
                               <1>
957
                               <1>
                                          ;movcl, [Last_DOS_DiskNo]
958
                               <1>
                                         ;add cl, 'A'
959
                               <1>
                                         ;mov [esi+LD_FS_Name], cl
                               <1>
960
                               <1> loc_validate_hd_fs_partition_retn:
961
962 00006B5A C3
                               <1>
                                         retn
963
                               <1>
964
                               <1> load_masterboot:
                                        ; 14/07/2020 (Reset function has been removed)
965
                               <1>
966
                               <1>
                                        ; 10/01/2016 (TRDOS 386 = TRDOS v2.0)
967
                               <1>
968
                               <1>
                                        ; 2005 - 2011
                                         ; input -> DL = drive number
969
                               <1>
970
                                        mov ah, ODh ; Alternate disk reset
                               <1>;
971
                               <1>;
                                        call int13h
```

```
972
                                <1> ;
                                       jnc short pass_reset_error
 973
                                <1> ;harddisk error:
 974
                                <1> ; retn
 975
                                <1> ;pass_reset_error:
                                      mov ebx, MasterBootBuff
 976 00006B5B BB[865B0100]
                                <1>
 977 00006B60 66B80102
                                <1>
                                         mov
                                                ax, 0201h
                                     mov cx, 1
 978 00006B64 66B90100
                                <1>
                                      xor
                                         xor dh, dh
call int13h
 979 00006B68 30F6
                                <1>
980 00006B6A E8BED6FFFF
                                <1>
981 00006B6F 720C
                                <1>
                                         jс
                                                short harddisk_error
982
                                <1>
 983 00006B71 66813D[845D0100]55- <1>
                                         cmp
                                               word [MBIDCode], 0AA55h
                                <1>
 983 00006B79 AA
                                                short load masterboot ok
 984 00006B7A 7401
                                <1>
                                         jе
 985 00006B7C F9
                                <1>
                                          stc
                                <1> harddisk_error:
986
                                <1> load masterboot ok:
 988 00006B7D C3
                                <1>
                                         retn
 989
                                <1>
 990
                                <1> get_free_FAT_sectors:
                                      ; \overline{2}1/1\overline{2}/2017
 991
                                <1>
                                         ; 29/02/2016
 992
                                <1>
                                         ; 13/02/2016
 993
                                <1>
                                        ; 04/02/2016
 994
                                <1>
                                        ; 07/01/2016 (TRDOS 386 = TRDOS v2.0)
 995
                                <1>
 996
                                <1>
                                         ; 11/07/2010
                                <1>
                                         ; 21/06/2009
 998
                                <1>
                                        ; INPUT: ESI = Logical DOS Drive Description Table address
999
                                <1>
                                         ; OUTPUT: STC => Error
                                          ; cf = 0 and EAX = Free FAT sectors
1000
                                <1>
1001
                                <1>
                                          ; Also, related parameters and FAT buffer will be reset and updated
1002
                                <1>
1003 00006B7E 31C0
                                <1>
                                          xor
                                                eax, eax
1004
                                <1>
                                          ;mov [esi+LD_FreeSectors], eax ; Reset
1005
                                <1>
                                                    byte [esi+LD_FATType], 2
1006 00006B80 807E0302
                                <1>
                                          jna short loc gfc get fat free clusters
1007 00006B84 7654
                                <1>
1008
                                <1>
1009
                                <1>
                                          ; 29/02/2016
                                          dec eax ; OFFFFFFFFh
1010 00006B86 48
                               <1>
1011 00006B87 89463A
                               <1>
                                          mov
                                                [esi+LD_BPB+BPB_Reserved], eax ; Free cluster count (reset)
1012 00006B8A 89463E
                                <1>
                                          mov
                                                [esi+LD BPB+BPB Reserved+4], eax ; First Free Cluster (reset)
1013 00006B8D 40
                                <1>
                                         inc
                                                eax ; 0
1014
                                <1>
                                                ax, [esi+LD_BPB+BPB FSInfo]
1015 00006B8E 668B4636
                                <1>
                                         mov
1016 00006B92 03466C
                                <1>
                                         add
                                                eax, [esi+LD_StartSector]
                                <1>
1017
1018 00006B95 BB[965E0100]
1019 00006B9A B901000000
                                <1>
                                                ebx, DOSBootSectorBuff
                                         mov
                                <1>
                                          mov
                                                ecx, 1
                                          call disk_read
1020 00006B9F E8B88C0000
                               <1>
                                          jnc short loc_gfc_check_fsinfo_signs
1021 00006BA4 7301
                                <1>
1022
                                <1> retn_gfc_get_fsinfo_sec:
1023 00006BA6 C3
                                <1>
                                         retn
1024
                                <1>
1025
                                <1> loc_gfc_check_fsinfo_signs:
1026 00006BA7 BB[965E0100]
                                <1> mov ebx, DOSBootSectorBuff; 13/02/2016
1027 00006BAC 813B52526141
                                <1>
                                          cmp
                                                 dword [ebx], 41615252h
1028 00006BB2 7524
                                <1>
                                          jne
                                               short retn_gfc_get_fsinfo_stc
1029
                                <1>
                                         ;add ebx, 484
1030
                                         ;cmp dword [ebx], 61417272h
                                <1>
1031 00006BB4 81BBE4010000727241- <1>
                                       cmp dword [ebx+484], 61417272h
1031 00006BBD 61
                                <1>
1032 00006BBE 7518
                                <1>
                                         jne
                                                short retn_gfc_get_fsinfo_stc
                                         ;add ebx, 4
1033
                                <1>
                                         ;mov eax, [ebx]
1034
                                <1>
1035 00006BC0 8B83E8010000
                                <1>
                                         mov
                                                eax, [ebx+488]
1036
                                <1>
                                         ; 29/02/2016
1037 00006BC6 89463A
                                <1>
                                         mov [esi+LD_BPB+BPB_Reserved], eax ; Free cluster count
1038 00006BC9 8B93EC010000
                                <1>
                                         mov
                                                edx, [ebx+492]
                                         mov [esi+LD_BPB+BPB_Reserved+4], eax ; First Free Cluster
1039 00006BCF 89463E
                                <1>
1040
                                <1>
                                         ; 21/12/2017
1041 00006BD2 89C3
                                <1>
                                         mov ebx, eax ; (initial value = OFFFFFFFFh)
                                                ebx ; 0FFFFFFFF -> 0
1042 00006BD4 43
                                <1>
                                         inc
                                          jnz short short retn from get free fat32 clusters
1043 00006BD5 7513
                                <1>
1044 00006BD7 C3
                                <1>
                                          retn
1045
                                <1>
1046
                                <1> retn gfc get fsinfo stc:
1047 00006BD8 F9
                                <1>
                                          stc
1048 00006BD9 C3
                                <1>
1049
                                <1>
1050
                                <1> loc_gfc_get_fat_free_clusters:
1051
                                 <1>
                                       ;mov eax, 2
1052 00006BDA B002
                                <1>
                                          mov
1053
                                <1>
                                          ;mov [FAT_CurrentCluster], eax
                                <1> loc_gfc_loop_get_next_cluster:
1054
1055 00006BDC E8EB4F0000
                                <1>
                                         call get_next_cluster
1056 00006BE1 730E
                               <1>
                                          jnc short loc_gfc_free_fat_clusters_cont
1057 00006BE3 21C0
                                <1>
                                          and eax, eax
1058 00006BE5 7411
                                <1>
                                                short loc_gfc_pass_inc_free_cluster_count
                                          jz
                                <1>
1060
                                <1> retn_from_get_free_fat_clusters:
1061 00006BE7 8B4674
                                <1>
                                       mov eax, [esi+LD FreeSectors] ; Free clusters !
                                <1> retn_from_get_free_fat32_clusters:
1062
                                                   ebx, byte [esi+LD BPB+BPB SecPerClust]
1063 00006BEA 0FB65E13
                                <1>
1064 00006BEE F7E3
                                <1>
                                            mul
                                                      ebx
1065
                                <1>
                                          ;mov [esi+LD_FreeSectors], eax ; Free sectors
1066
                                <1> retn_get_free_sectors_calc:
1067 00006BF0 C3
                                <1>
                                         retn
1068
                                <1>
1069
                                <1> loc_gfc_free_fat_clusters_cont:
1070 00006BF1 09C0
                                <1>
                                          or eax, eax
1071 00006BF3 7503
                                <1>
                                                short loc_gfc_pass_inc_free_cluster_count
                                          jnz
1072 00006BF5 FF4674
                                <1>
                                          inc
                                               dword [esi+LD_FreeSectors] ; Free clusters !
1073
                                <1>
1074
                                <1> loc_gfc_pass_inc_free_cluster_count:
```

```
1075
                                <1>
                                          ;mov eax, [FAT_CurrentCluster]
1076 00006BF8 89C8
                                <1>
                                         mov
                                                eax, ecx ; [FAT CurrentCluster]
1077 00006BFA 3B4678
                                                eax, [esi+LD Clusters]
                                <1>
                                          cmp
1078 00006BFD 77E8
                               <1>
                                                short retn_from_get_free_fat_clusters
                                          jа
1079 00006BFF 40
                                <1>
                                          inc
                                                eax
1080
                                <1>
                                          ;mov
                                                [FAT_CurrentCluster], eax
1081 00006C00 EBDA
                                <1>
                                                short loc_gfc_loop_get_next_cluster
                                          jmp
1082
                                <1>
1083
                                <1> floppy_drv_init:
                                        ; 09/12/2017
1084
                                <1>
1085
                                <1>
                                         ; 06/07/2016
1086
                                <1>
                                         ; 10/01/2016 (TRDOS 386 = TRDOS v2.0)
                                         ; 24/07/2011
1087
                                <1>
1088
                                <1>
                                        ; 04/07/2009
                                        ; INPUT ->
1089
                                <1>
                                         ; DL = Drive number (0,1)
1090
                                <1>
                                         ; OUTPUT ->
1091
                                <1>
                                        ;
                                <1>
1092
                                               BL = drive name
1093
                                <1>
                                                BH = drive number
                                                ESI = Logical DOS drv description table
1094
                                <1>
                                         ;
1095
                                <1>
                                                EAX = Volume serial number
                                <1>
1097 00006C02 BE[565D0000]
                                       mov esi, fd0_type ; 10/01/2016
                                <1>
                                     mov edi, Logical_DOSDisks
1098 00006C07 BF00010900
                                <1>
                                         or
1099 00006C0C 08D2
                                <1>
                                                dl, dl
1100 00006C0E 7407
                                <1>
                                          jz
                                                short loc_drv_init_fd0_fd1
                                     add edi, 100h
inc esi; fdl_type; 10/01/2016
1101 00006C10 81C700010000
                               <1>
1102 00006C16 46
                                <1>
1103
                                <1> loc_drv_init_fd0_fd1:
                               <1> mov byte [edi+LD_MediaChanged], 0
1104 00006C17 C6477E00
1105 00006C1B 803E01
                               <1>
                                         cmp byte [esi], 1; type (>0 if it is existing)
                                                ; 4 = 1.44 MB, 80 track, 3 1/2"
1106
                                <1>
                             <1> jb short read_fd_boot_se
<1> mov [edi+LD_PhyDrvNo], dl
1107 00006C1E 7221
                                                short read_fd_boot_sector_retn
1108 00006C20 885702
1109
                               <1> read fd boot sector:
                               <1> xor dh, dh
1110 00006C23 30F6
1111 00006C25 B904000000
                           <1>
                                         mov ecx, 4 ; Retry Count
                               <1> read_fd_boot_sector_again:
1112
                                     push ecx
1113 00006C2A 51
                                <1>
                                         ;mov cx, 1
1114
                               <1>
                                     mov cl, 1
mov ax, 020
mov ebx, DO
call int13h
1115 00006C2B B101
                               <1>
1116 00006C2D 66B80102
                               <1>
                                                ax, 0201h; Read 1 sector
                               <1>
1117 00006C31 BB[965E0100]
                                                ebx, DOSBootSectorBuff
1118 00006C36 E8F2D5FFFF
                               <1>
                                       pop
1119 00006C3B 59
                                <1>
                                                ecx
1120 00006C3C 7304
                                <1>
                                          jnc
                                                short use_fd_boot_sector_params
1121 00006C3E E2EA
                                <1>
                                          loop read_fd_boot_sector_again
1122
                                <1>
1123
                                <1> read_fd_boot_sector_stc_retn:
1124 00006C40 F9
                                <1>
                                         stc
                                <1> read_fd_boot_sector_retn:
1125
1126 00006C41 C3
                                <1>
                                          retn
1127
                                <1>
1128
                                <1> use_fd_boot_sector_params:
                                         ;mov esi, DOSBootSectorBuff
1129
                                <1>
1130 00006C42 89DE
                                <1>
                                          mov
                                                esi, ebx
1131 00006C44 6681BEFE01000055AA <1>
                                               word [esi+BS Validation], OAA55h
                                          cmp
                                          jne short read_fd_boot_sector_stc_retn
1132 00006C4D 75F1
                               <1>
1133 00006C4F 66817E035346
                                <1>
                                         cmp
jne
                                                   word [esi+bs_FS_Identifier], 'SF'
1134 00006C55 0F85A2000000
                                                   use_fd_fatfs_boot_sector_params
                                <1>
1135
                                <1>
1136 00006C5B 8A462D
                                <1>
                                         mov
                                                al, [esi+bs FS LBA Ready]
1137 00006C5E 884705
                                <1>
                                                [edi+LD_FS_LBAYes], al
                                          mov
1138
                                <1>
1139
                                <1>
                                          ; 03/01/2010 CHS -> DOS FAT/BPB compatibility fix
1140 00006C61 8A4608
                                <1>
                                          mov
                                                al, [esi+bs_FS_MediaAttrib]
                                                [edi+LD_FS_MediaAttrib], al
1141 00006C64 884706
                                <1>
                                          mov
1142
                                          ;
  mov al, [esi+bs_FS_VersionMaj]
                                <1>
1143 00006C67 8A460A
                                <1>
                                                byte [edi+LD FS VersionMajor], al
1144 00006C6A 884707
                                <1>
                                          mov
1145 00006C6D 668B4606
                                <1>
                                                ax, [esi+bs_FS_BytesPerSec]
1146 00006C71 66894711
                                <1>
                                                [edi+LD_FS_BytesPerSec], ax
                                          mov
                                                al, [esi+bs_FS_SecPerTrack]
1147 00006C75 8A462E
                                <1>
                                          mov
1148 00006C78 28E4
                                                ah, ah; 09/12/2017
                                <1>
1149 00006C7A 6689471E
                                <1>
                                                [edi+LD_FS_SecPerTrack], ax
                                         mov
1150 00006C7E 8A462F
                                <1>
                                          mov
                                                al, [esi+bs_FS_Heads]
1151 00006C81 66894720
                                <1>
                                                [edi+LD_FS_NumHeads], ax
                                         mov
1152
                                <1>
1153 00006C85 8B4628
                                <1>
                                                eax, [esi+bs_FS_UnDelDirD]
                                          mov
1154 00006C88 894722
                                <1>
                                                [edi+LD FS UnDelDirD], eax
                                          mov
1155 00006C8B 8B4618
                                <1>
                                          mov
                                                eax, [esi+bs_FS_MATLocation]
1156 00006C8E 89470C
                                <1>
                                          mov
                                                 [edi+LD_FS_MATLocation], eax
1157 00006C91 8B461C
                                <1>
                                          mov
                                                eax, [esi+bs_FS_RootDirD]
1158 00006C94 894708
                                <1>
                                                [edi+LD FS_RootDirD], eax
1159 00006C97 8B460C
                                <1>
                                          mov
                                                eax, [esi+bs FS BeginSector]
1160 00006C9A 89476C
                               <1>
                                          mov
                                                [edi+LD_FS_BeginSector], eax
1161 00006C9D 8B4610
                               <1>
                                                eax, [esi+bs FS VolumeSize]
                                         mov
                               <1>
1162 00006CA0 894770
                                        mov
                                                [edi+LD FS VolumeSize], eax
1163
                                <1>
1164 00006CA3 89FE
                                <1>
                                                esi, edi
                                         mov
                                                eax, [esi+LD FS MATLocation]
1165 00006CA5 8B460C
                                <1>
                                          mov
1166
                                <1>
                                          ;add eax, [edi+LD FS BeginSector]
                                <1> read_fd_MAT_sector_again:
1167
                                     ;mov ebx, DOSBootSectorBuff
1168
                                <1>
1169
                                <1>
                                          ; mov
                                               ecx, 1
1170 00006CA8 B101
                                <1>
                                          mov
                                                cl, 1
1171 00006CAA E8B38B0000
                               <1>
                                         call chs read
                                      mov esi, ebx
1172 00006CAF 89DE
                               <1>
1173 00006CB1 7301
                                <1>
                                         jnc
                                                short use_fdfs_mat_sector_params
                                         ;jmp short read_fd_boot_sector_retn
1174
                               <1>
1175 00006CB3 C3
                               <1>
                                         retn
1176
                                <1> use_fdfs_mat_sector_params:
                               <1>
1177 00006CB4 8B460C
                                         mov eax, [esi+FS MAT DATLocation]
1178 00006CB7 894714
                               <1>
                                                [edi+LD FS DATLocation], eax
1179 00006CBA 8B4610
                                <1>
                                                eax, [esi+FS_MAT_DATScount]
                                          mov
```

```
1180 00006CBD 894718
                                                  <1>
                                                                           [edi+LD_FS_DATSectors], eax
                                                                 mov
1181 00006CC0 8B4714
                                                  <1>
                                                                 mov
                                                                            eax, [edi+FS MAT FreeSectors]
                                                 <1>
1182 00006CC3 894774
                                                                           [edi+LD_FS_FreeSectors], eax
                                                                 mov
                                                 <1>
1183 00006CC6 8B4618
                                                                  mov
                                                                            eax, [esi+FS MAT FirstFreeSector]
1184 00006CC9 894778
                                                  <1>
                                                                           [edi+LD_FS_FirstFreeSector], eax
                                                                 mov
1185
                                                  <1>
1186 00006CCC 89FE
                                                 <1>
                                                                  mov
                                                                            esi, edi
                                                <1>
                                                                           eax, [esi+LD_FS_RootDirD]
1187 00006CCE 8B4608
                                                                 mov
1188
                                                  <1> read_fd_RDT_sector_again:
                                                 <1> ;mov ebx, DOSBootSectorBuff
1189
                                                 <1>
1190
                                                                  ;mov cx, 1
1191 00006CD1 B101
                                                  <1>
                                                                 mov
                                                                           cl, 1
                                                                 call chs_read
                                                 <1>
1192 00006CD3 E88A8B0000
                                                <1> mov esi, ebx
<1> jc short read_fd_F
<1> use_fdfs_RDT_sector_params:
1193 00006CD8 89DE
1194 00006CDA 7220
                                                                           short read_fd_RDT_sector_retn
1195
                                          <1> use_fdfs_RDT_sector_params:
<1> mov eax, [esi+FS_RDT_VolumeSerialNo]
<1> mov [edi+LD_FS_VolumeSerial], eax
<1> push edi
<1> ;mov ecx, 16
<1> mov cl, 16
<1> add esi, FS_RDT_VolumeName
<1> add edi, LD_FS_VolumeName
<1> rep movsd; 64 bytes
<1> pop esi
<1> mov byte [esi+LD_FATType], 0
1196 00006CDC 8B461C
1197 00006CDF 894728
1198 00006CE2 57
1199
1200 00006CE3 B110
1201 00006CE5 83C640
1202 00006CE8 83C72C
1203 00006CEB F3A5
1204 00006CED 5E
1205 00006CEE C6460300
1206 00006CF2 C64604A1
                                                 <1>
                                                                  mov
                                                                           byte [esi+LD FSType], 0A1h
                                                 <1>
1207 00006CF6 E9A5000000
                                                                  jmp
                                                                                 loc_cont_use_fd_boot_sector_params
1208
                                                  <1>
1209
                                                  <1> read_fd_RDT_sector_stc_retn:
1210 00006CFB F9
                                                  <1>
                                                                stc
1211
                                                   <1> read fd RDT sector retn:
1212 00006CFC C3
                                                  <1>
                                                                  retn
1213
                                                  <1>
                                                  <1> use_fd_fatfs_boot_sector_params:
1214
1215 00006CFD 807E2629
                                                  <1>
                                                                  cmp
                                                                           byte [esi+BS_BootSig], 29h
1216 00006D01 75F8
                                                 <1>
                                                                           short read fd RDT sector stc retn
                                              <1>
1217 00006D03 807E15F0
                                                          cmp ~;
jb short read_1
push edi
add edi, LD_BPB
;mov ecx, 16
mov cl, 16
mov sd; 64
                                                                           byte [esi+BPB_Media], 0F0h
                                                                            short read fd RDT sector retn
1218 00006D07 72F3
                                                  <1>
1219 00006D09 57
                                                 <1>
                                             <1>
1220 00006D0A 83C706
                                            ;mov ecx, 16

important color c
1221
                                                  <1>
1222 00006D0D B110
1223 00006D0F F3A5
1224 00006D11 5E
1225 00006D12 31C0
1226 00006D14 89466C
1227 00006D17 668B461C
1228 00006D1B 8A4E16
1229 00006D1E F7E1
1230
1231 00006D20 668B5614
1232 00006D24 66895660
1233
1234 00006D28 6601D0
1235 00006D2B 894664
1236 00006D2E 894668
1237 00006D31 668B5617
1238
                                                                 ;;shl edx, 5; * 32 (Size of a directory entry)
1239
                                                               ;;add edx, 511
1240
                                                   <1>
1241
                                                   <1>
                                                                  ;add dx, 511
1242
                                                   <1>
                                                                  ;;shr edx, 9; edx = ((edx*32)+511) / 512
                                                   <1>
1243
1244 00006D35 6683C20F
                                                  <1>
                                                                  add dx, 15; 06/07/2016 (+(512/32)-1)
                                                                           dx, 4 ; / 16 (==16 entries per sector)
1245 00006D39 66C1EA04
                                                  <1>
                                                                  shr
                                                                 add [esi+LD DATABegin], edx; + rd sectors
1246 00006D3D 015668
                                                  <1>
1247
                                                  <1>
                                                                 ;movzx eax, word [esi+LD_BPB+BPB_TotalSec16]
1248 00006D40 668B4619
                                                  <1>
                                                                  mov ax, [esi+LD_BPB+BPB_TotalSec16]
1249 00006D44 894670
                                                  <1>
                                                                           [esi+LD TotalSectors], eax
                                                                 mov
1250 00006D47 2B4668
                                                  <1>
                                                                  sub eax, [esi+LD_DATABegin]
1251
                                                  <1>
                                                                  ;movzx ecx, byte [esi+LD BPB+BPB SecPerClust]
1252 00006D4A 8A4E13
                                                  <1>
                                                                  mov cl, [esi+LD_BPB+BPB_SecPerClust]
1253 00006D4D 80F901
                                                 <1>
                                                                  cmp cl, 1
1254 00006D50 7605
                                                  <1>
                                                                  jna
                                                                          short save_fd_fatfs_cluster_count
1255
                                                                  ;sub edx, edx
                                                  <1>
1256 00006D52 6629D2
                                                  <1>
                                                                  sub dx, dx; 0
1257
                                                  <1>
                                                                  ;sub dl, dl; 06/07/2016
1258 00006D55 F7F1
                                                   <1>
                                                                  div
                                                                           ecx
1259
                                                   <1> save_fd_fatfs_cluster_count:
1260 00006D57 894678
                                                   <1>
                                                                  mov [esi+LD_Clusters], eax
1261
                                                    <1>
                                                                  ; Maximum Valid Cluster Number = EAX +1
1262
                                                    <1>
                                                    <1>
1263
                                                                  ; with 2 reserved clusters= EAX +2
1264
                                                   <1>
1265
                                                   <1> reset_FAT_buffer_decriptors:
1266 00006D5A 29C0
                                                   <1>
                                                                 sub eax, eax; 0
                                                                             [FAT_BuffValidData], al ; 0
1267 00006D5C A2[9A600100]
                                                   <1>
                                                                  mov
1268 00006D61 A2[9B600100]
                                                                             [FAT BuffDrvName], al; 0
                                                   <1>
                                                                  mov
1269 00006D66 A3[9E600100]
                                                   <1>
                                                                           [FAT BuffSector], eax; 0
                                                                  mov
1270
                                                   <1>
                                                   <1> read fd_FAT_sectors:
1271
1272 00006D6B BB001C0900
                                                                 mov ebx, FAT Buffer
                                                   <1>
1273 00006D70 668B4614
                                                  <1>
                                                                          ax, [esi+LD_BPB+BPB_RsvdSecCnt]
                                                                  mov
                                                                  ;mov ecx, 3
1274
                                                   <1>
1275 00006D74 B103
                                                                           cl, 3 ; 3 sectors
                                                   <1>
                                                                  mov
                                                                  call chs read
1276 00006D76 E8E78A0000
                                                  <1>
                                                                 jc short read_fd_FAT_sectors_retn
1277 00006D7B 7240
                                                   <1>
1278
                                                   <1> use_fd_FAT_sectors:
1279 00006D7D 8A4602
                                                   <1>
                                                              mov al, [esi+LD_PhyDrvNo]
1280 00006D80 0441
                                                   <1>
                                                                  add
                                                                            al, 'A'
1281 00006D82 A2[9B600100]
                                                   <1>
                                                                  mov
                                                                           [FAT BuffDrvName], al
                                                                 mov
1282 00006D87 C605[9A600100]01
                                                   <1>
                                                                           byte [FAT_BuffValidData], 1
1283 00006D8E E82B000000
                                                   <1>
                                                                  call fd_init_calculate_free_clusters
1284 00006D93 7228
                                                   <1>
                                                                  jc short read_fd_FAT_sectors_retn
```

```
1285
                                <1>
1286
                                <1> loc use fd boot sector params FAT:
1287 00006D95 C6460301
                                      mov byte [esi+LD_FATType], 1; FAT 12
                               <1>
1288 00006D99 C6460401
                               <1>
                                         mov byte [esi+LD_FSType], 1
                               <1>
1289 00006D9D 8B462D
                                        mov eax, [esi+LD BPB+VolumeID]
                          <1> loc cont_use_fd_boot_sector_params:
1290
1291 00006DA0 8A7E02
1292 00006DA3 887E7D
                                               [esi+LD DParamEntry], bh
1293 00006DA6 88FB
1294 00006DA8 80C341
1295 00006DAB 881E
                                        mov byte [esi+LD_DiskType],
mov byte [esi+LD_LBAYes], 0
                                               byte [esi+LD_DiskType], 1
1296 00006DAD C6460101
1297 00006DB1 C6460500
1298 00006DB5 C6467C00
                              <1>
                                         mov byte [esi+LD_PartitionEntry], 0
                               <1>
1299 00006DB9 C6467E06
                                       mov byte [esi+LD_MediaChanged], 6 ; Volume Name Reset
1300
                               <1>
1301
                               <1> read_fd_FAT_sectors_retn:
1302 00006DBD C3
                               <1>
                                        retn
1303
                                <1>
                                <1> fd_init_calculate_free_clusters:
1304
                                       ; 09/12/201<del>7</del>
1305
                                <1>
1306
                                <1>
                                         ; 10/01/2016 (TRDOS 386 = TRDOS v2.0)
1307
                                <1>
                                         ; 04/07/2009
1308
                                <1>
                                       ; INPUT ->
                                       ; ESI = Logical DOS drive description table address
                                <1>
1309
                                         ; OUTPUT ->
1310
                                <1>
                                        ; [ESI+LD_FreeSectors] will be set
1311
                                <1>
1312
                                <1>
1313 00006DBE 29C0
                                <1>
                                         sub
                                               eax, eax
                                               [esi+LD_FreeSectors], eax ; 0
1314 00006DC0 894674
                               <1>
                                         mov
1315 00006DC3 B002
                               <1>
                                         mov al, 2; eax = 2
1316
                                <1>
                               <1> fd_init_loop_get_next_cluster:
1317
1318 00006DC5 E830000000
                               <1>
                                       call fd init get next cluster
1319 00006DCA 722D
                                <1>
                                               short fd_init_calculate_free_clusters_retn
1320
                                <1>
1321
                                <1> fd init free fat clusters:
1322
                                <1>
                                       ;cmp eax, 0
1323
                                <1>
                                              short fd_init_pass_inc_free_cluster_count
                                         ;ja
1324
                                         ;and eax, eax
                               <1>
                                         ;jnz short fd_init_pass_inc_free_cluster_count
1325
                               <1>
                                        and ax, ax
jnz short fd_init_pass_inc_free_cluster_count
1326 00006DCC 6621C0
                               <1>
1327 00006DCF 7504
                               <1>
                                         ;inc dword [esi+LD FreeSectors]
1328
                               <1>
1329 00006DD1 66FF4674
                                <1>
                                         inc word [esi+LD_FreeSectors]
1330
                                <1>
1331
                                <1> fd init pass inc free cluster count:
                                     ;mov eax, [FAT_CurrentCluster]
1332
                                <1>
1333 00006DD5 66A1[96600100]
                               <1>
                                         mov
                                               ax, [FAT_CurrentCluster]
1334
                                         ;cmp eax, [esi+LD_Clusters]
                                <1>
1335 00006DDB 663B4678
                               <1>
                                         cmp ax, [esi+LD_Clusters]
1336 00006DDF 7704
                                <1>
                                         jа
                                               short short retn_from_fd_init_calculate_free_clusters
1337
                               <1>
                                         ;inc eax
1338 00006DE1 6640
                               <1>
                                         inc ax
1339 00006DE3 EBE0
                               <1>
                                         jmp
                                               short fd_init_loop_get_next_cluster
1340
                               <1>
                               <1> retn from fd init calculate free clusters:
1342 00006DE5 8A4613
                                         mov al, [esi+LD_BPB+BPB_SecPerClust]
                               <1>
1343 00006DE8 3C01
                               <1>
                                         cmp
                                               al, 1
                                         jna short fd_init_calculate_free_clusters_retn
1344 00006DEA 760D
                               <1>
1345
                               <1>
                                        ;movzx eax, al
1346 00006DEC 30E4
                               <1>
                                        xor ah, ah; 09/12/2017
1347
                               <1>
                                         ;mov ecx, [esi+LD_FreeSectors]
1348 00006DEE 668B4E74
                               <1>
                                         mov cx, [esi+LD_FreeSectors]; Count of free clusters
1349
                                <1>
                                         ;mul ecx
1350 00006DF2 66F7E1
                                <1>
                                         mul
                                               CX
                               <1>
                                         ;mov [esi+LD FreeSectors], eax
1352 00006DF5 66894674
                                         mov
                               <1>
                                               [esi+LD_FreeSectors], ax
1353
                                <1> fd_init_calculate_free_clusters_retn:
1354 00006DF9 C3
                                <1>
                                         retn
1355
                                <1>
1356
                                <1> fd_init_get_next_cluster:
                                      ; 04/02/2016
1357
                                <1>
1358
                                <1>
                                         ; 02/02/2016
                                       ; 10/01/2016 (TRDOS 386 = TRDOS v2.0)
1359
                                <1>
1360
                                <1>
                                         ; 04/07/2009
                                        ; INPUT ->
1361
                                <1>
1362
                                <1>
                                       ; EAX = Current cluster
1363
                                <1>
                                             ESI = Logical DOS drive description table address
                                         ; EDX = 0
1364
                                <1>
                                         ; OUTPUT ->
1365
                                <1>
                                              EAX = Next cluster
1366
                                <1>
                                <1>
1368 00006DFA A3[96600100]
                                <1>
                                               [FAT_CurrentCluster], eax
                                <1> fd_init_get_next_cluster_readnext:
1369
                                         sub edx, edx; 0
1370 00006DFF 29D2
                                <1>
1371 00006E01 BB00040000
                                               ebx, 1024 ; 400h
                               <1>
                                         mov
1372 00006E06 F7F3
                               <1>
                                         div ebx
1373
                               <1>
                                         ; EAX = Count of 3 FAT sectors
                                         ; EDX = Buffer entry index
1374
                               <1>
1375 00006E08 89C1
                               <1>
                                         mov ecx, eax
                                         ;mov eax, 3
1376
                                <1>
1377 00006E0A B003
                               <1>
                                         mov
                                               al, 3
1378 00006E0C F7E2
                               <1>
                                                edx ; Multiply by 3
                                         mul
1379 00006E0E 66D1E8
                               <1>
                                         shr
                                               ax, 1; Divide by 2
1380 00006E11 89C3
                               <1>
                                         mov
                                               ebx, eax; Buffer byte offset
1381 00006E13 81C3001C0900
                                               ebx, FAT Buffer
                               <1>
                                         add
1382 00006E19 89C8
                               <1>
                                         mov
                                               eax, ecx
1383
                                <1>
                                         ;mov
                                               edx, 3
1384 00006E1B 66BA0300
                               <1>
                                               dx, 3
                                         mov
1385 00006E1F F7E2
                               <1>
                                         mul
                                              edx
1386
                               <1>
                                         ; EAX = FAT Beginning Sector
1387
                               <1>
                                         ; EDX = 0
1388 00006E21 8A0E
                                <1>
                                         mov cl, [esi+LD_Name]
1389
                                <1>
                                         ;cmp byte [FAT_BuffValidData], 0
```

```
;jna short fd_init_load_FAT_sectors0
1391 00006E23 3A0D[9B600100]
                                <1>
                                          cmp
                                               cl, [FAT BuffDrvName]
1392 00006E29 751E
                                               short fd init load FAT sectors0
                                <1>
                                          jne
1393 00006E2B 3B05[9E600100]
                                <1>
                                          cmp eax, [FAT_BuffSector]
1394 00006E31 751C
                                <1>
                                               short fd init load FAT sectors1
                                         jne
1395
                                <1>
                                          ;mov eax, [FAT_CurrentCluster]
1396 00006E33 A0[96600100]
                                        mov al, [FAT CurrentCluster]
                                <1>
                                      ;shr eax, 1
1397
                                <1>
1398 00006E38 D0E8
                                <1>
                                          shr
                                                al, 1
                             <1>
                                         mov
1399 00006E3A 668B03
                                               ax, [ebx]
1400 00006E3D 7306
                               <1>
                                          jnc short fd_init_gnc_even
1401 00006E3F 66C1E804
                                <1>
                                          shr
                                               ax, 4
                                <1> fd_init_gnc_clc_retn:
1402
1403 00006E43 F8
                                <1>
                                          clc
1404 00006E44 C3
                                <1>
                                          retn
1405
                                <1>
                                <1> fd_init_gnc_even:
1407 00006E45 80E40F
                                <1>
                                          and ah, 0Fh
1408 00006E48 C3
                                <1>
                                          retn
1409
                                <1>
                                <1> fd_init_load_FAT_sectors0:
1410
1411 00006E49 880D[9B600100]
                                <1>
                                        mov [FAT BuffDrvName], cl
                                <1> fd_init_load_FAT_sectors1:
1412
1413 00006E4F C605[9A600100]00
                                <1> mov byte [FAT_BuffValidData], 0
mov [FAT_BuffSector], eax add eax, [esi+LD_FATBegin]
1414 00006E56 A3[9E600100]
                                <1>
                                         cmp cx, 3
jna short fdinit_pass_fix_sector_count_3
1422 00006E72 7605
                                <1>
1423
                                <1>
                                         ;mov ecx, 3
                                         mov ecx, 3
1424 00006E74 B903000000
                                <1>
                                <1> fdinit_pass_fix_sector_count_3:
1425
1426 00006E79 E8E4890000
                                <1> call chs read
1427 00006E7E 730D
                                <1>
                                          jnc
                                               short fd_init_FAT_sectors_no_load_error
1428 00006E80 C605[9A600100]00
                                <1>
                                                byte [FAT_BuffValidData], 0
                                         mov
1429
                                <1>
                                                ; Drv not ready or read Error !
1430 00006E87 B80F000000
                                <1>
                                         mov
                                               eax, ERR_DRV_NOT_RDY ; 15
1431
                                 <1>
                                         ;xor edx, edx
1432 00006E8C C3
                                <1>
                                          retn
1433
                                 <1>
1434
                                 <1> fd_init_FAT_sectors_no_load_error:
                                       mov byte [FAT_BuffValidData], 1
mov eax, [FAT_CurrentCluster]
1435 00006E8D C605[9A600100]01
                                <1>
1436 00006E94 A1[96600100]
                                <1>
1437 00006E99 E961FFFFFF
                                                 fd_init_get_next_cluster_readnext
                                <1>
                                          jmp
1438
                                <1>
1439
                                <1> get_FAT_volume_name:
                                       ; 10/01/2016 (TRDOS 386 = TRDOS v2.0)
1440
                                 <1>
1441
                                 <1>
                                         ; 12/09/2009
                                         ; INPUT ->
1442
                                 <1>
1443
                                 <1>
                                               BH = Logical DOS drive number (0,1,2,3,4...)
1444
                                 <1>
                                        ;
                                                 BL = 0
                                          ; OUTPUT ->
1445
                                 <1>
                                               CF = 0 -> ESI = Volume name address
1446
                                 <1>
                                         ;
                                 <1>
                                                CF = 1 -> Root volume name not found
1447
1448
                                 <1>
1449
                                 <1>
                                          ;mov ah, OFFh
1450
                                 <1>
                                          ;mov al, [Last_Dos_DiskNo]
1451
                                 <1>
                                               al, bh
                                          ;cmp
1452
                                 <1>
                                          ;jb
                                                 short loc_gfvn_dir_load_err
1453
                                 <1>
1454 00006E9E 89DE
                                <1>
                                                esi, ebx
                                          mov
1455 00006EA0 81E600FF0000
                                <1>
                                          and
                                                esi, OFFOOh; esi = bh
1456 00006EA6 81C600010900
                                <1>
                                          add
                                                esi, Logical DOSDisks
1457 00006EAC 8A06
                                <1>
                                                al, [esi+LD_Name]
                                          mov
1458 00006EAE 8A6603
                                <1>
                                                 ah, [esi+LD_FATType]
                                          mov
1459 00006EB1 80FC01
                                <1>
                                                 ah, 1
                                          cmp
1460 00006EB4 7210
                                <1>
                                          jb
                                                short loc_gfvn_dir_load_err
1461 00006EB6 3C41
                                                al, 'A'
                                <1>
                                          cmp
1462 00006EB8 720C
                                <1>
                                                 short loc_gfvn_dir_load_err
                                          jb
1463 00006EBA 80FC02
                                <1>
1464 00006EBD 7708
                                <1>
                                                 short get FAT32 root cluster
                                          jа
1465
                                <1>
1466 00006EBF E8634E0000
                                <1>
                                                 load FAT root directory
1467 00006EC4 730B
                                <1>
                                          jnc
                                                  short loc_get_volume_name
1468
                                <1>
1469
                                <1> loc_gfvn_dir_load_err:
1470 00006EC6 C3
                                 <1>
                                          retn
                                 <1>
                                <1> get_FAT32_root_cluster:
1473 00006EC7 8B4632
                                         mov eax, [esi+LD_BPB+BPB RootClus]
1474 00006ECA E8E34E0000
                                <1>
                                          call load_FAT_sub_directory
1475 00006ECF 7224
                                <1>
                                          jc short loc_get_volume_name_retn
                                <1>
1477
                                <1> loc_get_volume_name:
1478 00006ED1 BE00000800
                                <1>
                                          mov esi, Directory_Buffer
1479 00006ED6 6631C9
                                          xor cx, cx; 0
                               <1>
1480
                                <1> check_root_volume_name:
1481 00006ED9 8A06
                                <1>
                                         mov al, [esi]
1482 00006EDB 08C0
                               <1>
                                                al, al
                                          or
                                                 short loc get volume name retn
1483 00006EDD 7416
                                <1>
                                         jΖ
1484 00006EDF 807E0B08
                                <1>
                                                 byte [esi+0Bh], 08h
                                         cmp
                                                 short loc get volume name retn
1485 00006EE3 7410
                                <1>
                                          jе
1486 00006EE5 663B0D[AF600100] <1>
                                                 cx, [DirBuff LastEntry]
                                         cmp
1487 00006EEC 7308
                                <1>
                                         jnb
                                                 short pass_check_root_volume_name
1488 00006EEE 6641
                                <1>
                                          inc
1489 00006EF0 83C620
                                <1>
                                                 esi, 32
                                          add
1490 00006EF3 EBE4
                                <1>
                                          jmp
                                                 short check_root_volume_name
1491
                                <1>
                                <1> loc_get_volume_name_retn:
1492
1493 00006EF5 C3
                                <1>
                                 <1>
1494
```

<1>

1390

```
<1> pass_check_root_volume_name:
1496 00006EF6 803D[AB600100]03
                               <1>
                                       cmp
                                            byte [DirBuff FATType], 3
1497 00006EFD 7230
                               <1>
                                             short loc_get_volume_name_retn_xor
1498
                              <1>
                              <1> mov <1> mov <1>
1499 00006EFF BB001C0900
                                             ebx, FAT_Buffer
1500 00006F04 BE00010900
                                             esi, Logical_DOSDisks
                                             eax, eax
                                             ah, [DirBuff_DRV]
                                             ah, 'A'
                                             esi, eax
                                             eax, [DirBuff_Cluster]
                                       call get next cluster
                                             short loc_gfvn_load_FAT32_dir_cluster
                              <1>
1509 00006F22 83F801
                              <1>
                                      cmp
                                              eax, 1
1510 00006F25 F5
                              <1>
                                       cmc
                                     retn
1511 00006F26 C3
                              <1>
1512
                              <1>
1513
                              <1> loc_gfvn_load_FAT32_dir_cluster:
                              <1> call load_FAT_sub_directory
1514 00006F27 E8864E0000
1515 00006F2C 73A3
                              <1>
                                            short loc_get_volume_name
1516 00006F2E C3
                              <1>
                                       retn
1517
                              <1>
1518
                              <1> loc_get_volume_name_retn_xor:
                                     xor eax, eax
1519 00006F2F 31C0
                              <1>
1520 00006F31 C3
                              <1>
                                       retn
                              <1>
1522
                              <1> get_media_change_status:
                                    ; 10/01/2016 (TRDOS 386 = TRDOS v2.0)
1523
                               <1>
1524
                               <1>
                                       ; 09/09/2009
                                     ; INPUT:
1525
                               <1>
                                     ; DL = Drive number (physical)
1526
                               <1>
                                       ; OUTPUT: clc & AH = 6 media changed
1527
                               <1>
                                     ; clc & AH = 0 media not changed
1528
                               <1>
                                            stc -> Drive not ready or an error
1529
                               <1>
                                     ;
1530
                              <1>
                                   mov ah, 16h
call int13h
cmp ah, 06h
je short lo
1531 00006F32 B416
                              <1>
1532 00006F34 E8F4D2FFFF
                              <1>
                                       cmp ah, 06h
je short lo
1533 00006F39 80FC06
                              <1>
1534 00006F3C 7405
                                             short loc_gmc_status_retn
                              <1>
1535 00006F3E 08E4
                              <1> or <1> jz
                              <1>
                                             ah, ah
1536 00006F40 7401
                                             short loc gmc status retn
                              <1> loc_gmc_status_stc_retn:
1537
1538 00006F42 F9
                              <1> stc
1539
                               <1> loc_gmc_status_retn:
1540 00006F43 C3
                               <1>
                                      retn
                                 %include 'trdosk3.s'; 06/01/2016
2307
                               1
   2
                               <1>; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.0) - MAIN PROGRAM : trdosk3.s
   3
                               <1> ; Last Update: 31/12/2017
   4
   5
                               <1> ; Beginning: 06/01/2016
   6
   7
                               <1> ; ------
   8
                               <1>; Assembler: NASM version 2.11 (trdos386.s)
   9
                               10
                               <1>; Derived from TRDOS Operating System v1.0 (8086) source code by Erdogan Tan
                               <1>; MAINPROG.ASM (09/11/2011)
  11
                               12
                               <1> ; MAINPROG.ASM [ TRDOS KERNEL - COMMAND EXECUTER SECTION - MAIN PROGRAM ]
  13
                               <1>; (c) 2004-2011 Erdogan TAN [ 17/01/2004 ] Last Update: 09/11/2011
  14
                               <1>; CMD INTR.ASM [ TRDOS Command Interpreter Procedure ] Last Update: 09/11/2011
  15
                               <1> ; DIR.ASM [ DIRECTORY FUNCTIONS ] Last Update: 09/10/2011
  16
  17
                               <1>; FILE.ASM [ FILE FUNCTIONS ] Last Update: 09/10/2011
  18
                               <1>
  19
                               <1> change_current_drive:
                                   \frac{1}{7} 16/10/\frac{1}{2}016
  20
                                       ; 02/02/2016
  21
                               <1>
  22
                               <1>
                                       ; 15/01/2016 (TRDOS 386 = TRDOS v2.0)
                                      ; 18/08/2011
  23
                               <1>
                                      ; 09/09/2009
  24
                               <1>
                                      ; INPUT:
  25
                               <1>
                                       ; DL = Logical DOS Drive Number
  26
                               <1>
                                      ; OUTPUT:
  27
                               <1>
                                       ; cf=1 -> Not successful
; EAX = Error code
  28
                               <1>
  29
                               <1>
                                       ; cf=0 ->
  30
                               <1>
                                      ; EAX = 0 (successful)
  31
                               <1>
  32
                               <1>
  33 00006F44 31DB
                               <1>
                                       xor ebx, ebx
  34 00006F46 88D7
                               <1>
                                       mov bh, dl
                               <1>
                                       ;cmp dl, 1
  36
  37
                                       ;jna short loc_ccdrv_initial_media_change_check
                               <1>
  38
                               <1>
                                       ;cmp bh, [Last_Dos_DiskNo]
  39
                               <1>
                                       ;ja short loc_ccdrv_drive_not_ready_err
  40
                               <1>
  41
                               <1> loc ccdrv initial media change check:
  42 00006F48 BE00010900
                              <1>
                                             esi, Logical_DOSDisks
                                       mov
  43 00006F4D 01DE
                              <1>
                                       add
                                             esi, ebx
                              <1> loc_ccdrv_dos_drive_name_check:
  44
  45 00006F4F 80FA02
                              <1>
                                       cmp dl, 2
  46 00006F52 720F
                              <1>
                                       jb
                                             short loc_ccdrv_dos_drive_name_check_ok
                              <1>
                                             al, [esi+LD_Name] al, 'A'
  48 00006F54 8A06
                              <1>
                                       mov
  49 00006F56 2C41
                              <1>
                                       sub
  50 00006F58 38D0
                              <1>
                                             al, dl
                                       cmp
  51 00006F5A 7407
                              <1>
                                       jе
                                             short loc_ccdrv_dos_drive_name_check_ok
                              <1>
                              <1> loc_ccdrv_drive_not_ready_err:
  53
  54
                              <1>
                                      ; 16/10/2016 (15h -> 15)
  55 00006F5C B80F000000
                              <1>
                                       mov eax, 15; Drive not ready
                              <1> loc_change_current_drive_stc_retn:
  57 00006F61 F9
                              <1>
  58 00006F62 C3
                               <1>
                                       retn
```

1495

```
59
                               <1>
                              <1> loc ccdrv dos drive name check ok:
                                       mov ah, [esi+LD_MediaChanged]
 61 00006F63 8A667E
                              <1>
 62 00006F66 80FC06
                             <1>
                                        cmp ah, 6 ; VOLUME NAME CHECK/MOVE SIGN
 63 00006F69 7455
                              <1>
                                              short loc_ccdrv_get_FAT_volume_name_0
                              <1>
 65 00006F6B 80FA01
                             <1>
                                        cmp dl, 1
 66 00006F6E 777D
                              <1>
                                             short loc_gmcs_init_drv_hd
                                      ja
 67
                              <1>
                              <1> loc_gmcs_init_drv_fd:
 68
 69 00006F70 08E4
                              <1>
                                      or ah, ah
                                        ; AH = 1 is initialization sign (invalid fd parameter)
                              <1>
                                        jnz short loc_ccdrv_call_fd_init
 71 00006F72 7517
                              <1>
 72
                              <1>
 73 00006F74 E8B9FFFFF
                                       call get_media_change_status
                              <1>
                                              short loc_ccdrv_drive_not_ready_err
 74 00006F79 72E1
                              <1>
                              <1>
                                    and
 76 00006F7B 20E4
                              <1>
                                             ah, ah
 77 00006F7D 7476
                              <1>
                                              short loc_change_current_drv3
                                       jz
                              <1>
                              <1>
 79 00006F7F 80F406
                                        xor
                                             ah, 6
                                             short loc_ccdrv_drive_not ready err
 80 00006F82 75D8
                              <1>
                                      jnz
                              <1>
 81
                              <1> loc_ccdrv_call_fd_init_check_vol_id:
 83 00006F84 E8440A0000
                              <1>
                                    call get_volume_serial_number
 84 00006F89 730D
                              <1>
                                        jnc short loc_ccdrv_check_vol_serial
                              <1>
                              <1> loc_ccdrv_call_fd_init:
 86
                                     call floppy_drv_init
jnc short loc_reset_drv_fd_current_dir
 87 00006F8B E872FCFFFF
                              <1>
 88 00006F90 731A
                              <1>
 89
                              <1>
                              <1> loc ccdrv fdinit fail retn:
 90
                              <1> ; 16/10/2016 <1> mov 63 1
 91
 92 00006F92 B80F000000
                              <1>
                                       mov eax, 15; Drive not ready
                                      retn
 93 00006F97 C3
                              <1>
 94
                              <1>
                              <1> loc ccdrv check vol serial:
                              <1> mov [Current_VolSerial], eax
 96 00006F98 A3[7C590100]
 97
                              <1>
                                        ;mov
                                             dl, bh
                                        call floppy_drv_init
 98 00006F9D E860FCFFFF
                              <1>
 99 00006FA2 72EE
                                             short loc_ccdrv_fdinit_fail_retn
                              <1>
                              <1>
                                    cmp eax, [Current_VolSerial]
je short loc_change_current_drv2
                             <1>
101 00006FA4 3B05[7C590100]
102 00006FAA 7445
                              <1>
103
                              <1>
112
                              <1> loc_ccdrv_get_FAT_volume_name_0:
113 00006FC0 8A4603
                              <1> mov al, [esi+LD_FATType]
114 00006FC3 08C0
                              <1>
                                              al, al
115 00006FC5 742A
                              <1>
                                              short loc_change_current_drv2
                                       jг
                              <1>
116
                                     push esi
117 00006FC7 56
                              <1>
                                       cmp al, 2
118 00006FC8 3C02
                              <1>
119 00006FCA 7705
                              <1>
                                              short loc_ccdrv_get_FAT32_vol_name
                              <1>
                              <1> loc_ccdrv_get_FAT2_16_vol_name:
122 00006FCC 83C631
                              <1> add esi, LD_BPB + VolumeLabel
                                        jmp short loc_ccdrv_get_FAT_volume_name_1
123 00006FCF EB03
                              <1>
124
                              <1>
                              <1> loc_ccdrv_get_FAT32_vol_name:
126 00006FD1 83C64D
                            <1> add esi, LD_BPB + FAT32_VolLab
127
                              <1> loc_ccdrv_get_FAT_volume_name_1:
                              <1> push ebx
128 00006FD4 53
129 00006FD5 56
                                       push esi
                              <1>
                                       call get_FAT_volume_name pop edi
130 00006FD6 E8C3FEFFFF
                              <1>
131 00006FDB 5F
                              <1>
                                      pop ebx
132 00006FDC 5B
                              <1>
                                       ; BL = 0
                              <1>
133
                              <1>
134 00006FDD 720B
                                        jc short loc_change_current_drv1
135 00006FDF 20C0
                              <1>
                                       and al, al
136 00006FE1 7407
                              <1>
                                      jz short loc_change_current_drv1
137
                              <1>
                              <1> loc ccdrv move FAT volume name:
138
139 00006FE3 B90B000000
                              <1>
                                     mov ecx, 11
140 00006FE8 F3A4
                               <1>
                                        rep
                                              movsb
141
                               <1>
                               <1> loc_change_current_drv1:
143 00006FEA 5E
                               <1>
                                      pop esi
144 00006FEB EB04
                               <1>
                                              short loc_change_current_drv2
                                        jmp
                               <1>
                               <1> loc_gmcs_init_drv_hd:
146
147 00006FED 08E4
                               <1>
148 00006FEF 7404
                                        jz
                               <1>
                                             short loc change current drv3
                                        ; BL = 0, BH = Logical DOS drive number
149
                               <1>
150
                               <1> loc_change_current_drv2:
                                        mov byte [esi+LD_MediaChanged], 0
151 00006FF1 C6467E00
                               <1>
                               <1> loc_change_current_drv3:
                                       mov [Current_Drv], bh
153 00006FF5 883D[86590100]
                               <1>
154
                               <1>
155
                               <1>
                                        ;call restore_current_directory
156
                               <1>
                                        ;retn
157
                               <1>
158
                               <1> restore_current_directory:
                                       ; 11/02/2016
159
                               <1>
160
                               <1>
                                        ; 15/01/2016 (TRDOS 386 = TRDOS v2.0)
                                        ; 25/01/2010
161
                               <1>
                               <1>
                                        ; 12/10/2009
162
                               <1>
163
```

```
164
                                 <1>
                                         ; INPUT:
165
                                 <1>
                                          ; ESI = Logical DOS Drive Description Table
166
                                 <1>
167
                                 <1>
                                          ; OUTPUT:
                                 <1>
168
                                          ; ESI = Logical DOS Drive Description Table
169
                                 <1>
                                              EDI = offset Current_Dir_Drv
                                 <1>
171 00006FFB 8A4603
                                <1>
                                                 al, [esi+LD_FATType]
                                          mov
172 00006FFE A2[85590100]
                                <1>
                                                 [Current_FATType], al
                                <1>
174 00007003 8A26
                                <1>
                                          mov
                                                 ah, [esi+LD_Name]
175 00007005 8825[87590100]
                                <1>
                                                 [Current Dir Drv], ah
                                          mov
176
                                <1>
177 0000700B 20C0
                                <1>
                                           and
                                                 al, al
                                          jz
178 0000700D 741D
                                <1>
                                                 short loc_restore_FS_current_directory
179
                                <1>
                                <1> loc restore FAT current directory:
181 0000700F 8A667F
                                          mov ah, [esi+LD_CDirLevel]
                                <1>
                                                 [Current Dir Level], ah
182 00007012 8825[84590100]
                                <1>
                                          mov
183 00007018 08E4
                                <1>
                                                ah, ah
                                          or
184 0000701A 7416
                                <1>
                                          jz short loc_ccdrv_reset_cdir_FAT_12_16_32_fcluster
                                <1>
186 0000701C 0FB6D4
                                <1>
                                          movzx edx, ah
187 0000701F C0E204
                                <1>
                                        shl dl, 4; * 16
                                          add edx, esi
                                <1>
188 00007022 01F2
189 00007024 8B828C000000
                                <1>
                                          mov eax, [edx+LD_CurrentDirectory+12]
190 0000702A EB2C
                                <1>
                                          jmp short loc_ccdrv_reset_cdir_FAT_fcluster
191
                                <1>
                                <1> loc_restore_FS_current_directory:
                                        call load_current_FS_directory
193 0000702C E8BC4D0000
                                <1>
194 00007031 C3
                                <1>
195
                                <1>
196
                                <1> loc_ccdrv_reset_cdir_FAT_12_16_32_fcluster:
                                      cmp al, 3
197 00007032 3C03
                               <1>
                               <1>
198 00007034 7205
                                          jb
                                                short loc_ccdrv_reset_cdir_FAT_12_16_fcluster
199
                                <1> loc_ccdrv_reset_cdir_FAT32_fcluster:
200 00007036 8B4632
                               <1> mov eax, [esi+LD_BPB+FAT32_RootFClust]
201 00007039 EB04
                                <1>
                                        jmp short loc_ccdrv_check_rootdir_sign
202
                                <1> loc_ccdrv_reset_cdir_FAT_12_16_fcluster:
                                <1> xor al, al ; xor eax, eax
203 0000703B 30C0
204 0000703D 31D2
                                          xor edx, edx
                                <1>
                                <1> loc_ccdrv_check_rootdir_sign:
                                       cmp byte [esi+LD_CurrentDirectory], 0
jne short loc_ccdrv_reset_cdir_FAT_fcluster
206 0000703F 80BE8000000000
                                <1>
207 00007046 7510
                                <1>
                                <1> loc_ccdrv_set_rootdir_FAT_fcluster:
208
209 00007048 89868C000000
                                <1>
                                          mov [esi+LD_CurrentDirectory+12], eax
210 0000704E C78680000000524F4F- <1>
                                          mov dword [esi+LD_CurrentDirectory], 'ROOT'
210 00007057 54
                                <1>
211
                                <1>
212
                                <1> loc_ccdrv_reset_cdir_FAT_fcluster:
                                                [Current_Dir_FCluster], eax
213 00007058 A3[80590100]
                                <1>
214
                                <1>
                                                 edi, PATH_Array
215 0000705D BF[E3600100]
                                <1>
                                          mov
216 00007062 89F2
                                <1>
                                                 edx, esi
217 00007064 81C680000000
                                <1>
                                          add
                                                 esi, LD_CurrentDirectory
218 0000706A B920000000
                                <1>
                                          mov
                                                ecx, 32
219 0000706F F3A5
                                <1>
                                                movsd
                                          rep
                                <1>
220
221 00007071 E84C2D0000
                                <1>
                                          call change_prompt_dir_string
                                <1>
222
223 00007076 89D6
                                <1>
                                          mov
                                                 esi, edx
224
                                 <1>
225 00007078 29C0
                                <1>
                                           sub eax, eax
                                 <1>
226
                                           ;sub edx, edx
227 0000707A BF[87590100]
                                 <1>
                                          mov edi, Current_Dir_Drv
228
                                 <1>
229 0000707F A2[620D0100]
                                 <1>
                                          mov
                                                 [Restore_CDIR], al ; 0
230 00007084 C3
                                 <1>
                                          retn
231
                                 <1>
232
                                 <1> dos_prompt:
233
                                 <1>
                                         ; 06/05/2016
234
                                 <1>
                                          ; 30/01/2016
                                          ; 29/01/2016
235
                                 <1>
236
                                 <1>
                                         ; 16/01/2016 (TRDOS 386 = TRDOS v2.0)
                                         ; 15/09/2011
237
                                 <1>
238
                                 <1>
                                          ; 13/09/2009
                                          ; 2004-2005
239
                                 <1>
240
                                 <1>
                                 <1>
241
242 00007085 C705[40650100]-
                                 <1>
                                          mov dword [mainprog return addr], return from cmd interpreter
242 0000708B [39710000]
                                 <1>
                                 <1>
                                <1> loc_TRDOS_prompt:
245 0000708F BF[865A0100]
                                <1>
                                        mov edi, TextBuffer
246 00007094 C6075B
                                <1>
                                          mov
                                                byte [edi], "["
247 00007097 47
                                <1>
                                          inc
                                                edi
                               <1> inc edi
<1> mov esi, TRDOSPromptLabel
                         248 00007098 BE[B50D0100]
                               <1> get_next_prompt_label_char:
249
                               <1> cmp byte [esi], 20h
<1> jb short pass prom
250 0000709D 803E20
251 000070A0 7203
                                                short pass_prompt_label
252 000070A2 A4
                                      movsb
jmp short get_next_prompt_label_char
253 000070A3 EBF8
                               <1>
                               <1> pass_prompt_label:
254
                            <1> pass_prompt_taxel.
<1> mov byte [edi], "]"
255 000070A5 C6075D
                           <1>
<1>
256 000070A8 47
                                          inc
                                                 edi
257 000070A9 C60720
                              <1> mov byte [edi], 20h
<1> inc edi
<1> mov esi, Current_Dir_Drv
<1> movsw
                                          mov
                                                byte [edi], 20h
258 000070AC 47
259 000070AD BE[87590100] <1>
260 000070B2 66A5 <1>
                               <1>
261 000070B4 A4
                                         movsb
                            <1> loc_prompt_current_directory:
<1> cmp byte [esi], 20h
<1> jb short pass prompt
262
263 000070B5 803E20
264 000070B8 7203
                                           jb
                                                short pass_prompt_current_directory
265 000070BA A4
                               <1>
                                          movsb
266 000070BB EBF8
                                <1>
                                          jmp short loc_prompt_current_directory
```

```
267
                               <1> pass_prompt_current_directory:
                              268 000070BD C6073E
269 000070C0 47
270 000070C1 C60700
                              <1>
                                        mov byte [edi], 0
271 000070C4 BE[865A0100]
                              <1>
                                        mov
                                              esi, TextBuffer
272 000070C9 E8FFF2FFF
                              <1>
                                       call print_msg
                                     274
                               <1>
275
                               <1>
                                        ;call get_cpos ; get cursor position
                                        mov dx, [CURSOR_POSN]; video page 0
276 000070CE 668B15[DE580100]
                              <1>
277 000070D5 8815[E6590100]
                               <1>
                                      mov [CursorColumn], dl
                               <1>
                                        ; 30/01/2016 (to show cursor on the row, again)
279
                               <1>
280
                               <1>
                                        ; (Initial color attributes of video page 0 is 0)
                                        ; (see: 'StartPMP' in trdos386.s)
281
                               <1>
282
                               <1>
                               <1>
                                        ;mov edi, OB8000h; start of video page 0
                               <1>
284
                                        ;movzx ecx, dl ; column
285
                               <1>
                                        ;mov al, 80
                                        ;mul dh
286
                               <1>
287
                               <1>
                                        ;add ax, cx
                                        ;shl ax, 1; character + attribute
288
                               <1>
                                        ;add di, ax; (2*80*row) + (2*column)
289
                               <1>
290
                               <1>
                                        ;neg cl
291
                               <1>
                                        ;add cl, 80
292
                               <1>
                                        ; mov ax, 700h; ah = 7 (color attribute)
                               <1>
                                        ;rep stosw
294
                               <1>
                               <1> loc_rw_char:
295
296 000070DB E899000000
                              <1> call rw char
297
                              <1> loc_move_command:
298 000070E0 BE[365A0100]
                              <1>
                                    mov esi, CommandBuffer
299 000070E5 89F7
                              <1>
                                             edi, esi
                                        mov
                                      xor ecx, ecx
300 000070E7 31C9
                              <1>
301
                              <1> first_command_char:
                              <1> lodsb
302 000070E9 AC
303 000070EA 3C20
                            <1>
                                        cmp al, 20h
                             <1> ja short pass_space_control
<1> jb short loc_move_cmd_argume
<1> cmp esi, CommandBuffer + 79
<1> jb short first_command_char
304 000070EC 772E
305 000070EE 7241
                                              short loc_move_cmd_arguments_ok
306 000070F0 81FE[855A0100]
307 000070F6 72F1
308 000070F8 EB37
                                      jmp
                              <1>
                                             short loc move cmd arguments ok
309
                              <1>
310
                              <1> next_command_char:
311 000070FA AC
                              <1> lodsb
312 000070FB 3C20
                              <1>
                                        cmp al, 20h
313 000070FD 771D
                              <1>
                                              short pass_space_control
                                        jа
314 000070FF 7230
                              <1>
                                       jb
                                              short loc_move_cmd_arguments_ok
315
                              <1>
                             <1> loc_1st_cmd_arg: ; 30/01/2016
316
                            <1>
317 00007101 AC
                                        lodsb
318 00007102 3C20
                              <1>
                                        cmp al, 20h
319 00007104 74FB
                              <1>
                                        jе
                                              short loc_1st_cmd_arg
320 00007106 7229
                              <1>
                                               short loc_move_cmd_arguments_ok
                              <1>
                                        mov
322 00007108 C60700
                              <1>
                                                  byte [edi], 0
323 0000710B 47
                              <1>
                                        inc edi
324
                              <1>
325
                               <1> loc_move_cmd_arguments:
326 0000710C AA
                              <1> stosb
327 0000710D 81FE[855A0100]
                              <1>
                                        cmp esi, CommandBuffer + 79
                                      jnb
328 00007113 731C
                              <1>
                                              short loc_move_cmd_arguments_ok
329 00007115 AC
                              <1>
                                        lodsb
330 00007116 3C20
                              <1>
                                        cmp al, 20h
331 00007118 73F2
                              <1>
                                        jnb short loc_move_cmd_arguments
332 0000711A EB15
                              <1>
                                        jmp short loc_move_cmd_arguments_ok
                              <1>
                              <1> pass_space_control:
334
335 0000711C 3C61
                              <1>
                                    cmp al, 61h
336 0000711E 7206
                              <1>
                                        jb
                                              short pass_capitalize
337 00007120 3C7A
                                        cmp al, 7Ah
                              <1>
338 00007122 7702
                              <1>
                                              short pass capitalize
                                        jа
                                             al, ODFh
339 00007124 24DF
                              <1>
                                        and
340
                              <1> pass_capitalize:
341 00007126 AA
                                      stosb
                              <1>
342 00007127 FEC1
                              <1>
                                        inc
343 00007129 81FE[855A0100]
                              <1>
                                        cmp
                                               esi, CommandBuffer + 79
                                              short next command char
344 0000712F 72C9
                               <1>
                                        jb
345
                               <1>
346
                               <1> loc_move_cmd_arguments_ok:
347 00007131 C60700
                               <1>
                                       mov byte [edi], 0
                               <1>
                               <1> call_command_interpreter:
350 00007134 E8CF080000
                               <1>
                                              command_interpreter
351
                               <1>
                               <1> return_from_cmd_interpreter:
352
353 00007139 B950000000
                                       mov ecx, 80
                              <1>
354
                              <1>
                                        ;mov cx, 80
355 0000713E BF[365A0100]
                              <1>
                                              edi, CommandBuffer
                                        mov
356 00007143 30C0
                                        xor al, al
                               <1>
357 00007145 F3AA
                               <1>
                                        rep stosb
358
                               <1>
                                        ;cmp byte [Program_Exit], 0
359
                                        ;ja short loc_terminate_trdos
                               <1>
360
                               <1>
                                        ; 16/01/2016
361
                               <1>
362 00007147 803D[225F0000]03
                                        cmp byte [CRT_MODE], 3; 80*25 color
                              <1>
363 0000714E 741D
                               <1>
                                              short pass_set_txt_mode
                                        jе
364
                               <1>
                                        call set txt mode; set vide mode to 03h
365 00007150 E817A4FFFF
                               <1>
                                        ; 07/01/2017
366
                               <1>
367 00007155 30C0
                               <1>
                                        xor al, al
368
                               <1>
                               <1> loc_check_active_page:
369
                               <1>
                                      ;xor al, al
371 00007157 3805[EE580100]
                              <1>
                                        cmp [ACTIVE PAGE], al; 0
```

```
372 0000715D 0F842CFFFFFF
                              <1>
                                         je loc_TRDOS_prompt
                               <1>
                                        ; AL = 0 = video page 0
                                        call set_active_page
374 00007163 E81DA8FFFF
                               <1>
375 00007168 E922FFFFF
                               <1>
                                        jmp loc_TRDOS_prompt ; infinitive loop
376
                               <1>
377
                               <1> pass_set_txt_mode:
378 0000716D BE[FF190100]
                               <1>
                                     mov esi, nextline
379 00007172 E856F2FFFF
                                         call print_msg
                               <1>
380 00007177 EBDE
                               <1>
                                                short loc_check_active_page
                                         jmp
381
                               <1>
382
                               <1> rw_char:
                                      ; 13/05/2016
383
                               <1>
                                        ; 30/01/2016
384
                               <1>
                                      ; 29/01/2016
385
                               <1>
                                       ; 17/01/2016 (TRDOS 386 = TRDOS v2.0)
386
                               <1>
                                        ; 2004-2005
387
                               <1>
                               <1>
                               <1>
389
                                        ; DH = cursor row, DL = cursor column
                                         ; BH = 0 = video page number (active page)
390
                               <1>
391
                               <1>
                                         ; xor bh, bh; 0 = video page 0
392
                               <1>
393
                               <1>
394
                               <1> readnextchar:
395 00007179 30E4
                               <1>
                                                ah, ah
396 0000717B E89F9AFFFF
                               <1>
                                         call int16h
397 00007180 20C0
                               <1>
                                         and
                                              al, al
                                              short loc_arrow
398 00007182 7432
                              <1>
                                        jz
                              <1> cmp al, 0E0h
<1> je short loc
<1> cmp al, 08h
<1> jne short cha
399 00007184 3CE0
                              <1>
400 00007186 742E
                                               short loc_arrow
401 00007188 3C08
                              <1>
402 0000718A 7542
                                               short char_return
                               <1> loc back:
404 0000718C 3A15[E6590100]
                                              dl, [CursorColumn]
                               <1>
                                        cmp
405 00007192 76E5
                               <1>
                                              short readnextchar
                                         jna
                               <1> prev_column:
407 00007194 FECA
                               <1>
                                      dec dl
                               <1> set_cursor_pos:
                                      ;push dx
409
                               <1>
                                         push edx ; 29/12/2017
410 00007196 52
                               <1>
                                        ;xor bh, bh; 0 = video page 0
                               <1>
411
                                    ; DH = Row, DL = Column

call _set_cpos ; 17/01/2016

pop edx ; 29/12/2017
412
                               <1>
413 00007197 E8B5ABFFFF
                               <1>
414 0000719C 5A
                               <1>
                                        pop edx ; 29/12/2017
                                     ;pop dx
415
                               <1>
                                     ;movzx ebx, dl
mov bl, dl
sub bl, [CursorColumn]
416
                               <1>
417 0000719D 88D3
                               <1>
418 0000719F 2A1D[E6590100]
                               <1>
419 000071A5 B020
                               <1>
                                      mov al, 20h
                                               [CommandBuffer+ebx], al
420 000071A7 8883[365A0100]
                               <1>
                                        mov
                               <1>
                                        ; sub bh, bh ; video page 0
                                    ;mov cx, 1 mov bl, 7
422
                               <1>
423 000071AD B307
                               <1>
                                        mov
                                               bl, 7; color attribute
424 000071AF E88EAAFFFF
                               <1>
                                        call
                                               _write_c_current ; 17/01/2016
                               <1>
                                        ;mov dx, [CURSOR POSN]
426 000071B4 EBC3
                               <1>
                                               short readnextchar
                                        jmp
427
                               <1> loc_arrow:
428 000071B6 80FC4B
                               <1> cmp
                                               ah, 4Bh
429 000071B9 74D1
                                               short loc_back
                               <1>
                                         jе
430 000071BB 80FC53
                               <1>
                                               ah, 53h
                                         cmp
                                    je
cmp
                                               short loc_back
431 000071BE 74CC
                              <1>
432 000071C0 80FC4D
                              <1>
                                               ah, 4Dh
                                    jne
433 000071C3 75B4
                               <1>
                                               short readnextchar
434 000071C5 80FA4F
                              <1>
                                               dl, 79
                                        cmp
435 000071C8 73AF
                              <1>
                                               short readnextchar
                                         jnb
436 000071CA FEC2
                               <1>
                                         inc
                                               dl
437 000071CC EBC8
                               <1>
                                         jmp
                                               short set_cursor_pos
                               <1> char return:
439 000071CE 0FB6DA
                                     movzx ebx, dl
                               <1>
440 000071D1 2A1D[E6590100]
                              <1>
                                              bl, [CursorColumn]
441 000071D7 3C20
                               <1>
                                              al, 20h
                                         cmp
                                    jb
mov
442 000071D9 721D
                               <1>
                                               short loc_escape
443 000071DB 8883[365A0100]
                               <1>
                                               [CommandBuffer+ebx], al
444 000071E1 80FA4F
                               <1>
                                              dl, 79
                                        cmp
                                    jnb
445 000071E4 7393
                               <1>
                                               short readnextchar
                                    mov
                                               bx, 7 ; color attribute
446 000071E6 66BB0700
                               <1>
447 000071EA E8CCAAFFFF
                               <1>
                                        call
                                               _write_tty
448 000071EF 668B15[DE580100] <1>
                                        mov dx, [CURSOR POSN]; video page 0
                                       jmp
449 000071F6 EB81
                               <1>
                                                  readnextchar
                               <1> loc_escape:
450
451 000071F8 3C1B
                                        cmp
                               <1>
                                              al, 1Bh
452 000071FA 7418
                               <1>
                                               short rw_char_retn
                                <1>
454 000071FC 3C0D
                                         cmp al, ODh ; CR
                              <1>
455 000071FE 0F8575FFFFFF
                               <1>
                                         jne
                                                readnextchar
                               <1>
                                        ; 13/05/2016
456
457 00007204 66BB0700
                               <1>
                                        mov bx, 7; attribute/color (bl)
                              <1>
                                                    ; video page 0 (bh=0)
                            <1>
459 00007208 E8AEAAFFFF
                                       call _write_tty
                                      ;mov bx, 7; attribute/color; video page 0 (bh=0)
460
                              <1>
                              <1>
461
462 0000720D B00A
                                        mov al, OAh ; LF
                              <1>
463 0000720F E8A7AAFFFF
                               <1>
                                        call _write_tty
                              <1> rw_char_retn:
465 00007214 C3
                               <1>
466
                               <1>
                               <1> show date:
467
                                     ; 18/01/2016 (TRDOS 386 = TRDOS v2.0)
468
                               <1>
                                         ; 2004-2005
                               <1>
469
470
                               <1>
471
                               <1>
                                        ;mov ah, 04h
472
                               <1>
                                       call int1Ah;
473 00007215 E8A2E8FFFF
                              <1>
                                       call RTC 40; GET RTC DATE
                               <1>
475 0000721A 88D0
                                       mov al, dl
                               <1>
476 0000721C E8F599FFFF
                              <1>
                                       call bcd_to_ascii
```

```
477 00007221 66A3[A10E0100] <1>
                                      mov
                                             [Day], ax
479 00007227 88F0
                              <1>
                                       mov
                                              al, dh
480 00007229 E8E899FFFF
                              <1>
                                       call bcd_to_ascii
                                              [Month], ax
481 0000722E 66A3[A40E0100]
                              <1>
                                       mov
482
                              <1>
483 00007234 88E8
                                             al, ch
                              <1>
                                      mov
484 00007236 E8DB99FFFF
                                    call bcd_to_ascii
                              <1>
485 0000723B 66A3[A70E0100]
                              <1>
                                             [Century], ax
                                       mov
                              <1>
                                    mov
487 00007241 88C8
                              <1>
                                             al, cl
488 00007243 E8CE99FFFF
                              <1>
                                       call bcd to ascii
489 00007248 66A3[A90E0100]
                              <1>
                                       mov word [Year], ax
490
                              <1>
491 0000724E BE[910E0100]
                              <1>
                                       mov
                                             esi, Msg_Show_Date
492 00007253 E875F1FFFF
                              <1>
                                        call print_msg
                              <1>
494 00007258 C3
                               <1>
                                        retn
495
                               <1>
496
                               <1> set_date:
                                    ; 13/05/2016
497
                               <1>
                                        ; 18/01/2016 (TRDOS 386 = TRDOS v2.0)
498
                               <1>
                                      ; 2004-2005
499
                               <1>
                              <1>
                                     mov esi, Msg_Enter_Date
call print_msg
501 00007259 BE[750E0100]
                              <1>
 502 0000725E E86AF1FFFF
                              <1>
                              <1>
504
                              <1> loc_enter_day_1:
                    <1><1><1><1><1></1>
505 00007263 30E4
                              <1>
                             call int16h
<1> ; AL = ASCII Code of the Character
cmp al, 13
<1> je loc_set_date_retn
<1> cmp al, 27
<1> je loc_set_date_retn
<1> cmp al, 27
<1> je loc_set_date_retn
<1> mov [Day], al
<1> cmp al, '0'
<1> jb loc_set_date_stc_0
<1> cmp al, '3'
<1> cmp al, '3'
                                               ah, ah
506 00007265 E8B599FFFF
507
508 0000726A 3C0D
509 0000726C 0F84B7010000
510 00007272 3C1B
511 00007274 0F84AF010000
512 0000727A A2[A10E0100]
513 0000727F 3C30
                        514 00007281 0F82AD010000
515 00007287 3C33
516 00007289 0F87A5010000
517
518
519
520 0000728F B307
521 00007291 E825AAFFFF
522
523 00007296 30E4
                              <1> xor ah, ah
<1>
                                      call int16h
524 00007298 E88299FFFF
                                        ; AL = ASCII Code of the Character
537
                              <1> pass_set_day_31:
                              <1> ; 13/05/2016 
<1> ; mov bx 7
538
539
                              <1>
                                        ;mov bx, 7; attribute/color (bl)
                              540
                                                    ; video page 0 (bh)
541 000072CB B307
542 000072CD E8E9A9FFFF
                              <1> loc_enter_separator_1:
                             544 000072D2 28E4
545 000072D4 E84699FFFF
546
547 000072D9 3C1B
 548 000072DB 0F8448010000
549 000072E1 3C2D
550 000072E3 7408
551 000072E5 3C2F
552 000072E7 0F856C010000
553
                               <1> pass_set_date_separator_1:
                               <1> ; 13/05/2016
554
555
                               <1>
                                        ;mov bx, 7; attribute/color (bl)
                                                   ; video page 0 (bh)
556
                               <1>
                                   mov bl, 7
557 000072ED B307
                               <1>
 558 000072EF E8C7A9FFFF
                               <1>
                                        call
                                             _write_tty
                               <1> loc_enter_month_1:
560 000072F4 30E4
                               <1>
                                             ah, ah ; 0
561 000072F6 E82499FFFF
                              <1>
                                        call int16h
                                        ; AL = ASCII Code of the Character
562
                              <1>
563 000072FB 3C1B
                              <1>
                                       cmp al, 27
                             <1>
                              564 000072FD 0F8426010000
                                              loc_set_date_retn
565 00007303 A2[A40E0100]
566 00007308 3C30
567 0000730A 0F8264010000
 568 00007310 3C31
569 00007312 0F875C010000
                               <1>
                                        jа
                                              loc_set_date_stc_3
                                        ; 13/05/2016
570
                               <1>
                                       ;mov bx, 7 ; attribute/color (bl)
571
                               <1>
572
                                                    ; video page 0 (bh)
                               <1>
 573 00007318 B307
                                             bl, 7
                              <1>
                                        mov
                                       call write_tty
574 0000731A E89CA9FFFF
                              <1>
575
                              <1> loc_enter_month_2:
576 0000731F 30E4
                              <1> xor ah, ah
577 00007321 E8F998FFFF
                                        call int16h
                              <1>
                              <1>
                                        ; AL = ASCII Code of the Character
579 00007326 3C1B
                                        cmp al, 27
                              <1>
                              <1> je loc_set_da <1> mov [Month+1], al
                                       je loc_set_date_retn
580 00007328 0F84FB000000
581 0000732E A2[A50E0100]
```

```
582 00007333 3C30
583 00007335 0F8254010000
584 0000733B 3C39
584 0000733B 3C39
short pass_set_month_12
590
                                  <1> pass_set_month_12:
                                591
592
593
594 00007354 B307
595 00007356 E860A9FFFF
                                  <1> loc_enter_separator_2:
597 0000735B 28E4
                                  <1> sub ah, ah <1> call int16h
                             <1>
598 0000735D E8BD98FFFF
                                            call int16h
                                  <1> pass_set_date_separator_2:
607
                                  <1> ; 13/05/2016
608
                                  <1>
                                             ;mov bx, 7; attribute/color (bl)
                             <1> ; vio
<1> mov bl, 7
<1> call _write_tty
609
                                            ; video page 0 (bh)
610 00007376 B307
611 00007378 E83EA9FFFF
                                  <1> loc_enter_year_1:
612
                                  <1> xor ah, ah <1> call int16h
613 0000737D 30E4
                          614 0000737F E89B98FFFF
                                  <1>
615
616 00007384 3C1B
617 00007386 0F849D000000
618 0000738C A2[A90E0100]
619 00007391 3C30
620 00007393 0F822C010000
621 00007399 3C39
622 0000739B 0F8724010000
623
624
625
626 000073A1 B307
627 000073A3 E813A9FFFF
629 000073A8 30E4
630 000073AA E87098FFFF
                             call int16h
    ; AL = ASCII Code of the Character
    cmp al, 27
    je short loc_set_date_retn
    mov byte [Year+1], al
    cmp al, '0'
        jb loc_set_date_stc_7
    cmp al, '9'
        ja loc_set_date_stc_7
    ; 13/05/2016
    ; mov bx, 7; attribute/color (bl)
    ; video page 0 (bh)
632 000073AF 3C1B
633 000073B1 7476
634 000073B3 A2[AA0E0100]
635 000073B8 3C30
636 000073BA 0F8220010000
637 000073C0 3C39
632 000073AF 3C1B
637 000073C0 3C39
638 000073C2 0F8718010000
639
640
                                  <1>
641
                                                          ; video page 0 (bh)
                            <1> mov bl, 7
<1> call _write_tty
<1> loc_set_date_get_lchar_again:
642 000073C8 B307
643 000073CA E8ECA8FFFF
644
645 000073CF 28E4
                                 <1> sub ah, ah; 0
                                 call int16h

call int16h

call int16h

call int16h

cmp al, 13 ; ENTER key

cmp al, 27 ; ESC key

cmp al, 27 ; ESC key

je short loc_set_date_retn

call check_for_backspace

call check_for_backspace

call short loc_set_date_get_lchar_again

                                  <1>
                                            call int16h
646 000073D1 E84998FFFF
647
648 000073D6 3C0D
649 000073D8 7412
650 000073DA 3C1B
651 000073DC 744B
652
653 000073DE E82A010000
654 000073E3 75EA
655
                                  <1>
656
                                  <1> loc_set_date_bs_8:
657 000073E5 E811010000
                                          call write_backspace
                                  <1>
658 000073EA EBBC
                                   <1>
                                             jmp short loc_enter_year_2
659
                                   <1>
                                   <1> loc set date progress:
660
                                   <1> ; Get Current Date
661
662
                                   <1>
                                             ;mov ah, 04h
                                           ;call int1Ah
663
                                   <1>
                                             call RTC 40; GET RTC DATE
664 000073EC E8CBE6FFFF
                                   <1>
                                   <1>
                                             ; CH = century (in BCD)
666
                                   <1>
667 000073F1 66A1[A90E0100]
                                   <1>
                                             mov
                                                    ax, [Year]
668 000073F7 662D3030
                                                   ax, '00'
                                  <1>
                                           sub
                                                    al, 4 ; * 16
669 000073FB C0E004
                                  <1>
                                             shl
670 000073FE 88C1
                                  <1>
                                                    cl, al
                                             mov
671 00007400 00E1
                                  <1>
                                                   cl, ah
                                             add
672 00007402 66A1[A40E0100] <1>
                                             mov
                                                    ax, [Month]
673 00007408 662D3030
                                  <1>
                                             sub
                                                    ax, '00'
674 0000740C C0E004
                                                    al, 4 ; * 16
                                  <1>
                                             shl
                                                    dh, al
675 0000740F 88C6
                                  <1>
                                             mov
676 00007411 00E6
                                  <1>
                                             add
                                                    dh, ah
677 00007413 66A1[A10E0100]
                                  <1>
                                             mov
                                                    ax, [Day]
678 00007419 662D3030
                                  <1>
                                             sub
                                                    ax, '00'
                                                    al, 4 ; * 16
679 0000741D C0E004
                                             shl
                                  <1>
680 00007420 88C2
                                  <1>
                                             mov
                                                    dl, al
681 00007422 00E2
                                  <1>
                                                   dl, ah
                                             add
682
                                  <1>
683
                                   <1>
                                             ;mov ah, 05h
684
                                  <1>
                                             ;call int1Ah
685 00007424 E8C0E6FFFF
                                  <1>
                                             call RTC_50; SET RTC DATE
                                   <1>
686
```

```
<1> loc_set_date_retn:
687
                              <1> mov esi, nextline
<1> call print_msg
688 00007429 BE[FF190100]
689 0000742E E89AEFFFFF
690 00007433 C3
                              <1>
691
                              <1>
                              <1> loc_set_date_stc_0:
692
                              <1> ;xor bh, bh; video page 0
694 00007434 E862A9FFFF
                              <1>
                                        call beeper ; BEEP !
695 00007439 E925FEFFFF
                              <1>
                                       jmp loc_enter_day_1
                              <1> loc_set_date_stc_1:
                                   call check_for_backspace
697 0000743E E8CA000000
                              <1>
698 00007443 740A
                              <1>
                                        jе
                                              short loc set date bs 1
                                       ;xor bh, bh ; video page 0
                              <1>
                                     call beeper; BEEP!
jmp loc_enter_day_2
700 00007445 E851A9FFFF
                              <1>
701 0000744A E947FEFFFF
                              <1>
                              <1> loc_set_date_bs_1:
702
                              <1> call write_backspace
<1> jmp loc_enter_day_1
703 0000744F E8A7000000
704 00007454 E90AFEFFFF
                              <1> loc set date stc 2:
705
                              <1> call check_for_backspace
706 00007459 E8AF000000
707 0000745E 740A
                              <1>
                                        je short loc_set_date_bs_2
                                       ;xor bh, bh; video page 0 call beeper; BEEP!
                              <1>
709 00007460 E836A9FFFF
                              <1>
                              <1> jmp loc_enter_separator_1
710 00007465 E968FEFFFF
711
                              <1> loc_set_date_bs_2:
712 0000746A E88C000000
                              <1> call write_backspace
713 0000746F E922FEFFFF
                              <1>
                                        jmp loc enter day 2
714
                              <1> loc_set_date_stc_3:
                                    call check_for_backspace
715 00007474 E894000000
                              <1>
                                       je short loc_set_date_bs_3
716 00007479 740A
                              <1>
                                     717
                              <1>
                                       ;xor bh, bh ; video page 0
718 0000747B E81BA9FFFF
                              <1>
719 00007480 E96FFEFFFF
                              <1>
                              <1> loc set date bs 3:
                                    call write_backspace
721 00007485 E871000000
                              <1>
722 0000748A E943FEFFFF
                              <1>
                                       jmp loc_enter_separator_1
                              <1> loc_set_date_stc_4:
724 0000748F E87900000
                                   call check_for_backspace
                              <1>
725 00007494 740A
                                              short loc set date bs 4
                              <1>
                                        jе
                                       ;xor bh, bh ; video page 0
                              <1>
727 00007496 E800A9FFFF
                                     call beeper ; BEEP !
    jmp    loc_enter_month_2
                              <1>
728 0000749B E97FFEFFFF
                              <1>
                              <1> loc_set_date_bs_4:
730 000074A0 E856000000
                              <1>
                                   call write_backspace
731 000074A5 E94AFEFFFF
                              <1>
                                       jmp loc enter month 1
                              <1> loc_set_date_stc_5:
733 000074AA E85E000000
                              <1> call check_for_backspace
734 000074AF 740A
                              <1>
                                        je short loc_set_date_bs_5
735
                              <1>
                                        ;xor bh, bh ; video page 0
                                       call beeper; BEEP!
736 000074B1 E8E5A8FFFF
                              <1>
                                      jmp loc_enter_separator_2
737 000074B6 E9A0FEFFFF
                              <1>
                              <1> loc set date bs 5:
                              <1> call write_backspace
739 000074BB E83B000000
740 000074C0 E95AFEFFFF
                              <1>
                                        jmp loc_enter_month_2
                              <1> loc_set_date_stc_6:
742 000074C5 E843000000
                              <1> call check_for_backspace
743 000074CA 740A
                                       je short loc set date bs 6
                              <1>
744
                                       ;xor bh, bh ; video page 0
                              <1>
745 000074CC E8CAA8FFFF
                              <1>
                                       call beeper ; BEEP !
746 000074D1 E9A7FEFFFF
                              <1>
                                       jmp loc_enter_year_1
747
                              <1> loc_set_date_bs_6:
                                    call write_backspace
748 000074D6 E820000000
                              <1>
749 000074DB E97BFEFFFF
                              <1>
                                       jmp loc_enter_separator_2
750
                              <1> loc_set_date_stc_7:
                                    call check_for_backspace
751 000074E0 E828000000
                              <1>
752 000074E5 740A
                              <1>
                                        je
                                              short loc_set_date_bs_7
                                       ;xor bh, bh ; video page 0
                              <1>
754 000074E7 E8AFA8FFFF
                              <1>
                                        call beeper ; BEEP !
755 000074EC E9B7FEFFFF
                              <1>
                                       jmp loc_enter_year_2
                              <1> loc_set_date_bs_7:
                                     call write backspace
757 000074F1 E805000000
                              <1>
                                        jmp loc enter year 1
758 000074F6 E982FEFFFF
                              <1>
759
                              <1>
                              <1> write backspace:
760
                                     ; 18/01/2016 (TRDOS 386 = TRDOS v2.0)
761
                              <1>
762 000074FB B008
                                       mov al, 08h; BACKSPACE
                              <1>
763
                              <1>
                                       ; 13/05/2016
764 000074FD 66BB0700
                              <1>
                                       mov bx, 7; bl = attribute/color
                                                   ; bh = video page = 0
                              <1>
765
766 00007501 E8B5A7FFFF
                                        call _write_tty
                              <1>
                                        mov al, 20h; BLANK/SPACE char
767 00007506 B020
                              <1>
                                        ;mov bx, 7; attribute/color
                               <1>
                                       ;call _write_c_current
769
                              <1>
770
                              <1>
                                       ;retn
771 00007508 E935A7FFFF
                              <1>
                                       jmp _write_c_current
772
                              <1>
773
                              <1> check for backspace:
                                     ; 18/01/2016 (TRDOS 386 = TRDOS v2.0)
774
                              <1>
775 0000750D 663D080E
                              <1>
                                        cmp
                                             ax, 0E08h
776 00007511 7410
                             <1>
                                             short cfbs_retn
                                       iе
                             <1>
                                     cmp ax, 4BE0h
777 00007513 663DE04B
                                     je
778 00007517 740A
                              <1>
                                             short cfbs retn
779 00007519 663D004B
                              <1>
                                       cmp
                                             ax, 4B00h
780 0000751D 7404
                             <1>
                                     je
                                             short cfbs retn
781 0000751F 663DE053
                              <1>
                                             ax, 53E0h
                                       cmp
                              <1> cfbs_retn:
783 00007523 C3
                              <1>
784
                              <1>
                              <1> show_time:
785
                                   -; 18/01/2016 (TRDOS 386 = TRDOS v2.0)
786
                              <1>
787
                              <1>
                                        ; 2004-2005
788
                               <1>
789
                              <1>
                                       ;mov ah, 02h
                                      ;call int1Ah
                              <1>
790
791 00007524 E822E5FFFF
                              <1>
                                       call RTC_20; GET RTC TIME
```

```
792
                               <1>
  793 00007529 88E8
                               <1>
                                        mov
                                              al, ch
  794 0000752B E8E696FFFF
                                        call bcd to_ascii
                              <1>
  795 00007530 66A3[CF0E0100] <1>
                                             [Hour], ax
  796
                               <1>
                                        mov
  797 00007536 88C8
                               <1>
                                              al, cl
  798 00007538 E8D996FFFF
                              <1>
                                        call bcd to ascii
  799 0000753D 66A3[D20E0100]
                              <1>
                                             [Minute], ax
                                       mov
  800
                               <1>
  801 00007543 88F0
                               <1>
                                      mov
                                             al, dh
  802 00007545 E8CC96FFFF
                             <1>
                                       call bcd_to_ascii
  803 0000754A 66A3[D50E0100]
                               <1>
                                        mov
                                             [Second], ax
                              <1>
  805 00007550 BE[BF0E0100]
                              <1>
                                       mov esi, Msg_Show_Time
  806 00007555 E873EEFFFF
                               <1>
                                       call print msg
  807 0000755A C3
                               <1>
                                        retn
                               <1>
  809
                               <1> set_time:
                                     ; 13/05/2016
  810
                               <1>
                                        ; 18/01/2016 (TRDOS 386 = TRDOS v2.0)
  811
                               <1>
                                       ; 2004-2005
  812
                               <1>
  813
                               <1>
  814 0000755B BE[AE0E0100]
                               <1>
                                      mov esi, Msg_Enter_Time
  815 00007560 E868EEFFFF
                              <1>
                                      call print_msg
  816
                               <1>
                               <1> loc_enter_hour_1:
  817
  818 00007565 30E4
819 00007567 E8B396FFFF
                              <1> xor ah, ah
                              <1>
                                        call int16h
                                        ; AL = ASCII Code of the Character
  821 0000756C 3C0D
 821 0000756C 3C0D

822 0000756E 0F84AE010000

823 00007574 3C1B

824 00007576 0F84A6010000

825 00007571 A2[CF0E0100]
  826 00007581 3C30
  827 00007583 0F82A4010000
  828 00007589 3C32
  829 0000758B 0F879C010000
  830
                                        ;mov bx, 7 ; attribute/color (bl)
  831
                                             ; video page 0 (bh) bl, 7
  832
                              <1>
                    833 00007591 B307
  834 00007593 E823A7FFFF
835
                                        ; AL = ASCII Code of the Character
                               <1> ; \(\bar{1}\)3/05\(\bar{2}\)2016
  851
  852
                               <1>
                                        ;mov bx, 7; attribute/color (bl)
  853
                               <1>
                                                   ; video page 0 (bh)
                                        mov bl, 7
  854 000075CD B307
                             855 000075CF E8E7A6FFFF
                               <1> loc_enter_time_separator_1:
  856
                              <1> sub ah, ah; 0 <1> call int16h
  857 000075D4 28E4
                                       call int16h
  858 000075D6 E84496FFFF
                             <1>
 <1> ; vio
<1> mov bl, 7
<1> call _write_tty
                                                   ; video page 0 (bh)
  866
  867 000075EB B307
  868 000075ED E8C9A6FFFF
  869
                               <1> loc_enter_minute_1:
                                     xor ah, ah
  870 000075F2 30E4
                               <1>
  871 000075F4 E82696FFFF
                                        call int16h
                               <1>
                                     ; AL = ASCII Code of the Character
  872
                               <1>
  873 000075F9 3C1B
                               <1>
                                        cmp al, 27
  874 000075FB 0F8421010000
                                                 loc set time retn
                               <1>
                                          jе
  875 00007601 A2[D20E0100]
                               <1>
                                              [Minute], al
  876 00007606 3C30
                                              al, '0'
                               <1>
                                        cmp
  877 00007608 0F825F010000
                               <1>
                                         jb
                                              loc_set_time_stc_3
                                              al, '5'
  878 0000760E 3C35
                               <1>
                                        cmp
  879 00007610 0F8757010000
                                                 loc_set_time_stc_3
                               <1>
                                        jа
  880
                               <1>
                                        ; 13/05/2016
  881
                               <1>
                                        ;mov bx, 7 ; attribute/color (bl)
  882
                               <1>
                                                   ; video page 0 (bh)
                                              bl, 7
  883 00007616 B307
                               <1>
                                        mov
                                        call write_tty
  884 00007618 E89EA6FFFF
                               <1>
                               <1> loc enter minute 2:
  886 0000761D 30E4
                               <1>
                                        xor
                                             ah, ah
  887 0000761F E8FB95FFFF
                                        call int16h
                               <1>
                               <1>
                                        ; AL = ASCII Code of the Character
  889 00007624 3C1B
                                        cmp al, 27
                               <1>
  890 00007626 0F84F6000000
                               <1>
                                         jе
                                                 loc_set_time_retn
                                              [Minute+1], al
  891 0000762C A2[D30E0100]
                               <1>
                                        mov
                                              al, '0'
  892 00007631 3C30
                               <1>
                                        cmp
  893 00007633 0F824F010000
                               <1>
                                        jb
                                              loc_set_time_stc_4
                                              al, '9'
  894 00007639 3C39
                               <1>
                                        cmp
                                              loc_set_time_stc_4
  895 0000763B 0F8747010000
                               <1>
                                        jа
  896
                               <1>
                                        ; 13/05/2016
```

```
897
                                   <1>
                                             ;mov bx, 7; attribute/color (bl)
 898
                                   <1>
                                                          ; video page 0 (bh)
 899 00007641 B307
                                  <1>
                                             mov
                           <1>
 900 00007643 E873A6FFFF
                                             call write tty
                                  <1> loc_enter_time_separator_2:
 902 00007648 66C705[D50E0100]30- <1> mov word [Second], 3030h
 902 00007650 30
                      <1>
                                 903 00007651 28E4
                                                    ah, ah
                           <1>
 904 00007653 E8C795FFFF
                                             ; AL = ASCII Code of the Character
 906 00007658 3C0D
 906 00007658 3C0D
907 0000765A 0F8485000000
908 00007660 3C1B
909 00007662 0F84BA000000
                                                    loc set time progress
                                 <1> cmp al, 27
<1> je loc_set_time_retn
                                  comp al, ':'
<1> cmp al, ':'
<1> jne loc_set_time_stc_5
<1> ; 13/05/2016
<1> ;mov bx, 7; attribute/color (bl)
<1> ; video page 0 (bh)
 910 00007668 3C3A
 911 0000766A 0F8533010000
 913
 914
                                                          ; video page 0 (bh)
                                             mov bl, 7
                                <1> mov bl, 7 call _write_tty
 915 00007670 B307
 916 00007672 E844A6FFFF
                                  <1> loc enter second 1:
 918 00007677 30E4
                                  <1> xor ah, ah
                                <1>
 919 00007679 E8A195FFFF
                                             call int16h
 920
                                  <1>
                                             ; AL = ASCII Code of the Character
 921 0000767E 3C0D
                                  <1>
                                             cmp al, 13
                    je short loc_set_time_progress
 922 00007680 7463
                                 <1>
 923 00007682 3C1B
 924 00007684 0F8498000000
 925 0000768A A2[D50E0100]
 926 0000768F 3C30
 927 00007691 0F8227010000
 928 00007697 3C35
 929 00007699 0F871F010000
 930
 931
 933 0000769F B307
 934 000076A1 E815A6FFFF
ah, a
call int16h
; AL = ASCII
cmp al, 27
je short l
941 000076B1 3C30 <1> cmp
942 000076B3 0F8229010000 <1>
943 000076B9 3C39
944 000076BB 0F87210105
945
 936 000076A6 30E4
                                 <1> je short loc_set_time_ret
<1> cmp al, '0'
<1> jb loc_set_time_stc_7
                                                   short loc_set_time_retn
                                 coc_set_time_stc_7
cmp al, '9'
cl> ja loc_set_time_stc_7
cl> ja loc_set_time_stc_7
cl> ; 13/05/2016
cl> ;mov bx, 7; attribute/color (bl)
cl> ; video page 0 (bh)
 946
<1> loc_set_time_get_lchar_again:
                                        ; AL = ASCII Code of the Character cmp al, 13
je short loc_set_time_progress cmp al, 27
je short loc_set_time_retn
;
 954 000076CF 3C0D
                                  <1>
                                <1>
 955 000076D1 7412
 956 000076D3 3C1B
                                <1>
 957 000076D5 744B
                                  <1>
                                  <1>
                                        call check_for_backspace
jne short loc_set_time_get_lchar_again
                           <1>
 959 000076D7 E831FEFFFF
 960 000076DC 75EA
                                  <1>
 961
                                  <1>
                                  <1> loc_set_time_bs_8:
 963 000076DE E818FEFFFF
                                  <1>
                                          call write_backspace
 964 000076E3 EBC1
                                   <1>
                                             jmp short loc_enter_second_2
 965
                                   <1>
                                   <1> loc_set_time_progress:
 966
                                         ; Get Current Time
 967
                                   <1>
 968
                                   <1>
                                             ;mov ah, 02h
 969
                                   <1>
                                           call int1Ah;
                                           call RTC_20; GET RTC TIME
 970 000076E5 E861E3FFFF
                                   <1>
                                            ;DL = Daylight Savings Enable option (0-1)
 971
                                   <1>
                                   <1>
 973 000076EA 66A1[CF0E0100]
                                  <1>
                                            mov
                                                   ax, [Hour]
                                                   ax, '00'
 974 000076F0 662D3030
                                   <1>
                                             sub
 975 000076F4 C0E004
                                   <1>
                                                   al, 4 ; * 16
                                             shl
 976 000076F7 88C5
                                   <1>
                                             mov
                                                    ch, al
 977 000076F9 00E5
                                   <1>
                                             add
                                                    ch, ah
 978 000076FB 66A1[D20E0100]
                                 <1>
                                             mov
                                                    ax, [Minute]
                                                    ax, '00'
 979 00007701 662D3030
                                   <1>
                                                    al, 4 ; * 16
 980 00007705 C0E004
                                   <1>
                                             shl
 981 00007708 88C1
                                  <1>
                                             mov
                                                    cl, al
 982 0000770A 00E1
                                  <1>
                                             add
                                                    cl, ah
 983 0000770C 66A1[D50E0100] <1>
                                             mov
                                                    ax, [Second]
 984 00007712 662D3030
                                  <1>
                                                    ax, '00'
                                             sub
                                                    al, 4 ; * 16
 985 00007716 C0E004
                                  <1>
                                             shl
 986 00007719 88C6
                                  <1>
                                                    dh, al
                                             mov
                                   <1>
 987 0000771B 00E6
                                            add
                                                    dh, ah
 988
                                  <1>
 989
                                  <1>
                                             ;mov ah, 03h
 990
                                   <1>
                                             ;call int1Ah
 991 0000771D E858E3FFFF
                                             call RTC_30; SET RTC TIME
                                  <1>
                                   <1>
                                  <1> loc_set_time_retn:
 993
                                         mov esi, nextline call print_msg
 994 00007722 BE[FF190100]
                                   <1>
 995 00007727 E8A1ECFFFF
                                  <1>
 996 0000772C C3
                                   <1>
                                             retn
 997
                                   <1>
                                  <1> loc_set_time_stc_0:
998
                                        ;xor bh, bh ; video page 0
999
                                  <1>
1000 0000772D E869A6FFFF
                                   <1>
                                             call beeper ; BEEP !
```

```
1001 00007732 E92EFEFFF
                                          jmp
                                                     loc enter hour 1
                                 <1>
1002
                                  <1> loc set time stc 1:
                                 1003 00007737 E8D1FDFFFF
1004 0000773C 740A
                                 <1>
                                           je short loc_set_time_bs_1
                                         ;xor bh, bh; video page 0
call beeper; BEEP!
  jmp loc_enter_hour_2
1005
                                 <1>
1006 0000773E E858A6FFFF
                                 <1>
1007 00007743 E950FEFFFF
                                 <1>
                                  <1> loc_set_time_bs_1:
1008
                                        call write_backspace
1009 00007748 E8AEFDFFFF
                                 <1>
1010 0000774D E913FEFFFF
                                           jmp loc enter hour 1
                                  <1>
                                  <1> loc_set_time_stc_2:
1011
1012 00007752 E8B6FDFFFF
                                  <1> call check_for_backspace
<1> je short loc_set_time_bs_2
1013 00007757 740A
                                 <1>
                                           ;xor bh, bh ; video page 0
1014
                                  <1>
1015 00007759 E83DA6FFFF
                                       call beeper ; BEEP !
    jmp    loc_enter_time_separator_1
                                  <1>
1016 0000775E E971FEFFFF
                                 <1>
                                  <1> loc set time bs 2:
1018 00007763 E893FDFFFF
                                  <1> call write_backspace
1019 00007768 E92BFEFFFF
                                  <1>
                                           jmp loc_enter_hour_2
                                  <1> loc_set_time_stc_3:
1020
                                 <1> call check_for_backspace
1021 0000776D E89BFDFFFF
1022 00007772 740A
                                  <1>
                                           je short loc set time bs 3
                                       ;xor bh, bh; viueo recall beeper; BEEP !6
jmp loc_enter_minute_1
1023
                                  <1>
1024 00007774 E822A6FFFF
                                  <1>
1025 00007779 E974FEFFFF
                                  <1>
                                  <1> loc_set_time_bs_3:
1026
                                 <1> call write_backspace
1027 0000777E E878FDFFFF
1028 00007783 E94CFEFFFF
                                 <1>
                                           jmp loc_enter_time_separator_1
1029
                                  <1> loc_set_time_stc_4:
                                 <1> call check_for_backspace
<1> je short loc set time k
1030 00007788 E880FDFFFF
1031 0000778D 740A
                                           je short loc_set_time_bs_4
                                         ;xor bh, bh; video page 0
call beeper; BEEP!
  jmp loc_enter_minute_2
                                  <1>
1033 0000778F E807A6FFFF
                                 <1>
1034 00007794 E984FEFFFF
                                 <1>
                                  <1> loc_set_time_bs_4:
                                 <1> call write_backspace
<1> jmp loc_enter_minute_1
1036 00007799 E85DFDFFFF
1037 0000779E E94FFEFFFF
1038
                                  <1> loc_set_time_stc_5:
                                 <1> call check_for_backspace
<1> je short loc_set_time_bs_5
1039 000077A3 E865FDFFFF
1040 000077A8 740A
                                 <1>
1041
                                  <1>
                                           ;xor bh, bh ; video page 0
                                       call beeper ; BEEP !
    jmp    loc_enter_time_separator_2
1042 000077AA E8ECA5FFFF
                                  <1>
1043 000077AF E994FEFFFF
                                  <1>
                                  <1> loc_set_time_bs_5:
1045 000077B4 E842FDFFFF
                                  <1> call write_backspace
                                           jmp loc_enter_minute_2
1046 000077B9 E95FFEFFFF
                                  <1>
                                  <1> loc_set_time_stc_6:
1048 000077BE E84AFDFFFF
                                  <1> call check_for_backspace
1049 000077C3 7413
                                  <1>
                                           jе
                                                  short loc_set_time_bs_6
                                       ;xor bh, bh; video call beeper; BEEP!
                                           ;xor bh, bh ; video page 0
                                  <1>
1051 000077C5 E8D1A5FFFF
                                 <1>
                                           mov word [Second], 3030h
1052 000077CA 66C705[D50E0100]30- <1>
1052 000077D2 30
                                 <1>
1053 000077D3 E99FFEFFFF
                                 <1>
                                            jmp loc_enter_second_1
                                  <1> loc_set_time_bs_6:
                                       call write_backspace
1055 000077D8 E81EFDFFFF
                                 <1>
1056 000077DD E966FEFFFF
                                  <1>
                                           jmp loc_enter_time_separator_2
1057
                                  <1> loc_set_time_stc_7:
                                  <1> call check_for_backspace
<1> je short loc_set_time_bs_7
1058 000077E2 E826FDFFFF
1059 000077E7 740A
                                 <1>
1060
                                  <1>
                                           ;xor bh, bh ; video page 0
1061 000077E9 E8ADA5FFFF
                                 <1> call beeper ; BEEP !
<1> jmp loc_enter_second_2
1062 000077EE E9B3FEFFFF
1063
                                  <1> loc_set_time_bs_7:
1064 000077F3 E803FDFFFF
                                  <1>
                                       call write_backspace
1065 000077F8 E97AFEFFFF
                                  <1>
                                            jmp loc_enter_second_1
1066
                                  <1>
1067
                                  <1> print_volume_info:
                                        ; 01/03/2016
1068
                                  <1>
                                           ; 08/02/2016
1069
                                  <1>
1070
                                  <1>
                                          ; 06/02/2016
                                          ; 04/02/2016
1071
                                  <1>
1072
                                           ; 18/01/2016 (TRDOS 386 = TRDOS v2.0)
                                  <1>
                                          ; 25/10/2009
1073
                                  <1>
1074
                                  <1>
                                           ; "Volume Serial No: "
1075
                                  <1>
1076
                                  <1>
                                           ; INPUT : AL = DOS Drive Number
1077
                                  <1>
1078
                                  <1>
                                           ; OUTPUT : AH = FS Type
1079
                                           ; AL = DOS Drive Name
                                  <1>
                                           ; CF = 0 -> OK
1080
                                  <1>
                                            ; CF = 1 \rightarrow Drive not ready
1081
                                  <1>
<1>
                                                 short loc pvi set vol name
1092
                                 <1>
1093
                                <1> loc_pvi_set_vol_name:
                                 <1>
                                        mov
1094 00007814 A2[090F0100]
                                                 [Vol_Drv_Name], al
                                 <1>
1095 00007819 56
                                           push esi
                          <1>
<1>
1096 0000781A E858010000
                                           call move volume_name_and_serial_no ;;;
1097 0000781F 7302
                                           jnc short loc_pvi_mvn_ok
1098 00007821 5E
                                 <1>
                                           pop
1099 00007822 C3
                                <1>
                                           retn
1100
                                <1>
                                <1> loc_pvi_mvn_ok:
<1> mov esi, [esp]
<1> mov byte [esi]
1101
1102 00007823 8B3424
1103 00007826 807E04A1
                               <1>
                                            cmp byte [esi+LD_FSType], 0A1h
1104 0000782A 7509
                                 <1>
                                           jne short loc pvi fat vol size
```

```
movzx ebx, word [esi+LD_BPB+BPB_BytsPerSec]
1126 0000785C BF[D1600100] <1>
1127
1138
                                                          <1>
 1139
                                                           <1> loc_write_vol_size_str_ok:
 1140 00007876 893D[C3600100]
                                                          <1> mov [Vol_Tot_Sec_Str_Start], edi
<1>
                                                                             ;
                                                       <1> class of the contract of t
 1152
1153
                                               <1>
<1> loc_vol_freespace_recalc0:
<1> ; 01/03/2016
<1> cmp eax, 0FFFFFFFFH
<1> jb short loc_vol_freespace_mul32
<1> ;inc eax; 0
<1> and cl, cl; byte [esi+LD_FATType]
<1> jz short loc_vol_freespace_mul32
<1> push ebx
<1> mov bx, 0FF00h; recalculate free sectors
<1> call calculate_fat_freespace
 1168
 1169
 1170 000078BF 83F8FF
 1171 000078C2 720F
 1172
 1173 000078C4 20C9
 1174 000078C6 740B
 1175 000078C8 53
 1176 000078C9 66BB00FF
1177 000078CD E876490000
                                                                            call calculate_fat_freespace
 1178 000078D2 5B
                                                          <1>
                                                                           pop
                                                                                        ebx
 1179
                                                           <1>
                                                          <1> loc_vol_freespace_mul32:
 1180
                                                    <1> mul ebx <1> or edx,
 1181 000078D3 F7E3
 1182 000078D5 09D2
                                                                                        edx, edx
                                                                                      short loc_vol_fspace_in_kbytes
 1183 000078D7 7507
                                                           <1>
                                                                             jnz
                                                          <1> loc_vol_fspace_in_bytes:
 1184
                                                                     mov ecx, VolSize_Bytes
 1185 000078D9 B9[E70E0100]
                                                       <1>
 1186 000078DE EB0D
                                                                                         short loc write vol fspace str
                                                            <1>
                                                                              jmp
                                                            <1> loc_vol_fspace_in_kbytes:
 1187
 1188 000078E0 66BB0004
                                                            <1>
                                                                            mov bx, 1024
 1189 000078E4 F7F3
                                                            <1>
                                                                             div
                                                                                        ebx
 1190 000078E6 B9[DA0E0100]
                                                                                        ecx, VolSize KiloBytes
                                                           <1>
                                                                             mov
                                                                          xor edx, edx; 0
 1191 000078EB 31D2
                                                           <1>
                                                           <1> loc_write_vol_fspace_str:
 1192
 1193 000078ED 890D[BF600100]
                                                           <1>
                                                                            mov [VolSize_Unit2], ecx
 1194
                                                           <1>
 1195 000078F3 BF[E1600100]
                                                           <1>
                                                                             mov edi, Vol_Free_Sectors_Str_End
 1196
                                                            <1>
                                                                             ;mov byte [edi], 0
 1197 000078F8 B90A000000
                                                           <1>
                                                                             mov ecx, 10
 1198
                                                          <1> loc_write_vol_fspace_chr:
 1199 000078FD F7F1
                                                           <1>
                                                                        __div__ecx
                                                                                        dl, '0'
                                                          <1>
 1200 000078FF 80C230
                                                                             add
 1201 00007902 4F
                                                          <1>
                                                                            dec edi
                                                          <1>
 1202 00007903 8817
                                                                                        [edi], dl
                                                                            mov
 1203 00007905 85C0
                                                          <1>
                                                                             test eax, eax
 1204 00007907 7404
                                                          <1>
                                                                                        short loc_write_vol_fspace_str_ok
                                                                             iΖ
                                                          <1>
 1205 00007909 28D2
                                                                          sub
                                                                                        dl, dl ; 0
                                                                          jmp short loc_write_vol_fspace chr
 1206 0000790B EBF0
                                                           <1>
 1207
                                                           <1>
                                                           <1> loc write vol fspace str ok:
 1208
 1209 0000790D 893D[D3600100]
                                                           <1> mov [Vol_Free_Sectors_Str_Start], edi
```

```
1211 00007913 BE[F00E0100]
                                  <1>
                                           mov
                                                 esi, Volume in drive
1212 00007918 E8B0EAFFFF
                                           call print msg
                                 <1>
1213 0000791D BE[300F0100]
                                 <1>
                                                  esi, Vol_Name
                                            mov
1214 00007922 E8A6EAFFFF
                                 <1>
                                           call print msg
1215 00007927 BE[FF190100]
                                 <1>
                                           mov
                                                  esi, nextline
1216 0000792C E89CEAFFFF
                                           call print msq
                                 <1>
1217
                                  <1>
1218 00007931 BE[910F0100]
                                 <1>
                                           mov
                                                  esi, Vol_Total_Sector_Header
1219 00007936 E892EAFFFF
                                  <1>
                                           call print msq
1220 0000793B 8B35[C3600100]
                                  <1>
                                           mov
                                                  esi, [Vol_Tot_Sec_Str_Start]
1221 00007941 E887EAFFFF
                                  <1>
                                           call
                                                 print msg
                                                  esi, [VolSize_Unit1]
1222 00007946 8B35[BB600100]
                                  <1>
                                           mov
                                           call print msg
1223 0000794C E87CEAFFFF
                                  <1>
                                  <1>
1225 00007951 BE[A20F0100]
                                  <1>
                                           mov
                                                  esi, Vol_Free_Sectors_Header
1226 00007956 E872EAFFFF
                                  <1>
                                           call print msg
                                  <1>
                                                  esi, [Vol_Free_Sectors_Str_Start]
1227 0000795B 8B35[D3600100]
                                           mov
1228 00007961 E867EAFFFF
                                  <1>
                                            call
                                                  print_msg
                                                  esi, [VolSize_Unit2]
1229 00007966 8B35[BF600100]
                                  <1>
                                           mov
1230 0000796C E85CEAFFFF
                                  <1>
                                            call print_msg
                                  <1>
                                           ;
1232 00007971 5E
                                  <1>
                                           pop
                                                  esi
1233
                                  <1>
                                                  ah, [esi+LD_FSType]
1234
                                  <1>
                                            ; mov
1235
                                  <1>
                                                  al, [esi+LD_FATType]
                                            ; mov
1236 00007972 668B4603
                                  <1>
                                                  ax, [esi+LD_FATType]
                                            mov
1237
                                  <1>
1238 00007976 C3
                                  <1>
1239
                                  <1>
1240
                                  <1> move_volume_name_and_serial_no:
                                         \overline{\phantom{a}}; 08/\overline{0}2/20\overline{1}6 (TRDOS \overline{3}86 = TRDOS v2.0)
1241
                                  <1>
                                            ; this routine will be called by
1242
                                  <1>
1243
                                  <1>
                                           ; "print_volume_info" and "print_directory"
1244
                                  <1>
                                           ; INPUT ->
1245
                                  <1>
                                                 ESI = Logical DOS drv descripton table address
                                           ; OUTPUT ->
1246
1247
                                  <1>
                                                  *Volume name will be moved to text area
1248
                                  <1>
                                                  *Volume serial number will be converted to
1249
                                  <1>
                                                   text and will be moved to text area
                                           ; cf = 1 -> invalid/unknown dos drive
1250
                                  <1>
                                           ; cf = 0 \rightarrow ecx = 0
1251
                                  <1>
                                  <1>
1252
1253
                                  <1>
                                           ; (eax, edx, ecx, esi, edi will be changed)
1254
                                  <1>
1255 00007977 BF[300F0100]
                                  <1>
                                           mov edi, Vol_Name
                                  <1>
1257
                                  <1>
                                           ;mov ah, [esi+LD_FSType]
                                                  al, [esi+LD FATType]
1258
                                  <1>
                                                  ax, [esi+LD_FATType]
1259 0000797C 668B4603
                                 <1>
                                            mov
1260 00007980 80FCA1
                                                  ah, 0A1h
                                 <1>
                                            cmp
1261 00007983 7418
                                 <1>
                                            jе
                                                  short mvn 2
1262 00007985 08E4
                                 <1>
                                            or
                                                  ah, ah
                                                  short mvn_0
1263 00007987 7404
                                 <1>
                                            jz
1264 00007989 08C0
                                 <1>
                                                  al, al
                                            or
1265 0000798B 7504
                                 <1>
                                            jnz
                                                  short mvn_1
                                 <1> mvn 0:
1267 0000798D 8A06
                                 <1>
                                           mov
                                                  al, [esi]
1268 0000798F F9
                                 <1>
                                            stc
1269 00007990 C3
                                 <1>
                                            retn
1270
                                 <1> mvn_1:
1271 00007991 3C02
                                 <1>
                                           cmp
                                                  al, 2
1272 00007993 7717
                                 <1>
                                            jа
                                                  short mvn_3
1273
                                 <1>
                                                  al, al
1274
                                 <1>
                                           ;jz
                                                  short mvn_2
1275 00007995 8B462D
                                 <1>
                                            mov
                                                  eax, [esi+LD_BPB+VolumeID]
1276 00007998 83C631
                                                  esi, LD BPB+VolumeLabel
                                 <1>
                                            add
1277 0000799B EB15
                                 <1>
                                            jmp
                                                  short mvn_4
1278
                                 <1> mvn 2:
                                                  eax, [esi+LD_FS_VolumeSerial]
1279 0000799D 8B4628
                                 <1>
                                           mov
1280 000079A0 83C62C
                                 <1>
                                            add
                                                  esi, LD_FS_VolumeName
1281 000079A3 B910000000
                                  <1>
                                            mov
                                                  ecx, 16
1282 000079A8 F3A5
                                 <1>
                                            rep
                                                  movsd
1283 000079AA EB10
                                 <1>
                                            jmp
                                                  short mvn_5
1284
                                  <1> mvn_3:
1285 000079AC 8B4649
                                                  eax, [esi+LD_BPB+FAT32_VolID]
                                 <1>
                                           mov
1286 000079AF 83C64D
                                  <1>
                                            add
                                                  esi, LD BPB+FAT32 VolLab
                                  <1> mvn_4:
1288 000079B2 B90B000000
                                  <1>
                                                  ecx, 11
1289 000079B7 F3A4
                                 <1>
                                                  movsb
                                           rep
1290 000079B9 C60700
                                  <1>
                                           mov
                                                  byte [edi], 0
                                  <1> mvn 5:
                                            ;mov [Current_VolSerial], eax
                                  <1>
1293 000079BC E851B9FFFF
                                  <1>
                                            call dwordtohex
1294 000079C1 8915[850F0100]
                                  <1>
                                                 [Vol Serial1], edx
                                           mov
1295 000079C7 A3[8A0F0100]
                                  <1>
                                            mov
                                                 [Vol_Serial2], eax
                                           ; ecx = 0
                                  <1>
1297 000079CC C3
                                  <1>
                                           retn
1298
                                  <1>
1299
                                  <1> get volume serial number:
1300
                                           ; 19/01/2016 (TRDOS 386 = TRDOS v2.0)
                                  <1>
1301
                                            ; 08/08/2010
                                  <1>
1302
                                  <1>
1303
                                  <1>
                                           ; INPUT -> DL = Logical DOS Drive number
1304
                                  <1>
                                           ; OUTPUT -> EAX = Volume serial number
1305
                                  <1>
                                                       BL= FAT Type
1306
                                  <1>
                                                      BH = Logical DOS drv Number (DL input)
1307
                                  <1>
                                           ; cf = 1 -> Drive not ready
1308
                                  <1>
1309 000079CD 31DB
                                  <1>
                                           xor
                                                  ebx, ebx
                                                 bh, dl
1310 000079CF 88D7
                                  <1>
                                           mov
1311 000079D1 3815[610D0100]
                                 <1>
                                            cmp
                                                  [Last_DOS_DiskNo], dl
1312 000079D7 7304
                                 <1>
                                            jnb
                                                 short loc_gvsn_start
1313
                                 <1> loc_gvsn_stc_retn:
1314 000079D9 31C0
                                 <1>
                                           xor eax, eax
```

<1>

1210

```
1316 000079DC C3
                                 <1>
                                           retn
                                <1> loc_gvsn_start:
1317
1318 000079DD 56
                                <1> push esi
                          c1> push esi
c1> mov esi, Logical_DOSDisk:
c1> add esi, ebx
c1> mov bl, [esi+LD_FATType]
c1> and bl, bl
c1> jz short loc_gvsn_fs
c1> cmp bl, 2
c1> ja short loc_gvsn_fat32
c1> loc_gvsn_fat;
1319 000079DE BE00010900
                                                 esi, Logical_DOSDisks
1320 000079E3 01DE
1321 000079E5 8A5E03
1322 000079E8 20DB
1323 000079EA 740F
1324 000079EC 80FB02
1325 000079EF 7705
                                <1> loc gvsn fat:
1326
                                <1> add esi, LD_BPB + VolumeID
<1> imp short loc gysn return
1327 000079F1 83C62D
                                           jmp short loc_gvsn_return
1328 000079F4 EB0E
                                <1>
                                <1> loc_gvsn_fat32:
1329
                                <1> add esi, LD_BPB + FAT32_VolID
<1> jmp short loc_gvsn_return
1330 000079F6 83C649
1331 000079F9 EB09
                                                short loc_gvsn_return
                                <1> loc_gvsn_fs:
1332
1333 000079FB 807E04A1
                                 <1> cmp byte [esi+LD_FSType], 0A1h
                                                short loc_gvsn_stc_retn
1334 000079FF 75D8
                                <1>
                                           jne
                              <1> add esi, LD_FS_VolumeSerial
1335 00007A01 83C628
                                <1> loc_gvsn_return:
                                <1> mov eax, [esi]
1337 00007A04 8B06
1338 00007A06 5E
                                <1>
                                           pop
                                                 esi
1339 00007A07 C3
                                 <1>
                                           retn
1340
                                 <1>
1341
                                 <1> ; CMD INTR.ASM [ TRDOS Command Interpreter Procedure ]
                                 <1>; 09/\overline{1}1/2011
1342
1343
                                 <1>; 29/01/2005
1344
                                 <1>
1345
                                 <1> command_interpreter:
                                       ; 16/10/2016
1346
                                 <1>
                                           ; 12/10/2016
1347
                                 <1>
                                        ; 13/05/2016
1348
                                 <1>
                                        ; 07/05/2016
; 04/03/2016
; 04/02/2016
1349
                                 <1>
1350
                                 <1>
1351
                                 <1>
                                         ; 03/02/2016
1352
                                 <1>
                                         ; 30/01/2016
; 29/01/2016 (TRDOS 386 = TRDOS 2.0)
1353
                                 <1>
1354
                                 <1>
1355
                                 <1>
                                         ; 15/09/2011
                                          ; 29/01/2005
1356
                                 <1>
1357
                                 <1>
1358
                                 <1>
                                           ; Input: ecx = command word length (CL)
                                          ; CommandBuffer = Command string offset
1359
                                 <1>
1360
                                 <1>
1361 00007A08 C605[74610100]00
                                 <1>
                                           mov byte [Program_Exit],0
1362 00007A0F 80F904
                                           cmp
                                 <1>
                                                cl, 4
                                                  c_6
1363 00007A12 0F87B5020000
                                 <1>
                                            jа
1364 00007A18 0F8237010000
                                 <1>
                                             jb
                                                     c_2
1365
                                 <1> c_4:
1366
                                 <1>
1367
                                 <1> cmp_cmd_exit:
1368 00007A1E BF[CF0D0100]
                                 <1> mov edi, Cmd_Exit
1369 00007A23 E8C2030000
                                 <1>
                                          call cmp_cmd
                                         jc
1370 00007A28 7208
                                 <1>
                                                 short cmp_cmd_date
                                 <1>
1372 00007A2A C605[74610100]01 <1>
                                          mov
                                                     byte [Program_Exit], 1
1373 00007A31 C3
                                 <1>
                                             retn
1374
                                 <1>
1375
                                 <1> cmp_cmd_date:
                                 <1> mov cl, 4
<1> mov edi, Cmd_Date
1376 00007A32 B104
1377 00007A34 BF[EB0D0100]
                                <1>
                                       call cmp_cmd
1378 00007A39 E8AC030000
                                <1>
                                         jc short cmp_cmd_time
1379 00007A3E 720B
                                 <1>
1380
                                 <1>
1381 00007A40 E8D0F7FFFF
                                <1>
                                         call show date
1382 00007A45 E80FF8FFFF
                                 <1>
                                         call set_date
1383 00007A4A C3
                                 <1>
                                           retn
1384
                                 <1>
1385
                                 <1> cmp_cmd_time:
1386 00007A4B B104
                                 <1> mov cl, 4
1387 00007A4D BF[F00D0100]
                                                 edi, Cmd Time
                                <1>
                                          mov
1388 00007A52 E893030000
                                <1>
                                        call cmp_cmd
1389 00007A57 720B
                                 <1>
                                                 short cmp_cmd_show
                                         jc
1390
                                 <1>
                                        call show_time
1391 00007A59 E8C6FAFFFF
                                 <1>
                                         call set_time
1392 00007A5E E8F8FAFFFF
                                 <1>
1393 00007A63 C3
                                 <1>
                                           retn
1394
                                 <1>
1395
                                 <1> cmp_cmd_show:
1396 00007A64 B104
                                  <1>
                                           mov
1397 00007A66 BF[010E0100]
                                           mov edi, Cmd_Show
                                <1>
1398 00007A6B E87A030000
                                 <1>
                                           call cmp_cmd
                                           jnc show_file
1399 00007A70 0F83050A0000
                                 <1>
1400
                                 <1>
1401
                                 <1> cmp_cmd_echo:
1402 00007A76 B104
                                <1>
                                           mov cl, 4
1403 00007A78 BF[3D0E0100]
                                <1>
                                           mov
                                                 edi, Cmd_Echo
                                           call cmp_cmd
1404 00007A7D E868030000
                                <1>
                                <1>
1405 00007A82 7224
                                         jc short cmp_cmd_copy
                                 <1>
1406
1407
                                          ; 22/11/2017
                                 <1>
1408
                                <1>
                                         ; AL = 0
1409 00007A84 803E20
                                 <1>
                                          cmp byte [esi], 20h
1410 00007A87 7215
                                 <1>
                                           jb
                                                 short cmd_echo_nextline
                                           ; 14/04/2016
                                 <1>
1412 00007A89 56
                                          push esi
                                 <1>
1413
                                 <1> cmd_echo_asciiz:
1414
                                       ;inc esi
                                 <1>
                                           ;mov al, [esi]
1415
                                 <1>
1416
                                 <1>
                                           ; 22/11/2017
1417 00007A8A AC
                                <1>
                                           lodsb
1418 00007A8B 3C20
                                           cmp al, 20h
                                <1>
1419 00007A8D 73FB
                                 <1>
                                          jnb short cmd_echo_asciiz
```

1315 000079DB F9

<1>

stc

```
1428
                           <1>
                                   ;call print_msg
                                ;retn
jmp print_msg
1429
                           <1>
1430 00007AA3 E925E9FFFF <1>
1431 <1>
<1> cmp_cmd_copy:
                           <1> ___ mov __ cl, 4
                                   mov edi, Cmd_Copy call cmp_cmd
                                call cmp_cmd
jnc copy_file
<1> cmp_cmd_move:
                          <1> mov cl, 4
                                   mov
                                        edi, Cmd_Move
                                call cmp_cmd
jnc move_file
<1>
                           <1> cmp_cmd_path:
                           <1> mov cl, 4
<1> mov edi, Cmd_Path
                                call cmp_cmd
                                  jnc set_get_path
1450
                           <1> cmp_cmd_beep:
1459
                           <1> cmp_cmd_find:
edi, Cmd Find
                                   call cmp_cmd
                                   jc cmp_cmd_external
1464
                           <1>
                                   ;call find_and_list_files
1465
                           <1>
1466 00007B09 E9AF220000
                           <1>
                                    jmp find_and_list_files
                           <1>
                                    ;retn
1467
1468
                           <1>
                           <1> c_1:
1469
1470 00007B0E AD
                           <1>
                                    lodsd
1471
                           <1> cmp_cmd_help:
                                   cmp al, '?'
1472 00007B0F 3C3F
                           <1>
1473 00007B11 751D
                           <1>
                                    jne short cmp_cmd_remark
1474
                           <1>
                                    mov esi, Command_List
1475 00007B13 BE[C10D0100]
                           <1>
                           <1> cmd help next w:
1477 00007B18 E8B0E8FFFF
                                   call print_msg
                           <1>
1478
                           <1>
1479 00007B1D 803E20
                           <1>
                                         byte [esi], 20h; 0
                                    cmp
1480 00007B20 7232
                           <1>
                                   jb
                                         short cmd_help_retn
                           <1>
                                   push esi
1482 00007B22 56
                           <1>
1483 00007B22 56 (15)
1483 00007B23 BE[FF190100] (15)
1484 00007B28 E8A0E8FFFF (15)
                                   mov
                                         esi, nextline
                                    call print_msg
1485 00007B2D 5E
                           <1>
                                    pop
                                         esi
1486 00007B2E EBE8
                           <1>
                                    jmp
                                        short cmd_help_next_w
1487
                           <1>
1488
                           <1> cmp_cmd_remark:
1489 00007B30 3C2A <1>
1490 00007B32 0F8595020000 <1>
1491 00007B38 46 <1>
                                   cmp al, '*'
                           <1>
                                    jne cmp_cmd_external
                                   inc esi
1491 00007B38 46
                           <1>
                           1492 00007B39 BF[E8590100]
                                        edi, Remark
1493 00007B3E 8A06
1494 00007B40 3C20
1495 00007B42 7707
                                         short cmd remark write
                                    jа
1496 00007B44 89FE
                           <1>
                                   mov
                                        esi, edi ; Remark
1497 00007B46 E982E8FFFF
                           <1>
                                   jmp print_msg
1498
                           <1>
1499
                            <1> cmd remark write:
1500 00007B4B AA
                            <1>
                                   stosb
1501 00007B4C AC
                            <1>
                                    lodsb
1502 00007B4D 3C20
                           <1>
                                    cmp
                                        al, 20h
1503 00007B4F 73FA
                            <1>
                                        short cmd remark write
                                    jnb
1504 00007B51 C60700
                            <1>
                                        byte [edi], 0
                                   mov
1505
                            <1>
1506
                            <1> cmd help retn:
1507
                            <1> cmd remark retn:
                            <1> cd retn:
1508
1509 00007B54 C3
                            <1>
                                   retn
1510
                            <1>
                            <1> c 2:
1511
1512 00007B55 80F902
                                        cl, 2
                           <1>
                                    cmp
1513 00007B58 0F87AF000000
                           <1>
                                    ja c_3
1514 00007B5E BE[365A0100]
                            <1>
                                    mov esi, CommandBuffer
1515 00007B63 72A9
                           <1>
                                    jb
                                         short c_1
                            <1>
                           <1> cmp_cmd_cd:
1517
1518 00007B65 66AD
                           <1>
                                   lodsw
                                    cmp ax, 'CD'
1519 00007B67 663D4344
                           <1>
1520 00007B6B 7551
                                    jne short cmp_cmd_drive
                           <1>
                           <1> inc esi
<1> cd_0:
1521 00007B6D 46
1522
1523 00007B6E 668B06
                            <1>
                                  mov ax, [esi]
1524 00007B71 3C20
                            <1>
                                    cmp al, 20h
```

```
jna short cd_retn
; 10/02/2016
cmp ah, ':'
jne short cd_1
                                  <1>
1525 00007B73 76DF
                                  <1>
                             1527 00007B75 80FC3A
1528 00007B78 7504
                                 <1>
<1>
1529 00007B7A 46
                                             inc
                                                   esi
1530 00007B7B 46
                                             inc
                                                   esi
1531 00007B7C EB49
                                  <1>
                                            jmp
                                                  short cd 2
1532
                                  <1>
1533
                                  <1> cd_1: ; change current directory
                                           ; 29/11/2009
1534
                                   <1>
1535
                                   <1>
                                             ; AH = CDh ; to separate 'CD' command from others
1536
                                   <1>
                                                          ; for restoring current directory
1537
                                   <1>
                                                           ; OCDh sign is for saving cdir into
                                                           ; DOS drv description table cdir area
1538
                                   <1>
1539
                                   <1>
                                             mov
                                                    ah, OCDh; mov byte [CD_COMMAND], OCDh
1540 00007B7E B4CD
                                  <1>
                                  <1>
                                             call change_current_directory
1542 00007B80 E81D230000
                                  <1>
1543 00007B85 0F8337220000
                                  <1>
                                             jnc change prompt dir string
1544
                                  <1>
1545
                                  <1> cd_error_messages:
                                         cmp al, 3
je short cd_path_not_found
1546 00007B8B 3C03
                                  <1>
                             1547 00007B8D 740C

1549 00007B8F 3C0F
1550 00007B91 7459
1551 00007B93 3C11
1552 00007B95 7455
1553 00007B97 3C13
1554 00007B99 7466
1555
                                  <1>
                                  1556
1557 00007В9В 50
                                  <1>
1558
                                  <1>
                                            ;push ax
                                        mov esi, Msg_Dir_Not_Found
1559 00007B9C BE[64100100]
1560 00007BA1 E827E8FFFF
                                  <1>
                                  <1>
                                            call print_msg
                                        ;pop ax
                                  <1>
                                       pop eax; 29/12/2017
cmp ah, [Current_Dir_Level]
jnb change_prompt_dir_string
1562 00007BA6 58
1562 00007BA6 58
1563 00007BA7 3A25[84590100] <1>
00007BA7 3A25[84590100] <1>
1565 00007BB3 8825[84590100]
                                          mov [Current_Dir_Level], ah
                                  <1>
1566 00007BB9 E904220000
                                  <1>
                                           jmp change_prompt_dir_string
1567
                                  <1>
1568
                                  <1> cmp_cmd_drive: ; change current drive
                                          ; C:, D:, E: etc.
1569
                                  <1>
1570 00007BBE 80FC3A
                                  <1>
                                             cmp ah, ':'
                                             jne cmp_cmd_external
1571 00007BC1 0F8506020000
                                  <1>
1572
                                  <1>
                                  <1> cd 2: ; 'CD C:', 'CD D:' ...
1573
                                  <1> cmp byte [esi], 20h
<1> ja loc_cmd_failed
1574 00007BC7 803E20
1575 00007BCA 0F8707020000
                                 <1>
1580
                                  <1>
                                        cmp al, [Last_DOS_DiskNo]
ja short cd_drive_not_ready
1581 00007BDA 3A05[610D0100] <1>
1582 00007BE0 770A
                                  <1>
1583
                                  <1>
                                        mov dl, al
call change_current_drive
1584 00007BE2 88C2 <1>
1585 00007BE4 E85BF3FFFF <1>
                                           jc
1586 00007BE9 7201
                                  <1>
                                                   short cd_drive_not_ready
1587 00007BEB C3
                                  <1>
                                            retn
1588
                                  <1>
                                  <1> cd_drive_not_ready:
1589
1590 00007BEC BE[21100100]
                                  <1>
                                           mov esi, Msg_Not_Ready_Read_Err
                                             call print_msg
1591 00007BF1 E8D7E7FFFF
                                  <1>
1592
                                  <1>
1593
                                  <1> cd_fail_drive_restart:
                                 1594 00007BF6 8A15[86590100]
                                            ; call change current drive
1595
1596 00007BFC E943F3FFFF
                                                     change_current_drive
                                   <1>
                                             jmp
                                  <1>
1597
                                             ;retn
1598
                                  <1>
                                  <1> cd_command_failed:
1599
1600 00007C01 BE[02100100]
                                  <1> mov esi, Msg_Bad_Command
                                             call print msg
1601 00007C06 E8C2E7FFFF
                                   <1>
                                             jmp short cd fail drive restart
1602 00007C0B EBE9
                                  <1>
1603
                                   <1>
1604
                                   <1> c_3:
                                   <1> cmp_cmd_dir:
1605
1606 00007C0D BF[C10D0100]
                                   <1>
                                             mov
                                                    edi, Cmd_Dir
                                             call cmp_cmd
1607 00007C12 E8D3010000
                                   <1>
1608 00007C17 0F8380020000
                                   <1>
                                             jnc print_directory_list
1609
                                   <1>
1610
                                   <1> cmp_cmd_cls:
1611 00007C1D B103
                                  <1>
                                        mov cl, 3
                                             mov edi, Cmd_Cls call cmp_cmd
1612 00007C1F BF[FD0D0100]
                                  <1>
1613 00007C24 E8C1010000
                                  <1>
1614 00007C29 0F83B4E7FFFF
                                  <1>
                                             jnc clear_screen
1615
                                   <1>
1616
                                   <1> cmp_cmd_ver:
1617 00007C2F B103
                                            mov cl, 3
                                  <1>
1618 00007C31 BF[CB0D0100]
                                  <1>
                                                   edi, Cmd_Ver
1619 00007C36 E8AF010000
                                   <1>
                                             call cmp_cmd
1620 00007C3B 720A
                                   <1>
                                             jс
                                                    short cmp_cmd_mem
                                   <1>
1622 00007C3D BE[690D0100]
                                  <1>
                                           mov esi, mainprog_Version
1623
                                   <1>
                                             ;call print_msg
1624 00007C42 E986E7FFF
                                             jmp print_msg
                                   <1>
1625
                                   <1>
                                             ;retn
1626
                                   <1>
                                   <1> cmp_cmd_mem:
1627
1628 00007C47 B103
                                        mov cl, 3
                                   <1>
1629 00007C49 BF[330E0100]
                                  <1>
                                             mov edi, Cmd_Mem
```

```
call cmp_cmd
1631 00007C53 0F8304B6FFFF
                               <1>
                                         jnc memory_info
                               <1>
1633
                               <1> cmp_cmd_del:
                                      mov cl, 3
mov edi, Cmd_Del
1634 00007C59 B103
                               <1>
1635 00007C5B BF[060E0100]
                               <1>
                                    call cmp_cmd
1636 00007C60 E885010000
                               <1>
1637 00007C65 0F83280F0000
                               <1>
                                       jnc delete_file
1638
                               <1>
                               <1> cmp_cmd_set:
1639
1640 00007C6B B103
                               <1> mov cl, 3
1641 00007C6D BF[F90D0100]
                               <1>
                                               edi, Cmd Set
                                         mov
                                         call cmp_cmd
1642 00007C72 E873010000
                               <1>
1643 00007C77 0F83C9170000
                               <1>
                                         jnc set_get_env
1644
                               <1>
1645
                               <1> cmp_cmd_run:
1647 00007C7F BF[F50D0100]
1648 00007C84 E861010000
1649
                               <1> mov cl, 3
                               <1>
                                        mov edi, Cmd_Run call cmp_cmd
                               <1>
                                        ; 07/05/2016
                               <1>
                               <1> jc cmp_cmd_external
<1> jmp load_and_execute_file
<1> c_5:
1650 00007C89 0F823E010000
                               <1>
1651 00007C8F E90F1E0000
1652
                               <1> cmp_cmd_mkdir:
                               1654 00007C94 BF[1E0E0100]
1655 00007C99 E84C010000
1656 00007C9E 0F83990A0000
                               <1>
                                         jnc make_directory
                               <1>
1657
1658
                               <1> cmp_cmd_rmdir:
1659 00007CA4 B105
                               <1> mov cl, 5
1660 00007CA6 BF[180E0100]
                               <1>
                                         mov
                                              edi, Cmd_Rmdir
1661 00007CAB E83A010000
                               <1>
                                         call cmp_cmd
1662 00007CB0 0F83AA0B0000
                               <1>
                                                  delete_directory
                                         jnc
1663
                               <1>
1664
                               <1> cmp_cmd_chdir:
                               <1> mov cl, 5
1665 00007CB6 B105
1666 00007CB8 BF[550E0100]
                                         mov edi, Cmd Chdir
                               <1>
1667 00007CBD E828010000
                                         call cmp_cmd
                               <1>
1668 00007CC2 0F8205010000
                                               cmp cmd external
                               <1>
                                        jс
1669
                               <1>
1670 00007CC8 E9A1FEFFFF
                               <1>
                                         jmp
                                               cd_0
1671
                                <1>
                               <1> c_6:
1672
1673 00007CCD 80F906
                               <1>
                                         cmp cl, 6
1674 00007CD0 0F87E0000000
                               <1>
                                         ja c_8
                                         jb short c_5
1675 00007CD6 72BC
                               <1>
1676
                               <1> cmp_cmd_prompt:
                               <1> mov edi, Cmd_Prompt
<1> call cmp_cmd
1677 00007CD8 BF[D40D0100]
1678 00007CDD E808010000
                                     jc short cmp_cmd_volume
1679 00007CE2 722F
                               <1>
1680
                             <1> get_prompt_name_fchar:
                             <1> lodsb <1> cmp
1681 00007CE4 AC
1682 00007CE5 3C20
                                         cmp al, 20h
                                     je short get_prompt_name_fchar
ja short loc change prompt labo
1683 00007CE7 74FB
                             <1>
1684 00007CE9 7713
                               <1>
                                             short loc_change_prompt_label
1685
                               <1> default_command_prompt: ; 31/12/2017 ('sysprompt')
                               <1> mov esi, TRDOSPromptLabel
1686 00007CEB BE[B50D0100]
1687 00007CF0 C7065452444F
                               <1>
                                         mov dword [esi], "TRDO"
1688 00007CF6 66C746045300
                               <1>
                                               mov word [esi+4], "S"
1689
                               <1> loc_cmd_prompt_return:
1690 00007CFC C3
                               <1>
                                       retn
1691
                                <1>
1692
                               <1> set_command_prompt: ; 31/12/2017 ('sysprompt')
1693 00007CFD AC
                               <1>
                                      lodsb
1694
                               <1> loc_change_prompt_label:
                                      mov cx, 11
mov edi, TRDOSPromptLabel
1695 00007CFE 66B90B00
                               <1>
1696 00007D02 BF[B50D0100]
                               <1>
1697
                               <1> put_char_new_prompt_label:
1698 00007D07 AA
                               <1>
                                         stosb
1699 00007D08 AC
                               <1>
                                         lodsb
                                         cmp al, 20h
1700 00007D09 3C20
                               <1>
1701 00007D0B 7202
                                         jb short pass_put_new_prompt_label
loop put_char_new_prompt_label
                               <1>
1702 00007D0D E2F8
                               <1>
                               <1> pass_put_new_prompt_label:
1703
1704 00007D0F C60700
                               <1>
                                        mov byte [edi], 0
1705 00007D12 C3
                               <1>
                                         retn
1706
                                <1>
                                <1> cmp_cmd_volume:
1707
                                     mov cl, 6
1708 00007D13 B106
                                <1>
                                               edi, Cmd_Volume
1709 00007D15 BF[DB0D0100]
                               <1>
                                         mov
                                      call cmp_cmd
1710 00007D1A E8CB000000
                                <1>
1711 00007D1F 7255
                                <1>
                                       jc short cmp_cmd_attrib
1712
                                <1>
1713
                                <1> cmd vol1:
1714 00007D21 AC
                                <1>
                                         lodsb
                                              al, 20h
1715 00007D22 3C20
                                <1>
                                         cmp
1716 00007D24 7707
                                <1>
                                               short cmd vol2
                                         jа
1717 00007D26 A0[86590100]
                               <1>
                                               al, [Current_Drv]
                                         mov
                                               short cmd vol4
1718 00007D2B EB3D
                               <1>
                                         qmŗ
                                <1> cmd vol2:
1719
1720 00007D2D 3C41
                               <1>
                                         cmp
                                                al, 'A'
1721 00007D2F 0F82A2000000
                                               loc cmd failed
                                <1>
                                         jb
                                               al, 'z'
1722 00007D35 3C7A
                                <1>
                                         cmp
1723 00007D37 0F879A000000
                                <1>
                                               loc_cmd_failed
                                         jа
1724 00007D3D 3C5A
                                <1>
                                         cmp
                                               al, 'Z'
1725 00007D3F 760A
                                <1>
                                         jna
                                               short cmd_vol3
1726 00007D41 3C61
                                <1>
                                               al, 'a'
                                         cmp
1727 00007D43 0F828E000000
                                <1>
                                         jb
                                               loc_cmd_failed
1728 00007D49 24DF
                                <1>
                                         and
                                               al, ODFh
                                <1> cmd vol3:
1729
1730 00007D4B 8A26
                                               ah, [esi]
                               <1>
                                         mov
1731 00007D4D 80FC3A
                                <1>
                                         cmp ah, ':'
1732 00007D50 0F8581000000
                                         jne loc_cmd_failed
                               <1>
                                         sub al, 'A'
1733 00007D56 2C41
                                <1>
1734 00007D58 3A05[610D0100]
                                <1>
                                        cmp al, [Last_DOS_DiskNo]
```

<1>

1630 00007C4E E897010000

```
1735 00007D5E 760A
                                <1>
                                       jna short cmd_vol4
                                <1>
1737 00007D60 BE[21100100]
                               <1>
                                         mov
                                               esi, Msg_Not_Ready_Read_Err
1738 00007D65 E963E6FFFF
                               <1>
                                         jmp
                                               print_msg
1739
                                <1>
                               <1> cmd_vol4:
1740
1740
1741 00007D6A E88EFAFFFF
1742 00007D6F 0F8277FEFFFF
1743 00007D75 C3
                               <1> call print_volume_info
                               <1>
                                         jс
                                               cd_drive_not_ready
1743 00007D75 C3
                               <1>
1744
                               <1>
1745
                               <1> cmp_cmd_attrib:
                               <1> mov cl, 6 <1> mov edi, Cmd_Attrib
1746 00007D76 B106
1747 00007D78 BF[0A0E0100]
                              <1>
1748 00007D7D E868000000
                               <1>
                                         call cmp_cmd
1749 00007D82 0F831D0F0000
                               <1>
                                        jnc set_file_attributes
1750
                               <1>
                               <1> cmp_cmd_rename:
1751
1752 00007D88 B106
                               <1> mov cl, 6
1753 00007D8A BF[110E0100]
                               <1>
                                         mov
                                               edi, Cmd_Rename
                                         call cmp_cmd
1754 00007D8F E856000000
                               <1>
                                         jnc rename_file
1755 00007D94 0F8353110000
                               <1>
                               <1>
1757
                               <1> cmp_cmd_device:
1758 00007D9A B106
                               <1> mov cl, 6
                                         mov edi, Cmd_Device call cmp_cmd
1759 00007D9C BF[460E0100]
                               <1>
1760 00007DA1 E844000000
                               <1>
                                         jc short cmp_cmd_external
1761 00007DA6 7225
                               <1>
1762
                               <1>
1763 00007DA8 C3
                               <1>
                                         retn
1764
                                <1>
1765
                                <1> c_7:
1766
                                <1> cmp_cmd_devlist:
                               <1> mov edi, Cmd_DevList
1767 00007DA9 BF[4D0E0100]
1768 00007DAE E837000000
                               <1>
                                         call cmp_cmd
1769 00007DB3 7218
                               <1>
                                        jc short cmp_cmd_external
1770
                               <1>
1771
                               <1> loc cmd return:
1772 00007DB5 C3
                               <1>
                                         retn
1773
                               <1>
1774
                               <1> c_8:
1775 00007DB6 80F908
                             <1>
                                         cmp cl, 8
1776 00007DB9 7712
                                         ja short cmp_cmd_external ib short c.7
                               <1>
1777 00007DBB 72EC
                               <1>
                                         jb
                                               short c_7
1778
                               <1>
1779
                               <1> cmp_cmd_longname:
                                     mov edi, Cmd_LongName
1780 00007DBD BF[E20D0100]
                               <1>
                                         call cmp_cmd
1781 00007DC2 E823000000
                               <1>
                                         jnc __get_and_print_longname
1782 00007DC7 0F8350060000
                               <1>
1783
                                <1>
1784
                               <1> cmp_cmd_external:
1785
                               <1> ; 07/05/2016
1786
                                <1>
                                         ; 22/04/2016
1787 00007DCD BE[365A0100]
                               <1>
                                        mov esi, CommandBuffer
                                     jmp loc_run_check_filename
1788 00007DD2 E9CC1C0000
                               <1>
1789
                                <1>
                                <1> loc_cmd_failed:
1790
                               <1> cmp byte [CommandBuffer], 20h
1791 00007DD7 803D[365A0100]20
1792 00007DDE 76D5
                                         jna short loc_cmd_return
                                <1>
1793 00007DE0 BE[02100100]
                                <1>
                                               esi, Msg_Bad_Command
                                         mov
                               <1>; call print_msg
1794
1795
                                <1> ;loc_cmd_return:
1796
                                <1> ; retn
1797 00007DE5 E9E3E5FFFF
                                <1>
                                         jmp print_msg
1798
                                <1>
                                <1> cmp_cmd:
1799
                                     ; 29/01/2016 (TRDOS 386 = TRDOS v2.0)
1800
                                <1>
1801 00007DEA BE[365A0100]
                                <1>
                                          movesi, CommandBuffer
                                            ; edi = internal command word (ASCIIZ)
1802
                                <1>
1803
                                <1>
                                         ; ecx = command length (<=8)
1804
                               <1> cmp_cmd_1:
1805 00007DEF AC
                               <1> lodsb
1806 00007DF0 AE
                               <1>
                                         scasb
1807 00007DF1 750D
                                         jne short cmp_cmd_3
                               <1>
                                     loop cmp_cmd_1
1808 00007DF3 E2FA
                               <1>
1809 00007DF5 AC
                               <1>
                                         lodsb
                                         cmp al, 20h
ja short cmp_cmd_2
1810 00007DF6 3C20
                               <1>
1811 00007DF8 7703
                               <1>
1812 00007DFA 30C0
                                <1>
                                         xor al, al
1813
                                <1>
                                         ; ZF = 1 -> internal command word matches
1814 00007DFC C3
                                <1>
                                         retn
1815
                                <1> cmp_cmd_2:
                                         ; ZF = 0 (CF = 0) -> external command word
1816
                                <1>
1817 00007DFD 58
                                <1>
                                         pop
                                               eax ; no return to the caller from here
1818 00007DFE EBCD
                                <1>
                                         jmp
                                                cmp_cmd_external
                                <1> cmp_cmd_3:
1819
1820 00007E00 F9
                                <1>
                                         stc
                                         ; CF = 1 -> internal command word does not match
                               <1>
1822 00007E01 C3
                                <1>
                                         retn
1823
                                <1>
                                <1> loc_run_cmd_failed:
1824
1825
                                <1>
                                        ; 15/03/2016
                                         ; 15/02/2016 (TRDOS 386 = TRDOS v2.0)
1826
                                <1>
                                         ; 07/12/2009 (CMD INTR.ASM)
1827
                                <1>
1828
                                <1>
                                        ; 29/11/2009
1829
                                <1>
1830 00007E02 E863000000
                                         call restore_cdir_after_cmd_fail
                               <1>
                                <1>
1832
                                <1> loc_run_cmd_failed_cmp_al:
1833
                                <1>
                                        ; End of Restore_CDIR code (29/11/2009)
1834
                               <1>
1835 00007E07 3C01
                                               al, 1; Bad command or file name
                               <1>
                                         cmp
1836 00007E09 74CC
                               <1>
                                               loc cmd failed
                                         jе
                               <1> loc_run_dir_not_found:
1837
                                         cmp al, 3
1838 00007E0B 3C03
                               <1>
1839 00007E0D 750A
                                <1>
                                         jne short loc_run_file_notfound_msg
```

```
; Path not found (MS-DOS Error Code = 3)
1840
                                <1>
                                         mov esi, Msg_Dir_Not_Found
jmp print_msg
1841 00007EOF BE[64100100]
                                <1>
1842 00007E14 E9B4E5FFFF
                               <1>
1843
                               <1>
1844
                                <1> loc_run_file_notfound_msg:
                                       cmp al, 2; File not found jne short loc_run_file_drv_read_err
1845 00007E19 3C02
                               <1>
1846 00007E1B 750A
                               <1>
                                <1>
1847
1848
                                <1> loc_print_file_notfound_msg:
1849 00007E1D BE[7B100100]
                                         mov esi, Msg_File_Not_Found
                               <1>
1850
                               <1>
                                          ;call proc_printmsg
1851
                                <1>
                                          ;retn
1852 00007E22 E9A6E5FFFF
                               <1>
                                          jmp print_msg
1853
                                <1>
                                <1> loc run_file_drv_read_err:
1854
                                         ; Err: 17 (Read fault)
1855
                               <1>
                             <1>
1856 00007E27 3C11
                                          cmp al, 17; Drive not ready or read error
                               <1>
1857 00007E29 7404
                                                short loc_run_file_print_drv_read_err
                                         jе
1858
                               <1>
                               <1>
                                               al, 15; Drive not ready (or read error)
1859 00007E2B 3C0F
                                         cmp
                                        jne short loc_run_file_toobig
1860 00007E2D 750A
                               <1>
                                <1>
                               <1> loc_run_file_print_drv_read_err:
1862
1863 00007E2F BE[21100100] <1>
1864 00007E34 E994E5FFFF <1>
                                        mov esi, Msg_Not_Ready_Read_Err
1864 00007E34 E994E5FFFF
                               <1>
                                         jmp print_msg
1865
                               <1>
1866
                               <1> loc run file toobig:
1867 00007E39 3C08
                               <1>
                                      cmp al, 8 ; Not enough free memory to load&run file
1868 00007E3B 750A
                                <1>
                                               short loc_run_file_perm_denied
                                          jne
1869 00007E3D BE[C6100100]
                               <1>
                                               esi, Msg_Insufficient_Memory
                                         mov
1870 00007E42 E986E5FFFF
                               <1>
                                        jmp print_msg
1871
                                <1>
                               <1> loc_run_file_perm_denied:
1872
1873
                               <1> ; 29/12/2017
                               <1>
                                               al, ERR_PERM_DENIED ; 11 ; Permission denied
1874 00007E47 3C0B
                                         cmp
1875 00007E49 750A
                               <1>
                                          jne
                                               short loc_run_misc_error
1876 00007E4B BE[5A120100]
                               <1>
                                         mov esi, Msg Permission Denied
1877 00007E50 E978E5FFFF
                               <1>
                                        jmp print_msg
1878
                                <1>
                                        ; 15/03/2016
1879
                                <1>
                                <1> print misc error msg:
1880
1881
                                <1> loc run misc error:
1882
                                <1>
                                         ; AL = Error code
                                         call bytetohex
1883 00007E55 E878B4FFFF
                                <1>
1884 00007E5A 66A3[FA100100]
                                <1>
                                         mov [error_code_hex], ax
1885
                                <1>
1886 00007E60 BE[DD100100]
                                <1>
                                        mov esi, Msg_Error_Code
1887
                                <1>
                                       ;call print_msg
1888
                                <1>
                                         ;retn
1889
                                <1>
1890 00007E65 E963E5FFFF
                                <1>
                                        jmp print_msg
1891
                                <1>
1892
                                <1> restore_cdir_after_cmd_fail:
1893
                                <1>
                                       ; 15/02/2016 (TRDOS 386 = TRDOS v2.0)
1894 00007E6A 50
                                <1>
                                         push eax
1895 00007E6B 8A3D[E2600100]
                                <1>
                                         mov
                                                bh, [RUN_CDRV] ; it is set at the beginning
                                <1>
                                                           ; of the 'run' command.
1897 00007E71 3A3D[86590100]
                                               bh, [Current_Drv]
                                <1>
                                         cmp
1898 00007E77 7409
                                <1>
                                                short loc_run_restore_cdir
                                          jе
1899 00007E79 88FA
                                <1>
                                         mov
                                               dl, bh
1900 00007E7B E8C4F0FFFF
                                <1>
                                          call change_current_drive
1901 00007E80 EB19
                                <1>
                                         jmp
                                               short loc_run_err_pass_restore_cdir
1902
                                <1>
1903
                                <1> loc_run_restore_cdir:
1904 00007E82 803D[620D0100]00
                               <1> cmp byte [Restore_CDIR], 0
1905 00007E89 7610
                                <1>
                                          jna
                                               short loc_run_err_pass_restore_cdir
                                         xor bl, bl
1906 00007E8B 30DB
                               <1>
1907 00007E8D 0FB7F3
                               <1>
                                       movzx esi, bx
1908 00007E90 81C600010900
                                <1>
                                         add esi, Logical_DOSDisks
                                         call restore_current_directory
1909 00007E96 E860F1FFFF
                               <1>
1910
                                <1>
                                <1> loc_run_err_pass_restore_cdir:
1911
1912 00007E9B 58
                                <1>
                                        pop eax
1913 00007E9C C3
                                <1>
1914
                                <1>
1915
                                <1> print_directory_list:
1916
                                       ; 10/02/2016
1917
                                <1>
                                         ; 08/02/2016 (TRDOS 386 = TRDOS v2.0)
1918
                                <1>
                                         ; 06/12/2009 ('cmp_cmd_dir')
1919
                                <1>
1920 00007E9D 66C705[24620100]00- <1>
                                          mov word [AttributesMask], 0800h; ..except volume names..
1920 00007EA5 08
1921 00007EA6 A0[86590100]
                                <1>
                                          mov
                                                al, [Current_Drv]
1922 00007EAB A2[E2600100]
                                <1>
                                                [RUN_CDRV], al
                                <1> get_dfname_fchar:
1923
1924 00007EB0 AC
                                <1>
                                         lodsb
                                          cmp al, 20h
1925 00007EB1 3C20
                                <1>
1926 00007EB3 74FB
                                <1>
                                                short get dfname fchar
                                          jе
                                                loc_print_dir_call_all
1927 00007EB5 0F82A4000000
                               <1>
                                          jb
1928 00007EBB 3C2D
                                <1>
                                          cmp
1929 00007EBD 7542
                                               short loc_print_dir_call_flt
                                <1>
                                          jne
                                <1> get next attr char:
1931 00007EBF AC
                                <1>
                                         lodsb
1932 00007EC0 3C20
                                <1>
                                                al, 20h
1933 00007EC2 74FB
                                <1>
                                                short get next attr char
                                          jе
1934 00007EC4 0F820DFFFFFF
                                                loc cmd failed
                                <1>
                                          jb
1935 00007ECA 24DF
                                <1>
                                          and
                                          cmp
1936 00007ECC 3C44
                                <1>
                                               al, 'D'; directories only?
1937 00007ECE 7512
                                <1>
                                          jne
                                               short pass_only_directories
1938 00007ED0 AC
                                <1>
                                          lodsb
1939 00007ED1 3C20
                                <1>
                                          cmp
                                               al, 20h
1940 00007ED3 0F87FEFEFFF
                                <1>
                                                 loc cmd failed
                                          jа
1941 00007ED9 800D[24620100]10
                               <1>
                                          or
                                                byte [AttributesMask], 10h; ..directory..
                                                short get dfname fchar attr
1942 00007EE0 EB18
                                <1>
                                          jmp
                                <1> pass_only_directories:
```

```
cmp al, 'F' ; files only ?
1944 00007EE2 3C46
                                <1>
1945 00007EE4 0F85B0000000
                                <1>
                                          jne
                                                check_attr s
1946 00007EEA AC
                                         lodsb
                                <1>
                                          cmp al, 20h
1947 00007EEB 3C20
                               <1>
1948 00007EED 0F87E4FEFFFF
                                <1>
                                                loc cmd failed
                                         jа
1949 00007EF3 800D[25620100]10 <1>
                                         or byte [AttributesMask+1], 10h; ..except directories..
                               <1> get_dfname_fchar_attr:
                                     lodsb
1951 00007EFA AC
                               <1>
1952 00007EFB 3C20
                               <1>
                                         cmp al, 20h
1953 00007EFD 74FB
                               <1>
                                         iе
                                               short get dfname fchar attr
1954 00007EFF 725E
                               <1>
                                               short loc_print_dir_call_all
1955
                                <1>
                               <1> loc_print_dir_call_flt:
1956
                               <1> dec esi
1957 00007F01 4E
1958 00007F02 BF[26620100]
                               <1>
                                         mov
                                               edi, FindFile Drv
                                         call parse_path_name
1959 00007F07 E8AC250000
                               <1>
1960 00007F0C 7308
                               <1>
                                         jnc short loc_print_dir_change_drv_1
1961 00007F0E 3C01
                               <1>
                                         cmp
                                               al, 1
1962 00007F10 0F87ECFEFFFF
                                        ja
                               <1>
                                                loc_run_cmd_failed
1963
                               <1>
                                <1> loc_print_dir_change_drv_1:
<1> mov dl, [FindFile Drv]
1964
1965 00007F16 8A15[26620100]
                                <1> loc_print_dir_change_drv_2:
1966
                                    cmp dl, [RUN_CDRV]
1967 00007F1C 3A15[E2600100]
                                <1>
1968 00007F22 740B
                                <1>
                                         jе
                                               short loc_print_dir_change_directory
                                         call change_current_drive
1969 00007F24 E81BF0FFFF
                                <1>
                                       jc loc_run_cmd_failed
1970 00007F29 0F82D3FEFFFF
                                <1>
                                <1> loc_print_dir_change_directory:
1971
                                     cmp byte [FindFile_Directory], 20h; 0 or 20h ?
1972 00007F2F 803D[27620100]20
                               <1>
1973 00007F36 761D
                                               short pass_print_dir_change_directory
                                <1>
                                         jna
1974
                                <1>
1975 00007F38 FE05[620D0100]
                                <1>
                                         inc
                                               byte [Restore CDIR]
1976 00007F3E BE[27620100]
                                         mov
                                               esi, FindFile Directory
                               <1>
1977 00007F43 30E4
                                <1>
                                         xor ah, ah ; CD_COMMAND sign -> 0
                                         call change_current_directory
1978 00007F45 E8581F0000
                                <1>
1979 00007F4A 0F82B2FEFFFF
                               <1>
                                         jс
                                                  loc_run_cmd_failed
                                <1>
                                <1> loc_print_dir_change_prompt_dir_string:
1981
1982 00007F50 E86D1E0000
                                <1>
                                        call change_prompt_dir_string
1983
                               <1>
                               <1> pass_print_dir_change_directory:
1984
                               <1> mov esi, FindFile_Name
1985 00007F55 BE[68620100]
                                               byte [esi], 2\overline{0}h;; 0 or 20h?
1986 00007F5A 803E20
                               <1>
                                         cmp
1987 00007F5D 7706
                               <1>
                                               short loc_print_dir_call
1988
                                <1>
1989
                                <1> loc_print_dir_call_all:
                                <1> mov dword [esi], '*.*'
1990 00007F5F C7062A2E2A00
                                <1> loc_print_dir_call:
1991
                                         call print directory
1992 00007F65 E87E000000
                                <1>
                                <1>
1994 00007F6A 8A15[E2600100]
                                <1>
                                         mov dl, [RUN_CDRV] ; it is set at the beginning
1995 00007F70 3A15[86590100]
                                <1>
                                         cmp
                                               dl, [Current Drv]
1996 00007F76 7406
                                <1>
                                         jе
                                                short loc_print_dir_call_restore_cdir_retn
                                         call change_current_drive
1997 00007F78 E8C7EFFFFF
                                <1>
1998 00007F7D C3
                                <1>
                                         retn
1999
                                <1>
                                <1> loc_print_dir_call_restore_cdir_retn:
2001 00007F7E 803D[620D0100]00
                                         cmp byte [Restore_CDIR], 0
                               <1>
2002 00007F85 7610
                                <1>
                                               short pass_print_dir_call_restore_cdir_retn
2003
                               <1>
2004 00007F87 BE00010900
                               <1>
                                         mov
                                                esi, Logical_DOSDisks
2005 00007F8C 31C0
                                <1>
                                         xor
                                                eax, eax
2006 00007F8E 88D4
                               <1>
                                         mov
                                               ah, dl
2007 00007F90 01C6
                                <1>
                                               esi, eax
2008
                                <1>
2009 00007F92 E864F0FFFF
                                <1>
                                         call restore_current_directory
2010
                                <1>
2011
                                <1> pass_print_dir_call_restore_cdir_retn:
2012 00007F97 C3
                                <1>
                                         retn
2013
                                <1>
                                <1> check_attr_s_cap:
2014
2015 00007F98 24DF
                                <1> and al, ODFh
                                <1> check_attr_s:
2016
                                    cmp al, 'S'
2017 00007F9A 3C53
                                <1>
2018 00007F9C 7514
                                <1>
                                         jne
                                               short pass attr s
                                       or
lodsb
2019 00007F9E 800D[24620100]04
                               <1>
                                               byte [AttributesMask], 4 ; system
2020 00007FA5 AC
                                <1>
2021 00007FA6 3C20
                                <1>
                                         cmp al, 20h
2022 00007FA8 0F844CFFFFFF
                                <1>
                                                get_dfname_fchar_attr
                                         jе
                                         jb short loc_print_dir_call_all
2023 00007FAE 72AF
                                <1>
2024 00007FB0 24DF
                                <1>
                                        and al, ODFh
2025
                                <1> pass_attr_s:
                                               al, 'H'
2026 00007FB2 3C48
                                <1>
                                          cmp
2027 00007FB4 7514
                                <1>
                                          jne
                                                short pass_attr_h
2028 00007FB6 800D[24620100]02
                                <1>
                                         or
                                               byte [AttributesMask], 2 ; hidden
2029
                                <1> pass_attr_shr:
2030 00007FBD AC
                                <1>
                                         lodsb
2031 00007FBE 3C20
                                <1>
                                         cmp
                                                al, 20h
2032 00007FC0 0F8434FFFFFF
                                <1>
                                                  get_dfname_fchar_attr
                                          jе
                                                short loc_print_dir_call_all
2033 00007FC6 7297
                                         jb
                                <1>
2034 00007FC8 EBCE
                                <1>
                                                short check_attr_s_cap
2035
                                <1>
2036
                                <1> pass attr h:
2037 00007FCA 3C52
                                         cmp al, 'R'
                                <1>
2038 00007FCC 7509
                                <1>
                                                short pass_attr_r
                                          jne
2039 00007FCE 800D[24620100]01
                                <1>
                                         or
                                                byte [AttributesMask], 1; read only
2040 00007FD5 EBE6
                                <1>
                                               short pass_attr_shr
                                         jmp
2041
                                <1>
                                <1> pass_attr_r:
2042
                                        cmp al, 'A'
2043 00007FD7 3C41
                                <1>
                                                loc_cmd_failed
2044 00007FD9 0F85F8FDFFFF
                                <1>
                                          jne
2045 00007FDF 800D[24620100]20
                                <1>
                                               byte [AttributesMask], 20h; archive
2046 00007FE6 EBD5
                                <1>
                                         jmp
                                              short pass_attr_shr
2047
                                <1>
2048
                                <1> print_directory:
```

```
; 13/05/2016
2049
                               <1>
                                       ; 11/02/2016
2050
                               <1>
                                        ; 10/02/2016
2051
                               <1>
2052
                               <1>
                                       ; 08/02/2016 (TRDOS 386 = TRDOS v2.0)
2053
                               <1>
                                       ; 30/10/2010 ('proc_print_directory')
2054
                               <1>
                                        ; 19/09/2009
2055
                               <1>
                                        ; 2005
                                        ; INPUT ->
2056
                               <1>
2057
                               <1>
                                              ESI = Asciiz File/Dir Name Address
2058
                               <1>
2059 00007FE8 56
                               <1>
                                        push esi
2060
                               <1>
2061 00007FE9 29C0
                               <1>
                                        sub
                                               eax, eax
2062
                               <1>
                                              word [Dir_Count], ax ; 0
2063 00007FEB 66A3[B0620100]
                               <1>
                                        mov
2064 00007FF1 66A3[AE620100]
                                              word [File_Count], ax ; 0
                               <1>
                                        mov
2065 00007FF7 A3[B2620100]
                               <1>
                                        mov
                                              dword [Total FSize], eax ; 0
                               <1>
2066
2067 00007FFC E8E2E3FFFF
                               <1>
                                        call
                                              clear_screen
2068
                               <1>
2069 00008001 31C9
                               <1>
                                        xor
                                              ecx, ecx
                                              ch, [Current_Drv] ; DirBuff_Drv - 'A'
2070 00008003 8A2D[86590100]
                              <1>
                                        mov
                                                al, [Current_Dir_Drv]
2071 00008009 A0[87590100]
                                        mov
2071 00008009 A0[87590100]
2072 0000800E A2[1F0F0100]
2073 00008013 BE00010900
                              <1>
                                              [Dir_Drive_Name], al
2073 00008013 BE00010900
                                              esi, Logical_DOSDisks
                               <1>
                                        mov
2074 00008018 01CE
                               <1>
                                        add
                                              esi, ecx
                               <1>
2076 0000801A E858F9FFFF
                                        call move_volume_name_and_serial_no
                              <1>
                                       jnc
2077 0000801F 730C
                               <1>
                                              short print_dir_strlen_check
2078
                               <1>
2079 00008021 5E
                                      pop
                              <1>
                                               esi
2080 00008022 8A3D[EE580100]
                               <1>
                                        mov
                                               bh, [ptty] ; [ACTIVE_PAGE]
2081
                               <1>
                                        ;call beeper
2082
                               <1>
                                        ;retn
2083 00008028 E96E9DFFFF
                                        jmp
                               <1>
                                              beeper ; beep ! and return
2084
                               <1>
                               <1> print dir strlen check:
                                        mov esi, Current_Dir_Root
2086 0000802D BE[89590100]
                               <1>
2087 00008032 BF[BC0F0100]
                               <1>
                                             edi, Dir_Str_Root
                                        mov
2088
                               <1>
;xor ecx, ecx
                                        mov cl, [Current_Dir_StrLen]
                    inc cl
                                    cmp cl, 64
jna short pass_print_dir_strlen_shorting
inc esi
add esi, ecx
sub esi, 64
inc edi
2092 0000803F 80F940
                       2093 00008042 760D
2094 00008044 46
2095 00008045 01CE
2096 00008047 83EE40
2097 0000804A 47
                                        inc
                                              edi
                                    mov eax, '...'
2098 0000804B B82E2E2E20
2099 00008050 AB
                                       stosd
2100
                               <1>
2101
                               <1> pass_print_dir_strlen_shorting:
2102 00008051 F3A4
                              <1>
                                       rep movsb
2103
                               <1>
2104 00008053 BE[120F0100]
                               <1>
                                        mov
                                              esi, Dir_Drive_Str
2105 00008058 E870E3FFFF
                              <1>
                                        call print msg
2106
                               <1>
2107 0000805D BE[710F0100]
                               <1>
                                              esi, Vol_Serial_Header
                                        mov
                                     call print_msg
2108 00008062 E866E3FFFF
                               <1>
2109
                               <1>
2110 00008067 BE[B10F0100]
                               <1>
                                        mov
                                              esi, Dir Str Header
2111 0000806C E85CE3FFFF
                               <1>
                                        call print msg
2112
                               <1>
2113 00008071 BE[FD190100]
                               <1>
                                        mov
                                              esi, next2line
2114 00008076 E852E3FFFF
                               <1>
                                        call print_msg
2115
                               <1>
2116
                               <1> loc_print_dir_first_file:
                                    mov byte [PrintDir_RowCounter], 16
2117 0000807B C605[C5620100]10
                              <1>
                                        mov ax, [AttributesMask]
2118 00008082 66A1[24620100]
                               <1>
2119 00008088 5E
                                      pop esi
                               <1>
2120
                               <1>
2121 00008089 E859020000
                                       call find_first_file
    jc loc_dir_ok
                               <1>
2122 0000808E 0F826F010000
                               <1>
2123
                               <1>
                               <1> loc_dfname_use_this:
2124
2125
                               <1> ; bl = File Attributes (bh = Long Name Entry Length)
2126 00008094 F6C310
                               <1>
                                        test bl, 10h ; Is it a directory?
                                       jz short loc not dir
2127 00008097 741B
                               <1>
2128
                               <1>
2129 00008099 66FF05[B0620100] <1>
                                        inc word [Dir_Count]
                                               edx, esi ; FindFile_DirEntry address
2130 000080A0 89F2
                               <1>
                                        mov
2131 000080A2 BE[00110100]
                                        mov esi, Type_Dir; '<DIR>
                               <1>
2132 000080A7 BF[17110100]
                               <1>
                                        mov edi, Dir_Or_FileSize
                                        ; move 10 bytes
2133
                               <1>
2134 000080AC A5
                               <1>
                                        movsd
2135 000080AD A5
                              <1>
                                        movsd
2136 000080AE 66A5
                              <1>
                                        movsw
2137 000080B0 89D6
                               <1>
                                              esi, edx
                                        mov
2138 000080B2 EB36
                               <1>
                                               short loc_dir_attribute
                                        jmp
2139
                               <1>
                               <1> loc_not_dir:
2140
                                      inc word [File_Count]
2141 000080B4 66FF05[AE620100]
                               <1>
2142 000080BB 0105[B2620100]
                               <1>
                                         add
                                              [Total_FSize], eax
2143
                               <1>
                                               ecx, 10 ; 32 bit divisor
2144 000080C1 B90A000000
                               <1>
                                        mov
2145 000080C6 89CF
                               <1>
                                        mov
                                              edi, ecx
2146 000080C8 81C7[17110100]
                                              edi, Dir_Or_FileSize
                               <1>
                                        add
2147
                               <1> loc_dir_rdivide:
                                       sub
2148 000080CE 29D2
                               <1>
                                              edx, edx
2149 000080D0 F7F1
                                        div
                                               ecx ; remainder in dl (< 10)
                               <1>
                              <1>
2150 000080D2 80C230
                                        add
                                               dl, '0' ; to make visible (ascii)
                              <1>
2151 000080D5 4F
                                        dec
                                              edi
2152 000080D6 8817
                              <1>
                                              [edi], dl
                                        mov
2153 000080D8 21C0
                               <1>
                                        and
                                              eax, eax
```

```
2154 000080DA 75F2
                               <1>
                                         jnz short loc_dir_rdivide
2155
                                <1>
2156
                               <1> loc_dir_fill_space:
2157 000080DC 81FF[17110100]
                            <1>
                                              edi, Dir Or FileSize
2158 000080E2 7606
                               <1>
                                                 short loc_dir_attribute
                                         ina
2159 000080E4 4F
                               <1>
                                         dec
                                                 edi
2160 000080E5 C60720
                                                byte [edi], 20h
                               <1>
                                         mov
2161 000080E8 EBF2
                                                short loc_dir_fill_space
                               <1>
                                       jmp
2162
                               <1>
                               <1> loc_dir_attribute:
2163
2164 000080EA C705[22110100]2020- <1>
                                       mov dword [File Attribute], 20202020h
2164 000080F2 2020
                               <1>
2165
                               <1>
                      <1>
2166 000080F4 80FB20
                                         cmp
                                              bl, 20h ; Is it an archive file?
2167 000080F7 7207
                               <1>
                                               short loc_dir_pass_arch
                                         jb
2168 000080F9 C605[25110100]41 <1>
                                              byte [File Attribute+3], 'A'
                                         mov
                               <1>
2170
                               <1> loc_dir_pass_arch:
                                      and bl, 7
2171 00008100 80E307
                               <1>
2172 00008103 7428
                               <1>
                                         jz
                                               short loc_dir_file_name
2173 00008105 88DF
                               <1>
                                         mov
                                               bh, bl
2174 00008107 80E303
                               <1>
                                        and
                                               bl, 3
                                    and b1, 3
cmp bh, b1
jna short loc_dir_pass_s
mov byte [File_Attribute], 'S'
2175 0000810A 38DF
                               <1>
2176 0000810C 7607
                               <1>
2177 0000810E C605[22110100]53 <1>
2178
                               <1>
2179
                               <1> loc dir pass s:
2180 00008115 80E302
                                    and b1,2
                               <1>
2181 00008118 7407
                               <1>
                                         jz
                                                short loc_dir_pass_h
2182 0000811A C605[23110100]48 <1> mov
                                               byte [File_Attribute+1], 'H'
2183
                               <1> loc_dir_pass_h:
2184 00008121 80E701
                               <1> and bh,1
                                                short loc dir file name
2185 00008124 7407
                               <1>
                                         jz
2186 00008126 C605[24110100]52 <1> mov
                                                byte [File_Attribute+2], 'R'
                               <1> loc_dir_file_name:
2187
                               2188
2189
                                    mov ebx, [esi+16h]
mov ecx, esi ; FindFile_DirEntry address
mov edi, File_Name
; move 8 bytes
2190 0000812D 8B5E16
                                        mov
                               <1>
2191 00008130 89F1
                               <1>
2192 00008132 BF[0A110100]
                               <1>
2193
                               <1>
2194 00008137 A5
                               <1>
                                        movsd
2195 00008138 A5
                               <1>
                                        movsd
                                         mov byte [edi], 20h
2196 00008139 C60720
                               <1>
2197 0000813C 47
                               <1>
                                         inc edi
2198
                               <1>
                                         ; move 3 bytes
2199 0000813D 66A5
                               <1>
                                        movsw
2200 0000813F A4
                               <1>
                                         movsb
                                         mov esi, ecx
2201 00008140 89CE
                               <1>
2202
                               <1>
2203
                               <1> Dir_Time_start:
2204
                               <1> ; mov ax, dx
                                                           ; Time
2205 00008142 6689D8
                               <1>
                                         mov ax, bx
                              <1>
<1>
2206 00008145 66C1E805
                                        shr ax, 5
                                                          ; shift right 5 times
2207 00008149 6683E03F
                                        and ax, 0000111111b ; Minute Mask
                                                           ; Q([AL]/10)->AH
2208 0000814D D40A
                               <1>
                                        aam
                                                           ; R([AL]/10)->AL
                               <1>
2210
                               <1>
                                                           ; [AL]+[AH]= Minute as BCD
2211 0000814F 660D3030
                               <1>
                                               ax, '00'
                                                           ; Convert to ASCII
2212 00008153 86E0
                               <1>
                                        xchg
                                              ah, al
2213 00008155 66A3[35110100]
                               <1>
                                               [File_Minute], ax
2214
                               <1>
2215
                               <1>
                                         ;mov al, dh
2216 0000815B 88F8
                               <1>
                                         mov al, bh
                                               al, 3
                                                           ; shift right 3 times
2217 0000815D C0E803
                               <1>
                                         shr
                                                           ; [AL] + [AH] = Hours as BCD
2218 00008160 D40A
                               <1>
                                         aam
2219 00008162 660D3030
                                               ax, '00'
                               <1>
                                         or
2220 00008166 86E0
                               <1>
                                         xchg ah, al
2221 00008168 66A3[32110100]
                               <1>
                                               [File_Hour], ax
                                         mov
2222
                               <1>
2223 0000816E C1EB10
                                <1>
                                         shr
                                               ebx, 16
                                                                ; BX = Date
2224
                                <1>
2225
                                <1> Dir_Date_start:
                                        mov ax, bx
2226 00008171 6689D8
                               <1>
                                                          ; Date
2227 00008174 6683E01F
                               <1>
                                         and
                                               ax, 00011111b; Day Mask
                                                           ; Q([AL]/10)->AH
2228 00008178 D40A
                               <1>
                                         aam
2229
                               <1>
                                                           ; R([AL]/10)->AL
2230
                               <1>
                                                           ; [AL]+[AH]= Day as BCD
                                              ax, '00'
                                                          ; Convert to ASCII
2231 0000817A 660D3030
                               <1>
                                         or
2232 0000817E 86C4
                               <1>
                                         xchg al, ah
2233
                               <1>
2234 00008180 66A3[27110100]
                                               [File_Day], ax
                                <1>
                                         mov
                               <1>
2236 00008186 6689D8
                               <1>
2237 00008189 66C1E805
                               <1>
                                         shr
                                               ax, 5
                                                           ; shift right 5 times
                               <1>
2238 0000818D 6683E00F
                                         and
                                               ax, 00001111b; Month Mask
2239 00008191 D40A
                              <1>
                                         aam
                                               ax, '00'
2240 00008193 660D3030
                              <1>
                                        or
2241 00008197 86E0
                               <1>
                                         xchg
                                              ah, al
2242 00008199 66A3[2A110100]
                              <1>
                                        mov
                                               [File_Month], ax
2243
                               <1>
2244 0000819F 6689D8
                               <1>
                                         mov
                                               ax, bx
2245 000081A2 66C1E809
                               <1>
                                         shr
                                               ax, 9
                                               ax, 011111111b; Result = Year - 1980
2246 000081A6 6683E07F
                               <1>
                                               ax, 1980
2247 000081AA 6605BC07
                               <1>
                                         add
2248
                               <1>
2249 000081AE B10A
                               <1>
                                               cl, 10
                                         mov
2250 000081B0 F6F1
                                         div
                                              cl
ah, '0'
                               <1>
                                                         ; Q \rightarrow AL, R \rightarrow AH
2251 000081B2 80CC30
                               <1>
                                         or
2252 000081B5 8825[30110100]
                                               [File_Year+3], ah
                              <1>
                                         mov
2253 000081BB D40A
                               <1>
                                         aam
2254 000081BD 86E0
                               <1>
                                         xchg
                                              ah, al ; Convert to ASCII
2255 000081BF 80CC30
                               <1>
                                         or
2256 000081C2 8825[2F110100]
                               <1>
                                               [File_Year+2], ah
2257 000081C8 D40A
                               <1>
                                         aam
```

```
<1>
                                                            xchg al, ah
2259 000081CC 660D3030
2258 000081CA 86C4
                                                               or ax, '00'
mov [File_Year], ax
                                                <1>
                                            <1>
2260 000081D0 66A3[2D110100]
2261
                                                <1>
2262
                                                 <1> loc_show_line:
2263 000081D6 56
                                                <1> push esi
esi, File Name
                                                               mov
                                                         call print_msg
mov esi, nextl
call print_msg
pop esi
                                                                        esi, nextline
2268 000081EB 5E
                                                <1>
                                                <1>
2269
2270 000081EC FE0D[C5620100]
                                                <1>
                                                            dec byte [PrintDir_RowCounter]
                                                         jz pause_dir_scroll
2271 000081F2 0F84D4000000
                                                <1>
2272
                                                 <1>
                                                <1> loc_next_entry:
2273
                                                2274 000081F8 E899010000
2275 000081FD 0F8391FEFFFF
                                                <1>
                                                              jnc loc_dfname_use_this
2276
                                                <1>
2277
<1> loc dir ok:
                                                                           ecx, 10
2279 00008208 66A1[B0620100]
                                                <1>
                                                                         ax, [Dir Count]
                                                               mov
                                        2280 0000820E BF[4B110100]
                                                                         edi, Decimal_Dir_Count
2281 00008213 6639C8
                                                                         ax, cx ; 10
                                                                         short pass_ddc
2282 00008216 7216
2283 00008218 47
                                                                         edi
2284 00008219 6683F864
                                                                         ax, 100
2285 0000821D 720F
                                                                          short pass_ddc
2286 0000821F 47
                                                                         edi
2287 00008220 663DE803
                                                                         ax, 1000
2288 00008224 7208
                                                                          short pass_ddc
2289 00008226 47
                                                                         edi
2290 00008227 663D1027
                                                                         ax, 10000
                                            <1>
2291 0000822B 7201
                                                                jb
                                                                          short pass ddc
                                               <1> inc 
<1> pass_ddc:
2292 0000822D 47
2293
                                            <1> mov [6 <1> loc_ddc_rediv:
2294 0000822E 886F01
                                                                           [edi+1], ch ; 0
2295
                                       <1> xor edx, edx
2296 00008231 31D2
                                                                        cx ; 10
dl, '0'
2297 00008233 66F7F1
2298 00008236 80C230
2299 00008239 8817
                                                                           [edi], dl
2300 0000823B 4F
                                                                          edi
2301 0000823C 6609C0
                                                                          ax, ax
2302 0000823F 75F0
                                                                         short loc_ddc_rediv
2303
2304 00008241 66A1[AE620100]
                                                <1>
                                                            mov
                                                                            ax, [File Count]
                                         <1> mov
<1> mov
<1> cmp
<1> inc
<1> cmp
<1> inc
<1> cmp
<1> inc
<1> cmp
<1> jb
<1> inc
<1> cmp
<1> jb
<1> inc
<1> cmp
<1> jb
<1 inc
<1 cmp
<1 jb
<1 inc
<1 cmp
<1 jb
<1 inc
<1 cmp
<1 inc
<1 in
2305 00008247 BF[3A110100]
                                                                            edi, Decimal_File_Count
2306 0000824C 6639C8
                                                                            ax, cx ; 10
2307 0000824F 7216
                                                                            short pass_dfc
2308 00008251 47
                                                                            edi
2309 00008252 6683F864
                                                                            ax, 100
2310 00008256 720F
                                                                            short pass_dfc
2311 00008258 47
                                                                            edi
2312 00008259 663DE803
                                                                            ax, 1000
2313 0000825D 7208
                                                                            short pass_dfc
2314 0000825F 47
                                                                            edi
2315 00008260 663D1027
                                                                            ax, 10000
2316 00008264 7201
                                                                            short pass_dfc
2317 00008266 47
                                                <1>
                                                                inc
                                                                            edi
2318
                                                <1> pass_dfc:
2319
                                                 <1>
                                                               ;mov
                                                                           cx, 10
2320 00008267 886F01
                                                <1>
                                                                           [edi+1], ch ; 00
                                                                mov
2321
                                                <1> loc_dfc_rediv:
2322
                                                <1> ;xor dx, dx
2323 0000826A 30D2
                                                <1>
                                                                xor
                                                                         dl, dl
                                               <1> xor
<1> div
<1> add
<1> mov
<1> dec
<1> or
<1> jnz
                                            <1>
<1>
<1>
2324 0000826C 66F7F1
                                                                        CX
                                                                        dl, '0'
2325 0000826F 80C230
2326 00008272 8817
                                                                         [edi], dl
2327 00008274 4F
                                                                        edi
                                                                          ax, ax
2328 00008275 6609C0
2329 00008278 75F0
                                                                         short loc dfc rediv
                                                 <1>
2330
2331 0000827A BF[C4620100]
                                                <1>
                                                                mov
                                                                           edi, TFS Dec End
                                                               ;mov byte [edi], 0
2332
                                                 <1>
2333 0000827F A1[B2620100]
                                                 <1>
                                                               mov
                                                                           eax, [Total_FSize]
2334
                                                 <1>
                                                               ;mov
                                                                          ecx, 10
                                                 <1> rediv_tfs_hex:
2335
2336
                                                 <1>
                                                               ;sub edx, edx
2337 00008284 28D2
                                                 <1>
                                                                sub dl. dl
2338 00008286 F7F1
                                                                div
                                                 <1>
                                                                         ecx
2339 00008288 80C230
                                                  <1>
                                                                add
                                                                          dl, '0'
2340 0000828B 4F
                                                 <1>
                                                                dec
                                                                            edi
2341 0000828C 8817
                                                 <1>
                                                                            [edi], dl
2342 0000828E 21C0
                                                 <1>
                                                                and
                                                                          eax, eax
2343 00008290 75F2
                                                 <1>
                                                               jnz
                                                                          short rediv_tfs_hex
                                                 <1>
                                                             mov
2345 00008292 893D[B6620100]
                                                <1>
                                                                          [TFS_Dec_Begin], edi
                                                            mov esi, Decir
call print_msg
2346 00008298 BE[38110100]
                                                 <1>
                                                                          esi, Decimal_File_Count_Header
2347 0000829D E82BE1FFFF
                                                 <1>
2348 000082A2 BE[40110100]
                                                         mov
                                                <1>
                                                                          esi, str_files
                                                               call print_msg
2349 000082A7 E821E1FFFF
                                                 <1>
2350 000082AC BE[51110100]
                                                <1>
                                                                          esi, str_dirs
                                                               mov
2351 000082B1 E817E1FFFF
                                                <1>
                                                            call
                                                                         print_msg
                                                            mov
                                                                         esi, [TFS_Dec_Begin]
2352 000082B6 8B35[B6620100]
                                                <1>
2353 000082BC E80CE1FFFF
                                                <1>
                                                               call
                                                                        print_msg
2354 000082C1 BE[62110100]
                                                                         esi, str bytes
                                                <1>
                                                               mov
2355 000082C6 E802E1FFFF
                                                 <1>
                                                               call print msg
2356
                                                 <1>
2357 000082CB C3
                                                 <1>
                                                               retn
2358
                                                 <1>
2359
                                                 <1> pause_dir_scroll:
2360 000082CC 28E4
                                                <1>
                                                                sub ah, ah
2361 000082CE E84C89FFFF
                                                 <1>
                                                                call int16h
2362 000082D3 3C1B
                                                 <1>
                                                                cmp al, 1Bh
```

```
2363 000082D5 0F8428FFFFFF
                                <1>
                                          jе
                                                    loc_dir_ok
2364 000082DB C605[C5620100]10
                                <1>
                                          mov byte [PrintDir RowCounter], 16; Reset counter
2365 000082E2 E911FFFFFF
                                          jmp loc_next_entry
                                 <1>
2366
                                 <1>
2367
                                 <1> find_first_file:
                                       ; 11/02/2016
2368
                                 <1>
2369
                                          ; 10/02/2016
                                        ; 08/02/2016 (TRDOS 386 = TRDOS v2.0)
2370
                                 <1>
                                         ; 09/10/2011
; 17/09/2009
2371
                                 <1>
2372
                                 <1>
                                        ; 2005
2373
                                 <1>
                                         ; INPUT ->
2374
                                 <1>
2375
                                 <1>
                                                ESI = ASCIIZ File/Dir Name Address (in Current Directory)
                                                 AL = Attributes AND mask (The AND result must be equal to AL)
2376
                                 <1>
                                                      bit 0 = Read Only
bir 1 = Hidden
2377
                                 <1>
2378
                                 <1>
                                                      bit 2 = System
2379
                                 <1>
                                                      bit 3 = Volume Label
2380
                                 <1>
2381
                                 <1>
                                                      bit 4 = Directory
                                                      bit 5 = Archive
2382
                                 <1>
2383
                                 <1>
                                                      bit 6 = Reserved, must be 0
2384
                                 <1>
                                                      bit 7 = Reserved, must be 0
                                                 AH = Attributes Negative AND mask (The AND result must be ZERO)
2385
                                 <1>
                                          ;
2386
                                 <1>
                                          ; OUTPUT ->
2387
                                 <1>
                                               CF = 1 -> Error, Error Code in EAX (AL)
2388
                                 <1>
                                                 CF = 0 ->
2389
                                 <1>
2390
                                 <1>
                                                  ESI = Directory Entry (FindFile_DirEntry) Location
2391
                                 <1>
                                                      EDI = Directory Buffer Directory Entry Location
                                                     EAX = File Size
2392
                                 <1>
2393
                                 <1>
                                                      BL = Attributes of The File/Directory
2394
                                 <1>
                                                       BH = Long Name Yes/No Status (>0 is YES)
                                                        \mathrm{DX} \,>\, 0 : Ambiguous filename chars are used
2395
                                 <1>
2396
                                 <1>
2397
                                 <1>
                                          ; (EAX, EBX, ECX, EDX, ESI, EDI will be changed)
2398
                                 <1>
2399 000082E7 66A3[76620100]
                                <1>
                                                [FindFile AttributesMask], ax
2400 000082ED BF[78620100]
                                 <1>
                                          mov edi, FindFile_DirEntry; TR-DOS Fullfilename formatted buffer
2401 000082F2 31C0
                                 <1>
                                           xor
                                                 eax, eax
2402 000082F4 B90B000000
                                          mov ecx, 11
                                <1>
2403 000082F9 F3AB
                                <1>
                                        rep stosd ; 44 bytes
                                        ;stosw ; +2 bytes
2404
                                 <1>
2405
                                <1>
2406 000082FB BF[68620100]
2407 00008300 39FE
                                          mov edi, FindFile Name ; FFF structure, offset 66
                                <1>
2407 00008300 39FE
                                <1>
                                          cmp esi, edi
                                                 short loc fff mfn ok
2408 00008302 7408
                                <1>
                                           jе
2409 00008304 89FA
                                <1>
                                        mov edx, edi
2410
                                <1>
                                          ; move 13 bytes
2411 00008306 A5
                                <1>
                                          movsd
2412 00008307 A5
                                <1>
                                          movsd
                              <1>
2413 00008308 A5
                                          movsd
2414 00008309 AA
                                <1>
                                          stosb
2415 0000830A 89D6
                                <1>
                                          mov esi, edx
2416
                                <1> loc_fff_mfn_ok:
2417 0000830C BF[17620100]
2418 00008311 E8D7200000
                                2419 00008316 89FE
                                <1>
                                        mov esi, edi ; offset Dir_Entry_Name
2420
                                <1>
2421 00008318 66A1[76620100]
                                <1>
                                      mov ax, [Find
;xor ecx, ecx
                                                ax, [FindFile_AttributesMask]
2422
                                <1>
2423 0000831E 30C9
                                <1>
                                        xor cl, cl
                                      call locate_current_dir_file
jc short loc_fff_retn
; EDI = Directory Entry
2424 00008320 E8D11D0000
                                <1>
2425 00008325 726E
                                <1>
2426
                                <1>
2427
                                 <1>
                                         ; EBX = Directory Buffer Entry Index/Number
2428
                                 <1>
2429
                                <1> loc_fff_fnf_ln_check:
2430 00008327 30ED
                                <1>
                                      xor ch, ch
2431 00008329 80F60F
                                <1>
                                          xor
                                                dh, 0Fh
2432 0000832C 7408
                                <1>
                                          jz
                                                short loc_fff_longname_yes
2433 0000832E 882D[75620100]
                               <1>
                                        mov [FindFile_LongNameYes], ch ; 0
2434 00008334 EB0C
                                 <1>
                                        jmp short loc_fff_longname_no
2435
                                 <1>
                                 <1> loc fff longname yes:
2436
                                      ;inc byte [FindFile_LongNameYes]
2437
                                 <1>
2438 00008336 8A0D[82610100]
                                 <1>
                                          mov
                                                cl, [LFN_EntryLength]
2439 0000833C 880D[75620100]
                                 <1>
                                                [FindFile_LongNameEntryLength], cl ; FindFile_LongNameYes
                                          mov
2440
                                 <1>
2441
                                 <1> loc_fff_longname_no:
                                 <1> ;mov bx, [DirBuff_CurrentEntry]
2442
2443 00008342 66891D[A0620100]
                                 <1>
                                          mov
                                                [FindFile_DirEntryNumber], bx
2444 00008349 6689C2
                                 <1>
                                           mov
                                                 dx, ax ; Ambiguous Filename chars used sign > 0
                                 <1>
2446 0000834C A0[86590100]
                                 <1>
                                                 al, [Current_Drv]
                                                 [FindFile_Drv], al
2447 00008351 A2[26620100]
                                 <1>
                                           mov
2448
                                 <1>
2449 00008356 A1[80590100]
                                 <1>
                                                 eax, [Current Dir FCluster]
                                           mov
2450 0000835B A3[98620100]
                                                 [FindFile_DirFirstCluster], eax
                                 <1>
                                           mov
                                 <1>
2451
2452 00008360 A1[B1600100]
                                                 eax, [DirBuff Cluster]
                                 <1>
                                           mov
2453 00008365 A3[9C620100]
                                 <1>
                                                 [FindFile_DirCluster], eax
                                           mov
2454
                                 <1>
2455 0000836A 66FF05[A2620100]
                                                 word [FindFile_MatchCounter]
                                 <1>
                                           inc
                                 <1>
2456
2457 00008371 89FB
                                 <1>
                                                 ebx, edi
                                           mov
2458 00008373 89FE
                                                 esi, edi
                                 <1>
                                           mov
2459 00008375 BF[78620100]
                                 <1>
                                                 edi, FindFile_DirEntry
                                           mov
2460 0000837A 89F8
                                 <1>
                                           mov
                                                 eax, edi
2461 0000837C B108
                                 <1>
                                                 cl, 8
2462 0000837E F3A5
                                 <1>
                                           rep
                                                 movsd
                                                 esi, eax
2463 00008380 89C6
                                 <1>
                                           mov
2464 00008382 89DF
                                 <1>
                                                 edi, ebx
                                          mov
2465
                                 <1>
2466 00008384 A1[94620100]
                                 <1>
                                                 eax, [FindFile_DirEntry+28] ; File Size
                                 <1>
2467
```

```
2468 00008389 8A1D[83620100]
                                <1>
                                                bl, [FindFile_DirEntry+11] ; File Attributes
                                          mov
2469 0000838F 8A3D[75620100]
                                <1>
                                          mov
                                                bh, [FindFile LongNameYes]
2470
                                <1>
2471
                                <1>
                                          ;mov cx, [DirBuff_EntryCounter]
2472
                                 <1>
                                          ;mov [FindFile DirEntryNumber], cx
2473
                                 <1>
                                          ;mov cx, [FindFile DirEntryNumber]
2474
                                          ; ecx = 0
                                 <1>
2475
                                <1>
2476
                                <1> loc_fff_retn:
2477 00008395 C3
                                <1>
                                          retn
2478
                                <1>
2479
                                 <1> find next file:
                                      -; 1<del>5</del>/10/2016
2480
                                <1>
2481
                                <1>
                                         ; 10/02/2016
                                        ; 08/02/2016 (TRDOS 386 = TRDOS v2.0)
2482
                                 <1>
2483
                                 <1>
                                         ; 06/02/2011
                                        ; 17/09/2009
2484
                                 <1>
                                        ; 2005
                                 <1>
2485
                                        ; INPUT ->
2486
                                 <1>
                                               NONE, Find First File Parameters
2487
                                 <1>
                                         ;
                                         ; OUTPUT ->
2488
                                 <1>
                                                CF = 1 -> Error, Error Code in EAX (AL)
2489
                                 <1>
                                                CF = 0 \rightarrow
2490
                                 <1>
                                          ;
2491
                                 <1>
                                                    ESI = Directory Entry (FindFile_DirEntry) Location
2492
                                 <1>
                                                    EDI = Directory Buffer Directory Entry Location
2493
                                 <1>
                                                    EAX = File Size
2494
                                 <1>
                                                     BL = Attributes of The File/Directory
2495
                                 <1>
                                                      BH = Long Name Yes/No Status (>0 is YES)
2496
                                 <1>
                                                       DX > 0 : Ambiguous filename chars are used
2497
                                 <1>
2498
                                 <1>
                                          ; (EAX, EBX, ECX, EDX, ESI, EDI will be changed)
2499
                                 <1>
2500 00008396 66833D[A2620100]00 <1>
                                               word [FindFile MatchCounter], 0
                                          cmp
2501 0000839E 7707
                                <1>
                                                short loc start search next file
2502
                                <1>
2503
                                <1> loc_fnf_stc_retn:
2504 000083A0 F9
                                <1> stc
                                <1> loc_fnf_ax12h_retn:
2505
2506 000083A1 B80C000000
                                         mov eax, 12; No More files
                                 <1>
                                <1> ;loc_fnf_retn:
2507
2508 000083A6 C3
                                <1>
                                         retn
2509
                                 <1>
                                <1> loc_start_search_next_file:
2510
2511 000083A7 668B1D[A0620100]
                                <1> mov bx, [FindFile_DirEntryNumber]
2512 000083AE 6643
                                <1>
                                          inc
                                                bx
2513 000083B0 663B1D[AF600100]
                                                bx, [DirBuff LastEntry]
                                <1>
                                          cmp
2514 000083B7 7719
                                <1>
                                                short loc_cont_search_next_file
                                          jа
2515
                                <1>
2516
                                <1> loc_fnf_search:
2517 000083B9 BE[17620100]
                               <1> mov esi, Dir_Entry_Name
                                          mov ax, [FindFile_AttributesMask]
2518 000083BE 66A1[76620100] <1>
2519 000083C4 6631C9
                                <1>
                                          xor
                                                CX, CX
2520 000083C7 E82E1E0000
                                         call find directory entry
                                <1>
2521 000083CC 0F8355FFFFFF
                                <1>
                                         jnc loc_fff_fnf_ln_check
2522
                                <1>
2523
                                <1> loc_cont_search_next_file:
2524 000083D2 31DB
                                <1> xor ebx, ebx
                                                bh, [Current_Drv]
2525 000083D4 8A3D[86590100]
                                <1>
                                          mov
2526 000083DA BE00010900
                                <1>
                                                esi, Logical_DOSDisks
                                          mov
                                       add
2527 000083DF 01DE
                                <1>
                                                esi, ebx
2528
                                <1>
                                2529 000083E1 803D[84590100]00
                                <1>
                                         cmp
                                                byte [Current_Dir_Level], 0
2530 000083E8 7608
                                                short loc_fnf_check_FAT_type
                                          jna
2531 000083EA 807E0301
                                                byte [esi+LD_FATType], 1
                                                short loc_fnf_ax12h_retn
2532 000083EE 72B1
                                <1>
                                          jb
2533 000083F0 EB06
                                <1>
                                          jmp
                                                short loc_fnf_check_next_cluster
                                <1>
2535
                                <1> loc_fnf_check_FAT_type:
2536 000083F2 807E0303
                                <1>
                                         cmp byte [esi+LD_FATType], 3
2537 000083F6 72A9
                                <1>
                                                short loc_fnf_ax12h_retn
2538
                                <1>
2539
                                <1> loc fnf check next cluster:
2540 000083F8 A1[B1600100]
                                <1> mov eax, [DirBuff_Cluster]
2541 000083FD E8CA370000
                                <1>
                                         call get_next_cluster
2542 00008402 7306
                                <1>
                                          jnc short loc_fnf_load_next_dir_cluster
2543 00008404 09C0
                                <1>
                                          or
                                                eax, eax
                                         jz
2544 00008406 7498
                                <1>
                                                short loc fnf stc retn
2545
                                <1>
                                          ;mov eax, 17 ;Drive not ready or read error
2546 00008408 F5
                                <1>
                                          cmc
2547
                                <1> loc_fnf_retn:
2548 00008409 C3
                                 <1>
                                          retn
                                 <1>
                                 <1> loc_fnf_load_next_dir_cluster:
2551 0000840A E8A3390000
                                          call load FAT sub directory
2552 0000840F 72F8
                                <1>
                                          iс
                                                short loc_fnf_retn
2553 00008411 6631DB
                                <1>
                                          xor
                                                bx, bx
                                               [FindFile DirEntryNumber], bx
2554 00008414 66891D[A0620100]
                               <1>
                                          mov
2555 0000841B EB9C
                                <1>
                                                short loc_fnf_search
                                        jmp
2556
                                <1>
                                <1> get_and_print_longname:
2557
2558
                                 <1>
                                         ; 16/10/2016
2559
                                          ; 13/02/2016 (TRDOS 386 = TRDOS v2.0)
                                 <1>
2560
                                 <1>
                                          ; 24/01/2010
                                          ; 17/10/2009 (CMD INTR.ASM, 'cmp cmd longname')
2561
                                <1>
2562
                                <1> get_longname_fchar:
2563 0000841D 803E20
                                          cmp byte [esi], 20h
                                <1>
2564 00008420 7701
                                <1>
                                                short loc find longname
                                          jа
2565
                                <1>
                                          ;jb
                                               short loc_longname_retn
2566
                                <1>
                                          ;inc esi
                                          ;je short get_longname_fchar
2567
                                <1>
2568
                                <1> ;loc_longname_retn:
2569 00008422 C3
                                <1>
                                          retn
2570
                                <1> loc_find_longname:
2571 00008423 E839210000
                                <1>
                                          call find_longname
2572 00008428 7328
                                <1>
                                          jnc short loc_print_longname
```

```
2573
                                <1>
2574 0000842A 08C0
                                <1>
                                          or
                                                al, al
2575 0000842C 741A
                               <1>
                                          jz
                                                short loc_longname_not_found
2576
                               <1>
2577
                                <1>
                                         ; 16/10/2016 (15h -> 15, 17)
2578 0000842E 3C0F
                                <1>
                                          cmp al, 15
2579 00008430 0F84B6F7FFFF
                               <1>
                                                cd_drive_not_ready ; drive not ready
                                          jе
                                               ; or al, 17
2580
                                <1>
2581 00008436 3C11
                                <1>
                                                               ; read error
2582 00008438 0F84AEF7FFFF
                               <1>
                                          je
                                                cd_drive_not_ready
2583
                                <1>
2584
                                <1> loc_ln_file_dir_not_found:
2585 0000843E BE[8D100100]
                                <1>
                                       mov esi, Msg_File_Directory_Not_Found
                                          ;call print_msg
2586
                                <1>
                                          ;retn
2587
                                <1>
2588 00008443 E985DFFFFF
                                <1>
                                          jmp print_msg
                                <1>
2590
                                <1> loc_longname_not_found:
2591 00008448 BE[AC100100]
                                <1>
                                                   esi, Msg_LongName_Not_Found
                                          mov
                                          ;call print_msg
2592
                                <1>
2593
                                <1>
                                          ;retn
2594 0000844D E97BDFFFFF
                                <1>
                                         jmp print_msg
2595
                                <1>
2596
                                <1> loc_print_longname:
                                     ;mov esi, LongFileName
2597
                                <1>
2598 00008452 BF[865A0100]
                               <1>
                                          mov
                                                edi, TextBuffer
2599 00008457 57
                               <1>
                                         push edi
                               <1>
                                      с...<u>,</u>
ја
2600 00008458 3C00
                                         cmp al, 0
2601 0000845A 7708
                               <1>
                                                short loc_print_longname_1
2602
                               <1> loc_print_FS_longname: ; Singlix FS (64 byte ASCIIZ file name)
2603 0000845C AC
                               <1> lodsb
2604 0000845D AA
                                <1>
                                          stosb
2605 0000845E 08C0
                               <1>
                                         or
                                               al, al
2606 00008460 75FA
                               <1>
                                               short loc_print_FS_longname
                                        jmp
2607 00008462 EB07
                               <1>
                                               short loc_print_longname_2
2608
                               <1>
                               <1> loc_print_longname_1: ; MS Windows long name (UNICODE chars)
2609
                                       lodsw
2610 00008464 66AD
                               <1>
2611 00008466 AA
                                <1>
                                          stosb
2612 00008467 08C0
                               <1>
                                          or
                                                al, al
2613 00008469 75F9
                               <1>
                                          jnz
                                              short loc_print_longname_1
2614
                                <1>
                               <1> loc_print_longname_2:
2615
2616 0000846B 5E
                                     pop esi
                               <1>
                                         call print_msg
2617 0000846C E85CDFFFFF
                                <1>
2618 00008471 BE[FF190100]
                                <1>
                                         mov
                                                esi, nextline
                                         ;call print_msg
                                <1>
2620
                                <1>
                                         ;retn
2621 00008476 E952DFFFFF
                                <1>
                                         jmp print_msg
2622
                                <1>
2623
                                <1> show_file:
                                     ; 18/02/2016
2624
                                <1>
                                         ; 17/02/2016
2625
                                <1>
2626
                                <1>
                                       ; 15/02/2016 (TRDOS 386 = TRDOS v2.0)
                                       ; 13/09/2011 (CMD_INTR.ASM, 'cmp_cmd_show')
; 08/11/2009
2627
                                <1>
2628
                                <1>
2629
                                <1>
                                <1> loc_show_parse_path_name:
2630
                                       mov edi, FindFile_Drv call parse_path_name
2631 0000847B BF[26620100]
                                <1>
2632 00008480 E833200000
                                <1>
2633 00008485 0F824CF9FFF
                                <1>
                                          jc loc_cmd_failed
                                <1>
2635
                                <1> loc_show_check_filename_exists:
2636 0000848B BE[68620100]
                                <1>
                                     mov esi, FindFile_Name
2637 00008490 803E20
                                <1>
                                          cmp
                                                byte [esi], 20h
2638 00008493 0F863EF9FFFF
                                <1>
                                          jna
                                               loc_cmd_failed
2639
                                <1>
                                         ; 15/02/2016 (invalid file name check)
2640
                                <1>
2641 00008499 E807020000
                                <1>
                                         call check_filename
2642 0000849E 730A
                                <1>
                                               short loc_show_change_drv
                                          jnc
2643
                                <1>
2644 000084A0 BE[78110100]
                                <1>
                                                esi, Msg_invalid_name_chars
                                         mov
2645 000084A5 E923DFFFFF
                                <1>
                                          jmp
                                                print_msg
2646
                                <1>
2647
                                <1> loc_show_change_drv:
2648 000084AA 8A35[86590100]
                               <1> mov
                                               dh, [Current Drv]
2649 000084B0 8835[E2600100]
                               <1>
                                                [RUN CDRV], dh
                                         mov
                                         mov dl, [FindFile_Drv]
2650 000084B6 8A15[26620100]
                                <1>
                                         cmp dl, dh
je short loc_show_change_directory
2651 000084BC 38F2
                                <1>
2652 000084BE 740B
                                <1>
                                          call change_current_drive
2653 000084C0 E87FEAFFFF
                                <1>
                                <1>
                                          ;jc
                                                loc_file_rw_cmd_failed
2655 000084C5 0F8237F9FFFF
                                                loc_run_cmd_failed
                                <1>
                                 <1>
2657
                                <1> loc show change directory:
2658 000084CB 803D[27620100]20
                                                byte [FindFile_Directory], 20h
                                <1>
                                          cmp
2659 000084D2 7618
                                                short loc_findload_showfile
                                <1>
                                          jna
2660
                                <1>
2661 000084D4 FE05[620D0100]
                                <1>
                                          inc
                                                byte [Restore_CDIR]
2662 000084DA BE[27620100]
                                                esi, FindFile Directory
                                <1>
                                         mov
                                                ah, ah ; CD COMMAND sign -> 0
2663 000084DF 30E4
                                <1>
                                          xor
2664 000084E1 E8BC190000
                                <1>
                                          call
                                                change_current_directory
                                <1>
                                                loc_file_rw_cmd_failed
2665
                                          ;jc
                                                loc run cmd failed
2666 000084E6 0F8216F9FFFF
                                <1>
2667
                                <1>
2668
                                <1> ;loc_show_change_prompt_dir_string:
2669
                                <1>
                                          ;call change_prompt_dir_string
2670
                                <1>
2671
                                <1> loc_findload_showfile:
                                         ; 15/0<del>2</del>/2016
2672
                                <1>
2673 000084EC BE[68620100]
                                                esi, FindFile_Name
                                <1>
                                          mov
2674 000084F1 BF[17620100]
                                <1>
                                         mov
                                                edi, Dir Entry Name ; Dir Entry Format File Name
                                          call convert_file_name
2675 000084F6 E8F21E0000
                                <1>
2676 000084FB 89FE
                                <1>
                                          mov esi, edi ; offset Dir_Entry_Name
2677
                                <1>
```

```
2678 000084FD 28C0
                               <1>
                                       sub al, al; Attrib AND mask = 0
2679
                                <1>
                                         ; Directory attribute : 10h
2680
                               <1>
                                         ; Volume name attribute: 8h
2681 000084FF B418
                               <1>
                                         mov ah, 00011000b; 18h (Attrib NAND, AND --> zero mask)
                                <1>
2682
                                         ;
2683 00008501 6631C9
                               <1>
                                         xor
2684 00008504 E8ED1B0000
                               <1>
                                         call locate current dir file
2685
                                <1>
                                         ;jc loc_file_rw_cmd_failed
2686 00008509 0F82F3F8FFFF
                               <1>
                                               loc_run_cmd_failed
2687
                                <1>
                                <1> loc_show_load_file:
2688
                                       ; EDI = Directory Entry
2689
                                <1>
2690 0000850F 668B4714
                               <1>
                                         mov ax, [edi+DirEntry_FstClusHI] ; First Cluster High Word
2691 00008513 C1E010
                               <1>
                                         shl
2692 00008516 668B471A
                              <1>
                                         mov
                                               ax, [edi+DirEntry FstClusLO] ; First Cluster Low Word
2693 0000851A A3[D0620100]
                                               [Show Cluster], eax
                                         mov
2694 0000851F 8B471C
                                               eax, [edi+DirEntry FileSize] ; File Size
                                         mov
                                         and
2695 00008522 21C0
                                               eax, eax ; Empty file !
2696 00008524 0F8491000000
                                               end_of_show_file
                                         jz
                                         mov [Show FileSize], eax
2697 0000852A A3[D4620100]
                               <1>
                                     xor
                                               eax, eax
2698 0000852F 31C0
                               <1>
2699 00008531 A3[D8620100]
                               <1>
                                         mov
                                               [Show_FilePointer], eax ; 0
                               <1>
<1>
                                               [Show_ClusterPointer], ax ; 0
2700 00008536 66A3[DC620100]
                                         mov
                               <1>
                                               ebx, ebx
2701 0000853C 29DB
                                         sub
2702 0000853E 8A3D[86590100]
                               <1>
                                         mov
                                               bh, [Current_Drv]
2703 00008544 BE00010900
                               <1>
                                         mov
                                               esi, Logical_DOSDisks
2704 00008549 01DE
                               <1>
                                         add
                                               esi, ebx
2705 0000854B 8935[CC620100]
                               <1>
                                               [Show_LDDDT], esi ; Logical DOS Drv Description Table addr
                                       mov
2706
                                <1>
2707 00008551 807E0300
                                               byte [esi+LD FATType], 0
                               <1>
                                         cmp
2708 00008555 7713
                               <1>
                                         jа
                                               short loc_show_calculate_cluster_size
                                         ; Singlix FS
2709
                                <1>
2710
                                         ; First Cluster Number is FDT number (in compatibility buffer)
                               <1>
2711 00008557 8B15[D0620100]
                               <1>
                                         mov edx, [Show_Cluster] ; Compatibility dir. buffer value (FDT)
                                        mov [Show_FDT], edx xor eax, eax
2712 0000855D 8915[C8620100]
                               <1>
2713 00008563 31C0
                               <1>
                                         mov [Show_Cluster], eax ; Sector index = 0
2714 00008565 A3[D0620100]
                               <1>
2715
                                <1>
                                                              ; (next time it will be 1)
2716
                                <1> loc_show_calculate_cluster_size:
                                     mov bx, [esi+LD_BPB+BPB_BytsPerSec]; FAT 12-16-32 (512)
2717 0000856A 668B5E11
                               <1>
2718
                               <1>
                                         ; BX = 512 = [esi+LD_FS_BytesPerSec] ; Singlix FS
2719 0000856E 8A4613
                                <1>
                                         mov al, [esi+LD BPB+BPB SecPerClust]; FAT 12-16-32 (<= 128)
                                         ; AL = 1 = [esi+LD_FS_Reserved2] ; SectPerClust for Singlix FS
2720
                                <1>
2721 00008571 F7E3
                                <1>
                                         mul ebx
2722
                                <1>
2723
                                <1>
                                         ;cmp eax, 65536; non-compatible (very big) cluster size
2724
                                <1>
                                         ; ja short end of show file
2725 00008573 66A3[DE620100]
                                               [Show_ClusterSize], ax
                               <1>
                                         mov
2726
                                <1>
                                <1> loc_start_show_file:
2728 00008579 BE[FF190100]
                               <1>
                                         mov esi, nextline
2729 0000857E E84ADEFFFF
                                <1>
                                         call print msg
2730
                               <1>
2731 00008583 A1[D0620100]
                               <1>
                                                eax, [Show_Cluster]
                                       mov
2732 00008588 C605[E0620100]17
                               <1>
                                       mov
                                              byte [Show_RowCount], 23
2733
                                <1>
                                         ; 17/02/2016
                                <1>
2735 0000858F 8B35[CC620100]
                                <1>
                                       mov esi, [Show_LDDDT]
2736
                                <1>
2737
                                <1> loc_show_next_cluster:
                                      ; 15/02/2016
2738
                                <1>
2739 00008595 BB00000700
                                         mov ebx, Cluster Buffer; 70000h (for current TRDOS 386 version)
                                <1>
2740
                                <1>
                                         ; ESI = Logical DOS drv description table address
                                         call read_cluster
2741 0000859A E851380000
                                <1>
                                         ;jc loc_file_rw_cmd_failed
2742
                                <1>
2743 0000859F 0F825DF8FFFF
                                <1>
                                               loc_run_cmd_failed
                                         jс
2744
                                <1>
2745 000085A5 31DB
                                <1>
                                         xor ebx, ebx
2746
                                <1> loc_show_next_byte:
2747 000085A7 803D[E0620100]00
                               <1> cmp byte [Show_RowCount], 0
                                               short pass show wait for key
2748 000085AE 7521
                               <1>
                                         jne
2749 000085B0 30E4
                                <1>
                                         xor
                                               ah, ah
                                         call int16h
2750 000085B2 E86886FFFF
                               <1>
2751 000085B7 3C1B
                               <1>
                                         cmp
                                               al, 1Bh
2752 000085B9 750F
                               <1>
                                               short pass exit show
                                        jne
2753
                               <1> end_of_show_file:
2754
                               <1> pass show file:
2755 000085BB BE[FF190100]
                               <1> mov esi, nextline
                                         call print_msg
2756 000085C0 E808DEFFFF
                                <1>
2757 000085C5 E94B010000
                                         jmp loc_file_rw_restore_retn
                                <1>
2758
                                <1>
2759
                                <1> pass_exit_show:
2760 000085CA C605[E0620100]14
                                         mov byte [Show_RowCount], 20
                                <1>
                                <1> pass_show_wait_for_key:
2762 000085D1 81C300000700
                                <1>
                                         add ebx, Cluster_Buffer
2763 000085D7 8A03
                                <1>
                                         mov
                                               al, [ebx]
2764 000085D9 3C0D
                                <1>
                                         cmp al, 0Dh
2765 000085DB 0F8590000000
                                <1>
                                         jne loc show check tab space
2766 000085E1 FE0D[E0620100]
                                <1>
                                         dec byte [Show_RowCount]
                                <1> pass show dec rowcount:
                                         mov \overline{bl}, 7; (light gray character color, black background)
2768 000085E7 B307
                                <1>
2769 000085E9 8A3D[EE580100]
                                <1>
                                         mov
                                               bh, [ACTIVE_PAGE] ; [ptty]
                                         call _write_tty
2770 000085EF E8C796FFFF
                                <1>
                                <1> loc show check eof:
2772 000085F4 FF05[D8620100]
                                <1>
                                         inc dword [Show_FilePointer]
2773 000085FA A1[D8620100]
                                <1>
                                         mov
                                               eax, [Show_FilePointer]
2774 000085FF 3B05[D4620100]
                                <1>
                                               eax, [Show FileSize]
                                         cmp
2775 00008605 73B4
                                <1>
                                               short end_of_show_file
                                         jnb
2776 00008607 66FF05[DC620100]
                                <1>
                                         inc
                                               word [Show_ClusterPointer]
2777 0000860E 0FB71D[DC620100]
                                         movzx ebx, word [Show_ClusterPointer]
                                <1>
2778
                                <1>
2779
                                <1>
                                         ; 17/02/2016
                                         ; (sector boundary -9 bits- check, 512 = 0)
2780
                                <1>
2781 00008615 66F7C3FF01
                                <1>
                                         test bx, 1FFh; 1 to 511
2782 0000861A 758B
                                <1>
                                         jnz short loc_show_next_byte
```

```
2783
                                 <1>
2784
                                 <1>
                                          ; 16/02/2016
                                          mov esi, [Show_LDDDT]
2785 0000861C 8B35[CC620100]
                                 <1>
2786
                                 <1>
2787 00008622 807E0300
                                 <1>
                                                byte [esi+LD FATType], 0
                                          cmp
2788 00008626 7719
                                 <1>
                                          jа
                                                 short loc_show_check_fat_cluster_size
                                <1>
2790
                                          ; Singlix FS
                                 <1>
2791
                                <1>
                                          ; 1 sector, more... (cluster size = 1 sector)
2792 00008628 A1[D0620100]
                                <1>
                                          mov eax, [Show_Cluster]
2793 0000862D 40
                                <1>
                                          inc
                                                eax
                                                [Show_Cluster], eax
2794 0000862E A3[D0620100]
                                 <1>
                                          mov
                                <1>
2796 00008633 6621DB
                                <1>
                                          and bx, bx; 65536 -> 0
2797 00008636 0F856BFFFFF
                                          jnz loc_show_next_byte
                                <1>
2798 0000863C E954FFFFF
                                <1>
                                                loc_show_next_cluster
                                 <1>
                                 <1> loc_show_check_fat_cluster_size:
2800
                                        ; 17/02/2016
2801
                                 <1>
2802 00008641 663B1D[DE620100] <1>
                                          cmp bx, [Show ClusterSize]; cluster size in bytes
2803 00008648 0F8259FFFFF
                                      jb loc_show_next_byte
                                <1>
2804 0000864E 66C705[DC620100]00- <1>
                                          mov word [Show_ClusterPointer], 0
2804 00008656 00
                                <1>
2805
                                <1>
                                          mov eax, [Show_Cluster]
;mov esi, [Show_LDDDT]
2806 00008657 A1[D0620100]
                                <1>
2807
                                <1>
                                <1> loc show get next cluster:
2809 0000865C E86B350000
                                <1> call get_next_cluster
2810
                                <1>
                                                loc_file_rw_cmd_failed
                                          ;jc
                                        jс
2811 00008661 0F829BF7FFFF
                                                loc run cmd failed
                                <1>
                                <1> loc_show_update_ccluster:
2812
2813 00008667 A3[D0620100]
                                 <1> mov [Show_Cluster], eax
2814 0000866C E924FFFFFF
                                <1>
                                          jmp loc_show_next_cluster
2815
                                <1>
                                <1> loc show check tab space:
2816
                                       cmp al, 09h
2817 00008671 3C09
                                <1>
2818 00008673 0F856EFFFFFF
                                <1>
                                          jne pass_show_dec_rowcount
                                <1> loc show put tab space:
                                <1> mov bh, [ACTIVE_PAGE] ; [ptty]
<1> call get_cpos
2820 00008679 8A3D[EE580100]
2821 0000867F E8C692FFFF
                                <1>
                                      ; dl = cursor column
2822
                                <1>
                                      and dl, 7; 18/02/2016; shr bh, 1; [ACTIVE_PAGE]
mov bh, [ACTIVE_PAGE]
2823 00008684 80E207
                                <1>
2824
                                <1>
2825 00008687 8A3D[EE580100] <1>
                                         mov bl, 7; color attribute
2826 0000868D B307
                                <1>
                                 <1> loc_show_put_space_chars:
2827
                               2828 0000868F B020
                                          ;mov bh, [ACTIVE_PAGE] ; [ptty]
2829
2830
                                          ;mov bl, 7 ; color attribute
2832 00008691 52
2833 00008692 E82496FFFF
2834 00008697 5A
2835
2836
2837 00008698 80FA07
2838 0000869B 0F8353FFFFFF
                                <1>
                                          jnb
                                                loc_show_check_eof
2839 000086A1 FEC2
                                 <1>
                                          inc
2840 000086A3 EBEA
                                 <1>
                                                short loc_show_put_space_chars
                                          jmp
2841
                                 <1>
2842
                                 <1> check_filename:
                                      ; 10/10/2016
2843
                                 <1>
                                          ; 15/02/2016 (TRDOS 386 = TRDOS v2.0)
2844
                                 <1>
2845
                                 <1>
                                         ; 07/08/2010 (FILE.ASM, 'proc_check_filename')
                                        ; 10/07/2010
2846
                                 <1>
2847
                                 <1>
                                          ; Derived from 'proc_check_filename'
                                          ; in the old TRDOS.ASM (09/02/2005).
2848
                                 <1>
                                 <1>
2849
                                         ; INPUT ->
2850
                                 <1>
2851
                                 <1>
                                          ; ESI = Dot File Name Location
                                          ; OUTPUT ->
2852
                                 <1>
                                          ; cf = 1 -> error code in AL
2853
                                 <1>
                                                  AL = ERR_INV_FILE_NAME (=26)
2854
                                 <1>
2855
                                 <1>
                                                       Invalid file name chars
2856
                                 <1>
                                                cf = 0 \rightarrow valid file name
                                          ;
2857
                                 <1>
2858
                                 <1>
                                          ; (EAX, ECX, EDI will be changed)
2859
                                 <1>
2860
                                 <1> check_invalid_filename_chars:
                                       \frac{15}{02} 15/02/2016 (TRDOS 386 = TRDOS v2.0)
2861
                                 <1>
2862
                                 <1>
                                           ; 10/07/2010 (FILE.ASM, 'proc_check_invalid_filename_chars')
                                           ; 10/02/2010
2863
                                 <1>
                                           ; Derived from 'proc_check_invalid_filename_chars'
2864
                                 <1>
2865
                                 <1>
                                           ; in the old TRDOS.ASM (09/02/2005).
2866
                                 <1>
                                           ; INPUT ->
2867
                                 <1>
2868
                                 <1>
                                                 ESI = ASCIIZ FileName
                                           ; OUTPUT ->
2869
                                 <1>
2870
                                 <1>
                                                 cf = 1 \rightarrow invalid
2871
                                 <1>
                                                 cf = 0 \rightarrow valid
2872
                                 <1>
                                          ; (EAX, ECX, EDI will be changed)
2873
                                 <1>
2874
                                 <1>
2875 000086A5 56
                                 <1>
                                          push esi
2876
                                 <1>
2877 000086A6 BF[610E0100]
                                 <1>
                                            mov
                                                     edi, invalid_fname_chars
2878 000086AB AC
                                 <1>
                                           lodsb
2879
                                 <1> check_filename_next_char:
2880 000086AC B914000000
                                 <1>
                                           mov ecx, sizeInvFnChars
2881 000086B1 BF[610E0100]
                                 <1>
                                                edi, invalid fname chars
                                           mov
                                 <1> loc_scan_invalid_filename_char:
2882
2883 000086B6 AE
                                 <1>
                                           scasb
2884 000086B7 741F
                                 <1>
                                           jе
                                                 short loc_invalid_filename_stc
2885 000086B9 E2FB
                                 <1>
                                           loop loc_scan_invalid_filename_char
2886 000086BB AC
                                 <1>
                                           lodsb
```

```
2888 000086BE 77EC
                                <1>
                                          jа
                                                short check filename next char
                                <1>
2890
                               <1> check_filename_dot:
2891 000086C0 8B3424
                                <1>
                                       mov esi, [esp]
2892
                                <1>
                           <1> mov ah, 21h
<1> mov ecx, 8
<1> loc_check_filename_next_char:
2893 000086C3 B421
2894 000086C5 B908000000
2895
                              <1> lodsb
2896 000086CA AC
                             <1> cmp al, 2En
<1> jne short pass_ch
<1> loc_check_filename_ext_0:
2897 000086CB 3C2E
2898 000086CD 7511
                                                short pass_check_fn_dot_check
2899
                                        lodsb
2900 000086CF AC
                               <1>
2901 000086D0 38E0
                               <1>
<1>
                                          cmp al, ah; 21h
2902 000086D2 7205
                              <1> jb short lo
<1> cmp al, 2Eh
                                                short loc_invalid_filename
2903 000086D4 3C2E
                                       jne short loc_check_filename_ext_1
2904 000086D6 7519
                                <1>
2905
                                <1>
                                <1> loc invalid_filename_stc:
2906
2907
                                <1> loc_check_fn_stc_rtn:
2908 000086D8 F9
                                <1>
                                         stc
                                <1> loc invalid filename:
2909
2910
                                <1> ; 10/10/2016 (OBh -> 26)
2911 000086D9 B81A000000
                                <1>
                                         mov eax, ERR_INV_FILE_NAME ; (=26)
                                          ; Invalid file name chars
2912
                                <1>
                                <1> loc_check_fn_rtn:
                                <1>
                                      pop esi
2914 000086DE 5E
2915 000086DF C3
                                <1>
                                          retn
2916
                                <1>
2917
                                <1> pass_check_fn_dot_check:
2918 000086E0 38E0
                                <1>
                                      cmp al, ah; 21h
2919 000086E2 7224
                               <1>
                                                short loc_check_fn_clc_rtn
                                          jb
2920 000086E4 E2E4
                               <1>
                                          loop loc_check_filename_next_char
2921 000086E6 AC
                                <1>
                                          lodsb
                               <1>
2922 000086E7 38E0
                                          cmp al, ah; 21h
2923 000086E9 721D
                             <1>
                                          jb short loc_check_fn_clc_rtn
2924 000086EB 3C2E
                               <1>
                                          cmp al, 2Eh
                                                short loc check fn stc rtn
2925 000086ED 75E9
                                <1>
                                          jne
                                          jmp short loc_check_filename_ext_0
2926 000086EF EBDE
                                <1>
2927
                                <1>
                                <1> loc_check_filename ext 1:
2928
                                <1>
2929 000086F1 AC
                                          lodsb
                                          cmp al, ah; 21h
2930 000086F2 38E0
                                <1>
2931 000086F4 7212
                                <1>
                                          jb
                                                short loc_check_fn_clc_rtn
2932 000086F6 3C2E
                                                al, 2Eh
                                <1>
                                          cmp
2933 000086F8 74DE
                               <1>
                                                short loc_check_fn_stc_rtn
                                          jе
2934 000086FA AC
                                <1>
                                          lodsb
2935 000086FB 38E0
                                <1>
                                          cmp al, ah; 21h
2936 000086FD 7209
                                                short loc_check_fn_clc_rtn
                               <1>
                                          jb
                               <1>
2937 000086FF 3C2E
                                          cmp
                                                al, 2Eh
2938 00008701 74D5
                                <1>
                                          jе
                                                short loc_check_fn_stc_rtn
2939 00008703 AC
                                <1>
                                          lodsb
2940 00008704 38E0
                                <1>
                                          cmp al, ah; 21h
2941 00008706 73D0
                                <1>
                                          jnb
                                               short loc_check_fn_stc_rtn
2942
                                <1>
                                <1> loc check fn clc rtn:
2944 00008708 5E
                                <1>
                                          pop
                                                esi
2945 00008709 F8
                                <1>
                                          clc
2946 0000870A C3
                                <1>
                                          retn
2947
                                <1>
2948
                                <1> loc print deleted message:
                                <1> mov esi, Msg_Deleted
2949 0000870B BE[4D120100]
2950 00008710 E8B8DCFFFF
                                <1>
                                          call print_msg
2951
                                <1>
2952
                                <1>
                                          ;clc
                                <1>
                                <1> loc_file_rw_restore_retn:
2954
                                       ; \overline{15/02}/2016 (TRDOS 386 = TRDOS v2.0)
2955
                                <1>
                                          ; 28/02/2010 (CMD INTR.ASM)
2956
                                <1>
2957
                                <1> loc_file_rw_cmd_failed:
                                       pushf
2958 00008715 9C
                                <1>
2959 00008716 E84FF7FFFF
                                          call restore cdir_after cmd_fail
                                <1>
2960 0000871B 9D
                                <1>
                                          popf
                                          jс
2961 0000871C 720D
                                <1>
                                                short loc_file_rw_check_write_fault
2962 0000871E C3
                                <1>
                                          retn
2963
                                <1>
                                <1> loc permission denied:
2964
                                      ; 27/02/2016
2965
                                <1>
2966 0000871F BE[5A120100]
                                          mov esi, Msg Permission Denied
                                <1>
                                       call print_msg
2967 00008724 E8A4DCFFFF
                                <1>
2968 00008729 EBEA
                                        jmp short loc_file_rw_restore_retn
                                 <1>
2969
                                 <1>
2970
                                 <1> loc_file_rw_check_write_fault:
                                          ;cmp al, 1Dh ; Write Fault
2971
                                 <1>
2972 0000872B 3C12
                                 <1>
                                           cmp al, 18 ; 05/11/2016
                                                loc_run_cmd_failed cmp al
2973 0000872D 0F85D4F6FFFF
                                 <1>
                                          jne
2974 00008733 BE[42100100]
                                                 esi, Msg_Not_Ready_Write_Err
                                <1>
                                          mov
2975
                                 <1>
                                          ;call print_msg
2976
                                 <1>
                                          ;retn
2977 00008738 E990DCFFFF
                                 <1>
                                          jmp
                                                print_msg
2978
                                 <1>
2979
                                 <1> make_directory:
2980
                                 <1>
                                          ; 21/02/2016 (TRDOS 386 = TRDOS v2.0)
2981
                                 <1>
                                          ; 12/03/2011 (CMD_INTR.ASM, 'cmp_cmd_mkdir')
2982
                                 <1>
                                          ; 14/08/2010
                                          ; 10/07/2010
2983
                                 <1>
                                          ; 29/11/2009
2984
                                 <1>
2985
                                 <1>
2986
                                 <1> get_mkdir_fchar:
2987
                                 <1>
                                          ; esi = directory name
2988 0000873D 803E20
                                 <1>
                                          cmp byte [esi], 20h
2989 00008740 7701
                                 <1>
                                            ja short loc_mkdir_parse_path_name
2990
                                 <1>
2991
                                 <1> loc_mkdir_nodirname_retn:
```

<1>

cmp al, 1Fh ; 20h and above

2887 000086BC 3C1F

```
2992 00008742 C3
                                <1>
                                          retn
2993
                                <1>
                                <1> loc_mkdir_parse_path_name:
2994
2995 00008743 BF[26620100]
                                <1>
                                          mov edi, FindFile_Drv
2996 00008748 E86B1D0000
                                <1>
                                          call parse_path_name
2997 0000874D 0F8284F6FFFF
                                <1>
                                          jc loc_cmd_failed
                                <1>
2999
                                <1> loc_mkdir_check_dirname_exists:
3000 00008753 BE[68620100]
                               <1> mov esi, FindFile_Name
3001 00008758 803E20
                                <1>
                                                byte [esi], 20h
                                          cmp
3002 0000875B 0F8676F6FFF
                               <1>
                                          jna loc_cmd_failed
3003 00008761 8935[E4620100]
                                                [DelFile FNPointer], esi
                                <1>
                                         mov
                                         call check filename
3004 00008767 E839FFFFF
                                <1>
3005 0000876C 7259
                                <1>
                                                short loc_mkdir_invalid_dir_name_chars
3006
                                <1>
                                <1> loc_mkdir_drv:
3007
3008 0000876E 8A35[86590100]
                                        mov dh, [Current Drv]
                                <1>
3009 00008774 8835[E2600100]
                                <1>
                                                 [RUN CDRV], dh
                                          mov
3010
                                <1>
3011 0000877A 8A15[26620100]
                                <1>
                                                 dl, [FindFile_Drv]
                                          mov
3012 00008780 38F2
                                <1>
                                          cmp
                                                 dl, dh
3013 00008782 7407
                                                 short loc mkdir change directory
                                <1>
                                          jе
                                <1>
3014
                                          call change_current_drive
3015 00008784 E8BBE7FFFF
                                <1>
3016 00008789 728A
                                <1>
                                          jс
                                                loc_file_rw_cmd_failed
3017
                                <1>
                                <1> loc mkdir change directory:
3019 0000878B 803D[27620100]20 <1>
                                                byte [FindFile_Directory], 20h
                                          cmp
3020 00008792 7614
                                <1>
                                                short loc_mkdir_find_directory
3021
                                <1>
3022 00008794 FE05[620D0100] <1>
                                         inc
                                                byte [Restore_CDIR]
3023 0000879A BE[27620100]
                                                esi, FindFile Directory
                                <1>
                                         mov
                                         xor
3024 0000879F 30E4
                                <1>
                                                ah, ah ; CD_COMMAND sign -> 0
3025 000087A1 E8FC160000
                                <1>
                                       call change_current_directory
3026 000087A6 722E
                                <1>
                                                short loc_mkdir_check_error_code
3027
                                <1>
3028
                                <1> ;loc mkdir change prompt dir string:
                                         ;call change_prompt_dir_string
3029
                                <1>
3030
                                <1>
3031
                                <1> loc_mkdir_find_directory:
                                      ;mov esi, FindFile_Name
3032
                                <1>
3033 000087A8 8B35[E4620100]
                                <1>
                                          mov
                                                esi, [DelFile_FNPointer]
                                         ;xor eax, eax
3034
                                <1>
                                         xor ax, ax; any name (dir, file, volume)
3035 000087AE 6631C0
                                <1>
3036 000087B1 E831FBFFFF
                                <1>
                                         call find_first_file
3037 000087B6 721E
                                <1>
                                         jс
                                                short loc_mkdir_check_error_code
3038
                                <1>
3039
                                <1> loc_mkdir_directory_found:
                                         mov esi, Msg_Name_Exists call print_msg
3040 000087B8 BE[A5110100]
                                <1>
3041 000087BD E80BDCFFFF
                                <1>
3042
                                <1>
3043 000087C2 E94EFFFFFF
                                <1>
                                                    loc_file_rw_restore_retn
                                           jmp
3044
                                <1>
3045
                                <1> loc_mkdir_invalid_dir_name_chars:
                                      mov esi, Msg_invalid_name_chars call print_msg
3046 000087C7 BE[78110100]
                                <1>
3047 000087CC E8FCDBFFFF
                                <1>
                                <1>
3049 000087D1 E93FFFFFF
                                <1>
                                                 loc_file_rw_restore_retn
                                          jmp
3050
                                <1>
                                <1> loc_mkdir_check_error_code:
3051
3052 000087D6 3C02
                                <1>
                                      cmp al, 2
                                                short loc mkdir directory not found
3053
                                <1>
                                          ;je
3054 000087D8 7406
                                <1>
                                          jе
                                                short loc_mkdir_ask_for_yes_no
3055 000087DA F9
                                <1>
                                          stc
3056 000087DB E935FFFFFF
                                <1>
                                                  loc_file_rw_cmd_failed
                                          jmp
3057
                                <1>
                                <1> loc mkdir directory not found:
3058
                                <1> loc_mkdir_ask_for_yes_no:
3059
                                         mov esi, Msg_DoYouWantMkdir call print_msg
3060 000087E0 BE[C6110100]
                                <1>
3061 000087E5 E8E3DBFFFF
                                <1>
3062 000087EA 8B35[E4620100]
                                <1>
                                          mov
                                                esi, [DelFile_FNPointer]
3063 000087F0 E8D8DBFFFF
                                         call print_msg
                                <1>
                                         mov
3064 000087F5 BE[E5110100]
                                <1>
                                                 esi, Msg_YesNo
                                          call print_msg
3065 000087FA E8CEDBFFFF
                                <1>
3066
                                <1>
                                                byte [Y N nextline], 20h
3067 000087FF C605[EF110100]20
                                <1>
                                         mov
3068
                                 <1>
3069
                                <1> loc_mkdir_ask_again:
3070 00008806 30E4
                                <1>
                                         xor ah, ah
3071 00008808 E81284FFFF
                                          call int16h
                                <1>
3072 0000880D 3C1B
                                <1>
                                          cmp al, 1Bh
                                 <1>
                                                short loc_do_not_make_directory
                                          ;je
                                          jе
3074 0000880F 7439
                                <1>
                                                 short loc_mkdir_y_n_escape
3075 00008811 24DF
                                <1>
                                                 al, ODFh ; y \rightarrow Y, n \rightarrow N
3076 00008813 3C59
                                <1>
                                                al, 'Y' ; 'yes'
                                          cmp
3077 00008815 7404
                                                 short loc_mkdir_yes_make_directory
                                <1>
                                          jе
3078 00008817 3C4E
                                                al, 'N'; 'no'
                                <1>
                                          cmp
3079 00008819 75EB
                                <1>
                                                short loc_mkdir_ask_again
                                          jne
3080
                                <1>
                                 <1> loc_do_not_make_directory:
3081
3082
                                 <1> loc mkdir yes make directory:
3083 0000881B E82E000000
                                 <1>
                                          call y_n_answer ; 29/12/2017
                                          ;cmp al, 'Y'; 'yes'
                                <1>
3084
3085
                                <1>
                                          ;jnc loc_file_rw_restore_retn
cmp al, 'N'; 'no'
3086
                                <1>
3087 00008820 3C4E
                                <1>
                                          je loc_file_rw_restore_retn
3088 00008822 0F84EDFEFFFF
                                <1>
3089
                                <1>
                                 <1> loc mkdir call make sub directory:
3090
                                        mov esi, [DelFile_FNPointer]
3091 00008828 8B35[E4620100]
                                <1>
3092 0000882E B110
                                <1>
                                          mov cl, 10h; Directory attributes
                                          call make_sub_directory
3093 00008830 E8821D0000
                                 <1>
                                <1> loc_rename_file_ok: ; 06/03/2016
3095 00008835 0F82DAFEFFFF
                                <1> jc loc_file_rw_cmd_failed
3096
                                 <1> move_source_file_to_destination_OK:
```

```
3097 0000883B BE[F3110100] <1>
3098 00008840 E888DBFFFF <1>
3099 00008845 E9CBFEFFFF <1>
                                      mov esi, Msg_OK
call print_msg
jmp loc_file_rw_restore_retn
3100
                                <1>
                                <1> loc_mkdir_y_n_escape:
3101
                                       mov al, 'N'; 'no'
jmp short loc_do_not_make_directory
3102 0000884A B04E
                                <1>
3103 0000884C EBCD
                                <1>
3104
                                <1>
3105
                                <1> y_n_answer:
                                        ; 29/12/2017
mov [Y_N_nextline], al
3106
                                <1>
3107 0000884E A2[EF110100] <1>
3108 <1>
;push ax
3112 0000885E 58
                                <1>
                                          pop
                                                 eax
                                          ;pop ax
3113
                                <1>
3114 0000885F C3
                                        retn
                                 <1>
3115
                                 <1>
                                 <1> delete_directory:
3116
                                      ; 29/12/2017
3117
                                 <1>
                                      ; 15/10/2016
; 01/03/2016, 06/03/2016
; 27/02/2016, 28/02/2016, 29/02/2016
; 26/02/2016 (TRDOS 386 = TRDOS v2.0)
; 16/10/2010 (CMD_INTR.ASM, 'cmp_cmd_
; 05/06/2010
3118
                                 <1>
                                 <1>
3119
3120
                                 <1>
                                 <1>
3121
                                          ; 16/10/2010 (CMD_INTR.ASM, 'cmp_cmd_rmdir')
3122
                                 <1>
3123
                                 <1>
                                 <1>
3124
3125
                                 <1> get_fchar:
                                        ; esi = directory name
3126
                                <1>
3127 00008860 803E20
                                <1>
                                           cmp byte [esi], 20h
                                          ja short loc_rmdir_parse_path name
3128 00008863 7701
                                <1>
3129
                                <1>
3130
                                <1> loc_rmdir_nodirname_retn:
3131 00008865 C3
                                         retn
                                <1>
3132
                                <1>
                                <1> loc_rmdir_parse_path_name:
                                        mov edi, FindFile_Drv call parse_path_name
                                <1>
3134 00008866 BF[26620100]
3135 0000886B E8481C0000
                                <1>
                                           jc loc_cmd_failed
3136 00008870 0F8261F5FFFF
                                <1>
3137
                                <1>
                                 <1> loc rmdir check dirname exists:
3138
3139 00008876 BE[68620100]
                                3140 0000887B 803E20
                                <1>
                                           cmp byte [esi], 20h
                                       jna
mov
                                                 loc cmd failed
3141 0000887E 0F8653F5FFFF
                                <1>
3142 00008884 8935[E4620100]
                                <1>
                                                 [DelFile_FNPointer], esi
                                <1>
3144
                                 <1> loc_rmdir_drv:
                                                 dh, [Current Drv]
3145 0000888A 8A35[86590100]
                                 <1>
                                          mov
                                                 [RUN CDRV], dh
3146 00008890 8835[E2600100]
                                <1>
                                           mov
3147
                                 <1>
                                                 dl, [FindFile_Drv]
3148 00008896 8A15[26620100]
                                 <1>
3149 0000889C 38F2
                                <1>
                                                 dl, dh
3150 0000889E 740B
                                <1>
                                                 short loc_rmdir_change_directory
                                 <1>
3152 000088A0 E89FE6FFFF
                                 <1>
                                           call change_current_drive
                                                 loc file_rw_cmd_failed
3153 000088A5 0F826AFEFFFF
                                 <1>
                                          jс
3154
                                 <1>
3155
                                 <1> loc_rmdir_change_directory:
3156 000088AB 803D[27620100]20 <1> cmp byte [FindFile_Directory], 20h
3157 000088B2 7614
                                <1>
                                                 short loc_rmdir_find_directory
                                 <1>
                                <1>
<1>
<1>
3159 000088B4 FE05[620D0100]
                                                 byte [Restore_CDIR]
                                          inc
3160 000088BA BE[27620100]
                                          mov esi, FindFile_Directory
3161 000088BF 30E4
                                 <1>
                                           xor
                                                 ah, ah ; CD COMMAND sign -> 0
3162 000088C1 E8DC150000
                                 <1>
                                           call
                                                change_current_directory
                                                 short loc_rmdir_check_error_code
                                          jс
3163 000088C6 7211
                                <1>
3164
                                 <1>
3165
                                 <1> ;loc_rmdir_change_prompt_dir_string:
3166
                                 <1>
                                          ; call change prompt dir string
3167
                                 <1>
3168
                                 <1> loc rmdir find directory:
                                  ;mov esi, FindFile_Name
3169
                                          mov esi, [DelFile FNPointer]
3170 000088C8 8B35[E4620100]
                                <1>
                                      mov
                                          mov ax, 0810h; Only directories call find_first_file
3171 000088CE 66B81008
                                 <1>
3172 000088D2 E810FAFFFF
                                <1>
                                         jnc short loc_rmdir_ambgfn_check
3173 000088D7 730A
                                 <1>
3174
                                 <1>
3175
                                 <1> loc_rmdir_check_error_code:
3176 000088D9 3C02
                                 <1> cmp al, 2
                                                  short loc_rmdir_directory_not_found
3177 000088DB 740B
                                 <1>
                                           jе
3178 000088DD F9
                                 <1>
                                           stc
3179 000088DE E932FEFFFF
                                <1>
                                                 loc_file_rw_cmd_failed
                                           jmp
                                 <1>
                                 <1> loc rmdir ambgfn check:
3181
3182 000088E3 6621D2
                                 <1>
                                           and
                                                 dx, dx; Ambiguous filename chars used sign (DX>0)
3183 000088E6 740F
                                                 short loc rmdir directory found
                                <1>
3184
                                 <1>
3185
                                 <1> loc_rmdir_directory_not_found:
3186 000088E8 BE[64100100]
                                <1>
                                          mov esi, Msg_Dir_Not_Found
3187 000088ED E8DBDAFFFF
                                 <1>
                                           call print_msg
                                 <1>
3189 000088F2 E91EFEFFFF
                                 <1>
                                           jmp
                                                 loc_file_rw_restore_retn
3190
                                 <1>
3191
                                 <1> loc_rmdir_directory_found:
3192 000088F7 80E307
                                           and bl, 07h; Attributes
                                 <1>
                                           jnz loc_permission_denied
3193 000088FA 0F851FFEFFFF
                                 <1>
3194
                                 <1>
3195
                                 <1> loc_rmdir_save_lnel: ; 28/02/2016
                                          ; mov bh, [LongName EntryLength]
3196
                                 <1>
3197 00008900 883D[EE620100]
                                 <1>
                                           mov [DelFile_LNEL], bh ; Long name entry length (if > 0)
                                        ; edi = Directory Entry Offset (DirBuff)
3198
                                 <1>
                                          ; esi = Directory Entry (FFF Structure)
3199
                                 <1>
3200
                                 <1>
                                         ;mov [DelFile_DirEntryAddr], edi ; not required
3201
                                 <1>
                                         ;mov ax, [edi+20] ; First Cluster High Word
```

```
3202
                                <1>
                                         ;shleax, 16
3203
                                <1>
                                         ;mov ax, [edi+26] ; First Cluster Low Word
                                         ; ROOT Dir First Cluster = 0
3204
                                <1>
3205
                                <1>
                                         ;cmpeax, 2
                                         ;jb loc_update_direntry_1
3206
                                <1>
3207
                                <1>
                                <1> pass rmdir fc check:
3208
3209 00008906 57
                                       push edi ; * (29/02/2016)
                                <1>
3210
                                <1>
3211 00008907 BE[F9110100]
                                       mov esi, Msg_DoYouWantRmDir
                               <1>
3212 0000890C E8BCDAFFFF
                               <1>
                                     call print_msg
3213 00008911 8B35[E4620100]
                               <1>
                                        mov
                                              esi, [DelFile_FNPointer]
3214 00008917 E8B1DAFFFF
                                        call print_msg
                               <1>
3215 0000891C BE[E5110100]
                               <1>
                                        mov esi, Msg_YesNo
                                       call print_msg
3216 00008921 E8A7DAFFFF
                               <1>
3217
                               <1>
                               <1> loc_rmdir_ask_again:
3218
3219 00008926 30E4
                                     xor ah, ah
                               <1>
3220 00008928 E8F282FFFF
                               <1>
                                         call
                                              int16h
3221 0000892D 3C1B
                               <1>
                                        cmp al, 1Bh
                                               loc_rmdir_y_n_escape ; 06/03/2016
                                               short loc_rmdir_yes_delete_directory
                                              al, 'N'
3228 0000893C 3C4E
                               <1>
                                        cmp
3229 0000893E 75E6
                               <1>
                                       jne short loc_rmdir_ask_again
3230
                                <1>
3231
                               <1> loc_do_not_delete_directory:
3232
                               <1> loc_rmdir_yes_delete_directory:
                                     call y_n_answer; 29/12/2017 pop edi; * (29/02/2016)
3233 00008940 E809FFFFFF
                                <1>
3234 00008945 5F
                               <1>
                                         ;cmp al, 'Y'; 'yes'
3235
                               <1>
3236
                               <1>
                                         ;cmc
3237
                               <1>
                                         ;jnc loc_file_rw_restore_retn
                                       cmp al, 'N'; 'no'
3238 00008946 3C4E
                               <1>
                                        je loc_file_rw_restore_retn
3239 00008948 0F84C7FDFFFF
                               <1>
3240
                               <1>
                                         ; 29/12/2017
3241
                               <1>
3242 0000894E E869000000
                                       call delete_sub_directory
                               <1>
3243 00008953 7213
                                <1>
                                               short loc_rmdir_cmd_failed
3244
                               <1>
3245
                               <1> loc_rmdir_ok:
                                     mov esi, Msg_OK call print_msg
3246 00008955 BE[F3110100]
                               <1>
3247 0000895A E86EDAFFFF
                               <1>
3248 0000895F E9B1FDFFFF
                               <1>
                                         jmp loc_file_rw_restore_retn
3249
                               <1>
                               3250
3251 00008964 B04E
                                         jmp loc_do_not_delete_directory
3252 00008966 EBD8
                               <1>
3253
                               <1>
                               <1> loc_rmdir_cmd_failed:
3254
                                     ; 2<del>9</del>/12<del>/</del>2017
3255
                               <1>
3256 00008968 09C0
                               <1>
                                        or eax, eax; EAX = 0 -> Directory not empty!
3257 0000896A 7426
                               <1>
                                               short loc_rmdir_directory_not_empty
3258
                                <1>
3259
                                <1>
                                       ; EAX > 0 \rightarrow Error code in AL (or AX or EAX)
3260
                                <1>
3261 0000896C 833D[A2600100]01
                                              dword [FAT_ClusterCounter], 1
                                <1>
                                         cmp
3262 00008973 0F829CFDFFFF
                                <1>
                                         jb
                                               loc_file_rw_cmd_failed
3263 00008979 F9
                                <1>
                                         stc
                                <1> loc_rmdir_cmd_return:
3264
3265
                                <1>
                                       ; 01/03/2016
3266 0000897A 9C
                                       pushf
                                <1>
3267
                               <1>
                                         ; ESI = Logical DOS Drive Description Table address
3268 0000897B 66BB00FF
                               <1>
                                        mov
                                              bx, OFFOOh ; BH = FFh -> use ESI for Drive parameters
                               <1>
                                                   ; BL = 0 -> Recalculate free cluster count
3269
3270 0000897F 50
                               <1>
3271 00008980 E8C3380000
                               <1>
                                         call calculate_fat_freespace
3272 00008985 58
                                <1>
                                         pop
                                         popf
3273 00008986 9D
                                <1>
3274 00008987 0F8288FDFFFF
                                               loc_file_rw_cmd_failed
                               <1>
                                         jс
3275 0000898D E983FDFFFF
                               <1>
                                               loc_file_rw_restore_retn
3276
                                <1>
                               <1> loc_rmdir_directory_not_empty:
3277
                               <1> mov esi, Msg_Dir_Not_Empty
3278 00008992 BE[1A120100]
                                         call print_msg
3279 00008997 E831DAFFFF
                               <1>
3280
                                <1>
                                         ; 01/03/2016
                                         mov eax, [FAT_ClusterCounter]
3281 0000899C A1[A2600100]
                               <1>
3282 000089A1 09C0
                                <1>
                                     or eax, eax; 0?
3283 000089A3 0F846CFDFFFF
                                <1>
                                       jz
                                               loc_file_rw_restore_retn
                                         ; ESI = Logical DOS Drive Description Table address
3284
                                <1>
3285 000089A9 66BB01FF
                                <1>
                                              bx, OFFO1h; BH = FFh -> use ESI for Drive parameters
3286
                                <1>
                                                   ; BL = 1 -> add free clusters
3287 000089AD E896380000
                                <1>
                                         call calculate_fat_freespace
3288 000089B2 09C9
                                <1>
                                         or ecx, ecx
3289 000089B4 0F845BFDFFFF
                                <1>
                                                  loc_file_rw_restore_retn ; ecx = 0 -> OK
                                         jz
3290
                                <1>
                                         ; ecx > 0 -> Error (Recalculation is needed)
3291 000089BA EBBE
                                <1>
                                         jmp short loc_rmdir_cmd_return
3292
                                <1>
3293
                                <1>
                                <1> delete sub directory:
3294
3295
                                <1>
                                         ; 29/12/2017
3296
                                <1>
                                         ; (moved here from 'delete_directory' for 'sysrmdir' )
3297
                                <1>
3298
                                <1>
                                         ; EDI = Directory buffer entry offset/address
3299
                                <1>
3300
                                <1> loc_rmdir_delete_short_name_check_dir_empty:
3301 000089BC 668B4714
                                         mov ax, [edi+20]; First Cluster High Word
                                <1>
3302 000089C0 C1E010
                                <1>
                                          shl eax, 16
3303 000089C3 668B471A
                                <1>
                                              ax, [edi+26] ; First Cluster Low Word
                                         mov
3304
                                <1>
3305
                                <1>
                                         ;mov [DelFile_FCluster], eax
3306
                                <1>
```

```
3307
                                 <1>
                                           ;;mov bx, [DirBuff_EntryCounter]
3308
                                  <1>
                                           ;mov bx, [FindFile DirEntryNumber]; 27/02/2016
3309
                                 <1>
                                           ;mov [DelFile_EntryCounter], bx
3310
                                 <1>
3311 000089C7 29DB
                                 <1>
                                                 ebx, ebx
                                           sub
3312
                                 <1>
                                           ; 29/12/2017
3313 000089C9 891D[A2600100]
                                 <1>
                                           mov
                                                 [FAT ClusterCounter], ebx; 0; Reset
3314
                                 <1>
3315 000089CF 8A3D[26620100]
                                 <1>
                                                  bh, [FindFile Drv]
3316 000089D5 BE00010900
                                 <1>
                                                  esi, Logical DOSDisks
                                           mov
                                           add
3317 000089DA 01DE
                                 <1>
                                                 esi, ebx
3318
                                 <1>
3319 000089DC 66817F0CA101
                                 <1>
                                                  word [edi+DirEntry NTRes], 01A1h
                                           cmp
                                                  short loc rmdir check fs directory
3320 000089E2 745A
                                 <1>
3321
                                 <1>
                                                 byte [esi+LD FATType], 1
3322
                                 <1>
                                           ;cmp
3323
                                 <1>
                                           ;jb
                                                 short loc rmdir get last cluster 0
3324
                                 <1>
3325
                                 <1>
                                           ; 29/12/2017
3326 000089E4 83F802
                                 <1>
                                           cmp eax, 2
                                           jnb short loc_rmdir_get_last_cluster_1
3327 000089E7 7306
                                 <1>
                                           ; eax < 2
3328
                                 <1>
3329
                                 <1> loc rmdir_get_last_cluster_0:
3330
                                 <1>
                                           ;mov eax, ERR_INV_FORMAT ; invalid format!
3331 000089E9 B813000000
                                 <1>
                                           mov
                                                 eax, ERR_NOT_DIR ; not a valid directory!
3332
                                 <1>
                                           ;stc
3333 000089EE C3
                                 <1>
                                           retn
3334
                                 <1>
3335
                                 <1> loc_rmdir_get_last_cluster_1:
3336 000089EF 807E0303
                                           cmp byte [esi+LD FATType], 3 ; FAT32
                                 <1>
3337 000089F3 750C
                                 <1>
                                                short loc_rmdir_get_last_cluster_2
3338
                                 <1>
3339
                                           ; is it root directory ?
                                 <1>
                                           cmp eax, [esi+LD_BPB+BPB_RootClus]
3340 000089F5 3B4632
                                <1>
3341 000089F8 7507
                                 <1>
                                                 short loc_rmdir_get_last_cluster_2
3342
                                 <1>
3343
                                 <1>
                                           ; root directory can not be deleted !!
                                 <1> loc_rmdir_permission_denied:
3344
                                                eax, ERR PERM DENIED ; permission denied!
3345 000089FA B80B000000
                                 <1>
                                           mov
3346 000089FF F9
                                 <1>
                                           stc
3347 00008A00 C3
                                 <1>
                                           retn
3348
                                 <1>
3349
                                 <1> loc_rmdir_get_last_cluster_2:
                                           ; 29/12/2017
3350
                                 <1>
3351 00008A01 A3[E8620100]
                                 <1>
                                           mov [DelFile_FCluster], eax
3352
                                 <1>
                                 <1>
                                           ;mov dx, [DirBuff EntryCounter]
3354 00008A06 668B15[A0620100]
                                           mov
                                                  dx, [FindFile DirEntryNumber] ; 27/02/2016
                                 <1>
3355 00008A0D 668915[EC620100]
                                 <1>
                                           mov
                                                  [DelFile_EntryCounter], dx
                                 <1>
                                           mov
3357 00008A14 8B15[B1600100]
                                 <1>
                                                  edx, [DirBuff_Cluster]
3358 00008A1A 8915[18630100]
                                 <1>
                                           mov
                                                  [RmDir ParentDirCluster], edx
3359
                                 <1>
3360 00008A20 893D[14630100]
                                 <1>
                                                  [RmDir_DirEntryOffset], edi
                                           mov
3361
                                  <1>
3362
                                 <1>
                                           ; 01/03/2016
                                           ;mov dword [FAT ClusterCounter], 0 ; Reset
3363
                                 <1>
3364
                                 <1>
3365
                                 <1> loc_rmdir_get_last_cluster_3:
3366 00008A26 E89C390000
                                       call get last cluster
                                 <1>
3367
                                 <1>
                                           ;jc loc_rmdir_cmd_failed
3368 00008A2B 721E
                                                 short loc delete sub dir retn ; 29/12/2017
                                 <1>
                                 <1>
3369
3370 00008A2D 3B05[E8620100]
                                 <1>
                                                  eax, [DelFile_FCluster]
3371 00008A33 7517
                                 <1>
                                                 short loc_rmdir_multi_dir_clusters
                                           jne
3372
                                 <1>
3373 00008A35 C605[13630100]00
                                 <1>
                                           mov
                                                 byte [RmDir MultiClusters], 0
3374 00008A3C EB15
                                 <1>
                                           jmp
                                                  short pass_rmdir_multi_dir_clusters
3375
                                 <1>
3376
                                 <1> loc_rmdir_check_fs_directory:
                                          ; 29/12/2017
3377
                                 <1>
3378 00008A3E 807E04A1
                                           cmp byte [esi+LD FSType], 0A1h
                                 <1>
3379 00008A42 75B6
                                 <1>
                                           jne
                                                 short loc_rmdir_permission_denied
3380
                                 <1>
3381
                                 <1> loc rmdir delete fs directory:
                                           call delete_fs_directory
3382 00008A44 E876130000
                                 <1>
                                 <1>
                                           ; jnc loc print deleted message
3384 00008A49 7300
                                           jnc short loc delete sub dir retn ; 29/12/2017
                                 <1>
3385
                                 <1>
3386
                                 <1>
                                           ; EAX=0 -> Directory not empty !
3387
                                  <1>
                                           ; EAX>0 \rightarrow Disk r/w error or another (misc) error
3388
                                  <1>
                                                  eax, eax
3389
                                  <1>
                                           ;or
3390
                                  <1>
                                                 loc_rmdir_directory_not_empty_2
                                           ;jz
3391
                                  <1>
                                           ;;stc
3392
                                 <1>
                                           ;;jmp loc_file_rw_cmd_failed
3393
                                 <1>
3394
                                 <1> loc_delete_sub_dir_retn:
3395 00008A4B C3
                                 <1>
                                           retn
3396
                                 <1>
3397
                                 <1> loc_rmdir_multi_dir_clusters:
3398 00008A4C C605[13630100]01
                                 <1>
                                          mov byte [RmDir_MultiClusters], 1
3399
                                 <1>
3400
                                 <1> pass_rmdir_multi_dir_clusters:
3401 00008A53 A3[1C630100]
                                 <1>
                                           mov [RmDir_DirLastCluster], eax
3402 00008A58 890D[20630100]
                                 <1>
                                           mov
                                                 [RmDir_PreviousCluster], ecx
                                 <1>
3404
                                 <1> loc_rmdir_load_fat_sub_directory:
3405 00008A5E E84F330000
                                 <1>
                                        call load_FAT_sub_directory
                                           ;jc loc rmdir cmd failed
3406
                                 <1>
3407 00008A63 72E6
                                 <1>
                                                 short loc_delete_sub_dir_retn
3408
                                 <1>
                                 <1> loc_rmdir_find_last_dir_entry:
3409
3410 00008A65 56
                                 <1>
                                         push esi
                                 <1>
                                           mov esi, Dir_File_Name
3411 00008A66 BE[0A620100]
```

```
<1>
<1`
3412 00008A6B C6062A
                                        mov byte [esi], '*'
                               <1> mov byte [esi+8], '*'
<1> xor ebx, ebx; Entry offset = 0
3413 00008A6E C646082A
3414 00008A72 31DB
3415
                               <1> loc_rmdir_find_last_dir_entry_next:
                         3416 00008A74 66B80008
3417 00008A78 6631C9
                                                cx, cx; 0 = Find a valid file or dir name
3418 00008A7B E87A170000
3419 00008A80 7225
3420 00008A82 83FB01
                                                short loc_rmdir_directory_not_empty_1
3421 00008A85 771B
                              <1> loc_rmdir_dot_entry_check:
3422
                                <1> cmp ch, '.'; The first char of the dir entry
<1> jne short loc_rmdir_directory_not_empty_1
3423 00008A87 80FD2E
                           <1>
3424 00008A8A 7516
3425 00008A8C 08DB
                                                short loc_rmdir_dotdot_entry_check
3426 00008A8E 7506
3427 00008A90 807F0120
3428 00008A94 EB06
3429
                                <1> loc rmdir dotdot entry check:
3430
3431 00008A96 66817F012E20
                                <1> cmp word [edi+1], '. '
3432
                                <1> pass_rmdir_dot_entry_check:
3433 00008A9C 7504
                                <1>
                                      jne short loc_rmdir_directory_not_empty_1
3434 00008A9E FEC3
                                <1>
                                          inc
3435 00008AA0 EBD2
                                <1>
                                          jmp short loc_rmdir_find_last_dir_entry_next
3436
                                <1>
                                <1> loc_rmdir_directory_not_empty_1:
3437
3438 00008AA2 58
                                <1> pop eax; pushed esi
                              <1>
3439 00008AA3 31C0
                                          xor
                                                eax, eax ; 0
3440
                                <1> loc_rmdir_directory_not_empty_2:
3441
                                <1> loc_delete_sub_dir_stc_retn:
3442 00008AA5 F9
                                <1>
                                         stc
3443 00008AA6 C3
                                <1>
                                          retn
3444
                                <1>
3445
                                <1> loc_rmdir_empty_dir_cluster:
3446 00008AA7 5E
                                <1>
                                        pop esi
3447
                                <1>
                                <1> loc rmdir set prev cluster dir last cluster:
3449 00008AA8 803D[13630100]00 <1>
                                       cmp byte [RmDir_MultiClusters], 0
                                                short loc rmdir unlink dir last cluster
3450 00008AAF 7613
                                <1>
3451
3452 00008AB1 A1[20630100] <1>
3453 <1>
                                          mov eax, [RmDir_PreviousCluster]
                                          ;xor ecx, ecx
                                                ecx ; FFFFFFFFh
                                          dec
3455 00008AB7 E83A340000
                                <1>
                                          call update cluster
3456 00008ABC 7306
                                <1>
                                        jnc short loc_rmdir_unlink_dir_last_cluster
3457
                                <1>
                                          ; 01/03/2016
3458
                                <1>
                                 <1>
3459
                                          ;cmp eax, 1 ; eax = 0 \rightarrow end of cluster chain
3460
                                 <1>
                                          ; cmc
                                                short loc_rmdir_cmd_failed
3461
                                 <1>
                                          ;jc
3462
                                 <1>
                                          ;jmp short loc_rmdir_save_fat_buffer
3463
                                 <1>
                                          ; 29/12/2017
3464 00008ABE 21C0
                                <1>
                                          and eax, eax
3465 00008AC0 75E3
                                <1>
                                          jnz short loc_delete_sub_dir_stc_retn
3466 00008AC2 EB12
                                <1>
                                          jmp short loc_rmdir_save_fat_buffer
3467
                                <1>
                                <1> loc rmdir unlink dir last cluster:
3469 00008AC4 A1[1C630100]
3470 00008AC9 31C9
                                <1> mov eax, [RmDir_DirLastCluster]
                                <1>
                                                ecx, ecx ; 0
                                          xor
                                      jnc short loc_rmdir_unlink_stc_retn_0Bh
; Because of it is the last cluster
; 'update_cluster' must return with eocc e
or eax, eax
3471 00008ACB E826340000
                                <1>
3472 00008AD0 7327
                                <1>
3473
                                <1>
                                          ; 'update_cluster' must return with eocc error
3474
                                <1>
3475 00008AD2 09C0
                                <1>
                                          ;jz
3476
                                <1>
                                                short loc_rmdir_save_fat_buffer ; eocc
3477
                                <1>
                                          ;stc
                                                     short loc_rmdir_cmd_failed
3478
                                <1>
                                          ;jmp
                                          ; 29/12/2017
3479
                                <1>
3480 00008AD4 75CF
                                <1>
                                          jnz short loc_delete_sub_dir_stc_retn
3481
                                <1>
3482
                                <1> loc_rmdir_save_fat_buffer:
3483 00008AD6 803D[9A600100]02
                                <1>
                                      cmp byte [FAT_BuffValidData], 2
3484 00008ADD 7528
                                                short loc_rmdir_calculate_FAT_freespace
                                <1>
                                          jne
3485 00008ADF E8CF360000
                                <1>
                                          call save_fat_buffer
                                        ;jc
3486
                                <1>
                                                short loc_rmdir_cmd_failed
3487
                                <1>
                                          ; 29/12/2017
3488 00008AE4 7219
                                <1>
                                          jc short loc_rmdir_unlink_error_retn
3489
                                 <1>
                                          ; 01/03/2016
3490
                                 <1>
3491 00008AE6 803D[13630100]00
                                          cmp byte [RmDir MultiClusters], 0
                                <1>
3492 00008AED 7618
                                 <1>
                                              short loc_rmdir_calculate_FAT_freespace
3493
                                 <1>
                                          mov eax, [DelFile_FCluster]
3494 00008AEF A1[E8620100]
                                 <1>
3495 00008AF4 E92DFFFFFF
                                                 loc_rmdir_get_last_cluster_3
                                 <1>
                                           jmp
3496
                                 <1>
3497
                                 <1> loc_rmdir_unlink_stc_retn_0Bh:
                                      ; 1\overline{5}/10/20\overline{16} (\overline{0}Bh -> 28)
                                <1>
                                          mov eax, ERR INV FORMAT ; 28 = Invalid format
3499 00008AF9 B81C000000
                                <1>
3500
                                <1> loc_rmdir_unlink_stc_retn:
3501 00008AFE F9
                                <1> stc
                                <1> loc_rmdir_unlink_error_retn:
3502
3503 00008AFF C3
                                 <1>
                                          retn
                                <1>
3504
3505
                                <1> loc rmdir delete short name invalid data:
                                        mov eax, 29; Invalid data (15/10/2016)
3506 00008B00 B81D000000
                                <1>
3507
                                <1>
                                          ;jmploc rmdir cmd failed
3508
                                <1>
                                          ; 29/12/2017
3509
                                <1>
3510 00008B05 EBF7
                                <1>
                                          jmp short loc_rmdir_unlink_stc_retn
3511
                                <1>
3512
                                <1> loc_rmdir_calculate_FAT_freespace:
                                      ;mov eax, [FAT_ClusterCounter]
3513
                                 <1>
3514
                                <1>
                                          ; 29/12/2017
                                          sub eax, eax; 0
3515 00008B07 29C0
                                <1>
3516 00008B09 8705[A2600100]
                                <1>
                                          xchg eax, [FAT_ClusterCounter]
```

```
3517
                                 <1>
3518 00008B0F 66BB01FF
                                 <1>
                                          mov bx, 0FF01h
                                          ; BL = 1 \rightarrow Add EAX to free space count
3519
                                 <1>
3520
                                 <1>
                                          ; BH = FFh ->
3521
                                 <1>
                                          ; ESI = Logical DOS Drive Description Table address
3522 00008B13 E830370000
                                 <1>
                                           call calculate_fat_freespace
3523
                                 <1>
                                          and ecx, ecx; ecx = 0 -> valid free sector count
3524 00008B18 21C9
                                 <1>
3525 00008B1A 7409
                                <1>
                                                 short loc_rmdir_delete_short_name_continue
3526
                                <1>
3527
                                <1> loc_rmdir_recalculate_FAT_freespace:
                                          mov bx, 0FF00h; BL = 0 -> Recalculate free space
3528 00008B1C 66BB00FF
                                <1>
3529 00008B20 E823370000
                                <1>
                                           call calculate_fat_freespace
3530
                                <1>
                                <1> loc_rmdir_delete_short_name_continue:
3531
3532 00008B25 A1[18630100]
                                          mov eax, [RmDir ParentDirCluster]
                                <1>
3533 00008B2A 83F802
                                <1>
                                           cmp eax, 2
3534 00008B2D 7309
                                                short loc_rmdir_del_short_name_load_sub_dir
                                <1>
                                           jnb
                                          call load_FAT_root_directory
;jc loc_file_rw_cmd_failed
3535 00008B2F E8F3310000
                                <1>
3536
                                <1>
                                          ; 29/12/2017
3537
                                <1>
3538 00008B34 72C9
                                <1>
                                          jс
                                                short loc rmdir unlink error retn
                                                short loc_rmdir_del_short_name_ld_chk_fclust
3539 00008B36 EB07
                                <1>
                                          jmp
                                 <1>
3541
                                 <1> loc_rmdir_del_short_name_load_sub_dir:
                                          call load_FAT_sub_directory
3542 00008B38 E875320000
                                 <1>
                                          ;jc loc_file_rw_cmd_failed
3543
                                <1>
3544
                                          ; 29/12/2017
                                 <1>
3545 00008B3D 72C0
                                 <1>
                                                short loc_rmdir_unlink_error_retn
3546
                                 <1>
3547
                                 <1> loc_rmdir_del_short_name_ld_chk_fclust:
3548 00008B3F 0FB73D[14630100]
                                 <1>
                                          movzx edi, word [RmDir DirEntryOffset]
3549 00008B46 81C700000800
                                           add edi, Directory_Buffer
                                 <1>
3550
                                 <1>
                                                ax, [edi+20] ; First Cluster High Word
3551 00008B4C 668B4714
                                 <1>
                                          mov
                                                 eax, 16
3552 00008B50 C1E010
                                <1>
                                          shl
                                                ax, [edi+26] ; First Cluster Low Word
3553 00008B53 668B471A
                                <1>
                                          mov
3554
                                <1>
                                          ; Not necessary...
3555 00008B57 3B05[E8620100]
                                <1>
                                           cmp eax, [DelFile_FCluster]
3556 00008B5D 75A1
                                <1>
                                                short loc_rmdir_delete_short_name_invalid_data
                                          jne
3557
                                 <1>
3558 00008B5F C607E5
                                 <1>
                                          mov byte [edi], OE5h; 'Deleted' sign
                                          ; 27/02/2016
3559
                                 <1>
3560
                                 <1>
                                          ; TRDOS v1 has a bug here! it does not set
3561
                                 <1>
                                          ; 'DirBuff_ValidData' to 2; as result of this bug,
                                           ; 'save_directory_buffer' would not save the change !
3562
                                 <1>
3563 00008B62 C605[AC600100]02
                                          mov byte [DirBuff_ValidData], 2; change sign
                                 <1>
3564
                                 <1>
3565 00008B69 E8AE1D0000
                                 <1>
                                          call save_directory_buffer
                                          ;jc loc_file_rw_cmd_failed
                                 <1>
3566
3567
                                 <1>
                                           ; 29/12/2017
3568 00008B6E 728F
                                 <1>
                                           jc short loc_rmdir_unlink_error_retn
3569
                                 <1>
3570
                                 <1> loc_rmdir_del_long_name:
3571 00008B70 0FB615[EE620100]
                                <1>
                                          movzx edx, byte [DelFile_LNEL]
3572 00008B77 08D2
                                 <1>
                                           or
                                                dl, dl
3573 00008B79 7410
                                 <1>
                                                 short loc_rmdir_update_parent_dir_lmdt
3574
                                 <1>
3575 00008B7B 0FB705[EC620100]
                                 <1>
                                           movzx eax, word [DelFile_EntryCounter]
                                          sub eax, edx
3576 00008B82 29D0
                                 <1>
3577
                                 <1>
                                          ; 29/12/2017
3578 00008B84 7205
                                 <1>
                                                short loc_rmdir_update_parent_dir_lmdt
3579
                                 <1>
3580
                                 <1>
                                          ; EAX = Directory Entry Number of the long name last entry
3581 00008B86 E8EF1E0000
                                 <1>
                                          call delete_longname
3582
                                 <1>
3583
                                 <1> loc rmdir update parent dir lmdt:
3584 00008B8B E8271E0000
                                          call update_parent_dir_lmdt
                                 <1>
3585
                                 <1>
                                                short loc_file_rw_cmd_failed
                                           ;jc
3586
                                 <1>
                                          ; 29/12/2017
3587
                                 <1>
                                          ;jc short loc_rmdir_unlink_error_retn
3588
                                 <1>
                                 <1> loc_delete_sub_directory_ok:
3589
                                      ; 29<del>/</del>12/<del>2</del>017
3590
                                 <1>
3591 00008B90 31C0
                                 <1>
                                                eax, eax; 0; cf = 0
                                           xor
3592 00008B92 C3
                                 <1>
                                           retn
3593
                                 <1>
3594
                                 <1>
3595
                                 <1> delete_file:
3596
                                      ; 29/02/2016
                                 <1>
                                          ; 28/02/2016 (TRDOS 386 = TRDOS v2.0)
3597
                                 <1>
                                           ; 09/08/2010 (CMD_INTR.ASM, 'cmp_cmd_del')
3598
                                 <1>
3599
                                 <1>
                                           ; 28/02/2010
3600
                                 <1>
3601
                                 <1> get delfile fchar:
3602
                                 <1>
                                          ; esi = file name
3603 00008B93 803E20
                                 <1>
                                           cmp byte [esi], 20h
3604 00008B96 7701
                                 <1>
                                           ja short loc_delfile_parse_path_name
3605
                                 <1>
3606
                                 <1> loc_delfile_nofilename_retn:
3607 00008B98 C3
                                 <1>
                                          retn
3608
                                 <1>
3609
                                 <1> loc_delfile_parse_path_name:
                                         mov edi, FindFile Drv
3610 00008B99 BF[26620100]
                                 <1>
                                           call parse path name
3611 00008B9E E815190000
                                 <1>
                                                 loc cmd failed
3612 00008BA3 0F822EF2FFFF
                                 <1>
                                          jс
                                 <1>
3614
                                 <1> loc_delfile_check_filename_exists:
                                       mov
3615 00008BA9 BE[68620100]
                                 <1>
                                                esi, FindFile_Name
                                                 byte [esi], 20h
3616 00008BAE 803E20
                                 <1>
                                          cmp
3617 00008BB1 0F8620F2FFFF
                                 <1>
                                          jna
                                                loc_cmd_failed
3618 00008BB7 8935[E4620100]
                                 <1>
                                          mov
                                                [DelFile_FNPointer], esi
                                 <1>
                                 <1> loc_delfile_drv:
3621 00008BBD 8A15[26620100]
```

```
3622 00008BC3 8A35[86590100] <1>
                                                                       dh, [Current_Drv]
                                                             mov
3623 00008BC9 8835[E2600100]
                                                <1>
                                                              mov
                                                                        [RUN CDRV], dh
3624 00008BCF 38F2
                                               <1>
                                                              cmp
                                                                       dl, dh
3625 00008BD1 740B
                                               <1>
                                                                        short loc_delfile_change_directory
3626
                                                <1>
3627 00008BD3 E86CE3FFFF
                                                <1>
                                                               call change_current_drive
3628 00008BD8 0F8237FBFFFF
                                                                       loc_file_rw_cmd_failed
                                                <1>
3629
                                                <1>
3630
                                                <1> loc_delfile_change_directory:
                                                        cmp byte [FindFile Directory], 20h
3631 00008BDE 803D[27620100]20 <1>
3632 00008BE5 7618
                                                <1>
                                                              jna short loc_delfile_find
                                              3633
                                                <1>
3634 00008BE7 FE05[620D0100]
3635 00008BED BE[27620100]
3636 00008BF2 30E4
3637 00008BF4 E8A9120000
3638 00008BF9 0F8216FBFFFF
3639
                                                <1>
3640
                                                <1> ;loc_delfile_change_prompt_dir_string:
                                                            ;call change_prompt_dir_string
3641
                                                <1>
3642
                                                <1>
3643
                                                <1> loc_delfile_find:
                                               <1> ;mov esi, FindFile_Name
<1> mov esi, [DelFile_FNPo;
3644
3645 00008BFF 8B35[E4620100]
                                               <1>
                                                              mov esi, [DelFile_FNPointer]
                                                              mov ax, 1800h ; Except volume label and dirs
call find_first_file
3646 00008C05 66B80018
                                               <1>
                                                              mov
3647 00008C09 E8D9F6FFFF
                                               <1>
3648 00008C0E 0F8201FBFFFF
                                               <1>
                                                            jc loc_file_rw_cmd_failed
3649
                                                <1>
3650
                                                <1> loc_delfile_ambgfn_check:
3651 00008C14 6621D2
                                               <1> and dx, dx; Ambiguous filename chars used sign (DX>0)
3652 00008C17 740B
                                               <1>
                                                                       short loc_delfile_found
3653
                                                <1>
                                               <1> loc_file_not_found:
3654
                                                       mov eax, 2 ; File not found sign
3655 00008C19 B802000000
                                               <1>
3656 00008C1E F9
                                                <1>
3657 00008C1F E9F1FAFFFF
                                               <1>
                                                              jmp
                                                                      loc_file_rw_cmd_failed
3658
                                               <1>
                                                <1> loc_delfile_found:
3659
                                                        and bl, 07h; Attributes
3660 00008C24 80E307
                                                <1>
3661 00008C27 0F85F2FAFFFF
                                                              jnz loc_permission_denied
                                                <1>
3662
                                                <1>
                                                <1> ;loc delfile found save lnel:
3663
                                                <1>;
                                                            mov [DelFile_LNEL], bh ; Long name entry length (if > 0)
3664
3665
                                                <1>
                                                <1> loc_delfile_ask_for_delete:
3666
3667 00008C2D 57
                                                            push edi ; * (29/02/2016)
                                                <1>
                                                <1>
3668
3669 00008C2E BE[31120100]
                                              3670 00008C33 E895D7FFFF
3671 00008C38 8B35[E4620100]
                                                        call print_msg
mov esi, Msg_\
3672 00008C3E E88AD7FFFF
                                               <1>
3673 00008C43 BE[E5110100]
                                                <1>
                                                                       esi, Msg_YesNo
                                                             call print_msg
3674 00008C48 E880D7FFFF
                                               <1>
3675
                                               <1>
                                                <1> loc_delfile_ask_again:
3676
3677 00008C4D 30E4
                                     call int16h

call 
                                               <1> xor ah, ah
3678 00008C4F E8CB7FFFFF
3679 00008C54 3C1B
                                                             ;je short loc_do_not_delete_file
je short loc_delfile_y_n_escape
3680
                                                                       short loc_delfile_y_n_escape ; 06/03/2016
3681 00008C56 7449
3682 00008C58 24DF
                                                             mov [Y_N_nextline], al cmp al, 'Y'
3683 00008C5A A2[EF110100]
3684 00008C5F 3C59
3685 00008C61 7404
                                                                       short loc_yes_delete_file
                                                              je
3686 00008C63 3C4E
                                                <1>
                                                                       al, 'N'
                                                              cmp
3687 00008C65 75E6
                                                <1>
                                                               jne
                                                                       short loc_delfile_ask_again
3688
                                                <1>
3689
                                                <1> loc_do_not_delete_file:
3690
                                                <1> loc_yes_delete_file:
                                                <1> call y_n_answer; 29/12/2017
3691 00008C67 E8E2FBFFFF
3692 00008C6C 5F
                                                <1>
                                                              pop edi; * (29/02/2016)
                                                              ;cmp al, 'Y'; 'yes'
3693
                                                <1>
3694
                                                <1>
                                                              ; cmc
                                                              ;jncloc file rw restore retn
3695
                                               <1>
                                                             cmp al, 'N'; 'no'
  je loc_file_rw_restore_retn
3696 00008C6D 3C4E
                                                <1>
3697 00008C6F 0F84A0FAFFFF
                                               <1>
3698
                                                <1>
3699
                                                <1> loc_delete_file:
                                                        mov _ bh, [FindFile_Drv]
3700 00008C75 8A3D[26620100]
                                                <1>
                                                              ;mov bl, [DelFile_LNEL]
                                                <1>
3702 00008C7B 8A1D[75620100]
                                                              mov bl, [FindFile_LongNameEntryLength]
                                                <1>
                                                               ;mov cx, [DirBuff_EntryCounter]
3703
                                                 <1>
3704 00008C81 668B0D[A0620100]
                                                                      cx, [FindFile DirEntryNumber]
                                                <1>
                                                               mov
3705
                                                               ; (*) EDI = Directory buffer entry offset/address
                                                 <1>
3706 00008C88 E8D71F0000
                                                 <1>
                                                               call remove_file ; (FILE.ASM, 'proc_delete_file')
3707 00008C8D 0F8378FAFFFF
                                                                       loc print deleted message
                                                <1>
                                                               jnc
                                                <1>
3709
                                                <1>
                                                              ;cmp al, 05h
3710 00008C93 3C0B
                                                <1>
                                                                        al, ERR_PERM_DENIED ; 29/12/2017 (5 -> 11)
                                                               cmp
3711 00008C95 0F8484FAFFFF
                                                                        loc_permission_denied
                                               <1>
                                                               iе
3712 00008C9B F9
                                                <1>
                                                               stc
3713 00008C9C E974FAFFFF
                                                <1>
                                                               jmp
                                                                       loc_file_rw_cmd_failed
3714
                                                <1>
                                                <1> loc_delfile_y_n_escape:
3715
                                                              mov al, 'N'; 'no'
3716 00008CA1 B04E
                                                <1>
3717 00008CA3 EBC2
                                                <1>
                                                               jmp
                                                                       short loc_do_not_delete_file
                                                <1>
3719
                                                <1> set_file_attributes:
                                                           ; \overline{0}6/03/2016
3720
                                                <1>
3721
                                                              ; 04/03/2016 (TRDOS 386 = TRDOS v2.0)
                                                <1>
3722
                                                <1>
                                                            ; 10/07/2010 (TRDOS v1, CMD_INTR.ASM, 'cmp_cmd_attrib')
3723
                                                 <1>
                                                              ; 23/05/2010
3724
                                                <1>
                                                              ; 17/12/2000 (P2000.ASM)
3725
                                                 <1>
3726
                                                 <1>
                                                            ; esi = file or directory name
```

```
xor ax, ax
                              <1>
3727 00008CA5 6631C0
3728 00008CA8 66A3[82120100]
                              <1>
                                       mov
                                             [Attr Chars], ax
                                            [Attributes], al
3729 00008CAE A2[3C630100]
                              <1>
                                       mov
3730
                              <1>
3731
                              <1> get_attrib_fchar:
                              \langle 1 \rangle ; esi = file name
3732
3733 00008CB3 8A06
                                       mov al, [esi]
                              <1>
3734 00008CB5 3C20
                                       cmp al, 20h
                              <1>
                                      jna short loc_attr_file_nofilename_retn
3735 00008CB7 7623
                              <1>
3736
                              <1>
3737
                              <1> loc_scan_attrib_params:
cmp al, '-'
ja loc_att
                                             loc_attr_file_parse_path_name
                                             short loc attr space
                         cmp al, '+'
3742 00008CC3 3C2B
                                     jne loc_cmd failed
3743 00008CC5 0F850CF1FFFF
3744
                              <1> loc attr space:
3745
                          3746 00008CCB 8A6601
                            3747 00008CCE 80FC20
3748 00008CD1 770A
                                             short pass attr space
3749 00008CD3 0F82FEF0FFFF
3750 00008CD9 46
3751 00008CDA EBEF
3752
3753
3754 00008CDC C3
3755
                              <1>
3756
                              <1> pass_attr_space:
3757 00008CDD 80E4DF
                              <1> and ah, ODFh
3757 00008CDD 80E4DF
3758 00008CE0 80FC53
3759 00008CE3 0F87EEF0FFFF
3760 00008CE9 7204
                                       cmp ah, 'S' ja loc_cmd
                              <1>
                             <1>
                                             loc_cmd_failed
                             short pass attr system
                                       mov ah, 04h ; System
3761 00008CEB B404
                                      jmp short pass_attr_archive
                              <1>
3762 00008CED EB21
3763
                              <1>
3764
                              <1> pass_attr_system:
                              <1> cmp ah, 'H'
3765 00008CEF 80FC48
3766 00008CF2 7706
                              <1>
                                             short pass_attr_hidden
                                        jа
3767 00008CF4 7213
                                             short pass_attr_read_only
                              <1>
                                       jb
                                      mov ah, 02h ; Hidden jmp short pass_attr_archive
3768 00008CF6 B402
                              <1>
                             <1>
3769 00008CF8 EB16
3770
                              <1>
<1> pass_attr_hidden:
                              <1> cmp ah, 'R'
                                             loc cmd failed
                                        jа
                                       jb
                                             short pass_attr_read_only ; Read only
3775 00008D05 B401
                              <1>
                                       mov
                                             ah, 01h
3776 00008D07 EB07
                                      jmp short pass_attr_archive
                              <1>
3777
                              <1>
3778
                              <1> pass attr read only:
3779 00008D09 80FC41
                              <1> cmp ah, 'A'
3780 00008D0C 753B
                              <1>
                                        jne short loc_chk_attr_enter
                                      mov ah, 20h ; Archive
3781 00008D0E B420
                              <1>
3782
                              <1>
3783
                              <1> pass_attr_archive:
3784 00008D10 3C2D
                              <1> cmp al, '-'
3785 00008D12 7508
                              <1>
                                        jne
                                             short pass_reducing_attributes
                                       or [Attr Chars], ah
3786 00008D14 0825[82120100]
                              <1>
3787 00008D1A EB06
                              <1>
                                       jmp short loc_change_attributes_inc
3788
                              <1>
                              <1> pass_reducing_attributes:
3789
3790 00008D1C 0825[83120100]
                              <1>
                                      or [Attr_Chars+1], ah
3791
                              <1>
3792
                              <1> loc_change_attributes_inc:
3793 00008D22 46
                              <1> inc esi
                             <1>
3794 00008D23 8A6601
                                       mov
                                             ah, [esi+1]
                                             ah, 20h
3795 00008D26 80FC20
                              <1>
                                       cmp
                             <1>
3796 00008D29 7227
                                       jb
                                              short pass_change_attr
3797 00008D2B 74F5
                             <1>
                                       jе
                                              short loc_change_attributes_inc
                                             ah, '-'
3798 00008D2D 80FC2D
                              <1>
                                       cmp
3799 00008D30 770D
                              <1>
                                        jа
                                             short loc_chk_next_attr_char1
3800 00008D32 7405
                              <1>
                                        jе
                                              short loc_chk_next_attr_char0
3801 00008D34 80FC2B
                              <1>
                                             ah, '+'
                                       cmp
3802 00008D37 7506
                              <1>
                                       jne
                                             short loc_chk_next_attr_char1
3803
                              <1>
3804
                              <1> loc_chk_next_attr_char0:
                                    inc esi
3805 00008D39 46
                              <1>
3806 00008D3A 668B06
                              <1>
                                       mov ax, [esi]
3807 00008D3D EB9E
                              <1>
                                        jmp short pass_attr_space
3808
                               <1>
                               <1> loc_chk_next_attr_char1:
                                       cmp byte [esi], '-'
3810 00008D3F 803E2D
3811 00008D42 7799
                               <1>
                                        ja short pass_attr_space
3812 00008D44 E988000000
                              <1>
                                        jmp loc_attr_file_check_fname_fchar
                              <1>
3814
                              <1> loc_chk_attr_enter:
3815 00008D49 80FC0D
                              <1>
                                       cmp ah, 0Dh
3816 00008D4C 0F8585F0FFFF
                                            loc_cmd_failed
                              <1>
                                        jne
3817
                               <1>
3818
                               <1> pass_change_attr:
3819 00008D52 A0[82120100]
                              <1>
                                     mov al, [Attr_Chars]
3820 00008D57 F6D0
                              <1>
                                       not al
3821 00008D59 2005[3C630100]
                              <1>
                                       and
                                             [Attributes], al
3822 00008D5F A0[83120100]
                              <1>
                                       mov
                                             al, [Attr_Chars+1]
3823 00008D64 0805[3C630100]
                              <1>
                                      or [Attributes], al
3824
                               <1>
3825
                               <1> loc_show_attributes:
3826 00008D6A BE[FF190100]
                               <1> mov esi, nextline
                                        call print_msg
3827 00008D6F E859D6FFFF
                               <1>
3828
                               <1>
                              <1> loc_show_attributes_no_nextline:
3829
                                      mov dword [Attr Chars], 'NORM'
3830 00008D74 C705[82120100]4E4F- <1>
3830 00008D7C 524D
                               <1>
```

```
3831 00008D7E 66C705[86120100]41- <1>
                                      mov word [Attr_Chars+4], 'AL'
mov
                                              esi, Attr Chars
3833 00008D8C A0[3C630100] <1>
3834 00008D91 A804 <1>
                                        mov al, [Attributes]
                                        test al, 04h
                              <1>
3835 00008D93 7406
                                        jz
                                              short pass_put_attr_s
3836 00008D95 66C7065300
                              <1>
                                              word [esi], 0053h ; S
                                        mov
3837 00008D9A 46
                              <1>
                                      inc esi
3838
                              <1>
                             <1> pass_put_attr_s:
3839
                          <1> test al, 02h
<1> jz short pa
<1> mov word [es
3840 00008D9B A802
3841 00008D9D 7406
                                              short pass put attr h
3842 00008D9F 66C7064800
                                              word [esi], 0048h ; H
3843 00008DA4 46
                              <1>
                                      inc esi
3844
                              <1>
                              <1> pass_put_attr_h:
3845
3846 00008DA5 A801
                            <1> test al, 01h
3847 00008DA7 7406
                              <1>
                                        jz
                                             short pass_put_attr_r
3848 00008DA9 66C7065200
                              <1>
                                        mov
                                              word [esi], 0052h ; R
                                      inc esi
3849 00008DAE 46
                              <1>
3850
                              <1>
3851
                              <1> pass put attr r:
                         3852 00008DAF 3C20
3853 00008DB1 7205
                                              short pass_put_attr_a
3854 00008DB3 66C7064100
                                        mov word [esi], 0041h ; A
                              <1>
3855
                              <1> pass_put_attr_a:
call print_msg
jmp loc_file_rw_restore_retn
3861 00008DCC E944F9FFFF
                              <1>
                              <1>
3862
3863
                              <1> loc_attr_file_check_fname_fchar:
3864 00008DD1 46
                              <1> inc esi
                              <1>
3865 00008DD2 803E20
                                        cmp
                                             byte [esi], 20h
3866 00008DD5 74FA
                                        je short loc_attr_file_check fname fchar
                              <1>
                                        jb pass_change_attr
3867 00008DD7 0F8275FFFFF
                              <1>
3868
                              <1>
3869
                              <1> loc_attr_file_parse_path_name:
3870 00008DDD BF[26620100]
                              <1> mov edi, FindFile_Drv
3871 00008DE2 E8D1160000
                                        call parse path name
                               <1>
3872 00008DE7 0F82EAEFFFFF
                                        jc loc_cmd_failed
                              <1>
3873
                              <1>
3874
                               <1> loc_attr_file_check_filename_exists:
                              <1> mov esi, FindFile_Name
3875 00008DED BE[68620100]
3876 00008DF2 803E20
                                              byte [esi], 20h
                              <1>
                                        cmp
3877 00008DF5 0F86DCEFFFFF
                              <1>
                                              loc_cmd_failed
                                        jna
                                              [DelFile FNPointer], esi
3878 00008DFB 8935[E4620100]
                              <1>
                                       mov
                               <1>
3880
                               <1> loc_attr_file_drv:
                               <1> mov dh, [Current_Drv]
3881 00008E01 8A35[86590100]
3882 00008E07 8835[E2600100]
                                              [RUN\_CDRV], dh
                              <1>
                                        mov
3883
                               <1>
3884 00008E0D 8A15[26620100]
                               <1>
                                              dl, [FindFile_Drv]
                                        mov
3885 00008E13 38F2
                               <1>
                                        cmp
                                              dl, dh
3886 00008E15 740B
                                              short loc attr file change directory
                               <1>
                                        jе
3887
                               <1>
3888 00008E17 E828E1FFFF
                               <1>
                                        call change_current_drive
                                      jc loc_file_rw_cmd_failed
3889 00008E1C 0F82F3F8FFFF
                               <1>
3890
                               <1>
3891
                               <1> loc_attr_file_change_directory:
3892 00008E22 803D[27620100]20
                                        cmp byte [FindFile Directory], 20h
                               <1>
3893 00008E29 7618
                               <1>
                                        jna short loc_attr_file_find
3894
                               <1>
3895 00008E2B FE05[620D0100]
                               <1>
                                        inc
                                              byte [Restore_CDIR]
                              <1>
3897 00008E31 BE[27620100]
                               <1>
                                              esi, FindFile_Directory
                                        mov
3898 00008E36 30E4
                               <1>
                                              ah, ah ; CD_COMMAND sign -> 0
                                        xor
                                        call change current directory
3899 00008E38 E865100000
                               <1>
3900 00008E3D 0F82D2F8FFFF
                               <1>
                                              loc_file_rw_cmd_failed
3901
                               <1>
3902
                               <1> ;loc_attr_file_change_prompt_dir_string:
3903
                               <1>
                                       ;call change_prompt_dir_string
3904
                               <1>
                               <1> loc_attr_file_find:
3905
                               <1> ;mov esi, FindFile_Name
                                        mov esi, [DelFile_FNPointer]
3907 00008E43 8B35[E4620100]
                               <1>
3908 00008E49 66B80008
                               <1>
                                        mov
                                              ax, 0800h ; Except volume labels
3909 00008E4D E895F4FFFF
                               <1>
                                        call find first file
3910 00008E52 0F82BDF8FFFF
                                              loc file rw cmd failed
                               <1>
3911
                               <1>
                               <1> loc attr file ambgfn check:
3912
3913 00008E58 6609D2
                                              dx, dx; Ambiguous filename chars used sign (DX>0)
                                              (Note: It was BX in TRDOS v1)
3914
                               <1>
3915
                               <1>
                                        ;jz
                                             short loc_attr_file_found
3916 00008E5B 0F85B8FDFFFF
                               <1>
                                                loc_file_not_found ; 06/03/2016
                                         jnz
3917
                               <1>
3918
                               <1>
                                        ;mov eax, 2 ; File not found sign
3919
                               <1>
                                        ;stc
3920
                               <1>
                                        ;jmp
                                             loc_file_rw_cmd_failed
3921
                               <1>
3922
                               <1> loc_attr_file_found:
3923
                               <1>
                                       ; EDI = Directory buffer entry offset/address
3924
                               <1>
                                        ; BL = File (or Directory) Attributes
                                        ; (Note: It was 'CL' in TRDOS v1)
3925
                               <1>
                                        ; mov bl, [EDI+0Bh]
3926
                               <1>
3927
                               <1>
3928 00008E61 66833D[82120100]00 <1>
                                        cmp
                                              word [Attr_Chars], 0
                                              \verb|short loc_attr_file_change_attributes| \\
3929 00008E69 770B
                               <1>
                                        jа
3930 00008E6B 881D[3C630100]
                               <1>
                                        mov
                                               [Attributes], bl
3931 00008E71 E9F4FEFFFF
                               <1>
                                              loc_show_attributes
                                        qmŗ
3932
                               <1>
3933
                               <1> loc_attr_file_change_attributes:
3934 00008E76 A0[82120100]
                                        mov al, [Attr Chars]
                               <1>
```

```
3935 00008E7B F6D0
                               <1>
                                         not
                                               al
3936 00008E7D 20C3
                               <1>
                                         and
                                               bl, al
3937 00008E7F A0[83120100]
                               <1>
                                               al, [Attr_Chars+1]
                                         mov
3938 00008E84 08C3
                               <1>
3939
                               <1>
3940 00008E86 66817F0CA101
                               <1>
                                         cmp
                                               word [edi+DirEntry_NTRes], 01A1h ; Singlix FS
3941 00008E8C 741D
                                               short loc attr file fs check
                               <1>
                                         jе
3942
                               <1>
3943 00008E8E 881D[3C630100]
                               <1>
                                                [Attributes], bl
                                         mov
3944 00008E94 885F0B
                                <1>
                                         mov
                                               [edi+0Bh], bl ; Attributes (New!)
3945
                                <1>
3946
                                <1>
                                         ; 04/03/2016
3947
                                <1>
                                         ; TRDOS v1 has a bug here! it does not set
3948
                                <1>
                                         ; 'DirBuff_ValidData' to 2; as result of this bug,
3949
                                <1>
                                         ; 'save_directory_buffer' would not save the new attributes !
3950
                                <1>
3951 00008E97 C605[AC600100]02
                                <1>
                                               byte [DirBuff_ValidData], 2
3952
                                <1>
3953 00008E9E E8791A0000
                                <1>
                                         call save_directory_buffer
3954 00008EA3 0F826CF8FFFF
                                <1>
                                               loc_file_rw_cmd_failed
                                         jс
3955
                                <1>
3956 00008EA9 EB33
                                              short loc print attr changed message
                                <1>
                                         jmp
3957
                                <1>
3958
                                <1> loc_attr_file_fs_check:
                               <1> sub eax, eax
3959 00008EAB 29C0
3960 00008EAD 8A25[AA600100]
                               <1>
                                         mov ah, [DirBuff_DRV]
3961 00008EB3 BE00010900
                               <1>
                                         mov esi, Logical_DOSDisks
                              3962 00008EB8 01C6
3963 00008EBA 807E04A1
3964 00008EBE 7309
3965
3966 00008EC0 66B81D00
3967 00008EC4 E94CF8FFFF
3968
                               <1>
                               <1> loc_attr_file_change_fs_file_attributes:
3969
                               ; BL = New MS-DOS File Attributes
3970
3971 00008EC9 88D8
                                         mov al, bl ; File/Directory Attributes
                               <1>
                                         xor ah, ah ; Attributes in MS-DOS format sign
call change_fs_file_attributes
3972 00008ECB 30E4
                               <1>
                                         xor
3973 00008ECD E873050000
                               <1>
3974 00008ED2 0F823DF8FFFF
                               <1>
                                         jc loc_file_rw_cmd_failed
3975
                               <1>
3976 00008ED8 881D[3C630100]
                               <1>
                                        mov [Attributes], bl
3977
                                <1>
3978
                               <1> loc_print_attr_changed_message:
                                     mov esi, Msg_New call print_msg
3979 00008EDE BE[70120100]
                               <1>
3980 00008EE3 E8E5D4FFFF
                               <1>
                                         jmp loc show attributes no nextline
3981 00008EE8 E987FEFFFF
                                <1>
3982
                                <1>
                                <1> rename file:
3983
                                <1> ; 13/11/2017
3984
                                       ; 06/11/2016
3985
                                <1>
                                       ; 05/11/2016
; 16/10/2016
; 08/03/2016
3986
                                <1>
3987
                                <1>
3988
                                <1>
                                       ; 06/03/2016 (TRDOS 386 = TRDOS v2.0)
3989
                                <1>
3990
                                <1>
                                         ; 20/11/2010 (TRDOS v1, CMD_INTR.ASM, 'cmp_cmd_rename')
3991
                                <1>
                                         ; 16/11/2010
3992
                                <1>
3993
                                <1> get_rename_source_fchar:
                               \langle 1 \rangle ; esi = file name
3994
3995 00008EED 803E20
                               <1>
                                         cmp byte [esi], 20h
                                         jna short loc_rename_nofilename retn
3996 00008EF0 7614
                                <1>
3997
                               <1>
3998 00008EF2 8935[64630100]
                               <1>
                                       mov [SourceFilePath], esi
3999
                                <1>
4000
                                <1> rename_scan_source_file:
4001 00008EF8 46
                               <1> inc esi
4002 00008EF9 803E20
                               <1>
                                         cmp byte [esi], 20h
                               4003 00008EFC 7409
                                               short rename_scan_destination_file_1
                                         ;jb short loc_rename_nofilename_retn
4004
4005 00008EFE 0F82D3EEFFFF
4006 00008F04 EBF2
                                        jmp short rename_scan_source_file
4007
                               <1>
4008
                               <1> loc_rename_nofilename_retn: ; 08/03/2016
4009 00008F06 C3
                                       _ retn_
                               <1>
4010
                               <1>
4011
                               <1> rename_scan_destination_file_1:
4012 00008F07 C60600
                               <1>
                                       mov byte [esi], 0
4013
                               <1>
4014
                               <1> rename scan destination file 2:
4015 00008F0A 46
                               <1>
                                         inc esi
4016 00008F0B 803E20
                                <1>
                                         cmp
                                               byte [esi], 20h
4017 00008F0E 74FA
                                                short rename scan destination file 2
                                <1>
                                         jе
                                <1>
                                                short loc_rename_nofilename_retn
4018
                                         ;jb
4019 00008F10 0F82C1EEFFFF
                                <1>
                                         jb
                                                loc cmd failed
4020
                                <1>
4021 00008F16 8935[68630100]
                                <1>
                                                [DestinationFilePath], esi
                                         mov
4022
                                <1>
4023
                                <1> rename_scan_destination_file_3:
4024 00008F1C 46
                                <1>
                                         inc
                                              esi
4025 00008F1D 803E20
                                               byte [esi], 20h
                                <1>
                                         cmp
4026 00008F20 77FA
                                <1>
                                         jа
                                               short rename_scan_destination_file_3
4027
                                <1>
4028 00008F22 C60600
                                <1>
                                               byte [esi], 0
4029
                                <1>
                                <1> loc_rename_save_current_drive:
4030
4031 00008F25 8A35[86590100]
                                <1>
                                         mov dh, [Current Drv]
                                               byte [RUN_CDRV], dh
4032 00008F2B 8835[E2600100]
                                <1>
                                         mov
4033
                                <1>
                                <1> loc_rename_sf_parse_path_name:
4034
4035 00008F31 8B35[64630100]
                                <1>
                                       mov esi, [SourceFilePath]
4036 00008F37 BF[26620100]
                                <1>
                                         mov
                                               edi, FindFile Drv
4037 00008F3C E877150000
                                         call parse_path_name
                                <1>
4038 00008F41 0F8290EEFFFF
                                <1>
                                         jc loc_cmd_failed
                                <1>
```

```
<1> loc_rename_sf_check_filename_exists:
4040
                                               esi, FindFile Name
4041 00008F47 BE[68620100]
                                 <1>
                                          mov
4042 00008F4C 803E20
                                                 byte [esi], 20h
                                 <1>
                                          cmp
4043 00008F4F 0F8682EEFFFF
                                 <1>
                                                loc_cmd_failed
                                 <1>
4044
4045
                                 <1>
                                           ;mov
                                                [DelFile_FNPointer], esi
4046
                                 <1>
4047
                                 <1> loc_rename_sf_drv:
4048
                                 <1>
                                          ;mov dh, [Current_Drv]
                                          ;mov [RUN_CDRV], dh
4049
                                 <1>
4050
                                 <1>
4051 00008F55 8A15[26620100]
                                 <1>
                                                dl, [FindFile Drv]
                                          mov
4052 00008F5B 38F2
                                 <1>
                                                dl, dh ; dh = [Current_Drv]
                                          cmp
4053 00008F5D 740B
                                 <1>
                                                 short rename_sf_change_directory
4054
                                 <1>
                                          call change current drive
4055 00008F5F E8E0DFFFFF
                                 <1>
                                          jс
4056 00008F64 0F82ABF7FFFF
                                 <1>
                                                 loc file rw cmd failed
4057
                                 <1>
4058
                                 <1> rename_sf_change_directory:
4059 00008F6A 803D[27620100]20
                                       cmp byte [FindFile_Directory], 20h
                                <1>
                                                short rename_sf_find
4060 00008F71 7618
                                 <1>
                                 <1>
4062 00008F73 FE05[620D0100]
                                <1>
                                          inc
                                                byte [Restore_CDIR]
4063 00008F79 BE[27620100]
                                 <1>
                                          mov esi, FindFile_Directory
4064 00008F7E 30E4
                                          xor
                                 <1>
                                                ah, ah ; CD_COMMAND sign -> 0
4065 00008F80 E81D0F0000
                                 <1>
                                           call
                                                change_current_directory
4066 00008F85 0F828AF7FFFF
                                 <1>
                                                 loc_file_rw_cmd_failed
                                          jс
4067
                                 <1>
                                 <1> ;rename_sf_change_prompt_dir_string:
4068
                                          ;call change_prompt_dir_string
4069
                                 <1>
4070
                                 <1>
4071
                                 <1> rename sf find:
                                        ;mov esi, [DelFile FNPointer]
4072
                                 <1>
4073 00008F8B BE[68620100]
                                 <1>
                                          mov esi, FindFile Name
4074
                                 <1>
                                                ax, 0800h; Except volume labels
4075 00008F90 66B80008
                                 <1>
                                          mov
4076 00008F94 E84EF3FFFF
                                          call find first file
                                 <1>
4077 00008F99 0F8276F7FFFF
                                 <1>
                                          jc loc_file_rw_cmd_failed
4078
                                 <1>
4079
                                 <1> loc_rename_sf_ambgfn_check:
4080 00008F9F 6621D2
                                 <1>
                                        and dx, dx; Ambiguous filename chars used sign (DX>0)
4081
                                 <1>
                                          ;
                                                 (Note: It was BX in TRDOS v1)
4082
                                                 short loc_rename_sf_found
                                 <1>
                                          ;jz
4083 00008FA2 0F8571FCFFFF
                                 <1>
                                          jnz loc_file_not_found
4084
                                 <1>
4085
                                 <1>
                                           ;mov eax, 2 ; File not found sign
4086
                                 <1>
                                          ;stc
4087
                                 <1>
                                          ;jmp loc_file_rw_cmd_failed
4088
                                 <1>
4089
                                 <1> loc_rename_sf_found:
4090
                                 <1>
                                          ; EDI = Directory buffer entry offset/address
4091
                                 <1>
                                          ; BL = File (or Directory) Attributes
                                                 (Note: It was 'CL' in TRDOS v1)
4092
                                 <1>
4093
                                 <1>
                                          ; mov bl, [EDI+0Bh]
                                 <1>
4094
4095 00008FA8 F6C307
                                 <1>
                                          test bl, 07h; Attributes, S-H-R
4096 00008FAB 0F856EF7FFFF
                                 <1>
                                          jnz loc_permission_denied
4097
                                 <1>
4098 00008FB1 BE[26620100]
                                 <1>
                                          mov
                                                     esi, FindFile_Drv
                                                  edi, SourceFile_Drv
4099 00008FB6 BF[6C630100]
                                 <1>
4100 00008FBB B920000000
                                 <1>
                                          mov ecx, 32
4101 00008FC0 F3A5
                                 <1>
                                          rep movsd
4102
                                 <1>
4103
                                 <1> loc_rename_df_parse_path_name:
                                <1> mov esi, [DestinationFilePath]
4104 00008FC2 8B35[68630100]
4105 00008FC8 BF[26620100]
                                 <1>
                                          mov
                                                 edi, FindFile_Drv
4106 00008FCD E8E6140000
                                 <1>
                                          call parse path name
4107 00008FD2 7219
                                 <1>
                                                 short loc_rename_df_cmd_failed
                                         jс
4108
                                 <1>
                                          ;mov dh, [RUN_CDRV]
4109
                                 <1>
                                                 dh, [Current Drv]
4110 00008FD4 8A35[86590100]
                                 <1>
4111
                                 <1>
                                          ; 'rename' command is valid only for same dos drive and same dir!
4112
                                 <1>
                                          ; ('move' command must be used if source file and destination file
4113
                                 <1>
4114
                                          ; directories are not same!)
                                 <1>
4115 00008FDA 8A15[26620100]
                                          mov dl, [FindFile Drv]
                                 <1>
4116 00008FE0 38F2
                                 <1>
                                                dl, dh; are source and destination drives different ?!
                                           cmp
4117 00008FE2 7509
                                 <1>
                                           jne short loc_rename_df_cmd_failed ; yes!
4118
                                 <1>
4119
                                 <1> rename_df_check_dirname_exists:
4120 00008FE4 803D[27620100]00
                                           cmp byte [FindFile_Directory], 0
                                 <1>
4121 00008FEB 760B
                                 <1>
                                                 short rename_df_check_filename_exists
                                           jna
4122
                                 <1>
4123
                                           ; different source file and destination file directories !
                                 <1>
4124
                                 <1> loc rename df cmd failed:
                                                eax, 1; TRDOS 'Bad command or file name' error
4125 00008FED B801000000
                                 <1>
                                          mov
4126 00008FF2 F9
                                 <1>
                                           stc
4127 00008FF3 E91DF7FFFF
                                 <1>
                                           jmp
                                                loc_file_rw_cmd_failed
                                 <1>
4128
4129
                                 <1> rename_df_check_filename_exists:
                                                 esi, FindFile Name
4130 00008FF8 BE[68620100]
                                 <1>
                                           mov
4131 00008FFD E8A3F6FFFF
                                 <1>
                                           call check filename
4132 00009002 0F82BFF7FFFF
                                 <1>
                                           jс
                                                 loc_mkdir_invalid_dir_name_chars
                                 <1>
4134
                                 <1>
                                                 [DelFile FNPointer], esi
                                           ; mov
                                                 byte [esi], 20h
4135
                                 <1>
                                           ;cmp
                                                 short loc rename df find
4136
                                 <1>
                                           ;ja
4137
                                 <1>
4138
                                 <1>
                                                dh, [Current_Drv] ; dh has not been changed
4139
                                 <1>
4140
                                 <1> rename_df_drv_check_writable:
4141 00009008 0FB6F6
                                 <1>
                                           movzx esi, dh
4142
                                 <1>
                                           ;movzx esi, byte [Current_Drv]
4143 0000900B 81C600010900
                                 <1>
                                           add esi, Logical_DOSDisks
                                 <1>
4144
```

```
dl, dh ; dl = [Current Drv]
4145 00009011 88F2
                                <1>
                                          mov
4146 00009013 8A7601
                                <1>
                                         mov
                                                dh, [esi+LD DiskType]
                                <1>
4148 00009016 80FE01
                               <1>
                                          cmp dh, 1; 0 = Invalid
4149 00009019 7310
                                <1>
                                          jnb short rename_df_compare_sf_df_name
4150
                                <1>
                                          ; 16/10/2016 (13h -> 30)
4151
4152 0000901B B81E000000 <1>
4153 00009020 8B1D[68630100] <1>
                                          mov eax, 30 ; 'Disk write-protected' error
                                                ebx, [DestinationFilePath]
                                          mov
                              <1>
4154 00009026 E9EAF6FFFF
                                          jmp loc_file_rw_cmd_failed
4155
                                <1>
4156
                                <1> rename_df_compare_sf_df_name:
4157 0000902B BE[68620100]
4158 00009030 BF[AE630100]
4159 00009035 B90C000000
                               <1> mov esi, FindFile Name
                               <1>
                                                edi, SourceFile_Name
                               <1> mov ecx, 12
<1> rename_df_compare_sf_df_name_next:
4160
                            <1> lodsb <1> scasb <1> jne <1> or
4161 0000903A AC
4162 0000903B AE
4163 0000903C 7506
                                          jne short loc_rename_df_find
4164 0000903E 08C0
                                                al, al
                                      jz short loc_rename_df_cmd_failed
4165 00009040 74AB
                               <1>
4166 00009042 E2F6
                                <1>
                                         loop rename_df_compare_sf_df_name_next
                                <1>
4167
                                <1> loc_rename_df_find:
4168
                                      ;mov esi, [DelFile_FNPointer]
4169
                                <1>
4170 00009044 BE[68620100]
                                <1>
                                          mov
                                                esi, FindFile_Name
                                <1>
4172 00009049 6631C0
                                         xor ax, ax ; Any
call find_first_file
                                <1>
4173 0000904C E896F2FFFF
                                <1>
                                         ;jnc short loc_rename_df_found
4174
                                <1>
                                        ; 29/12/2017
4175
                                <1>
4176 00009051 0F83C8F6FFFF
                                <1>
                                        jnc loc_permission_denied
4177
                                <1>
4178
                                <1> loc_rename_df_check_error_code:
                                      ;cmp eax, 2
4179
                                <1>
                                                al, 2 ; Not found error
4180 00009057 3C02
                                <1>
                                          cmp
4181 00009059 7406
                               <1>
                                          jе
                                                short rename_df_move_find_struct_to_dest
4182 0000905B F9
                                <1>
                                          stc
                                          jmp loc file rw cmd failed
4183 0000905C E9B4F6FFFF
                                <1>
                                <1>
4184
                                <1> ;loc rename df found:
4185
                                      ; 05/11/2016
4186
                                 <1>
4187
                                <1>
                                          ; Permission denied error
4188
                                 <1>
                                          ;mov eax, ERR_PERM_DENIED ; 29/12/2017
                                         ;stc
4189
                                 <1>
                                         ;jmp loc_permission_denied ; 06/11/2016
4190
                                <1>
4191
                                <1>
                                <1> rename_df_move_find_struct_to_dest:
4192
                                     mov esi, FindFile_Drv
mov edi, DestinationFile_Drv
4193 00009061 BE[26620100]
                                <1>
4194 00009066 BF[EC630100]
                                <1>
4195 0000906B B92000000
                                <1>
                                         mov ecx, 32
4196 00009070 F3A5
                                <1>
                                        rep movsd
4197
                                <1>
4198
                                <1> loc_rename_df_process_q_sf:
4199
                                <1> ; mov ecx, 12
4200 00009072 B10C
                                <1>
                                          mov
                                                cl, 12
                                         mov esi, SourceFile_Name
4203
                                <1> rename_df_process_q_nml_1_sf:
                               <1> lodsb
4204 0000907E AC
                                          cmp al, 20h
4205 0000907F 3C20
                               <1>
4206 00009081 7603
                                <1>
                                           jna short rename_df_process_q_nml_2_sf
4207 00009083 AA
                                <1>
                                          stosb
4208 00009084 E2F8
                                <1>
                                          loop rename_df_process_q_nml_1_sf
4209
                                <1>
4210
                                <1> rename_df_process_q_nml_2_sf:
4211 00009086 C60700
                                <1>
                                        mov byte [edi], 0
4212
                                <1>
4213
                                <1> loc_rename_df_process_q_df:
                               <1> ;mov ecx, 12 <1> mov cl, 12
4214
4215 00009089 B10C
                                          mov cl, 12
                                      mov esi, DestinationFile_Name
mov edi, Rename_NewName
4216 0000908B BE[2E640100]
                                <1>
4217 00009090 BF[C2120100]
                                <1>
4218
                                <1> rename_df_process_q_nml_1_df:
4219 00009095 AC
                                      lodsb
                                <1>
4220 00009096 3C20
                                <1>
                                          cmp al, 20h
4221 00009098 7603
                               <1>
                                          jna short loc_rename_df_process_q_nml_2_df
4222 0000909A AA
                                          stosb
                                <1>
4223 0000909B E2F8
                                <1>
                                          loop rename_df_process_q_nml_1_df
4224
                                <1>
                                 <1> loc_rename_df_process_q_nml_2_df:
4225
4226 0000909D C60700
                                 <1>
                                          mov byte [edi], 0
4227
                                 <1>
                                 <1> loc_rename_confirmation_question:
4228
4229 000090A0 BE[89120100]
                                 <1>
                                          mov esi, Msg_DoYouWantRename
4230 000090A5 E823D3FFFF
                                <1>
                                          call print_msg
                                <1>
                                                al, [SourceFile_DirEntry+11] ; Attributes
4232 000090AA A0[C9630100]
                                <1>
                                          mov
4233 000090AF 2410
                                <1>
                                                 al, 10h
4234 000090B1 750C
                                <1>
                                                short rename_confirmation_question_dir
                                          jnz
4235
                                 <1>
                                 <1> rename confirmation_question_file:
4236
                                          mov esi, Rename_File
4237 000090B3 BE[A0120100]
                                <1>
4238 000090B8 E810D3FFFF
                                          call print_msg
                                <1>
4239 000090BD EB0A
                                 <1>
                                          jmp short rename_confirmation_question_as
4240
                                 <1>
                                 <1> rename confirmation question dir:
4242 000090BF BE[A6120100]
                                          mov esi, Rename Directory
                                <1>
                                          call print_msg
4243 000090C4 E804D3FFFF
                                <1>
4244
                                <1>
4245
                                <1> rename_confirmation_question_as:
4246 000090C9 BE[B1120100]
                                <1>
                                          mov esi, Rename OldName
                                          call print_msg
4247 000090CE E8FAD2FFFF
                                <1>
4248 000090D3 BE[BE120100]
                                <1>
                                                esi, Msg_File_rename_as
                                <1>
                                          call print_msg
4249 000090D8 E8F0D2FFFF
```

```
4250 000090DD BE[E5110100] <1>
                                             mov esi, Msg_YesNo
4251 000090E2 E8E6D2FFFF
                                  <1>
                                             call print msg
                                  <1>
4253
                                  <1> loc_rename_ask_again:
4254 000090E7 30E4
                                  <1>
                                         xor ah, ah
4255 000090E9 E8317BFFFF
                                  <1>
                                             call
                                                   int16h
4256 000090EE 3C1B
                             dal, 1Bh

<p
                                  <1>
                                             cmp al, 1Bh
4257 000090F0 740F
                                                    short loc_do_not_rename_file
4258 000090F2 24DF
4259 000090F4 A2[EF110100]
4260 000090F9 3C59
4261 000090FB 7404
                                                    short loc_yes_rename_file
                                                    al, 'N'
4262 000090FD 3C4E
                                             cmp
                                                    short loc rename ask again
4263 000090FF 75E6
                                  <1>
4264
                                   <1>
                                   <1> loc_do_not_rename_file:
4265
                                   <1> loc_yes_rename_file:
                                             call y_n_answer; 29/12/2017
4267 00009101 E848F7FFFF
                                   <1>
                                              ;cmp al, 'Y'; 'yes'
4268
                                   <1>
4269
                                   <1>
                                             ; cmc
4270
                                   <1>
                                              ;jnc loc_file_rw_restore_retn
4271 00009106 3C4E
                                             cmp al, 'N'; 'no'
                                   <1>
                                           je loc_file_rw_restore_retn
4272 00009108 0F8407F6FFFF
                                   <1>
                                   <1>
4274 0000910E BE[C2120100]
                                   <1>
                                            mov esi, Rename_NewName
4275 00009113 668B0D[E6630100]
                                  <1>
                                             mov
                                                    cx, [SourceFile_DirEntryNumber]
4276 0000911A 66A1[D2630100]
                                  <1>
                                             mov ax, [SourceFile_DirEntry+20] ; First Cluster, HW
                                           shl
4277 00009120 C1E010
                                   <1>
                                                    eax, 16 : 13/11/2017
4278 00009123 66A1[D8630100]
                                   <1>
                                             mov ax, [SourceFile_DirEntry+26] ; First Cluster, LW
                                         movzx ebx, byte [SourceFile_LongNameEntryLength]
call rename directory entry
4279
                                   <1>
4280 00009129 0FB61D[BB630100]
                                   <1>
                                             call rename_directory_entry
jmp loc_rename_file_ok
4281 00009130 E8CB1B0000
                                   <1>
4282 00009135 E9FBF6FFFF
                                   <1>
4283
                                   <1> ;loc_rename_file_ok:
                                   <1> ;     jc     loc_run_cmd_failed
<1> ;     mov     esi, Msg_OK
4284
4285
                                   <1>;
                                              mov
                                             call proc printmsg
4286
                                   <1>;
4287
                                   <1>;
                                             jmp loc_file_rw_restore_retn
4288
                                   <1>
4289
                                   <1> move_file:
4290
                                   <1> ; 11/03/2016
4291
                                   <1>
                                             ; 09/03/2016
                                   <1>
                                             ; 08/03/2016 (TRDOS 386 = TRDOS v2.0)
4292
                                           ; 21/05/2011 (TRDOS v1, CMD_INTR.ASM, 'cmp_cmd_move')
4293
                                   <1>
                                           ; 23/04/2011
4294
                                   <1>
4295
                                   <1>
                                   <1> get move source fchar:
                                         ; esi = \overline{\text{file}} name
4297
                                   <1>
                                              cmp byte [esi], 20h
4298 0000913A 803E20
                                   <1>
                                             jna short loc_move_nofilename_retn
4299 0000913D 7614
                                   <1>
4300
                                   <1>
4301 0000913F 8935[64630100]
                                   <1>
                                             mov [SourceFilePath], esi
                                   <1>
4302
4303
                                   <1> move_scan_source_file:
                                   <1> inc esi
4304 00009145 46
4305 00009146 803E20
                                  <1>
                                              cmp
                                                    byte [esi], 20h
4306 00009149 7409
                                  <1>
                                             je short move scan destination 1
                                             ;jb short loc_move_nofilename_retn
4307
                                  <1>
4308 0000914B 0F8286ECFFFF
                                   <1>
                                             jb
                                                    loc_cmd_failed
                                             jmp short move_scan_source_file
4309 00009151 EBF2
                                  <1>
4310
                                   <1>
4311
                                   <1> loc move nofilename retn:
4312 00009153 C3
                                   <1>
                                             retn
4313
                                   <1>
4314
                                   <1> move_scan_destination_1:
4315 00009154 C60600
                                   <1>
                                            mov byte [esi], 0
4316
                                  <1>
4317
                                  <1> move_scan_destination_2:
4318 00009157 46
                                   <1>
                                             inc
4319 00009158 803E20
                                  <1>
                                                    byte [esi], 20h
                                              cmp
4320 0000915B 74FA
                                   <1>
                                              jе
                                                     short move_scan_destination_2
4321
                                   <1>
                                             ;jb
                                                    short loc_move_nofilename_retn
4322 0000915D 0F8274ECFFFF
                                   <1>
                                                    loc_cmd_failed
                                             jb
4323
                                   <1>
4324 00009163 8935[68630100]
                                   <1>
                                             mov
                                                    [DestinationFilePath], esi
4325
                                   <1>
4326
                                   <1> move_scan_destination_3:
                                   <1> inc esi
4327 00009169 46
4328 0000916A 803E20
                                   <1>
                                                    byte [esi], 20h
                                              cmp
4329 0000916D 77FA
                                   <1>
                                             ja
                                                    short move scan destination 3
4330 0000916F C60600
                                   <1>
                                             mov byte [esi], 0
4331
                                   <1>
                                   <1> loc_move_scan_destination_OK:
4333 00009172 8B35[64630100]
                                   <1>
                                             mov esi, [SourceFilePath]
4334 00009178 8B3D[68630100]
                                   <1>
                                                    edi, [DestinationFilePath]
                                             mov
4335
                                   <1>
                                                   al, 1 ; move procedure Phase 1
4336 0000917E B001
                                   <1>
                                             mov
4337 00009180 E8F71B0000
                                   <1>
                                             call move_source_file_to_destination_file
4338 00009185 7328
                                   <1>
                                                    short move_source_file_to_destination_question
                                             jnc
4339
                                   <1>
4340
                                   <1> loc_move_cmd_failed_1:
4341 00009187 08C0
                                   <1>
                                             or
                                                    al, al
4342 00009189 0F8448ECFFFF
                                                    loc_cmd_failed
                                   <1>
                                              jz
4343 0000918F 3C11
                                   <1>
                                              cmp
                                                    al, 11h
4344 00009191 740D
                                   <1>
                                                    short loc_msg_not_same_device
                                              jе
4345
                                   <1>
                                              ;cmp
                                                   al, 05h
4346
                                   <1>
                                             ;cmp al, ERR PERM DENIED ; 29/12/2017
4347
                                             ;jne loc_run_cmd_failed
                                   <1>
                                                    loc_permission denied
4348
                                   <1>
                                              ;jmp
4349 00009193 3C0B
                                                     al, ERR PERM DENIED
                                   <1>
                                              cmp
4350 00009195 0F8484F5FFFF
                                   <1>
                                             jе
                                                    loc_permission_denied
4351 0000919B E962ECFFFF
                                   <1>
                                                    loc_run_cmd_failed
                                             jmp
4352
                                   <1>
                                              ;mov esi, Msg Permission denied
4353
                                   <1>
4354
                                   <1>
                                              ;call print msg
```

```
4355
                                 <1>
                                           ;jmp loc_file_rw_restore_retn
4356
                                 <1>
                                 <1> loc_msg_not_same_device:
4357
4358 000091A0 BE[CF120100]
                                <1>
                                          mov esi, msg_not_same_drv
4359 000091A5 E823D2FFFF
                                <1>
                                           call print_msg
4360 000091AA E966F5FFFF
                                <1>
                                           jmp loc_file_rw_restore_retn
                                <1>
                                <1> move_source_file_to_destination_question:
4362
4363 000091AF A0[6C630100]
                                <1>
                                          mov al, [SourceFile_Drv]
4364 000091B4 0441
                                          add al, 'A'
                                <1>
4365 000091B6 A2[31130100]
                                          mov [msg_source_file_drv], al
                                <1>
4366 000091BB A0[EC630100]
                                <1>
                                          mov al, [DestinationFile_Drv]
                                          add al, 'A'
4367 000091C0 0441
                                <1>
4368 000091C2 A2[50130100]
                                <1>
                                          mov
                                                [msg_destination_file_drv], al
4369
                                 <1>
4370 000091C7 57
                                <1>
                                        push edi; *
4371
                                <1>
4372 000091C8 BE[15130100]
                                <1>
                                                esi, msg_source_file
                                          mov
4373 000091CD E8FBD1FFFF
                                <1>
                                          call print_msg
                                          mov esi, SourceFile Directory
4374 000091D2 BE[6D630100]
                                <1>
                                          cmp byte [esi], 20h
4375 000091D7 803E20
                                <1>
4376 000091DA 7605
                                                short msftdfq sfn
                                <1>
                                          jna
                                          call print_msg
4377 000091DC E8ECD1FFFF
                                <1>
                                <1> msftdfq_sfn:
                                <1> mov esi, SourceFile_Name
4379 000091E1 BE[AE630100]
4380 000091E6 E8E2D1FFFF
                                <1>
                                          call print_msg
                               call print_msg
<1> mov esi, msg_destination_file
<1> call print_msg
<1> mov esi, DestinationFile_Direct
<1> cmp byte [esi], 20h
<1> jna short msftdfq_dfn
<1> call print msg
4381 000091EB BE[34130100]
4382 000091F0 E8D8D1FFFF
4383 000091F5 BE[ED630100]
                                                 esi, DestinationFile_Directory
4384 000091FA 803E20
4385 000091FD 7605
4386 000091FF E8C9D1FFFF
                                <1>
                                          call print msg
                                <1> msftdfq_dfn:
4387
4388 00009204 BE[2E640100]
                                <1> mov esi, DestinationFile_Name
                                          call print_msg
4389 00009209 E8BFD1FFFF
                                <1>
                               4390 0000920E BE[53130100]
                                                esi, msg_copy_nextline
4391 00009213 E8B5D1FFFF
4392 00009218 BE[53130100]
                                        call print_msg
4393 0000921D E8ABD1FFFF
                                <1>
4394
                                <1>
4395
                                <1> loc_move_ask_for_new_file_yes_no:
                                <1> mov esi, Msg_DoYouWantMoveFile
<1> call print_msg
4396 00009222 BE[E1120100]
4397 00009227 E8A1D1FFFF
4398 0000922C BE[E5110100]
                                <1>
                                          mov esi, Msg_YesNo
4399 00009231 E897D1FFFF
                                <1>
                                         call print msg
                                <1> loc_move_ask_for_new_file_again:
4400
4401 00009236 30E4
                                <1> xor ah, ah
4402 00009238 E8E279FFFF
                                <1>
                                          call int16h
4403 0000923D 3C1B
                                <1>
                                          cmp
                                                al, 1Bh
                                          ;je short loc_do_not_move_file
                                <1>
                                     je
and
4405 0000923F 7441
                                <1>
                                                 short loc_move_y_n_escape
4406 00009241 24DF
                                <1>
                                                al, ODFh
                                          mov [Y_N_nextline], al cmp al, 'Y'
4407 00009243 A2[EF110100]
                                <1>
4408 00009248 3C59
                                <1>
4409 0000924A 7404
                                 <1>
                                                 short loc_yes_move_file
                                          jе
4410 0000924C 3C4E
                                 <1>
                                           cmp
                                                 al, 'N'
4411 0000924E 75E6
                                 <1>
                                          jne short loc_move_ask_for_new_file_again
4412
                                 <1>
4413
                                 <1> loc_do_not_move_file:
4414
                                 <1> loc_yes_move_file:
4415 00009250 E8F9F5FFFF
                                <1>
                                        call y_n_answer ; 29/12/2017
                                          pop edi; *
;cmp al, 'Y'; 'yes'
4416 00009255 5F
                                 <1>
4417
                                <1>
4418
                                <1>
4419
                                <1>
                                          ;jncloc_file_rw_restore_retn
                                          cmp al, 'N'; 'no'
je loc_file_rw_restore_retn
4420 00009256 3C4E
                                <1>
4421 00009258 0F84B7F4FFFF
                                <1>
4422
                                <1>
4423
                                <1> loc_move_yes_move_file:
4424 0000925E B002
                                <1> mov al, 2; move procedure Phase 2
4425 00009260 E8171B0000
                                 <1>
                                           call move_source_file_to_destination_file
                                           ;jc short loc_move_cmd_failed 2
                                 <1>
4427 00009265 0F83D0F5FFFF
                                           jnc move_source_file_to_destination_OK
                                 <1>
4428
                                 <1>
4429
                                 <1> ;move_source_file_to_destination_OK:
                                          mov esi, Msg OK
4430
                                 <1>;
4431
                                 <1>;
                                           call print msg
                                          jmp loc file rw restore retn
4432
                                 <1>;
4433
                                 <1>
4434
                                 <1> loc move cmd failed 2:
4435 0000926B 3C27
                                 <1>
                                          cmp al, 27h
                                                 loc run cmd failed
4436 0000926D 0F858FEBFFFF
                                 <1>
                                           jne
4437
                                 <1>
4438 00009273 BE[FA120100]
                                 <1>
                                                 esi, msg insufficient disk space
4439 00009278 E850D1FFFF
                                 <1>
                                           call print_msg
4440
                                 <1>
4441 0000927D E993F4FFFF
                                <1>
                                                loc_file_rw_restore_retn
                                           qmţ
4442
                                <1>
                                 <1> loc_move_y_n_escape:
4443
                                          mov al, 'N'; 'no'
4444 00009282 B04E
                                 <1>
4445 00009284 EBCA
                                 <1>
                                                short loc_do_not_move_file
4446
                                 <1>
                                 <1> copy_file:
4447
                                        ; 15/10/2016
4448
                                 <1>
4449
                                 <1>
                                          ; 24/03/2016
                                          ; 21/03/2016
4450
                                 <1>
                                          ; 15/03/2016 (TRDOS 386 = TRDOS v2.0)
4451
                                 <1>
4452
                                 <1>
                                          ; 21/05/2011 (TRDOS v1, CMD_INTR.ASM, 'cmp_cmd_copy')
4453
                                 <1>
                                          ; 01/08/2010
4454
                                 <1>
4455
                                 <1> get_copy_source_fchar:
4456
                                 <1>
                                          ; esi = file name
4457 00009286 803E20
                                 <1>
                                           cmp byte [esi], 20h
4458 00009289 7614
                                 <1>
                                           jna short loc_copy_nofilename_retn
4459
                                 <1>
```

```
4460 0000928B 8935[64630100]
                                        mov [SourceFilePath], esi
                              <1>
4461
                                <1>
                              <1> copy_scan_source_file:
4462
4463 00009291 46
                              <1>
                                        inc esi
                                         cmp byte [esi], 20h
je short copy_scan_
4464 00009292 803E20
                               <1>
4465 00009295 7409
                               <1>
                                               short copy_scan_destination_1
                                    ;jb short loc_copy_
jb loc_cmd_failed
                                        ;jb short loc_copy_nofilename_retn
                               <1>
4467 00009297 0F823AEBFFFF
                              <1>
                                       jmp short copy_scan_source_file
4468 0000929D EBF2
                               <1>
                               <1>
                               <1> loc_copy_nofilename_retn:
4470
4471 0000929F C3
                               <1>
                                        retn
4472
                               <1>
4473
                               <1> copy_scan_destination_1:
                              <1>
                                       mov byte [esi], 0
4474 000092A0 C60600
4475
                               <1>
                              <1> copy_scan_destination_2:
4477 000092A3 46
                                     inc esi
                               <1>
4478 000092A4 803E20
                               <1>
                                               byte [esi], 20h
                                         cmp
4479 000092A7 74FA
                                         je
                               <1>
                                              short copy_scan_destination_2
4480
                               <1>
                                        ;jb short loc_copy_nofilename_retn
4481 000092A9 0F8228EBFFFF
                                        jb
                               <1>
                                               loc_cmd_failed
                               <1>
4483 000092AF 8935[68630100]
                               <1>
                                       mov [DestinationFilePath], esi
4484
                               <1>
4485
                               <1> copy_scan_destination_3:
4486 000092B5 46
                               <1> inc esi
4487 000092B6 803E20
                              <1>
                                              byte [esi], 20h
                                         cmp
                                               short copy_scan_destination 3
4488 000092B9 77FA
                               <1>
                                         jа
                                       mov byte [esi], 0
4489 000092BB C60600
                              <1>
4490
                               <1>
                               <1> loc_copy_save_current_drive:
4492 000092BE 8A35[86590100]
                               <1> mov dh, [Current_Drv]
4493 000092C4 8835[E2600100]
                              <1>
                                              [RUN_CDRV], dh
                                        mov
4494
                               <1>
4495
                               <1> copy_source_file_to_destination_phase_1:
4496 000092CA 8B35[64630100] <1>
                                        mov esi, [SourceFilePath]
4497 000092D0 8B3D[68630100]
                              <1>
                                         mov
                                              edi, [DestinationFilePath]
4498
                               <1>
                                        mov al, 1 ; copy procedure Phase 1
4499 000092D6 B001
                               <1>
4500 000092D8 E83C1D0000
                                      call copy_source_file_to_destination_file
                               <1>
                                       jnc short copy_source_file_to_destination_question
4501 000092DD 732B
                               <1>
4502
                               <1>
                               <1> loc copy cmd failed 1:
4503
                                     \overline{18/03/2016} (restore current drive and directory)
4504
                               <1>
4505 000092DF 08C0
                               <1>
                                         or al, al
4506 000092E1 7507
                               <1>
                                        jnz short loc_copy_cmd_failed_2
4507
                               <1>
4508 000092E3 FEC0
                               <1>
                                                al ; mov al, 1 ; Bad command or file name !
4509 000092E5 E918EBFFFF
                              <1>
                                         jmp loc_run_cmd_failed
4510
                               <1>
4511
                               <1> loc_copy_cmd_failed_2:
                                      cmp al, 27h; Insufficient disk space
4512 000092EA 3C27
                               <1>
4513 000092EC 740D
                               <1>
                                         je short loc_file_write_insuff_disk_space_msg
4514
                               <1>
4515
                               <1>
                                         ; 29/12/2017
                               <1>
                                        ;cmp al, 05h
                                              al, ERR_PERM_DENIED
4517 000092EE 3C0B
                               <1>
                                         cmp
4518 000092F0 0F850CEBFFFF
                               <1>
                                         jne loc_run_cmd_failed
                               <1>
4519
4520 000092F6 E924F4FFFF
                               <1>
                                         jmp loc_permission_denied
4521
                               <1>
4522
                               <1> loc_file_write_insuff_disk_space_msg:
4523 000092FB BE[FA120100]
                               <1>
                                       mov esi, msg_insufficient_disk_space
                                         call print_msg
4524 00009300 E8C8D0FFFF
                               <1>
4525 00009305 E90BF4FFFF
                               <1>
                                         jmp loc_file_rw_restore_retn
4526
                               <1>
4527
                               <1> copy_source_file_to_destination_question:
4528 0000930A 57
                               <1>
                                        push edi; *
4529
                               <1>
4530
                               <1>
                                         ; dh = source file attributes
                                         ; dl > 0 -> destination file found
4531
                               <1>
4532 0000930B 20D2
                                         and dl, dl
                               <1>
                                             short copy_source_file_to_destination_pass_owrq
4533 0000930D 7449
                               <1>
                               <1>
4534
                               <1> loc_copy_ask_for_owr_yes_no:
4535
4536 0000930F BE[56130100]
                              <1> mov esi, Msg_DoYouWantOverWriteFile
                                         call print_msg
4537 00009314 E8B4D0FFFF
                               <1>
                                              esi, DestinationFile Name
4538 00009319 BE[2E640100]
                               <1>
                                         mov
4539 0000931E E8AAD0FFFF
                               <1>
                                        call print msg
4540 00009323 BE[E5110100]
                               <1>
                                        mov esi, Msg YesNo
4541 00009328 E8A0D0FFFF
                                <1>
                                         call
                                              print_msg
4542
                                <1>
4543
                                <1> loc_copy_ask_for_owr_again:
4544 0000932D 30E4
                                <1>
                                        xor ah, ah
                                         call int16h
4545 0000932F E8EB78FFFF
                               <1>
4546 00009334 3C1B
                               <1>
                                         cmp al, 1Bh
                                        ;je
4547
                               <1>
                                                  loc do not copy file
                                                   short loc_copy_y_n_escape
4548 00009336 7419
                               <1>
                                           jе
                                         and al, ODFh
4549 00009338 24DF
                               <1>
4550 0000933A A2[EF110100]
                               <1>
                                         mov
                                                [Y_N_nextline], al
                                              al, 'Y'
4551 0000933F 3C59
                                <1>
                                         cmp
4552 00009341 0F84B1000000
                               <1>
                                               loc_yes_copy_file
                                         jе
                                               al, 'N'
4553 00009347 3C4E
                               <1>
                                         cmp
4554 00009349 0F84A9000000
                                               loc_do_not_copy_file
                                <1>
                                         jе
4555 0000934F EBDC
                                <1>
                                         jmp
                                               short loc_copy_ask_for_owr_again
                                <1>
                                <1> loc_copy_y_n_escape:
4557
                                         mov al, 'N'; 'no'
4558 00009351 B04E
                                <1>
4559 00009353 E9A0000000
                               <1>
                                         jmp loc_do_not_copy_file
4560
                               <1>
4561
                               <1> copy_source_file_to_destination_pass_owrq:
                                              al, [SourceFile_Drv]
4562 00009358 A0[6C630100]
                               <1>
                                        mov
4563 0000935D 0441
                               <1>
                                         add
                                              al, 'A'
4564 0000935F A2[31130100]
                               <1>
                                         mov
                                              [msg_source_file_drv], al
```

```
al, [DestinationFile_Drv]
 4566 00009369 0441
                                                         <1>
                                                                          add al, 'A'
 4567 0000936B A2[50130100]
                                                        <1>
                                                                          mov
                                                                                     [msg_destination_file_drv], al
                                                       4568
 4569 00009370 BE[15130100]
                                                                          mov esi, msg_source_file
call print_msg
 4570 00009375 E853D0FFFF
 4571 0000937A BE[6D630100]
                                                                          mov esi, SourceFile Directory
 4572 0000937F 803E20
                                                <1> jna short csft
<1> call print_msg
 4573 00009382 7605
 4574 00009384 E844D0FFFF
                                                        <1> csftdfq_sfn:
 4575
4576 00009389 BE[AE630100]

4577 0000938E E83AD0FFFF

4578 00009393 BE[34130100]

4579 00009398 E830D0FFFF

4580 0000939D BE[ED630100]

4581 000093A2 803E20

4582 000093A5 7605
                                                         <1> mov esi, SourceFile_Name
<1> call print_msg
                                                        <1>
                                                       <1> call print_msg
<1> mov esi, msg_destination_file
<1> call print_msg
<1> mov esi, DestinationFile_Direct
<1> cmp byte [esi], 20h
<1> jna short csftdfq_dfn
<1> call print_msg
                                                                                     esi, DestinationFile Directory
 4582 000093A5 7605
 4583 000093A7 E821D0FFFF
4584

4585 000093AC BE[2E640100]

4586 000093B1 E817D0FFFF

4587 000093B6 BE[53130100]

4588 000093BB E80DD0FFFF

4589 000093C0 BE[53130100]

4580 000093C5 E803D0FFFF
                                                        <1> csftdfq_dfn:
                                                        <1> mov esi, DestinationFile_Name
                                                                         call print_msg
mov esi, msg_copy_nextline
                                                         <1>
                                                        <1>
                                                        call print_msg
copy_nextline
call print_msg
call print_msg
call print_msg
                                                        <1>
                                                         <1> loc_copy_ask_for_new_file_yes_no:
 4592
 4593 000093CA BE[75130100]
4594 000093CF E8F9CFFFFF
4595 000093D4 BE[E5110100]
4596 000093D9 E8EFCFFFFF
                                                        <1> mov esi, Msg_DoYouWantCopyFile
<1> call print_msg
                                                       <1>
                                                        <1>
                                                                          mov esi, Msg_YesNo
                                                         <1>
                                                                       call print_msg
                                                        <1>
 4597
 4598
                                                        <1> loc_copy_ask_for_new_file_again:
4598
4599 000093DE 30E4
4600 000093E0 E83A78FFFF
4601 000093E5 3C1B
4602 000093E7 740F
4603 000093E9 24DF
4604 000093EB A2[EF110100]
4605 000093F0 3C59
4606 000093F2 7404
4607 000093F4 3C4E
4609 000093F6 75F6

41 loc_copy_ask_for_new_file_again:
41 call int16h
42 call int16h
43 int16h
44 call int16h
45 call int16h
46 ca
                                                                                     short loc_do_not_copy_file
                                                                       jne short loc copy ask for new file again
 4608 000093F6 75E6
                                                        <1>
 4609
                                                         <1>
                                                         <1> loc_do_not_copy_file:
 4610
 4611
                                                        <1> loc_yes_copy_file:
                                                    <1>
 4612 000093F8 E851F4FFFF
                                                                    call y_n_answer ; 29/12/2017
                                                                          pop edi; *
;cmp al, 'Y'; 'yes'
 4613 000093FD 5F
                                                         <1>
                                                        <1>
 4614
                                                                   ;cmp
 4615
                                                        <1>
                                                                          ;jnc loc_file_rw_restore_retn
cmp al, 'N'; 'no'
 4616
                                                         <1>
 4617 000093FE 3C4E
                                                         <1>
 4618 00009400 0F840FF3FFFF
                                                        <1>
                                                                          je loc_file_rw_restore_retn
 4619
                                                         <1>
 4620
                                                         <1> copy_source_file_to_destination_pass_q:
                                                         <1>
 4621 00009406 B002
                                                                          mov al, 2 ; copy procedure Phase 2
 4622 00009408 E80C1C0000
                                                         <1>
                                                                           call copy_source_file_to_destination_file
 4623
                                                         <1>
                                                                          ;jc short loc_file_write_check_disk_space_err
 4624
                                                         <1>
 4625
                                                         <1>
                                                                          ; 24/03/2016
                                                         <1>
                                                                          ;push cx
 4627 0000940D 51
                                                         <1>
                                                                          push ecx ; 29/12/2017
 4628 0000940E BE[53130100]
                                                         <1>
                                                                          mov esi, msg_copy_nextline
 4629 00009413 E8B5CFFFFF
                                                         <1>
                                                                          call print_msg
 4630 00009418 58
                                                          <1>
                                                                                     eax ; 29/12/2017
                                                                          pop
 4631
                                                          <1>
                                                                          ;;pop cx
 4632
                                                         <1>
                                                                          ;pop ax
 4633
                                                         <1>
 4634
                                                         <1>
                                                                          ;or cl, cl
 4635 00009419 08C0
                                                         <1>
                                                                                     short copy_source_file_to_destination_OK
 4636 0000941B 7419
                                                         <1>
                                                                          jΖ
                                                         <1>
 4637
 4638
                                                         <1>
                                                                          ; 15/10/2016 (1Dh -> 18)
 4639
                                                         <1>
                                                                          ; 18/03/2016 (1Dh)
 4640
                                                         <1>
                                                                           ;cmp cl, 18 ; write error
 4641 0000941D 3C12
                                                         <1>
                                                                           cmp
                                                                                     al, 18
 4642 0000941F 7506
                                                         <1>
                                                                           jne
                                                                                    short copy_source_file_to_destination_not_OK
 4643
                                                         <1>
                                                         <1>
 4644
                                                                           ;mov al, cl; error number (write fault!)
 4645 00009421 F9
                                                         <1>
                                                                           stc
 4646 00009422 E9EEF2FFFF
                                                                                      loc file rw cmd failed
                                                          <1>
                                                                           jmp
 4647
                                                          <1>
 4648
                                                          <1> copy_source_file_to_destination_not_OK:
                                                                                    esi, Msg_read_file_error_before_EOF
 4649 00009427 BE[8E130100]
                                                          <1>
                                                                           mov
 4650 0000942C E89CCFFFFF
                                                          <1>
                                                                           call
                                                                                    print_msg
 4651 00009431 E9DFF2FFFF
                                                          <1>
                                                                                    loc_file_rw_restore_retn
                                                                           jmp
 4652
                                                          <1>
 4653
                                                          <1> copy_source_file_to_destination_OK:
 4654 00009436 BE[F3110100]
                                                          <1>
                                                                           mov esi, Msg OK
 4655 0000943B E88DCFFFFF
                                                          <1>
                                                                           call print_msg
                                                           <1>
 4657 00009440 E9D0F2FFFF
                                                          <1>
                                                                           jmp
                                                                                      loc_file_rw_restore_retn
 4658
                                                          <1>
                                                           <1> ;loc_file_write_check_disk_space_err:
 4659
                                                                           ;cmp al, 27h ; Insufficient disk space
 4660
                                                          <1>
                                                                           ;je loc file write_insuff_disk_space_msg
 4661
                                                           <1>
                                                          <1>
                                                                              ;jb loc_file_rw_cmd_failed
 4662
 4663
                                                          <1>
                                                          <1>
 4664
                                                                           ; call print misc error msg ; 15/03/2016
 4665
                                                          <1>
                                                                              ;jmp loc_file_rw_restore_retn
 4666
                                                           <1>
                                                          <1> change_fs_file_attributes:
 4667
 4668
                                                          <1>
                                                                          ; 04/03/2016 ; Temporary
                                                                           ; AL = File or directory attributes
                                                          <1>
 4669
```

4565 00009364 A0[EC630100]

<1>

mov

```
4670
                                <1>
                                         ; AH = 0 -> Attributes are in MS-DOS format
4671
                                <1>
                                         ; AH > 0 -> Attributes are in SINGLIX format
                                         ;push ebx
4672
                                <1>
4673
                                <1>
                                         ; ... do somethings here ...
4674
                                <1>
                                         ;pop ebx
4675
                                <1>
                                         ; BL = File or directory attributes
4676 00009445 C3
                                <1>
                                <1>
4677
4678
                                <1> set_get_env:
4679
                                <1>
                                         ; 11/04/2016 (TRDOS 386 = TRDOS v2.0)
                                         ; 02/09/2011 (TRDOS v1, CMD_INTR.ASM, 'cmp_cmd_set')
4680
                                <1>
4681
                                         ; 2005 - 28/08/2011
                                <1>
                                <1> get_setenv_fchar:
4682
4683
                                <1>
                                       ; esi = environment variable/string
                                         mov al, [esi]
cmp al, 20h
4684 00009446 8A06
                               <1>
4685 00009448 3C20
                               <1>
4686 0000944A 771E
                               <1>
                                         jа
                                               short loc_find_env
4687
                               <1>
4688 0000944C BE00300900
                                        mov esi, Env Page
                               <1>
4689
                               <1> loc_print_setline:
4690 00009451 803E00
                               <1> cmp byte [esi], 0
4691 00009454 7613
                               <1>
                                         jna
                                               short loc setenv retn
                          <1>
                                         call print_msg
4692 00009456 E872CFFFFF
                                     push esi
mov esi, nextline
4693 0000945B 56
                               <1>
4694 0000945C BE[FF190100]
                               <1>
4695 00009461 E867CFFFFF
                               <1>
                                         call print_msg
4696 00009466 5E
                               <1>
                                         pop esi
4697 00009467 EBE8
                               <1>
                                        jmp
                                               short loc_print_setline
4698
                               <1>
4699
                               <1> loc_setenv_retn:
4700 00009469 C3
                               <1>
                                         retn
4701
                                <1>
                               <1> loc_find_env:
4702
                                     cmp al, '='
4703 0000946A 3C3D
                               <1>
4704 0000946C 0F8465E9FFFF
                               <1>
                                               loc_cmd_failed
4705
                               <1>
4706 00009472 56
                               <1>
                                        push esi
4707
                               <1> loc_repeat_env_equal_check:
4708 00009473 46
                               <1>
                                         inc
                                               esi
4709 00009474 803E3D
                               <1>
                                               byte [esi], '='
                                         cmp
4710 00009477 7431
                               <1>
                                         jе
                                               short pass_env_equal_check
4711 00009479 803E20
                               <1>
                                        cmp
                                               byte [esi], 20h
                               <1>
4712 0000947C 73F5
                                         jnb
                                               short loc_repeat_env_equal_check
4713 0000947E C60600
                               <1>
                                         mov
                                               byte [esi], 0
                                         pop
4714 00009481 5E
                               <1>
                                               esi
                               <1>
4715 00009482 BF[865A0100]
                                         mov
                                               edi, TextBuffer ; out buffer
4716 00009487 B9FF000000
                                               ecx, 255; maximum size (limit)
                               <1>
                                         mov
                                         xor
4717 0000948C 30C0
                               <1>
                                              al, al ; 0 -> use [ESI]
4718 0000948E E89E000000
                               <1>
                                         call get_environment_string
                                         jс
4719 00009493 72D4
                               <1>
                                               short loc_setenv_retn
4720
                               <1>
4721 00009495 BE[865A0100]
                               <1>
                                         mov
                                               esi, TextBuffer
                                         call print_msg
4722 0000949A E82ECFFFFF
                               <1>
4723 0000949F BE[FF190100]
                               <1>
                                         mov
                                               esi, nextline
4724 000094A4 E824CFFFFF
                                <1>
                                         call print_msg
4725
                                <1>
4726 000094A9 C3
                               <1>
                                         retn
4727
                                <1>
4728
                                <1> pass_env_equal_check:
                               <1> inc esi
4729 000094AA 46
4730 000094AB 803E20
                               <1>
                                         cmp
                                               byte [esi], 20h
4731 000094AE 73FA
                               <1>
                                         jnb
                                               short pass_env_equal_check
                                         mov byte [esi], 0
4732 000094B0 C60600
                               <1>
4733
                               <1>
                               <1> loc_call_set_env_string:
4734
                                     pop esi
4735 000094B3 5E
                               <1>
4736 000094B4 E83B010000
                                         call set environment string
                               <1>
4737 000094B9 73AE
                               <1>
                                         jnc short loc_setenv_retn
4738
                               <1>
                               <1> loc_set_cmd_failed:
4739
4740 000094BB 3C08
                               <1>
                                     cmp al, 08h
4741 000094BD 0F8514E9FFFF
                               <1>
                                              loc_cmd_failed
                                         jne
4742
                               <1>
4743 000094C3 BE[CE130100]
                               <1>
                                         mov esi, Msg_No_Set_Space
                                         call print msg
4744 000094C8 E800CFFFFF
                                <1>
4745
                                <1>
4746 000094CD C3
                                <1>
                                         retn
4747
                                <1>
                                <1> set_get_path:
4748
                                        \frac{1}{1000}; \frac{1}{1000} (TRDOS 386 = TRDOS v2.0)
4749
                                <1>
4750
                                <1>
                                         ; 03/09/2011 (TRDOS v1, CMD_INTR.ASM, 'cmp_cmd_path')
4751
                                <1>
4752
                                <1> get_path_fchar:
4753
                                <1>
                                        ; esi = path
4754 000094CE 803E20
                                <1>
                                         cmp byte [esi], 20h
4755 000094D1 7737
                                <1>
                                         jа
                                               short loc_set_path
                               <1>
4757 000094D3 BE00300900
                               <1>
                                         mov esi, Env_Page
4758
                                <1> loc_print_path:
                                        cmp byte [esi], 0
4759 000094D8 803E00
                               <1>
4760 000094DB 762C
                                <1>
                                               short loc_path_retn
4761
                                <1>
4762 000094DD BE[2E0E0100]
                                                esi, Cmd_Path ; 'PATH' address
                                <1>
                                         mov
                                               edi, TextBuffer; out buffer
4763 000094E2 BF[865A0100]
                                <1>
                                       mov
4764 000094E7 30C0
                                <1>
                                        xor
                                               al, al ; use [ESI]
4765 000094E9 B9FF000000
                                               ecx, 255; maximum size (limit)
                                <1>
                                         mov
4766 000094EE E83E000000
                                <1>
                                         call get environment string
4767 000094F3 7214
                                <1>
                                                short loc_path_retn
                                       jс
4768
                                <1>
4769 000094F5 BE[865A0100]
                                <1>
                                               esi, TextBuffer
                                        mov
4770 000094FA E8CECEFFFF
                                <1>
                                        call print_msg
4771 000094FF BE[FF190100]
                                <1>
                                         mov
                                               esi, nextline
                                         call print_msg
4772 00009504 E8C4CEFFFF
                                <1>
4773
                                <1>
4774
                                <1> loc_path_retn:
```

```
<1>
                                          retn
4776
                                <1>
                                <1> loc_set_path:
4777
4778 0000950A 56
                               <1> push esi
                                <1> loc_set_path_find_end:
4779
4780 0000950B 46
                               <1> inc esi
4781 0000950C 803E20
                              <1>
                                          cmp byte [esi], 20h
                            <1> jnb short lo
<1> mov byte [es
<1> loc_set_path_header:
4782 0000950F 73FA
                                                short loc_set_path_find_end
4783 00009511 C60600
                                                byte [esi], 0
4785 00009514 5E
                              <1> pop esi
4786
                                <1> set path x: ; 31/12/2017 ('syspath')
                               <1> dec esi
4787 00009515 4E
4788 00009516 C6063D
                              <1>
                                          mov byte [esi], '='
                                     mov byte
dec esi
mov byte
dec esi
mov byte
4789 00009519 4E
                                <1>
                             <1>
4790 0000951A C60648
                                                byte [esi], 'H'
4791 0000951D 4E
                              <1>
                                <1>
4792 0000951E C60654
                                                byte [esi], 'T'
4793 00009521 4E
                                <1>
                                          dec
                                                esi
                                          mov
4794 00009522 C60641
                               <1>
                                                byte [esi], 'A'
                                <1>
4795 00009525 4E
                                          dec
                                                esi
                                                byte [esi], 'P'
4796 00009526 C60650
                                <1>
                                         mov
4797
                                <1>
4798
                                <1> loc_path_call_set_env_string:
4799 00009529 E8C6000000
                                <1> call set_environment_string
4800 0000952E 728B
                                <1>
                                          jc short loc_set_cmd_failed
                                <1>
4802 00009530 C3
                                <1>
                                          retn
4803
                                 <1>
4804
                                 <1> get_environment_string:
4805
                                 <1>
                                       ; 12/04/2016
4806
                                 <1>
                                          ; 11/04/2016
                                          ; 05/04/2016 (TRDOS 386 = TRDOS v2.0)
4807
                                 <1>
4808
                                 <1>
                                        ; 02/09/2011 (TRDOS v1, MAINPROG.ASM)
                                        ; 28/08/2011
4809
                                 <1>
4810
                                 <1>
                                          ; INPUT->
                                                EDI = Output buffer
4811
                                 <1>
                                          ;
4812
                                 <1>
                                        ;
                                                CX = Buffer length (<= ENV_PAGE_SIZE)</pre>
4813
                                 <1>
4814
                                 <1>
                                                AL > 0 = AL = String sequence number
4815
                                 <1>
                                                AL = 0 \rightarrow ESI = ASCIIZ Set word
4816
                                 <1>
                                         ;
                                                       (environment variable)
                                          ; OUTPUT ->
                                 <1>
4817
4818
                                 <1>
                                                ESI is not changed
                                                EDI is not changed
4819
                                 <1>
4820
                                 <1>
                                                EAX = String length (with zero tail)
4821
                                 <1>
                                                EDX = Environment variables page address
                                 <1>
                                                CF = 1 \rightarrow Not found (EAX not valid)
4822
                                          ;
4823
                                 <1>
4824
                                 <1>
                                          ; (Modified registers: EAX, EDX)
4825
                                 <1>
4826 00009531 BA00300900
                                 <1>
                                          mov
                                                edx, Env Page
4827 00009536 803A00
                                <1>
                                          cmp
                                                byte [edx], 0
4828 00009539 7474
                                <1>
                                                short get_env_string_with_word_stc_retn
                                 <1>
4830 0000953B 66890D[F0640100]
                                <1>
                                          mov
                                                [env_var_length], cx
                                 <1>
                                          push ecx; *
4832 00009542 51
                                <1>
4833 00009543 56
                                <1>
                                          push esi; **
                                <1>
4835 00009544 08C0
                                <1>
                                          or
                                                al, al
4836 00009546 7449
                                <1>
                                          jz
                                                short get_env_string_with_word
4837
                                <1>
4838
                                <1> get_env_string_with_seq_number:
                                         \overline{\text{mov}} cl, 1
4839 00009548 B101
                                <1>
4840 0000954A 88C5
                                <1>
                                          mov
                                                ch, al
                                          xor eax, eax
4841 0000954C 31C0
                               <1>
4842 0000954E 89D6
                                <1>
                                          mov esi, edx ; Env_Page
4843
                                <1>
4844
                                <1> get_env_string_seq_number_check:
4845 00009550 38CD
                                <1>
                                        cmp ch, cl
4846 00009552 7726
                                <1>
                                          jа
                                                short get_env_string_seq_number_next
4847
                                <1>
4848
                                <1> get_env_string_move_to_buff:
4849 00009554 57
                                <1>
                                         push edi; ***
4850
                                <1>
4851 00009555 29D2
                                <1>
                                          sub edx, edx
4852
                                <1>
4853
                                 <1> get_env_string_seq_number_repeat1:
4854 00009557 42
                                         inc edx
                                 <1>
4855 00009558 AC
                                 <1>
                                          lodsb
4856 00009559 AA
                                 <1>
                                          stosb
                                <1>
4858 0000955A 66FF0D[F0640100]
                                <1>
                                                 word [env_var_length]
4859 00009561 7508
                                <1>
                                          jnz short get_env_string_seq_number_repeat3
4860
                                <1>
                               <1> get_env_string_seq_number_repeat2:
4862 00009563 20C0
                              <1>
                                          and al, al
4863 00009565 7408
                                <1>
                                          jz
                                                short get_env_string_seq_number_ok
4864 00009567 42
                               <1>
                                          inc
                                                edx
                               <1>
4865 00009568 AC
                                          lodsb
                                <1>
4866 00009569 EBF8
                                          jmp short get_env_string_seq_number_repeat2
4867
                               <1>
                               <1> get_env_string_seq_number_repeat3:
4868
4869 0000956B 08C0
                                <1>
                                          or
                                                al, al
                               <1>
4870 0000956D 75E8
                                          jnz
                                                short get_env_string_seq_number_repeat1
                               <1>
4872
                                <1> get_env_string_seq_number_ok:
                                      pop edi; ***
4873 0000956F 5F
                                <1>
                                                eax, edx; Length of the environment string
                               <1>
4874 00009570 89D0
                                          mov
4875
                                <1>
                                                       ; (ASCIIZ, includes ZERO tail)
4876 00009572 BA00300900
                               <1>
                                               edx, Env_Page
                                          mov
4877
                                <1>
4878
                                <1> get_env_string_stc_retn:
4879 00009577 5E
                                <1>
                                        pop esi; **
```

4775 00009509 C3

```
4880 00009578 59
                              <1>
                                        pop
                                              ecx ; *
4881 00009579 C3
                              <1>
                                        retn
4882
                              <1>
4883
                              <1> get_env_string_seq_number_next:
                              <1>
<1>
4884 0000957A AC
                                        lodsb
4885 0000957B 08C0
                                        or
                                              al, al
4886 0000957D 75FB
                                              short get_env_string_seq_number_next
                              <1>
                                        jnz
4887
                              <1>
4888 0000957F 81FE00320900
                              <1>
                                             esi, Env_Page + Env_Page_Size ; +512 (+4096)
4889 00009585 F5
                              <1>
                                        cmc
4890 00009586 72EF
                              <1>
                                              short get_env_string_stc_retn
4891
                              <1>
4892 00009588 AC
                              <1>
                                        lodsb
                           <1>
4893 00009589 3C01
                                       cmp al, 1
4894 0000958B 72EA
                              <1>
                                              short get_env_string_stc_retn
                                        iЬ
4895 0000958D FEC1
                              <1>
                                        inc
                                              cl
4896 0000958F EBBF
                              <1>
                                        jmp short get_env_string_seq_number_check
4897
                              <1>
4898
                              <1> get_env_string_with_word:
                                       xor ecx, ecx
4899 00009591 31C9
                              <1>
4900
                              <1>
                              <1> get env string calc word length:
4901
4902 00009593 AC
                              <1>
                                       lodsb
                           <1>
<1>
<1>
4903 00009594 3C20
                                        cmp
                                             al, 20h
                              <1>
4904 00009596 7211
                                              short get_env_string_calc_word_length_ok
                                        jb
                           <1>
4905
                                        ;inc cx
4906 00009598 FEC1
                                       inc
4907
                              <1>
                           4908 0000959A 3C61
                              <1>
                                             al, 'a'
                                        cmp
4909 0000959C 72F5
                                        jb
                                              short get_env_string_calc_word_length
4910 0000959E 3C7A
                                        cmp al, 'z'
4911 000095A0 77F1
                                              short get_env_string_calc_word_length
                                        jа
                                        and
                                             al, ODFh
4912 000095A2 24DF
                           <1>
<1>
<1>
4913 000095A4 8846FF
                                        mov [esi-1], al
4914 000095A7 EBEA
                              <1>
                                      jmp short get_env_string_calc_word_length
4915
                              <1>
4916
                              <1> get_env_string_calc_word_length_ok:
4917 000095A9 08C9
                              <1> or cl, cl
4918 000095AB 7506
                              <1>
                                             short get_env_string_calc_word_length_save
4919
                              <1>
4920 000095AD 5E
                              <1>
                                       pop esi; **
4921
                              <1>
4922
                              <1> get_env_string_stc_retn1:
4923 000095AE 59
                              <1>
                                      pop ecx; *
4924
                              <1>
4925
                              <1> get_env_string_with_word_stc_retn:
                          <1> c
<1>
4926 000095AF 31C0
                                    xor eax, eax
                              <1>
4927 000095B1 F9
                                        stc
4928 000095B2 C3
                              <1>
                                        retn
                         4930
4931 000095B3 871C24
4932 000095B6 89DE
4933
                                              ; Start of the env string (to be searched)
                              <1>
<1>
4934
                         4935 000095B8 57
                                        push edi; ***
4936 000095B9 89D7
                                       mov
                                             edi, edx ; Env_Page
4937
4938
                              <1> get_env_string_compare:
                            4939 000095BB 57
                                        push ecx; *****; Variable name length
4940 000095BC 51
4941
                              <1>
                              <1> get_env_string_compare_rep:
4942
                            <1> get
<1>
4943 000095BD AC
                                      lodsb
4944 000095BE AE
                              <1>
                                        scasb
4945 000095BF 7511
                              <1>
                                        jne short get_env_string_compare_next1
4946 000095C1 E2FA
                              <1>
                                      loop get_env_string_compare_rep
4947
                              <1>
4948 000095C3 803F3D
                              <1>
                                              byte [edi], '='
                                        cmp
4949 000095C6 750A
                              <1>
                                             short get_env_string_compare_next1
                                        jne
4950
                              <1>
4951 000095C8 59
                              <1>
                                              ecx ; *****
                                        pop
                                              edi ; ****
4952 000095C9 5F
                              <1>
                                        pop
                                              esi, edi
4953 000095CA 89FE
                              <1>
                                        mov
                                             edi ; ***
4954 000095CC 5F
                              <1>
                                        pop
4955 000095CD 871C24
                                        xchg ebx, [esp] ; **
                              <1>
4956 000095D0 EB82
                              <1>
                                        jmp short get_env_string_move_to_buff
4957
                              <1>
4958
                               <1> get_env_string_compare_next1:
4959 000095D2 89FE
                              <1> mov esi, edi
                                             ecx ; ****
4960 000095D4 59
                               <1>
                                        pop
4961 000095D5 5F
                                        pop
                                              edi ; ****
                               <1>
                               <1> get_env_string_compare_next2:
4963 000095D6 81FEFF310900
                                        cmp esi, Env Page + Env Page Size - 1; +511 (+4095)
                               <1>
4964 000095DC 7310
                               <1>
                                        jnb
                                              short get_env_string_compare_not_ok
4965 000095DE 20C0
                              <1>
                                        and
4966 000095E0 AC
                              <1>
                                        lodsb
                                      jnz
4967 000095E1 75F3
                              <1>
                                              short get_env_string_compare_next2
4968 000095E3 08C0
                              <1>
                                        or
4969 000095E5 7407
                              <1>
                                              short get_env_string_compare_not_ok
                                        jΖ
4970 000095E7 4E
                              <1>
                                        dec
                                              esi ; 12/04/2016
4971 000095E8 89F7
                              <1>
                                        mov
                                              edi, esi
4972 000095EA 89DE
                              <1>
                                        mov
                                              esi, ebx
4973 000095EC EBCD
                              <1>
                                             short get_env_string_compare
                                        jmp
4974
                               <1>
4975
                              <1> get_env_string_compare_not_ok:
4976 000095EE 5F
                              <1>
                                             edi ; ***
                                       pop
4977 000095EF 89DE
                              <1>
                                        mov
                                              esi, ebx
4978 000095F1 5B
                               <1>
                                        pop
                                              ebx ; **
4979 000095F2 EBBA
                              <1>
                                              short get_env_string_stc_retn1
                                        jmp
4980
                               <1>
4981
                               <1> set_environment_string:
                                       ; 13/04/2016
4982
                               <1>
4983
                               <1>
                                        ; 12/04/2016
4984
                               <1>
                                        ; 11/04/2016
```

```
; 06/04/2016
4985
                                 <1>
4986
                                 <1>
                                          ; 05/04/2016 (TRDOS 386 = TRDOS v2.0)
                                          ; 02/09/2011 (TRDOS v1, MAINPROG.ASM)
4987
                                 <1>
4988
                                 <1>
                                         ; 29/08/2011
                                        ; 29/08/2011
4989
                                 <1>
4990
                                 <1>
                                          ; INPUT->
4991
                                                ESI = ASCIIZ environment string
                                 <1>
                                        ; OUTPUT ->
4992
                                 <1>
4993
                                 <1>
                                                ESI is not changed
4994
                                 <1>
                                                CF = 1 \rightarrow Could not set,
4995
                                 <1>
                                                     insufficient environment space
4996
                                 <1>
                                          ; (EAX, EDX will be changed)
4997
                                 <1>
4998
                                 <1>
                                               (EAX = Start address of the env string if > 0)
4999
                                 <1>
                                               (EDX = Environment string length)
5000
                                 <1>
5001
                                 <1>
5002 000095F4 56
                                 <1>
                                          push esi; *
5003
                                <1>
5004 000095F5 31C0
                                <1>
                                          xor eax, eax
5005
                                <1>
                                <1> set_env_chk_validation1:
5006
5007 000095F7 FEC4
                                <1>
                                          inc ah ; variable (string) length
5008 000095F9 AC
                                <1>
                                          cmp al, '='
5009 000095FA 3C3D
                                <1>
                                                short set_env_chk_validation2
5010 000095FC 7415
                                <1>
                                          jе
5011 000095FE 3C20
                                <1>
                                          cmp al, 20h
5012 00009600 720F
                                <1>
                                        jb
                                                short set_env_string_stc
5013
                                <1>
                                          ; 06/04/2016
5014
                                <1>
5015 00009602 3C61
                                <1>
                                          cmp al, 'a'
5016 00009604 72F1
                                <1>
                                          jb
                                                short set_env_chk_validation1
                                                al, 'z'
5017 00009606 3C7A
                                <1>
                                          cmp
5018 00009608 77ED
                                <1>
                                                short set_env_chk_validation1
                                          jа
                                          sub al, 'a'-'A'
mov [esi-1], al
5019 0000960A 2C20
                                <1>
5020 0000960C 8846FF
                                <1>
5021 0000960F EBE6
                                          jmp short set_env_chk_validation1
                                <1>
5022
                                <1>
5023
                                <1> set_env_string_stc:
5024 00009611 5E
                                        pop esi; *
                                <1>
5025
                                <1>
                                          ;stc
5026 00009612 C3
                                <1>
                                          retn
5027
                                <1>
                                <1> set env chk validation2:
5028
                                      push ecx; **
push ebx; ***
5029 00009613 51
                                <1>
5030 00009614 53
                                <1>
                                          push edi ; ****
5031 00009615 57
                               <1>
                                <1>
5032
5033
                                <1>
                                          ; 12/04/2016
                                          mov ebx, [esp+12]
5034 00009616 8B5C240C
                               <1>
5035
                                <1>
                                <1> set_env_chk_validation2w:
5036
5037 0000961A 89F7
                                <1> mov edi, esi
5038 0000961C 4F
                                <1>
                                                 edi
5039
                                <1>
5040 0000961D 807FFF20
                                <1>
                                          cmp
                                                byte [edi-1], 20h
5041 00009621 771A
                                <1>
                                                short set env chk validation2z
                                          jа
5042
                                <1>
                                          push esi
5043 00009623 56
                                <1>
5044 00009624 89FE
                                <1>
                                          mov
                                                esi, edi
5045 00009626 4E
                                <1>
                                          dec
                                                esi
5046
                                <1>
5047
                                <1> set_env_chk_validation2x:
5048 00009627 4E
                                <1>
                                          dec
                                                esi
5049
                                <1>
5050 00009628 39DE
                                <1>
                                          cmp
                                                 esi, ebx
5051 0000962A 7207
                                <1>
                                          jb
                                                short set_env_chk_validation2y
5052
                                <1>
5053 0000962C 4F
                                <1>
                                          dec
                                                 edi
5054
                                <1>
5055 0000962D 8A06
                                <1>
                                          mov
                                                 al, [esi]
5056 0000962F 8807
                                 <1>
                                          mov
                                                 [edi], al
5057
                                <1>
5058 00009631 EBF4
                                <1>
                                                 short set_env_chk_validation2x
5059
                                <1>
                                <1> set_env_chk_validation2y:
5060
5061 00009633 5E
                                <1>
                                                esi
                                          pop
5062
                                <1>
5063
                                <1>
                                          ;mov byte [ebx], 20h
5064
                                <1>
5065 00009634 43
                                <1>
                                          inc
                                                 ebx
5066 00009635 895C240C
                                                 [esp+12], ebx
                                 <1>
                                          mov
5067
                                 <1>
5068 00009639 FECC
                                 <1>
                                                 ah ; 13/04/2016
5069
                                 <1>
5070 0000963B EBDD
                                                 short set env chk validation2w
                                 <1>
                                          jmp
                                <1>
5072
                                <1> set_env_chk_validation2z:
5073 0000963D BA00300900
                                <1>
                                                edx, Env_Page
                                          mov
5074 00009642 89D7
                                               edi, edx
                                <1>
                                          mov
5075
                                <1>
5076
                                <1> set env chk validation3:
5077 00009644 AC
                                <1>
                                          lodsb
                                          cmp al, 20h
5078 00009645 3C20
                                <1>
5079 00009647 74FB
                                <1>
                                                short set env chk validation3
5080
                                <1>
5081 00009649 9C
                                <1>
                                          pushf
5082
                                <1>
5083
                                <1>
                                          ; 12/04/2016
5084
                                <1> set_env_chk_validation3n:
5085 0000964A 3C61
                                <1>
                                          cmp
                                               al, 'a'
5086 0000964C 720C
                                <1>
                                          jb
                                                short set_env_chk_validation3c
                                                al, 'z'
                                <1>
5087 0000964E 3C7A
                                          cmp
5088 00009650 7705
                                                 short set env chk validation3x
                                <1>
                                          jа
                                                al, 'a'-'Ā'
5089 00009652 2C20
                                <1>
                                          sub
```

```
5090 00009654 8846FF
                           <1>
                                         mov [esi-1], al
5091
                               <1>
5092
                              <1> set_env_chk_validation3x:
5093 00009657 AC
                             <1>
                                         lodsb
5094 00009658 EBF0
                               <1>
                                         jmp short set_env_chk_validation3n
5095
                               <1>
                               <1> set_env_chk_validation3c:
5096
5097 0000965A 3C20
                               <1>
                                        cmp al, 20h
5098 0000965C 73F9
                               <1>
                                               short set_env_chk_validation3x
                                         jnb
                               <1>
5100 0000965E 803F00
                               <1>
                                         cmp
                                               byte [edi], 0
5101 00009661 7731
                               <1>
                                         jа
                                               short set env chk validation4
5102
                               <1>
5103 00009663 9D
                               <1>
                                         popf
5104 00009664 7228
                               <1>
                                         jb
                                               short set_env_string_nothing
5105
                               <1>
5106 00009666 B900020000
                                               ecx, Env Page Size; 512 (4096)
                               <1>
                                         mov
5107
                               <1>
                                               esi, ebx ; 12/04/2016
5108 0000966B 89DE
                               <1>
5109
                               <1>
5110
                               <1> set_env_string_copy_to_envb:
5111 0000966D AC
                               <1>
                                         lodsb
5112 0000966E 3C20
                               <1>
                                         cmp al, 20h
5113 00009670 720A
                               <1>
                                               short set_env_string_copy_to_envb_z
5114 00009672 AA
                               <1>
                                         stosb
5115 00009673 E2F8
                               <1>
                                         loop set_env_string_copy_to_envb
5116
                               <1>
                               <1>
                                         ; 11/04/2016
5117
                                         mov edi, edx ; Env_Page
mov ecx, Env_Page_Size
5118 00009675 89D7
                               <1>
5119 00009677 B900020000
                        <1>
5120
                               <1>
5121
                               <1> set_env_string_copy_to_envb_z:
5122 0000967C 52
                               <1> push edx ; Start address of the variable
                            <1>
5123 0000967D BA00020000
                                         mov edx, Env_Page_Size
5124 00009682 29CA
                               <1>
                                         sub
                                              edx, ecx; variable (string) length
5125
                               <1>
5126 00009684 28C0
                               <1>
                                              al, al ; 0
5127 00009686 F3AA
                               <1>
                                        rep stosb; clear remain bytes of the env page
5128
                               <1>
5129 00009688 58
                                         pop eax ; Start address of the variable
                               <1>
5130
                               <1>
5131
                               <1> set_env_string_allocate_envb_retn: ; stc or clc return
                               <1>
                                         pop edi; ****
5132 00009689 5F
                                              ebx ; ***
5133 0000968A 5B
                               <1>
                                         pop
5134 0000968B 59
                                              ecx ; **
                               <1>
                                         pop
5135 0000968C 5E
                               <1>
                                         pop
                                               esi ; *
5136 0000968D C3
                               <1>
                                         retn
5137
                               <1>
5138
                               <1> set_env_string_nothing:
5139 0000968E 31C0
                               <1> xor eax, eax
                                         xor edx, edx; 11/04/2016
5140 00009690 31D2
                               <1>
5141 00009692 EBF5
                               <1>
                                         jmp short set_env_string_allocate_envb_retn
                               <1>
5142
5143
                               <1> set_env_chk_validation4:
                                     ; 11/04/2016
5144
                               <1>
5145 00009694 9D
                               <1>
                                         popf
5146
                               <1>
5147 00009695 89D6
                               <1>
                                       mov esi, edx ; Env_Page
5148
                               <1>
5149
                               <1> set_env_chk_validation5:
5150 00009697 89DF
                               <1>
                                         mov edi, ebx ; ASCIIZ environment string address
5151 00009699 OFB6CC
                                         movzx ecx, ah ; Variable (string) length (with '=')
                               <1>
5152
                               <1>
5153
                               <1> set_env_chk_validation5_loop:
                                     lodsb
5154 0000969C AC
                               <1>
5155 0000969D AE
                               <1>
                                         scasb
5156 0000969E 750A
                               <1>
                                         jne short set env chk validation6
5157 000096A0 E2FA
                                         loop set_env_chk_validation5_loop
                               <1>
5158
                               <1>
5159 000096A2 3C3D
                               <1>
                                              al, '='
                                         cmp
5160 000096A4 0F8483000000
                               <1>
                                         jе
                                                 set_env_change_variable
5161
                               <1>
                               <1> set_env_chk_validation6:
5162
5163 000096AA 08C0
                               <1>
                                    or al, al ; 0
5164 000096AC 7403
                               <1>
                                               short set_env_chk_validation7
5165
                               <1>
5166 000096AE AC
                               <1>
                                         lodsb
5167 000096AF EBF9
                               <1>
                                         jmp short set_env_chk_validation6
5168
                               <1>
5169
                               <1> set_env_chk_validation7:
5170 000096B1 88E1
                                <1>
                                         mov cl, ah
5171 000096B3 01F1
                                <1>
                                         add
                                               ecx, esi
5172 000096B5 81F9FF310900
                                <1>
                                              ecx, Env_Page + Env_Page_Size - 1
5173
                                <1>
                                               ; 511 (4095)
                                               ; strlen + '=' + 0
5174
                                <1>
5175 000096BB 72DA
                                               short set_env_chk_validation5
                               <1>
                                         jb
5176
                               <1>
5177
                               <1> set_env_chk_validation8: ; variable not found
5178 000096BD 0FB6F4
                               <1>
                                         movzx esi, ah ; variable name length (with '=')
                                         add esi, ebx; position just after of the '='
5179 000096C0 01DE
                              <1>
5180
                               <1>
                               <1> set env chk validation8 loop:
5181
5182 000096C2 AC
                               <1>
                                        lodsb
5183 000096C3 3C20
                              <1>
                                              al, 20h
5184 000096C5 74FB
                               <1>
                                               short set_env_chk_validation8_loop
                                         iе
                               <1>
5185 000096C7 72C5
                                               short set_env_string_nothing
                                         jb
                               <1>
                               <1> set_env_chk_validation9:
5187
5188 000096C9 AC
                                         lodsb
                               <1>
5189 000096CA 3C20
                               <1>
                                              al, 20h
                                         cmp
5190 000096CC 73FB
                                               short set_env_chk_validation9
                               <1>
5191
                                <1>
5192
                               <1>
                                         ; End of ASCIIZ environment string
5193
                                <1>
5194
                                <1> set_env_add_variable:
```

```
5195 000096CE 29DE
                                <1>
                                          sub
                                                 esi, ebx; variable+definition length
5196
                                 <1>
5197 000096D0 56
                                               esi ; *****
                                <1>
                                          push
5198
                                <1>
5199 000096D1 89D6
                                <1>
                                                esi, edx ; Environment page address
                                          mov
5200
                                <1>
5201 000096D3 B900020000
                                                ecx, Env Page Size; 512 (4096)
                                <1>
                                          mov
5202
                                <1>
5203
                                <1> set_env_add_variable_loop:
5204 000096D8 AC
                                <1>
                                          lodsb
5205 000096D9 20C0
                                <1>
                                          and al, al
5206 000096DB 7406
                                                short set env add variable chk1 ; 0
                                <1>
                                          jz
5207 000096DD E2F9
                                <1>
                                          loop set_env_add_variable_loop
5208
                                <1>
5209
                                <1>
                                          ; 11/04/2016
                                          mov [esi-1], cl; 0
5210 000096DF 884EFF
                                <1>
5211 000096E2 41
                                <1>
                                          inc
                                                ecx
5212
                                <1>
5213
                                <1> set_env_add_variable_chk1:
5214 000096E3 49
                                <1>
                                        dec ecx
5215 000096E4 7408
                               <1>
                                                 short set_env_add_variable_nspc
5216 000096E6 AC
                                <1>
                                          lodsb
5217 000096E7 08C0
                               <1>
                                          or
                                                al, al
5218 000096E9 740C
                               <1>
                                          jz
                                                short set_env_add_variable_chk2 ; 00
5219 000096EB 49
                                <1>
                                          dec
                                                ecx
5220 000096EC 75EA
                                <1>
                                          jnz
                                                short set_env_add_variable_loop
                                <1>
5222
                                <1> set_env_add_variable_nspc: ; no space on environment page
5223 000096EE 58
                                <1>
                                               eax ; *****
                                          pop
5224 000096EF B808000000
                                <1>
                                                eax, 8; No space for new environment string
                                          mov
5225 000096F4 F9
                                <1>
                                          stc
                                          jmp
5226 000096F5 EB92
                                <1>
                                                    short set_env_string_allocate_envb_retn
5227
                                <1>
5228
                                <1> set_env_add_variable_chk2:
                                          mov ecx, [esp] ; ****
5229 000096F7 8B0C24
                                <1>
5230 000096FA 4E
                                <1>
                                          dec
                                                esi ; beginning address of the new variable
5231 000096FB 89F0
                               <1>
                                                eax, esi
                                          mov
                                                eax, ecx; string length (with CR)
5232 000096FD 01C8
                                <1>
                                          add
5233 000096FF 81C200020000
                                                edx, Env Page Size ; 512 (4096)
                                <1>
                                          add
5234 00009705 39D0
                                          cmp eax, edx
                                <1>
5235 00009707 77E5
                                <1>
                                                short set_env_add_variable_nspc
5236 00009709 49
                                <1>
                                          dec
                                                ecx ; except CR at the end
5237 0000970A 89CA
                               <1>
                                                edx, ecx ; 12/04/2016
                                          mov
5238 0000970C 89F7
                               <1>
                                                edi, esi
                                          mov
                                                [esp], edi ; ***** ; Start address of new variable
5239 0000970E 893C24
                                <1>
                                          mov
5240 00009711 89DE
                                <1>
                                          mov
                                                esi, ebx ; ASCIIZ environment string address
5241 00009713 F3A4
                               <1>
                                          rep
                                                movsb
5242 00009715 28C0
                                <1>
                                          sub al, al
5243 00009717 AA
                                <1>
                                          stosb
                                          pop eax; *****; Beginning address of new variable
5244 00009718 58
                                <1>
5245 00009719 81FF00320900
                               <1>
                                                    edi, Env_Page + Env_Page_Size ; 12/04/2016
5246 0000971F 0F8364FFFFFF
                                <1>
                                          jnb
                                                    set_env_string_allocate_envb_retn ; OK !
                                          mov [edi], cl; 0
5247 00009725 880F
                                <1>
5248 00009727 F8
                                <1>
                                          clc ; 13/04/2016
5249 00009728 E95CFFFFFF
                                <1>
                                          jmp
                                                 set_env_string_allocate_envb_retn ; OK !
5250
                                <1>
5251
                                <1> set env change variable:
                                        ; 06/04/2016
5252
                                 <1>
5253
                                 <1>
                                          ; esi = Variable's address in environment page (after '=')
5254
                                 <1>
                                          ; edi = ASCIIZ environment string address (after '=')
5255
                                 <1>
5256
                                 <1>
                                          ; ah = variable length from start to the '='
5257 0000972D 8825[F0640100]
                                <1>
                                          mov [env_var_length], ah
5258
                                <1>
5259 00009733 28C9
                                 <1>
                                          sub
                                                cl, cl; ecx = 0
5260
                                <1>
5261 00009735 57
                                          push edi ; ****
                                <1>
                                <1>
5262
5263 00009736 89F7
                                <1>
                                                edi, esi ; 11/04/2016
5264
                                <1>
                                <1> set env change variable calc1:
5265
5266 00009738 AC
                                <1>
                                          lodsb
5267 00009739 08C0
                                <1>
                                          or
                                                 al, al
5268 0000973B 7403
                                <1>
                                          jΖ
                                                 short set_env_change_variable_calc2
5269
                                <1>
5270 0000973D 41
                                                 ecx ; length of environment string (after the '=')
                                <1>
                                          inc
5271
                                <1>
5272 0000973E EBF8
                                <1>
                                          jmp
                                                short set_env_change_variable_calc1
5273
                                <1>
5274
                                <1> set env_change variable_calc2:
5275 00009740 8B3424
                                <1>
                                          mov esi, [esp] ; ASCIIZ environment string address
                                 <1>
5277 00009743 29D2
                                <1>
                                          sub
                                                edx, edx
                                <1>
5278
5279
                                <1> set_env_change_variable_calc3:
5280 00009745 AC
                                <1>
                                          lodsb
5281 00009746 3C20
                                <1>
                                                al, 20h
                                          cmp
5282 00009748 7203
                                <1>
                                                 short set_env_change_variable_calc4
5283
                                <1>
                                <1>
5284 0000974A 42
                                                edx ; length of ASCIIZ string (after the '=')
                                          inc
5285
                                <1>
5286 0000974B EBF8
                                <1>
                                                short set_env_change_variable_calc3
                                          qmp
                                <1>
5287
5288
                                <1> set_env_change_variable_calc4:
5289 0000974D C646FF00
                                <1>
                                          mov byte [esi-1], 0 ; put ZERO instead of CR
5290
                                <1>
5291 00009751 5E
                                <1>
                                          pop esi ; ***** ; ASCIIZ string address (after '=')
5292
                                <1>
5293
                                <1>
                                          ; EDI = Old variable's address (after '=')
5294
                                <1>
5295
                                <1>
                                          ; compare the new string with the old string
5296 00009752 39CA
                                <1>
                                          cmp
                                               edx, ecx
5297 00009754 7717
                                <1>
                                                short set_env_change_variable_calc5 ; longer
                                                    set env change variable calc9; shorter
5298 00009756 0F828F000000
                                <1>
                                <1>
5299
```

```
5300
                                <1>
                                          ; same length (simple copy)
5301 0000975C 0FB6C4
                                <1>
                                          movzx eax, ah
                               <1>
5302 0000975F 01C2
                                          add edx, eax
5303 00009761 F7D8
                               <1>
                                          neg eax
5304 00009763 01F8
                                <1>
                                          add
                                               eax, edi
5305
                                <1>
                                          ; EAX = Start address of the variable
                                          ; EDX = Variable length (without ZERO at the end of variable)
5306
                                <1>
5307
                                <1>
                                         rep movsb clc ; 13/04/2016
5308 00009765 F3A4
                                <1>
5309 00009767 F8
                                <1>
5310 00009768 E91CFFFFF
                                <1>
                                                 set_env_string_allocate_envb_retn ; OK !
                                <1>
                                <1> set_env_change_variable_calc5:
5312
5313
                                <1>
                                         ; 11/04/2016
5314 0000976D 52
                                          push edx ; ****
                                <1>
5315 0000976E 29CA
                                <1>
                                          sub
                                                edx, ecx; difference; (the new string is longer)
5316 00009770 89F3
                               <1>
                                          mov
                                               ebx, esi
5317 00009772 89FE
                                <1>
                                         mov esi, edi
5318
                                <1>
                                <1> set_env_change_variable_calc6:
5319
5320 00009774 AC
                                <1>
                                          lodsb
5321 00009775 20C0
                                <1>
                                          and
                                                al, al
5322 00009777 75FB
                                <1>
                                                short set_env_change_variable_calc6
                                          jnz
                                <1>
5324 00009779 81FE00320900
                                <1>
                                          cmp esi, Env_Page + Env_Page_Size ; 512 (4096)
                                          jnb
5325 0000977F 0F8369FFFFFF
                                <1>
                                                 set_env_add_variable_nspc
                                <1>
5327 00009785 89F9
                                <1>
                                          mov
                                                ecx, edi ; current (old) variable's address
5328 00009787 89F7
                                <1>
                                                edi, esi ; next variable's address
                                          mov
5329
                                <1>
5330 00009789 AC
                                <1>
                                          lodsb
5331 0000978A 08C0
                                <1>
                                          or
                                                al, al
5332 0000978C 7416
                                                short set_env_change_variable_calc8 ; 00
                                <1>
                                          jΖ
5333
                                <1>
5334
                                <1> set_env_change_variable_calc7:
5335 0000978E AC
                                <1>
                                          lodsb
5336 0000978F 20C0
                                <1>
                                          and al, al
5337 00009791 75FB
                                <1>
                                          jnz short set_env_change_variable_calc7
5338
                                <1>
5339 00009793 81FE00320900
                                <1>
                                          cmp esi, Env_Page + Env_Page_Size ; 512 (4096)
5340 00009799 0F834FFFFFF
                                          jnb set_env_add_variable_nspc
                                <1>
5341
                                <1>
5342 0000979F AC
                                <1>
                                          lodsb
                                          or al, al
5343 000097A0 08C0
                                <1>
5344 000097A2 75EA
                                <1>
                                          jnz short set_env_change_variable_calc7
5345
                                <1>
                                <1> set env change variable calc8:
5346
5347 000097A4 4E
                                <1>
                                          dec esi; address of the second (last) 0 of the 00
5348
                                <1>
5349 000097A5 01F2
                                                edx, esi; final position of the last 0
                                <1>
                                          add
5350
                                <1>
5351 000097A7 81FA00320900
                                <1>
                                          cmp
                                               edx, Env Page + Env Page Size; 512 (4096)
5352 000097AD 0F833BFFFFFF
                                <1>
                                          jnb
                                                  set_env_add_variable_nspc
5353
                                <1>
5354 000097B3 89C8
                                <1>
                                                eax, ecx ; old variable's address (after '=')
                                          mov
5355
                                <1>
5356 000097B5 89F1
                                <1>
                                                ecx, esi
                                          mov
                                                ecx, edi ; count of bytes to move forward
5357 000097B7 29F9
                                <1>
                                          sub
5358
                                <1>
5359
                                          ; 13/04/2016
                                <1>
5360 000097B9 C60200
                               <1>
                                                byte [edx], 0
                                          mov
5361 000097BC 89D7
                                <1>
                                          mov
                                                edi, edx
5362 000097BE 29F2
                               <1>
                                                edx, esi; difference (additional byte count)
                                          sub
5363 000097C0 4F
                                <1>
                                                edi ; the last zero address (first byte of the 00)
5364 000097C1 89FE
                                <1>
                                                esi, edi
                                          mov
5365 000097C3 29D6
                                <1>
                                          sub
                                                esi, edx ; - displacement
5366
                                <1>
5367 000097C5 FA
                                <1>
                                          cli
                                                ; disable interrupts
5368 000097C6 FD
                                <1>
                                                ; backward
5369
                                <1>
5370 000097C7 F3A4
                                <1>
                                                movsb ; move ECX bytes from DS:ESI to ES:EDI
5371
                                <1>
5372 000097C9 FC
                                                ; forward (default)
                                <1>
                                          cld
5373 000097CA FB
                                <1>
                                                ; enable interrupts
5374
                                <1>
5375 000097CB 89C7
                                <1>
                                          mov
                                                edi, eax
5376 000097CD 59
                                <1>
                                                ecx; *****; byte count (after '=')
                                          pop
5377 000097CE 89CA
                                <1>
                                          mov
                                                edx, ecx
5378 000097D0 89DE
                                <1>
                                                esi, ebx ; ASCIIZ string address (after '=')
5379 000097D2 89FB
                                <1>
                                          mov
                                                ebx, edi
5380
                                <1>
5381 000097D4 F3A4
                                 <1>
                                          rep
                                                movsb
5382
                                 <1>
5383 000097D6 880F
                                 <1>
                                                 [edi], cl; 0; end of variable
5384
                                 <1>
5385 000097D8 0FB605[F0640100]
                                          movzx eax, byte [env var length]
                                <1>
                                          add edx, eax; variable length (total)
5386 000097DF 01C2
                                <1>
5387 000097E1 F7D8
                                <1>
                                          neg
                                                eax
5388 000097E3 01D8
                                <1>
                                                eax, ebx; start address of the variable
                                          add
                                          clc ; 13/04/2016
5389 000097E5 F8
                                <1>
5390 000097E6 E99EFEFFFF
                                <1>
                                          jmp set_env_string_allocate_envb_retn ; OK !
5391
                                <1>
5392
                                <1> set_env_change_variable_calc9:
5393
                                <1>
                                         ; 11/04/2016
5394 000097EB 21D2
                                <1>
                                          and
                                               edx, edx ; is empty ?
5395 000097ED 753B
                                <1>
                                          jnz
                                                short set_env_change_variable_calc15
                                <1>
5397 000097EF 0FB6DC
                                <1>
                                          movzx ebx, ah
5398 000097F2 F7DB
                                <1>
                                          neg
                                                ebx
5399 000097F4 01FB
                                <1>
                                          add
                                                ebx, edi
5400
                                <1>
5401
                                <1>
                                          ; EBX = Start address of the variable (in env page)
5402
                                <1>
                                          ; EDX = Variable length = 0
5403
                                <1>
5404 000097F6 89FE
                                <1>
                                               esi, edi
                                          mov
```

```
5405
                                <1>
5406
                               <1> set env change variable calc10:
5407 000097F8 AC
                               <1>
                                        lodsb
5408 000097F9 08C0
                               <1>
                                               al, al
5409 000097FB 75FB
                               <1>
                                         jnz short set_env_change_variable_calc10
5410
                               <1>
5411 000097FD B9FF310900
                                               ecx, Env Page + Env Page Size - 1
                               <1>
5412
                               <1>
5413 00009802 39CE
                               <1>
                                               esi, ecx; +511 (+4095)
                                         cmp
5414 00009804 7604
                               <1>
                                         jna
                                               short set_env_change_variable_calc11
5415
                               <1>
5416 00009806 89CE
                               <1>
                                         mov
                                               esi, ecx
5417 00009808 8806
                               <1>
                                               [esi], al ; 0
                                         mov
5418
                               <1>
                               <1> set env change_variable_calc11:
5419
                               <1>
5420 0000980A 89DF
                                         mov edi, ebx; old variable's start address
                               <1>
5422
                               <1> set_env_change_variable_calc12:
5423 0000980C AC
                               <1>
                                        lodsb
                           5424 0000980D AA
                                         stosb
                                         and al, al
5425 0000980E 20C0
5426 00009810 75FA
                                    jnz
cmp
                                         jnz
                                               short set_env_change_variable_calc12
5427 00009812 39CE
                                               esi, ecx
                            <1>
                                     ja
5428 00009814 7706
                                               short set_env_change_variable_calc13
5429 00009816 AC
                               <1>
                                         lodsb
5430 00009817 AA
                               <1>
                                         stosb
                            <1>
<1>
5431 00009818 20C0
                                         and al, al
                                         jnz short set_env_change_variable_calc12
5432 0000981A 75F0
5433
                               <1>
5434
                              <1> set_env_change_variable_calc13:
                            <1>
<1>
5435 0000981C 29F9
                                         sub ecx, edi
                              <1>
<1>
                                               short set env_change_variable_calc14
5436 0000981E 7203
                                         jb
                                         inc ecx; 1-5\overline{12} (\overline{1}-4096)
5437 00009820 41
                            <1>
                                       rep stosb; al = 0
5438 00009821 F3AA
5439
                               <1>
                               <1> set_env_change_variable_calc14:
5440
5441 00009823 29C0
                            <1> sub eax, eax; Start address of the variable
                                         ; EAX = 0 -> Variable is removed
5442
                               <1>
5443
                               <1>
                                         ; EDX = Variable length = 0
5444
                               <1>
5445 00009825 E95FFEFFFF
                               <1>
                                                   set_env_string_allocate_envb_retn ; OK !
5446
                               <1>
                               <1> set_env_change_variable_calc15:
                          <1> set_env_change_variable_
<1> push edx; *****
<1> neg edx
5447
5448 0000982A 52
                              <1>
<1>
5449 0000982B F7DA
5450 0000982D 01CA
                                         add
                                               edx, ecx; difference (the old string is longer)
                            <1>
<1>
5451 0000982F 89F3
                                         mov ebx, esi
5452 00009831 89FE
                                        mov esi, edi
5453
                               <1>
                               <1> set_env_change_variable_calc16:
5454
5455 00009833 AC
                               <1>
                                         lodsb
5456 00009834 20C0
                               <1>
                                         and
                                               al, al
5457 00009836 75FB
                               <1>
                                         jnz
                                               short set_env_change_variable_calc16
5458
                               <1>
5459 00009838 B900320900
                               <1>
                                        mov
                                               ecx, Env_Page + Env_Page_Size
5460
                               <1>
5461 0000983D 39CE
                                               esi, ecx; +512 (+4096)
                               <1>
                                         cmp
5462 0000983F 7605
                               <1>
                                               short set_env_change_variable_calc17
                                         jna
5463
                               <1>
5464 00009841 89CE
                               <1>
                                         mov
                                               esi, ecx
5465 00009843 8846FF
                             <1>
                                         mov
                                              [esi-1], al; 0
                               <1>
                            5467
5468 00009846 89F9
                               <1>
<1>
5469 00009848 89F7
                                               edi, esi ; next variable's address
                                         mov
5470
                               <1>
5471 0000984A AC
                                         lodsb
                               <1>
                                         or al, al
5472 0000984B 08C0
5473 0000984D 741D
                               <1>
                                               short set_env_change_variable_calc20
5474
                               <1>
5475
                               <1> set_env_change_variable_calc18:
                                     lodsb
5476 0000984F AC
                               <1>
5477 00009850 2000
                               <1>
                                         and al, al
5478 00009852 75FB
                               <1>
                                         jnz short set_env_change_variable_calc18
5479
                               <1>
5480 00009854 81FE00320900
                               <1>
                                               esi, Env_Page + Env_Page_Size
                                         cmp
5481 0000985A 720B
                               <1>
                                         jb
                                               short set env change variable calc19
5482 0000985C 740E
                                               short set env change variable calc20
                               <1>
                                         jе
5483
                               <1>
5484 0000985E BEFF310900
                               <1>
                                                esi, Env_Page + Env_Page_Size - 1
                                         mov
5485 00009863 8806
                               <1>
                                         mov
                                               [esi], al ; 0
5486 00009865 EB06
                                <1>
                                         jmp
                                                short set_env_change_variable_calc21
5487
                                <1>
5488
                                <1> set_env_change_variable_calc19:
5489 00009867 AC
                                <1>
                                         lodsb
5490 00009868 08C0
                                <1>
                                         or
                                                al, al
5491 0000986A 75E3
                               <1>
                                              short set_env_change_variable_calc18
                                         jnz
5492
                               <1>
5493
                                <1> set_env_change_variable_calc20:
5494 0000986C 4E
                               <1>
                                         dec esi; address of the second (last) 0 of the 00
5495
                                <1>
5496
                                <1> set_env_change_variable_calc21:
5497
                                         ; edx = difference (byte count)
                                <1>
5498
                                <1>
                                               eax, ecx; old variable's address (after '=')
5499 0000986D 89C8
                                <1>
                                         mov
5500
                                <1>
5501 0000986F 89F1
                                <1>
                                               ecx, esi
                                         mov
                                               ecx, edi ; count of bytes to move backward
5502 00009871 29F9
                                <1>
                                         sub
5503
                                <1>
5504 00009873 89FE
                                                esi, edi ; next variable's address
                               <1>
                                         mov
                                                edi, edx ; (displacement)
5505 00009875 29D7
                                <1>
                                         sub
5506
                                <1>
5507 00009877 F3A4
                                <1>
                                               movsb
                                         rep
5508
                                <1>
5509 00009879 880F
                                <1>
                                                [edi], cl ; 0 ; 00 ; end of environment variables
                                         mov
```

```
5510
                                  <1>
5511 0000987B 89C7
                                  <1>
                                            mov
                                                  edi, eax
                                                  edx; *****; byte count (after '=')
5512 0000987D 5A
                                 <1>
                                            pop
5513 0000987E 89D1
                                 <1>
                                                  ecx, edx
                                            mov
                                                  esi, ebx ; ASCIIZ string address (after '=')
5514 00009880 89DE
                                 <1>
                                            mov
5515 00009882 89FB
                                  <1>
                                            mov
                                                  ebx, edi
5516
                                  <1>
5517 00009884 F3A4
                                  <1>
                                                  movsb
                                            rep
5518
                                  <1>
5519 00009886 880F
                                                  [edi], cl ; 0 ; end of variable
                                  <1>
                                            mov
5520
                                  <1>
5521 00009888 0FB605[F0640100]
                                  <1>
                                            movzx eax, byte [env var length]
5522 0000988F 01C2
                                  <1>
                                            add edx, eax; variable length (total)
5523 00009891 F7D8
                                  <1>
5524 00009893 01D8
                                  <1>
                                            add
                                                 eax, ebx; start address of the variable
5525 00009895 F8
                                                 ; 13/04/2016
                                  <1>
                                            clc
5526 00009896 E9EEFDFFFF
                                  <1>
                                            jmp
                                                      set_env_string_allocate_envb_retn ; OK !
5527
                                  <1>
                                  <1> mainprog startup configuration:
5528
                                        ; \overline{2}2/11/20\overline{1}7
5529
                                  <1>
5530
                                  <1>
                                            ; 06/05/2016
                                           ; 14/04/2016 (TRDOS 386 = TRDOS v2.0)
5531
                                  <1>
5532
                                  <1>
                                           ; 17/09/2011 (TRDOS v1, MAINPROG.ASM)
5533
                                  <1>
                                  <1> loc_load_mainprog_cfg_file:
5534
5535 0000989B BE[A80D0100]
                                 <1>
                                           mov
                                                 esi, MainProgCfgFile
5536 000098A0 66B80018
                                 <1>
                                           mov
                                                 ax, 1800h ; Except volume label and dirs
                                            call find_first_file
5537 000098A4 E83EEAFFFF
                                 <1>
5538 000098A9 7256
                                  <1>
                                                  short loc_load_mainprog_cfg_exit
                                            jс
5539
                                  <1>
5540
                                  <1>
                                            ;or
                                                  eax, eax
5541
                                  <1>
                                           ;jz
                                                 short loc_load_mainprog_cfg_exit
5542
                                  <1>
5543
                                  <1> loc start mainprog configuration:
                                           ; ESI = FindFile DirEntry Location
5544
                                  <1>
                                            ; EAX = File Size
5545
                                  <1>
5546
                                  <1>
5547 000098AB A3[74590100]
                                                  [MainProgCfg_FileSize], eax
                                  <1>
                                           mov
5548
                                  <1>
5549 000098B0 668B5614
                                  <1>
                                                  dx, [esi+DirEntry_FstClusHI]
                                            mov
                                                  edx, 16
5550 000098B4 C1E210
                                  <1>
                                            shl
5551 000098B7 668B561A
                                  <1>
                                            mov
                                                  dx, [esi+DirEntry FstClusLO]
5552 000098BB 8915[A4640100]
                                 <1>
                                            mov
                                                  [csftdf_sf_cluster], edx
5553
                                  <1>
5554 000098C1 89C1
                                  <1>
                                           mov
                                                  ecx, eax
5555 000098C3 29C0
                                  <1>
                                            sub
                                                  eax, eax
5556
                                  <1>
                                           ; TRDOS 386 (TRDOS v2.0)
5557
                                  <1>
5558
                                  <1>
                                            ; Allocate contiguous memory block for loading the file
5559
                                  <1>
5560
                                  <1>
                                            ; eax = 0 (Allocate memory from the beginning)
5561
                                  <1>
                                            ; ecx = File (Allocation) size in bytes
5562
                                  <1>
5563 000098C5 E8B7BBFFFF
                                  <1>
                                            call allocate_memory_block
5564 000098CA 7235
                                  <1>
                                                  short loc_load_mainprog_cfg_exit
                                            jс
5565
                                  <1>
5566 000098CC A3[9C640100]
                                  <1>
                                                   [csftdf sf mem addr], eax; loading address
                                            mov
5567 000098D1 890D[A0640100]
                                  <1>
                                            mov
                                                   [csftdf_sf_mem_bsize], ecx; block size
5568
                                  <1>
5569 000098D7 31DB
                                  <1>
                                            xor
                                                   ebx, ebx
5570
                                  <1>
                                                  [csftdf_sf_rbytes], ebx; 0, reset
                                  <1>
5572 000098D9 8A3D[86590100]
                                  <1>
                                                  bh, [Current_Drv] ; [FindFile_Drv]
                                            mov
5573 000098DF BE00010900
                                  <1>
                                                   esi, Logical_DOSDisks
5574 000098E4 01DE
                                  <1>
                                            add
                                                  esi, ebx
5575
                                  <1>
5576 000098E6 8B1D[9C640100]
                                  <1>
                                            mov
                                                   ebx, [csftdf sf mem addr] ; memory block address
5577
                                  <1>
5578 000098EC 807E0300
                                  <1>
                                                  byte [esi+LD_FATType], 0
                                            cmp
5579 000098F0 7710
                                  <1>
                                                 short loc_mcfg_load_fat_file
                                            jа
5580
                                  <1>
5581 000098F2 C705[AC640100]0000- <1>
                                                   dword [csftdf_r_size], 65536
5581 000098FA 0100
                                 <1>
                                                      loc mcfg load fs file
5582 000098FC E9A1010000
                                  <1>
                                              jmp
5583
                                  <1>
5584
                                  <1> loc_load_mainprog_cfg_exit:
5585 00009901 C3
                                 <1>
                                           retn
5586
                                  <1>
                                  <1> loc mcfg load fat file:
5587
5588 00009902 0FB74611
                                          movzx eax, word [esi+LD BPB+BytesPerSec]
                                 <1>
5589 00009906 0FB64E13
                                  <1>
                                            movzx ecx, byte [esi+LD_BPB+SecPerClust]
5590 0000990A F7E1
                                  <1>
                                            mul
                                                   ecx
5591 0000990C A3[AC640100]
                                  <1>
                                            mov
                                                  [csftdf_r_size], eax
                                  <1>
5593
                                  <1> loc_mcfg_load_fat_file_next:
                                            call mcfg_read_fat_file_sectors
5594 00009911 E822010000
                                  <1>
5595 00009916 0F8206010000
                                 <1>
                                                      mcfg_deallocate_mem
                                            jс
5596
                                  <1>
5597 0000991C 09D2
                                  <1>
                                                   edx, edx ; edx > 0 -> EOF
                                            or
5598 0000991E 74F1
                                  <1>
                                                  short loc mcfg load fat file next
                                            jΖ
5599
                                  <1>
                                  <1> loc mcfg_load_fat_file_ok:
5600
5601
                                           ; 06/05/2016
                                  <1>
                                                  dword [mainprog return addr], loc mcfg ci return addr
5602 00009920 C705[40650100]-
                                  <1>
5602 00009926 [E3990000]
                                  <1>
5603
                                  <1>
5604 0000992A 8B35[9C640100]
                                  <1>
                                                   esi, [csftdf sf mem addr]
                                           mov
5605 00009930 8935[78590100]
                                  <1>
                                                   [MainProgCfg_LineOffset], esi
                                           mov
5606
                                  <1>
5607 00009936 A1[74590100]
                                  <1>
                                                   eax, [MainProgCfg_FileSize]
                                            mov
5608 0000993B 89C2
                                  <1>
                                           mov
                                                   edx, eax
5609 0000993D 01F2
                                  <1>
                                           add
                                                  edx, esi
5610
                                  <1>
                                  <1> loc mcfg process next line check:
5611
5612 0000993F 89C1
                                  <1>
                                           mov ecx, eax
```

```
5613
                               <1>
5614 00009941 803E2A
                               <1>
                                        cmp
                                              byte [esi], "*"; Remark sign
5615 00009944 7503
                              <1>
                                        jne
                                              short loc_mcfg_process_next_line
5616 00009946 46
                              <1>
                                        inc
                                              esi
5617 00009947 EB17
                              <1>
                                              short loc_move_mainprog_cfg_nl1
                                        jmp
5618
                              <1>
                              <1> loc mcfg_process_next_line:
5619
5620 00009949 83F94F
                              <1>
                                        cmp ecx, 79
5621 0000994C 7605
                               <1>
                                              short loc_start_mainprog_cfg_process
                                        jna
5622
                              <1>
5623 0000994E B94F000000
                               <1>
                                        mov
                                             ecx, 79
5624
                               <1>
                               <1> loc_start_mainprog_cfg_process:
5625
5626 00009953 BF[365A0100]
                                       mov edi, CommandBuffer
                              <1>
                               <1>
5627
                               <1> loc_move_mainprog_cfg_line:
5628
5629 00009958 AC
                                        lodsb
                              <1>
                           <1>
5630 00009959 3C20
                                        cmp al, 20h
5631 0000995B 720C
                                              short loc move mainprog cfg nl2
                              <1>
                                        jb
                                        stosb
5632 0000995D AA
                              <1>
5633 0000995E E2F8
                              <1>
                                        loop loc_move_mainprog_cfg_line
5634
                               <1>
                              <1> loc_move_mainprog_cfg_nl1:
5635
                           5636 00009960 39D6
                              <1>
                                        cmp esi, edx; + configuration file size
5637 00009962 7312
                                        inb
                                              short loc_end_of_mainprog_cfg_line
5638 00009964 AC
                                        lodsb
5639 00009965 3C20
                                        cmp al, 20h
                              <1>
5640 00009967 73F7
                                        jnb
                                              short loc_move_mainprog_cfg_nl1
5641
                               <1>
5642
                              <1> loc_move_mainprog_cfg_nl2:
                              <1>
5643 00009969 39D6
                                        cmp esi, edx
5644 0000996B 7309
                              <1>
                                        jnb
                                              short loc end of mainprog cfg line
5645 0000996D 8A06
                              <1>
                                              al, [esi]
                                        mov
5646 0000996F 3C20
                            <1>
                                        cmp al, 20h
5647 00009971 7703
                                        ja
                              <1>
                                              short loc_end_of_mainprog_cfg_line
                              <1>
5648 00009973 46
                                        inc
                                              esi
5649 00009974 EBF3
                               <1>
                                        jmp short loc_move_mainprog_cfg_nl2
5650
                               <1>
5651
                               <1> loc_end_of_mainprog_cfg_line:
5652 00009976 C60700
                                       mov byte [edi], 0
                               <1>
5653
                               <1>
                                             [MainProgCfg LineOffset], esi
5654 00009979 8935[78590100]
                               <1>
                                        mov
5655
                               <1>
                                        ; 22/11/2017
5656
                               <1>
5657 0000997F BE[3E5A0100]
                               <1>
                                        mov esi, CommandBuffer + 8
5658 00009984 29FE
                              <1>
                                        sub
                                              esi, edi
                                        jna short loc_move_mainprog_cfg_command
5659 00009986 7606
                              <1>
                           5660 00009988 30C0
                                       xor al, al
5661
                              <1> loc_mainprog_cfg_clear_chrs:
5662 0000998A AA
5663 0000998B 4E
                            <1>
                                        dec esi
5664 0000998C 75FC
                              <1>
                                        jnz short loc_mainprog_cfg_clear_chrs
                              <1>
5665
5666
                              <1> loc_move_mainprog_cfg_command:
5667 0000998E BE[365A0100]
                              <1> mov esi, CommandBuffer
5668 00009993 89F7
                              <1>
                                        mov
                                              edi, esi
5669 00009995 31DB
                              <1>
                                        xor ebx, ebx
5670
                              <1>
                                        ;xor ecx, ecx
5671 00009997 30C9
                               <1>
                                        xor cl, cl
5672
                              <1>
5673
                              <1> loc_move_mcfg_first_cmd_char:
                           5674 00009999 8A041E
                              <1> mov al, [esi+ebx]
5675 0000999C FEC3
                                        inc
                                              bl
                                        cmp al, 20h
5676 0000999E 3C20
                            <1>
                                        ja
5677 000099A0 7712
                              <1>
                                              short loc_move_mcfg_cmd_capitalizing
5678 000099A2 7237
                              <1>
                                        jb
                                               short loc_move_mcfg_cmd_arguments_ok
5679 000099A4 80FB4F
                              <1>
                                        cmp
                                              bl, 79
5680 000099A7 72F0
                               <1>
                                        jb
                                              short loc_move_mcfg_first_cmd_char
5681 000099A9 EB30
                               <1>
                                             short loc_move_mcfg_cmd_arguments_ok
                                        jmp
5682
                               <1>
5683
                               <1> loc_move_mcfg_next_cmd_char:
5684 000099AB 8A041E
                               <1>
                                       mov
                                              al, [esi+ebx]
5685 000099AE FEC3
                              <1>
                                        inc
                                              bl
5686 000099B0 3C20
                              <1>
                                        cmp al, 20h
5687 000099B2 7614
                              <1>
                                              short loc move mcfg cmd ok
                                        jna
5688
                              <1>
                              <1> loc move mcfg cmd capitalizing:
5689
5690 000099B4 3C61
                                        cmp al, \overline{61}h; 'a'
                              <1>
                                               short loc move mcfg cmd caps ok
5691 000099B6 7206
                              <1>
                                         jb
5692 000099B8 3C7A
                              <1>
                                              al, 7Ah ; 'z'
                                        cmp
5693 000099BA 7702
                               <1>
                                        jа
                                              short loc_move_mcfg_cmd_caps_ok
5694 000099BC 24DF
                               <1>
                                         and
                                               al, ODFh ; sub
                                                             al, 'a'-'A'
5695
                               <1> loc_move_mcfg_cmd_caps_ok:
5697 000099BE AA
                               <1>
                                        stosb
5698 000099BF FEC1
                               <1>
                                        inc
                                               cl
5699 000099C1 80FB4F
                              <1>
                                              bl, 79
                                        cmp
5700 000099C4 72E5
                              <1>
                                              short loc_move_mcfg_next_cmd_char
                                        jb
5701 000099C6 EB13
                               <1>
                                        jmp
                                              short loc_move_mcfg_cmd_arguments_ok
5702
                               <1>
                               <1> loc move mcfg cmd ok:
5703
5704 000099C8 30C0
                               <1>
                                       xor al, al; 0
5705
                               <1>
5706
                              <1> loc_move_mcfg_cmd_arguments:
                              <1>
5707 000099CA 8807
                                        mov
                                              [edi], al
                              <1>
5708 000099CC 47
                                        inc
                                               edi
                            <1>
5709 000099CD 80FB4F
                                              bl, 79
                                        cmp
                              <1>
5710 000099D0 7309
                                        jnb
                                              short loc_move_mcfg_cmd_arguments_ok
5711 000099D2 8A041E
                              <1>
                                        mov
                                               al, [esi+ebx]
5712 000099D5 FEC3
                              <1>
                                        inc
                                              bl
                              <1>
5713 000099D7 3C20
                                        cmp
                                              al, 20h
                                      jnb
5714 000099D9 73EF
                              <1>
                                              short loc_move_mcfg_cmd_arguments
5715
                              <1>
                               <1> loc_move_mcfg_cmd_arguments_ok:
5716
5717 000099DB C60700
                                        mov byte [edi], 0
                               <1>
```

```
5718
                                 <1>
5719
                                 <1> loc mcfg process cmd interpreter:
5720 000099DE E825E0FFFF
                                          call
                                                 command_interpreter
                                <1>
5721
                                <1>
5722
                                <1> loc_mcfg_ci_return_addr:
5723 000099E3 A1[74590100]
                                      mov eax, [MainProgCfg_FileSize]
                                <1>
5724 000099E8 89C2
                                <1>
                                          mov
                                                edx, eax
5725 000099EA 8B35[78590100]
                                                esi, [MainProgCfg_LineOffset]
                                <1>
                                          mov
5726 000099F0 01F2
                                <1>
                                          add
                                                edx, esi
5727 000099F2 0305[9C640100]
                                <1>
                                          add
                                                eax, [csftdf_sf_mem_addr]
5728 000099F8 29F0
                                <1>
                                          sub
                                                eax, esi
                                         ja
5729 000099FA 0F873FFFFFF
                                 <1>
                                                loc_mcfg_process_next_line_check
5730
                                <1>
                                        call mcfg deallocate mem
5731 00009A00 E81D000000
                                <1>
5732
                                 <1>
5733 00009A05 B94F000000
                                <1>
                                          mov
                                                 ecx, 79; 80?
                                          mov edi, CommandBuffer
5734 00009A0A BF[365A0100]
                                <1>
5735 00009A0F 30C0
                                <1>
                                          xor al, al
5736 00009A11 F3AA
                                <1>
                                                stosb
                                          rep
5737
                                <1>
                                          ; 06/05/2016
5738
                                <1>
                                          mov esi, nextline
call print_msg
5739 00009A13 BE[FF190100]
                                <1>
5740 00009A18 E8B0C9FFFF
                                <1>
5741 00009A1D E963D6FFFF
                                <1>
                                          jmp dos_prompt
5742
                                <1>
5743
                                 <1> mcfg_deallocate_mem:
                                <1>     mov     eax, [csftdf_sf_mem_addr] ; start address
5744 00009A22 A1[9C640100]
5745 00009A27 8B0D[A0640100]
                                <1>
                                          mov
                                                ecx, [csftdf_sf_mem_bsize] ; block size
5746
                                 <1>
                                          ;call deallocate_memory_block
5747
                                <1>
                                          ;retn
5748 00009A2D E95CBCFFFF
                                <1>
                                         jmp deallocate_memory_block
5749
                                 <1>
5750
                                <1> mcfg_read_file_sectors:
                                      ; 1\overline{4}/04/\overline{2}016
5751
                                <1>
5752 00009A32 807E0300
                                <1>
                                          cmp byte [esi+LD_FATType], 0
                                           jna short mcfg_read_fs_file_sectors
5753 00009A36 7669
                                <1>
5754
                                 <1>
5755
                                 <1> mcfg_read_fat_file_sectors:
                                       ; return:
5756
                                 <1>
                                          ; CF = 0 \& EDX > 0 \rightarrow END OF FILE
5757
                                 <1>
5758
                                 <1>
                                        ; CF = 0 \& EDX = 0 \rightarrow not EOF
                                          ; CF = 1 -> read error (error code in AL)
5759
                                 <1>
5760
                                 <1>
5761
                                 <1> mcfg_read_fat_file_secs_0:
5762 00009A38 8B15[74590100]
                                <1> mov edx, [MainProgCfg_FileSize]
5763 00009A3E 2B15[B4640100]
                                <1>
                                          sub
                                                edx, [csftdf_sf_rbytes]
5764 00009A44 3B15[AC640100]
                                <1>
                                          cmp edx, [csftdf r size]
5765 00009A4A 7306
                                <1>
                                       jnb short mcfg_read_fat_file_secs_1
5766 00009A4C 8915[AC640100]
                                <1>
                                         mov
                                                [csftdf_r_size], edx
5767
                                <1>
5768
                                <1> mcfg_read_fat_file_secs_1:
                                <1> mov eax, [csftdf_r_size]
<1> sub edx, edx
5769 00009A52 A1[AC640100]
5770 00009A57 29D2
                                <1>
5771 00009A59 0FB74E11
                               <1>
                                          movzx ecx, word [esi+LD_BPB+BytesPerSec]
5772 00009A5D 01C8
                                <1>
                                          add eax, ecx
5773 00009A5F 48
                                <1>
                                          dec
                                                eax
5774 00009A60 F7F1
                                <1>
                                                ecx, eax ; sector count
5775 00009A62 89C1
                                <1>
                                          mov
5776 00009A64 A1[A4640100]
                                <1>
                                                eax, [csftdf_sf_cluster]
                                          mov
5777
                                <1>
5778
                                 <1>
                                          ; EBX = memory block address (current)
                                 <1>
5780 00009A69 E88C230000
                                 <1>
                                          call read_fat_file_sectors
5781 00009A6E 7230
                                 <1>
                                                 short mcfg_read_fat_file_secs_3
5782
                                 <1>
5783
                                 <1>
                                          ; EBX = next memory address
                                 <1>
5785 00009A70 A1[B4640100]
                                <1>
                                          mov
                                                eax, [csftdf_sf_rbytes]
5786 00009A75 0305[AC640100]
                                 <1>
                                                 eax, [csftdf_r_size]
                                          add
5787 00009A7B 8B15[74590100]
                                <1>
                                                edx, [MainProgCfg_FileSize]
                                          mov
5788 00009A81 39D0
                                 <1>
                                           cmp
                                                 eax, edx
5789 00009A83 731B
                                 <1>
                                          jnb
                                                 short mcfg_read_fat_file_secs_3 ; edx > 0
5790 00009A85 A3[B4640100]
                                <1>
                                          mov
                                                [csftdf_sf_rbytes], eax
5791
                                <1>
                                          push ebx; *
5792 00009A8A 53
                                 <1>
                                          ; get next cluster (csftdf_r_size! bytes)
5793
                                 <1>
5794 00009A8B A1[A4640100]
                                <1>
                                          mov eax, [csftdf sf cluster]
                                          call get_next_cluster
5795 00009A90 E837210000
                                 <1>
5796 00009A95 5B
                                                ebx ; *
                                 <1>
                                          pop
5797 00009A96 7301
                                 <1>
                                          jnc short mcfg_read_fat_file_secs_2
5798
                                 <1>
5799
                                 <1>
                                           ;mov eax, 17; Read error !
5800 00009A98 C3
                                 <1>
                                          retn
                                 <1>
                                 <1> mcfg read fat file secs 2:
5802
5803 00009A99 29D2
                                 <1>
                                          sub edx, edx; 0
                                          mov [csftdf_sf_cluster], eax ; next cluster
5804 00009A9B A3[A4640100]
                                <1>
5805
                                 <1>
5806
                                 <1> mcfg_read_fat_file_secs_3:
5807 00009AA0 C3
                                 <1>
                                          retn
5808
                                 <1>
5809
                                 <1> mcfg_read_fs_file_sectors:
5810 00009AA1 C3
                                 <1>
                                          retn
                                 <1>
5811
                                 <1> loc_mcfg_load_fs_file:
5812
5813 00009AA2 C3
                                 <1>
                                          retn
5814
                                 <1>
5815
                                 <1> load and execute file:
                                          ; \overline{0}4/01/20\overline{1}7
5816
                                 <1>
                                          ; 06/05/2016, 07/05/2016, 11/05/2016
5817
                                 <1>
5818
                                 <1>
                                          ; 23/04/2016, 24/04/2016
5819
                                 <1>
                                          ; 22/04/2016 (TRDOS 386 = TRDOS v2.0)
                                          ; 05/11/2011
5820
                                 <1>
                                          ; (TRDOS v1, CMDINTR.ASM, 'cmp cmd run', 'cmp cmd external')
5821
                                 <1>
                                 <1>
                                          ; ('loc run check filename')
5822
```

```
; 29/08/2011
5823
                                 <1>
                                          ; 10/09/2011
5824
                                 <1>
                                          ; INPUT->
5825
                                 <1>
5826
                                 <1>
                                                 ESI = Path Name address (CommandBuffer address)
5827
                                 <1>
                                          ; OUTPUT ->
5828
                                 <1>
                                                none (error message will be shown if an error will occur)
5829
                                 <1>
5830
                                 <1>
                                          ; (EAX, EBX, ECX, EDX, ESI, EDI, EBP will be changed)
5831
                                 <1>
5832
                                 <1> loc_run_check_filename:
                                           cmp byte [esi], 20h
5833 00009AA3 803E20
                                 <1>
5834 00009AA6 0F822BE3FFFF
                                 <1>
                                           jb
                                                 loc cmd failed
5835 00009AAC 7703
                                                 short loc_run_check_filename_ok
                                 <1>
                                           jа
5836 00009AAE 46
                                 <1>
                                           inc
                                                 esi
5837 00009AAF EBF2
                                 <1>
                                                short loc run check filename
                                           jmp
5838
                                 <1>
                                 <1> loc run check filename ok:
                                          mov byte [CmdArgStart], 0; reset
5840 00009AB1 C605[E7590100]00
                                <1>
                                           push esi; *
5841 00009AB8 56
                                 <1>
5842
                                <1> loc_run_get_first_arg_pos:
5843 00009AB9 46
                                <1>
                                        inc esi
                                                al, [esi]
al, 20h
5844 00009ABA 8A06
                                <1>
                                          mov
5845 00009ABC 3C20
                                <1>
                                          cmp
                                                 short loc_run_get_first_arg_pos
5846 00009ABE 77F9
                                <1>
                                           ja
5847 00009AC0 C60600
                                <1>
                                         mov byte [esi], 0
                                <1> loc_run_get_external_arg_pos:
5848
                                <1> ; 11/05/2016
5849
                                          inc esi
5850 00009AC3 46
                                <1>
                                     mov al, [esi]
cmp al, 20h
jna short loc_run_parse_path_name
mov eax, esi
sub eax, CommandBuffer
mov byte [CmdArgStart], al
5851 00009AC4 8A06
                                <1>
5852 00009AC6 3C20
                                <1>
5853 00009AC8 760C
                                <1>
5854 00009ACA 89F0
                                <1>
5855 00009ACC 2D[365A0100]
                                <1>
5856 00009AD1 A2[E7590100]
                                <1>
5857
                                 <1> loc_run_parse_path_name:
5858 00009AD6 5E
                                <1> pop esi; *
5859 00009AD7 BF[26620100]
                                                edi, FindFile Drv
                                <1>
                                          mov
5860 00009ADC E8D7090000
                                          call parse_path_name
                                <1>
5861 00009AE1 0F82F0E2FFFF
                                <1>
                                          jc loc_cmd_failed
5862
                                <1>
                                 <1> loc_run_check_filename_exists:
5863
5864 00009AE7 BE[68620100]
                                                esi, FindFile Name
                                 <1>
                                          mov
5865 00009AEC 803E20
                                                 byte [esi], 20h
                                 <1>
                                           cmp
5866 00009AEF 0F86E2E2FFFF
                                <1>
                                          jna loc_cmd_failed
5867
                                 <1>
5868
                                 <1> loc_run_check_exe_filename_ext:
5869 00009AF5 E890020000
                                <1>
                                      call check prg filename ext
5870 00009AFA 0F82D7E2FFFF
                                <1>
                                           jс
                                                loc_cmd_failed
5871
                                 <1>
                                 <1> loc_run_check_exe_filename_ext_ok:
5872
5873 00009B00 66A3[3E650100]
                                <1>
                                         mov word [EXE_ID], ax
5874
                                 <1>
                                 <1> loc_run_drv:
5875
                                      mov byte [Run_Manual_Path], 0
5876 00009B06 C605[3D650100]00
                                 <1>
                                                 eax, [Current_Dir_FCluster]
5877 00009B0D A1[80590100]
                                 <1>
                                          mov
5878 00009B12 A3[38650100]
                                 <1>
                                          mov
                                                    [Run_CDirFC], eax
                                 <1>
                                          ;
5880 00009B17 8A35[86590100]
                                                 dh, [Current_Drv]
                                 <1>
                                          mov
5881 00009B1D 8835[E2600100]
                                 <1>
                                                 [RUN_CDRV], dh
                                          mov
                                 <1>
5883 00009B23 8A15[26620100]
                                 <1>
                                           mov
                                                 dl, [FindFile_Drv]
5884 00009B29 38F2
                                 <1>
                                           cmp
                                                 dl, dh
5885 00009B2B 7412
                                 <1>
                                                 short loc_run_change_directory
                                           jе
5886
                                 <1>
5887 00009B2D 8005[3D650100]02
                                 <1>
                                          add
                                                 byte [Run_Manual_Path], 2
5888
                                 <1>
5889 00009B34 E80BD4FFFF
                                 <1>
                                           call change current drive
5890 00009B39 0F82C3E2FFFF
                                 <1>
                                                 loc_run_cmd_failed
                                           jс
5891
                                 <1>
                                 <1> loc_run_change_directory:
5892
5893 00009B3F 803D[27620100]20
                                 <1>
                                           cmp
                                                 byte [FindFile_Directory], 20h
                                                 short loc_run_find_executable file
5894 00009B46 7623
                                 <1>
                                           jna
5895
                                 <1>
5896 00009B48 FE05[3D650100]
                                 <1>
                                           inc
                                                 byte [Run_Manual_Path]
5897
                                 <1>
5898 00009B4E FE05[620D0100]
                                                 byte [Restore_CDIR]
                                 <1>
                                           inc
                                 <1>
5900 00009B54 BE[27620100]
                                                 esi, FindFile_Directory
                                 <1>
                                           mov
5901 00009B59 30E4
                                 <1>
                                                 ah, ah ; CD_COMMAND sign -> 0
                                           xor
5902 00009B5B E842030000
                                 <1>
                                           call change current directory
5903 00009B60 0F829CE2FFFF
                                 <1>
                                                 loc run cmd failed
5904
                                 <1>
                                 <1> loc_run_change_prompt_dir_string:
5906 00009B66 E857020000
                                          call change_prompt_dir_string
5907
                                 <1>
5908
                                 <1> loc run find executable file:
5909 00009B6B 66C705[3C650100]00- <1>
                                         mov word [Run_Auto_Path], 0
5909 00009B73 00
                                 <1>
5910
                                 <1>
5911
                                 <1> loc_run_find_executable_file_next:
5912 00009B74 BE[68620100]
                                 <1>
                                         mov esi, FindFile_Name
                                 <1> loc run find program file next:
5913
                                          mov ax, 1800h; Except volume label and dirs
5914 00009B79 66B80018
                                 <1>
                                           call find_first_file
5915 00009B7D E865E7FFFF
                                 <1>
5916
                                 <1>
                                          ; ESI = Directory Entry (FindFile_DirEntry) Location
5917
                                 <1>
                                          ; EDI = Directory Buffer Directory Entry Location
                                 <1>
                                          ; EAX = File size
5919 00009B82 0F835C010000
                                          jnc loc_load_and_run_file
                                 <1>
5920
                                 <1>
5921 00009B88 3C02
                                                 al, 2; file not found
                                 <1>
                                           cmp
5922 00009B8A 0F8572E2FFFF
                                 <1>
                                          jne
                                                 loc_run_cmd_failed
5923
                                 <1>
                                                 ax, word [EXE ID]
5924 00009B90 66A1[3E650100]
                                 <1>
                                          mov
5925 00009B96 80FC2E
                                 <1>
                                                 ah, '.'; File name has extension sign
                                           cmp
5926 00009B99 7424
                                 <1>
                                                 short loc run check auto path
                                           jе
```

```
5927
                                <1>
5928 00009B9B 08C0
                                <1>
                                          or
                                                al, al
5929 00009B9D 7520
                               <1>
                                          jnz
                                               short loc_run_check_auto_path
5930
                               <1>
5931 00009B9F 80FC08
                               <1>
                                               ah, 8 ; count of file name chars
                                          cmp
5932 00009BA2 771B
                               <1>
                                          jа
                                                short loc_run_check_auto_path
                               <1>
5934
                               <1> loc_run_change_file_ext_to_prg:
5935 00009BA4 0FB6DC
                               <1> movzx ebx, ah ; count of file name chars
5936 00009BA7 BE[68620100]
                               <1>
                                         mov esi, FindFile Name
                                         add
5937 00009BAC 01F3
                               <1>
                                               ebx, esi
5938 <1>
5939 00009BAE C7032E505247 <1>
                                         ; 07/05/2016
                                         mov dword [ebx], '.PRG'
5940 00009BB4 66C705[3E650100]50- <1> mov word [EXE_ID], 'P.'
                       <1>
5940 00009BBC 2E
5941 00009BBD EBBA
                               <1>
                                         jmp short loc_run_find_program_file_next
5942
                                5943
5944
5945
                                <1>
                                         ; If the path is given, value of byte [Run Manual Path]
                                        ; will not be ZERO. If so, file searching by using
5946
                                <1>
                                        ; Automatic Path (via 'PATH' environment variable) ; will not be applicable, because the program file
5947
                                <1>
                                <1>
5948
5949
                                <1>
                                        ; is already/absolutely not found.
5950
                                <1>
5951 00009BBF A0[3D650100]
                                <1>
                                         mov al, [Run_Manual_Path]
5952 00009BC4 08C0
                                <1>
                                         or al, al
5953 00009BC6 0F850BE2FFFF
                                          jnz loc_cmd_failed
                                <1>
5954
                                <1>
5955
                                <1> loc_run_check_auto_path_again:
5956 00009BCC 66833D[3C650100]FF <1>
                                       cmp word [Run_Auto_Path], 0FFFFh
5957
                                <1>
                                                ; OFFFFh = Not a valid run path (in ENV block)
                               5958 00009BD4 0F83FDE1FFFF
5959
5960 00009BDA BE[2E0E0100]
5961 00009BDF BF[865A0100]
5962 00009BE4 E848F9FFFF
5963 00009BE9 730E
                                         call get_environment_string
                               <1>
5963 00009BE9 730E
                               <1>
                                         jnc short loc_run_chk_filename_ext_again
5964 00009BEB 66C705[3C650100]FF- <1>
                                               word [Run_Auto_Path], OFFFFh; invalid
                                         mov
                       <1>
5964 00009BF3 FF
                                         jmp loc_cmd_failed
5965 00009BF4 E9DEE1FFFF
                               <1>
5966
                                <1>
                               <1> loc_run_chk_filename_ext_again:
5967
5968 00009BF9 89C1
                               <1>
                                         mov ecx, eax; string length (with zero tail)
5969 00009BFB 49
                               <1>
                                          dec
                                               ecx ; without zero tail
5970 00009BFC 66A1[3E650100]
                               <1>
                                          mov
                                               ax, [EXE_ID]
5971 00009C02 80FC2E
                                         cmp ah, '.'
                               <1>
5972 00009C05 740E
                                        je short loc_run_chk_auto_path_pos
                               <1>
5973
                               <1>
                               <1> loc_run_change_file_ext_to_noext_again:
5974
                           <1>
<1>
<1>
                                     movzx ebx, ah
5975 00009C07 0FB6DC
5976 00009C0A BE[68620100]
                                         mov esi, FindFile Name
5977 00009C0F 01F3
                                         add ebx, esi
                               <1>
5978 00009C11 29C0
                               <1>
                                         sub eax, eax
5979 00009C13 8903
                                <1>
                                         mov [ebx], eax; 0; erase extension (.PRG)
5980
                                <1>
                                <1> loc_run_chk_auto_path_pos:
                                <1> ;movzx eax, word [Run_Auto_Path]
5982
5983 00009C15 66A1[3C650100]
                                <1>
                                         mov ax, [Run_Auto_Path]
                                         cmp
                                               eax, ecx; ecx = string length (except zero tail)
5984 00009C1B 39C8
                               <1>
5985 00009C1D 0F83B4E1FFFF
                               <1>
                                        jnb loc_cmd_failed
                                         ;or eax, eax or ax, ax
                                <1>
5987 00009C23 6609C0
                               <1>
                                               ax, ax
5988 00009C26 7502
                               <1>
                                        jnz short loc_run_auto_path_pos_move
5989 00009C28 B005
                                <1>
                                        mov al, 5
5990
                               <1>
                               <1> loc_run_auto_path_pos_move:
5992 00009C2A 89FE
                               <1>
                                         mov esi, edi ; offset TextBuffer
5993 00009C2C 01C6
                               <1>
                                          add
                                               esi, eax
5994
                               <1>
5995
                               <1> loc_run_auto_path_pos_space_loop:
5996 00009C2E AC
                               <1>
                                         lodsb
5997 00009C2F 3C20
                               <1>
                                          cmp al, 20h
                                                short loc run auto path pos space loop
5998 00009C31 74FB
                               <1>
                                          jb
5999 00009C33 0F829EE1FFFF
                               <1>
                                               loc_cmd_failed
6000 00009C39 AA
                               <1>
                                          stosb
                               <1> loc_run_auto_path_pos_move_next:
6002 00009C3A AC
                               <1>
                                         lodsb
                                          cmp al, ';'
je short le
6003 00009C3B 3C3B
                                <1>
6004 00009C3D 7414
                               <1>
                                                short loc_run_auto_path_pos_move_last_byte
6005 00009C3F 3C20
                               <1>
                                          cmp
                                              al, 20h
                                                short loc_run_auto_path_pos_move_next
6006 00009C41 74F7
                                <1>
                                          jе
6007 00009C43 7203
                                <1>
                                          jb
                                                short loc_byte_ptr_end_of_path
6008 00009C45 AA
                                <1>
                                          stosb
6009 00009C46 EBF2
                                <1>
                                          jmp
                                               short loc_run_auto_path_pos_move_next
6010
                                <1>
                                <1> loc_byte_ptr_end_of_path:
6012 00009C48 66C705[3C650100]FF- <1>
                                          mov word [Run_Auto_Path], OFFFFh; end of path
6012 00009C50 FF
                                <1>
6013 00009C51 EB0D
                                <1>
                                               short loc_run_auto_path_move_ok
                                          jmp
6014
                                <1>
                                <1> loc_run_auto_path_pos_move_last_byte:
6015
6016 00009C53 89F0
                                <1>
                                         mov eax, esi
6017 00009C55 2D[865A0100]
                                <1>
                                                eax, TextBuffer
6018 00009C5A 66A3[3C650100]
                                <1>
                                               [Run_Auto_Path], ax ; next path position
                                         mov
6019
                                <1>
                                <1> loc_run_auto_path_move_ok:
6020
6021 00009C60 4F
                                <1>
                                         dec edi
                                                al, '/'
6022 00009C61 B02F
                                <1>
                                          mov
6023 00009C63 3807
                                          cmp
                               <1>
                                               [edi], al
6024 00009C65 7403
                               <1>
                                          jе
                                                short loc_run_auto_path_move_file_name
6025 00009C67 47
                                <1>
                                                edi
                                          inc
6026 00009C68 8807
                                <1>
                                          mov
                                                [edi], al
6027
                                <1>
6028
                                <1> loc_run_auto_path_move_file_name:
```

```
6029 00009C6A 47
                                <1>
                                          inc
                                                edi
6030 00009C6B BE[68620100]
                                <1>
                                                esi, FindFile Name
6031
                                <1>
6032
                                <1> loc_run_auto_path_move_fn_loop:
6033 00009C70 AC
                                <1>
                                          lodsb
6034 00009C71 AA
                                <1>
                                          stosb
6035 00009C72 08C0
                               <1>
                                          or
                                                al, al
6036 00009C74 75FA
                                <1>
                                        jnz
                                                short loc_run_auto_path_move_fn_loop
6037
                                <1>
6038 00009C76 BE[865A0100]
                                <1>
                                         mov
                                                esi, TextBuffer
6038 00009C76 BE[865A0100]
6039 00009C7B BF[26620100]
6040 00009C80 E833080000
                                <1>
                                          mov
                                               edi, FindFile_Drv
6040 00009C80 E833080000
                                <1>
                                          call parse path name
6041 00009C85 0F824CE1FFFF
                                <1>
                                                loc_cmd_failed
                                          jс
6042
                                <1>
6043 00009C8B 8A35[86590100]
                                <1>
                                                dh, [Current_Drv]
                                         mov
6044 00009C91 8A15[26620100]
                                <1>
                                          mov
                                                dl, [FindFile_Drv]
6045 00009C97 38F2
                                <1>
                                          cmp
                                                dl, dh
6046 00009C99 740B
                                                short loc_run_change_directory_again
                                <1>
                                          jе
6047
                                <1>
6048 00009C9B E8A4D2FFFF
                                <1>
                                          call change_current_drive
6049 00009CA0 0F825CE1FFFF
                                <1>
                                                loc_run_cmd_failed
                                <1>
                                <1> loc_run_change_directory_again:
6051
6052 00009CA6 803D[27620100]20
                               <1>
                                         cmp byte [FindFile_Directory], 20h
6053 00009CAD 761D
                                                short loc_load_executable_cdir_chk_again
                                <1>
                                          jna
6054
                                <1>
6055 00009CAF FE05[620D0100]
                                <1>
                                          inc
                                                byte [Restore CDIR]
6056 00009CB5 BE[27620100]
                                <1>
                                      mov
                                                esi, FindFile_Directory
6057 00009CBA 30E4
                                <1>
                                          xor
                                                ah, ah ; CD_COMMAND sign -> 0
6058 00009CBC E8E1010000
                                          call change current directory
                                <1>
6059 00009CC1 0F823BE1FFFF
                                <1>
                                                loc_run_cmd_failed
6060
                                <1>
                                <1> loc_run_chg_prompt_dir_str_again:
6061
6062 00009CC7 E8F6000000
                                <1>
                                         call change prompt dir string
6063
                                <1>
6064
                                <1> loc_load_executable_cdir_chk_again:
6065 00009CCC A1[80590100]
                               <1> mov eax, [Current_Dir_FCluster]
6066 00009CD1 3B05[38650100]
                                                eax, [Run_CDirFC]
                               <1>
                                          cmp
                                                loc run find executable file next
6067 00009CD7 0F8597FEFFFF
                                <1>
                                          jne
6068 00009CDD 30C0
                                <1>
                                                al, al ; 0
                                          xor
6069 00009CDF E9E8FEFFFF
                                <1>
                                        jmp loc_run_check_auto_path_again
6070
                                <1>
                                <1> loc_load_and_run_file:
6071
                                     ; \overline{1}3/1\overline{1}7
6072
                                <1>
                                          ; 04/01/2017
6073
                                <1>
6074
                                <1>
                                          ; 23/04/2016
                                        mov esi, FindFile_Name
6075 00009CE4 BE[68620100]
                                <1>
6076 00009CE9 BF[865A0100]
                                <1>
                                        mov edi, TextBuffer
6077
                                <1>
                                          ; 24/04/2016
                                <1>
6079 00009CEE 31D2
                                <1>
                                          xor edx, edx
6080 00009CF0 668915[4A040300]
                                <1>
                                          mov
                                                word [argc], dx ; 0
6081 00009CF7 8915[8C030300]
                                <1>
                                          mov
                                                dword [u.nread], edx ; 0
6082
                                <1>
6083
                                <1> loc_load_and_run_file_1:
6084 00009CFD AC
                                <1>
                                          lodsb
6085 00009CFE AA
                                <1>
                                          stosb
6086 00009CFF FF05[8C030300]
                                <1>
                                          inc dword [u.nread]
6087 00009D05 20C0
                                <1>
                                          and
                                                al, al
6088 00009D07 75F4
                                <1>
                                          jnz short loc_load_and_run_file_1
6089
                                <1>
6090 00009D09 A0[E7590100]
                                <1>
                                          mov
                                                al, [CmdArgStart]
6091 00009D0E 20C0
                                <1>
                                          and
                                                al, al
6092 00009D10 7445
                                <1>
                                                short loc_load_and_run_file_7
                                          jz
6093
                                <1>
6094 00009D12 0FB6F0
                                <1>
                                          movzx esi, al ; 11/05/2016
6095 00009D15 B950000000
                                <1>
                                          mov ecx, 80
6096 00009D1A 29F1
                                          sub
                                <1>
                                                ecx, esi
6097 00009D1C 81C6[365A0100]
                                <1>
                                               esi, CommandBuffer
                                          add
6098
                                <1>
6099 00009D22 66FF05[4A040300]
                                <1>
                                          inc word [argc] ; 11/05/2016
6100
                                 <1>
                                <1> loc_load_and_run_file_2:
6101
6102 00009D29 AC
                                <1>
                                      lodsb
                                <1>
6103 00009D2A 3C20
                                          cmp al, 20h
6104 00009D2C 7717
                                <1>
                                          jа
                                                short loc_load_and_run_file_5
6105 00009D2E 721E
                                <1>
                                                short loc load and run file 6
6106
                                <1>
                                <1> loc load and run file 3:
6107
6108 00009D30 803E20
                                        cmp byte [esi], 20h
                                <1>
                                                short loc_load_and_run_file_4
6109 00009D33 7707
                                <1>
6110 00009D35 7217
                                 <1>
                                          jb
                                                 short loc_load_and_run_file_6
6111 00009D37 46
                                <1>
                                          inc
                                                 esi
6112 00009D38 E2F6
                                 <1>
                                          loop loc_load_and run file 3
6113 00009D3A EB12
                                <1>
                                          jmp short loc_load_and_run_file_6
6114
                                <1>
                                <1> loc load and run file 4:
6116 00009D3C 28C0
                                          sub al, al; 0
                                <1>
6117 00009D3E 66FF05[4A040300]
                                <1>
                                                 word [argc]
                                          inc
                                <1> loc_load_and_run_file_5:
6118
6119 00009D45 AA
                                <1>
                                          stosb
6120 00009D46 FF05[8C030300]
                                          inc dword [u.nread]
                                 <1>
6121 00009D4C E2DB
                                          loop loc load and run file 2
                                <1>
                                <1>
6122
                                 <1> loc_load_and_run_file_6:
6123
6124 00009D4E 30C0
                                <1>
                                          xor al, al; 0
6125 00009D50 AA
                                <1>
                                          stosb
6126 00009D51 FF05[8C030300]
                                                dword [u.nread]
                                <1>
                                          inc
6127
                                 <1> loc_load_and_run_file_7:
6128 00009D57 8807
                                <1>
                                        mov [edi], al ; 0
                                                word [argc] ; 24/04/2016
6129 00009D59 66FF05[4A040300]
                                <1>
                                          inc
6130 00009D60 FF05[8C030300]
                                 <1>
                                          inc
                                                dword [u.nread] ; 24/04/2016
6131 00009D66 BE[865A0100]
                                <1>
                                          mov
                                                esi, TextBuffer
6132 00009D6B 8B15[94620100]
                                 <1>
                                                 edx, [FindFile_DirEntry+DirEntry_FileSize]
6133 00009D71 66A1[8C620100]
                                <1>
                                                ax, [FindFile DirEntry+DirEntry FstClusHI]
                                         mov
```

```
shl eax, 16; 13/11/2017
mov ax, [FindFile_DirEntry+DirEntry_FstClusLO]
; EAX = First Cluster number
; EDX = File Size
6134 00009D77 C1E010
                               <1>
6135 00009D7A 66A1[92620100]
                               <1>
6136
                               <1>
6137
                               <1>
6138
                               <1>
                                       ; ESI = Argument list address
6139
                               <1>
                                        ; [argc] = argument count
                                        ; [u.nread] = argument list length
6140
                               <1>
                                      call load_and_run_file ; trdosk6.s
   ;jc loc_run_cmd_failed ; 04/01/2017
6141 00009D80 E89D420000
                               <1>
6142
                               <1>
                               <1> loc_load_and_run_file_8: ; 06/05/2016
6143
6144 00009D85 E98BE9FFFF
                               <1>
                                       jmp loc_file_rw_restore_retn
6145
                               <1>
6146
                               <1> check_prg_filename_ext:
                                      ; 23/04/2016 (TRDOS 386 = TRDOS v2.0)
6147
                               <1>
6148
                               <1>
                                        ; 10/09/2011
                                       ; (TRDOS v1, CMDINTR.ASM, 'proc_check_exe_filename_ext')
; 14/11/2009
6149
                               <1>
6150
                                <1>
                                      ; INPUT ->
                               <1>
6151
6152
                               <1>
                                            ESI = Dot File Name
                                        ; OUTPUT ->
6153
                               <1>
                                      ; cf = 0 -> EXE_ID in AL
6154
                               <1>
                                       ;
6155
                               <1>
                                              ESI = Last char + 1 position
                                              cf = 1 -> Invalid executable file name
6156
                               <1>
                                        ;
6157
                               <1>
                                             or no file name extension if AH<=8
                               <1>
6158
                                              AL = Last file name char
                                              cf = 0 \rightarrow AL='P' (PRG), AL=0 (no extension)
6159
                               <1>
6160
                               <1>
                               <1>
                                       ; (Modified registers: EAX, ESI)
6161
                               <1>
6162
6163 00009D8A 30E4
                               <1>
                                        xor ah, ah
6164
                               <1> loc_run_check_filename_ext:
6165 00009D8C AC
                               <1>
                                        lodsb
6166 00009D8D 3C21
                                         cmp al, 21h
                               <1>
6167 00009D8F 7229
                               <1>
                                              short loc check exe fn retn
6168 00009D91 FEC4
                               <1>
                                        inc ah
                                              al, '.'
6169 00009D93 3C2E
                               <1>
                                         cmp
                                        jne short loc_run_check_filename_ext
6170 00009D95 75F5
                              <1>
6171
                               <1>
6172
                               <1> loc_run_check_filename_ext_dot:
6173 00009D97 80FC02
                              <1> cmp ah, 2; .??? is not valid
6174 00009D9A 88C4
                                         mov ah, al ; '.'
                              <1>
6175 00009D9C 7219
                               <1>
                                         jb
                                               short loc_check_prg_fn_retn
6176
                               <1>
                              <1> loc run check filename ext dot ok:
6177
6178 00009D9E AC
                                       lodsb
                               <1>
6179 00009D9F 24DF
                               <1>
                                         and al, ODFh
6180
                              <1>
6181
                              <1> loc_run_check_filename_ext_prg:
6182 00009DA1 3C50
                              <1>
                                        cmp al, 'P'
6183 00009DA3 7212
                                              short loc_check_prg_fn_retn
                              <1>
                                         jb
6184 00009DA5 7711
                              <1>
                                               short loc_check_prg_fn_stc
6185 00009DA7 AC
                               <1>
                                        lodsb
6186 00009DA8 24DF
                              <1>
                                        and al, ODFh
6187 00009DAA 3C52
                              <1>
                                        cmp al, 'R'
6188 00009DAC 750A
                               <1>
                                        jne short loc_check_prg_fn_stc
6189 00009DAE AC
                               <1>
                                        lodsb
6190 00009DAF 24DF
                              <1>
                                        and al, ODFh
6191 00009DB1 3C47
                                         cmp al, 'G'
                               <1>
6192 00009DB3 7503
                               <1>
                                        jne short loc_check_prg_fn_stc
6193
                               <1>
6194 00009DB5 B050
                               <1>
                                        mov al, 'P'
6195
                               <1> loc_check_prg_fn_retn:
6196 00009DB7 C3
                               <1>
                                        retn
6197
                               <1>
                               <1> loc_check_prg_fn_stc:
6198
6199 00009DB8 F9
                               <1>
                                        stc
6200 00009DB9 C3
                               <1>
                                        retn
6201
                               <1>
6202
                               <1> loc_check_exe_fn_retn:
6203 00009DBA 28C0
                               <1> sub al, al; 0
6204 00009DBC C3
                               <1>
6205
                               <1>
6206
                               <1> find_and_list_files:
6207 00009DBD C3
                               <1> retn
6208
                               <1> set_exec_arguments:
6209 00009DBE C3
                               <1> retn
                               <1> delete_fs_directory:
6211 00009DBF 31C0
                               <1> xor eax, eax
6212 00009DC1 C3
                               <1>
                                         retn
2308
                                  %include 'trdosk4.s'; 24/01/2016
                                1
                                <1> ; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.0) - Directory Functions : trdosk4.s
  2
                                <1> ; Last Update: 29/12/2017
                                <1> ; ------
  5
  6
                                <1> ; Beginning: 24/01/2016
  7
  8
                                <1>; Assembler: NASM version 2.11 (trdos386.s)
  9
                                <1>; Derived from TRDOS Operating System v1.0 (8086) source code by Erdogan Tan
 10
                                <1>; DIR.ASM (09/10/2011)
 11
                                12
 13
                                <1>
                                <1>; DIR.ASM [ TRDOS KERNEL - COMMAND EXECUTER SECTION - DIRECTORY FUNCTIONS ]
 14
 15
                                <1>; (c) 2004-2010 Erdogan TAN [ 17/01/2004 ] Last Update: 09/10/2011
 16
                                <1> ; FILE.ASM [ FILE FUNCTIONS ] Last Update: 09/10/2011
 17
 18
                                <1> change_prompt_dir_string:
                                        ; 05/10/201<del>6</del>
 19
                                <1>
                                         ; 24/01/2016 (TRDOS 386 = TRDOS v2.0)
 20
                                <1>
 21
                                <1>
                                        ; 27/03/2011
 22
                                <1>
                                        ; 09/10/2009
 23
                                <1>
                                         ; INPUT/OUTPUT => none
 24
                                <1>
                                        ; this procedure changes current directory string/text
 25
                                <1>
                                         ; 2005
```

```
<1>
 26
 27 00009DC2 BE[E3600100]
                                 <1>
                                          mov
                                                esi, PATH Array
                                 <1> change_prompt_dir_str: ; 05/10/2016 (call from 'set_working path')
 29 00009DC7 BF[8A590100]
                                <1> mov edi, Current_Directory
 30 00009DCC 8A25[84590100]
                                <1>
                                                 ah, [Current_Dir_Level]
                                          mov
                                                set_current_directory_string
 31 00009DD2 E80700000
                                <1>
                                           call
 32 00009DD7 880D[E5590100]
                                                 [Current_Dir_StrLen], cl
                                <1>
                                 <1>
 33
 34 00009DDD C3
                                 <1>
                                          retn
 35
                                 <1>
                                 <1> set_current_directory_string:
 36
                                        ; 24/01/2016 (TRDOS 386 = TRDOS v2.0)
 37
                                          ; 27/03/2011
 38
                                 <1>
 39
                                 <1>
                                         ; 09/10/2009
                                        ; INPUT:
 40
                                 <1>
                                          ; ESI = Path Array Address
 41
                                 <1>
                                          ; EDI = Current Directory String Buffer
 42
                                 <1>
                                 <1>
                                               AH = Current Directory Level
 43
 44
                                 <1>
                                          ; OUTPUT => EAX, EBX, ESI will be changed
 45
                                 <1>
                                          ; EDI will be same with input
 46
                                 <1>
                                          ; ECX = Current Directory String Length
                                 <1>
                                          push edi
 48 00009DDE 57
                                <1>
 49 00009DDF 80FC00
                                <1>
                                          cmp
                                                ah, 0
                                                 short pass_write_path
 50 00009DE2 7652
                                <1>
                                           jna
 51 00009DE4 83C610
                                <1>
                                           add
                                                 esi, 16
 52 00009DE7 89F3
                                <1>
                                          mov ebx, esi
                                <1> loc_write_path:
<1> mov ecx
 53
                                          mov ecx, 8
 54 00009DE9 B908000000
                                <1>
 55
                                <1> path_write_dirname1:
 56 00009DEE AC
                                <1>
                                           lodsb
                                          cmp al, 20h
jna short pass_write_dirname1
 57 00009DEF 3C20
                                <1>
                             58 00009DF1 7612
 59 00009DF3 AA
                                          cmp edi, End_Of_Current_Dir_Str
jnb short pass_write_path
 60 00009DF4 81FF[E4590100]
 61 00009DFA 733A
 62 00009DFC E2F0
 63 00009DFE 803E20
 64 00009E01 7624
                                                short pass_write_dirname2
                                                short loc_put_dot_cont_ext
 65 00009E03 EB0A
                                <1> pass_write_dirname1:
                                <1>
 67 00009E05 89DE
                                          mov esi, ebx add esi, 8
 68 00009E07 83C608
                                <1>
                                           add
 69 00009E0A 803E20
                              <1>
                                           cmp byte [esi], 20h
 70 00009E0D 7618
                                <1>
                                           jna
                                                short pass_write_dirname2
                                <1> loc_put_dot_cont_ext:
                                <1> mov byte [edi], "."
 72 00009E0F C6072E
                                <1>
                                           ;mov ecx, 3
 73
                                           mov cl, 3
 74 00009E12 B103
                                <1>
 75
                                <1> loc_check_dir_name_ext:
 76 00009E14 AC
                                <1> lodsb
                               inc
<1> cmp
<1> jna
<1> mov
<1> cmp
<1> inc
                                          inc edi
cmp al, 20h
 77 00009E15 47
                                <1>
 78 00009E16 3C20
 79 00009E18 760D
                                          jna short pass_write_dirname2
                                                 [edi], al
 80 00009E1A 8807
 81 00009E1C 81FF[E4590100]
                                                 edi, End_Of_Current_Dir_Str
 82 00009E22 7312
                                                 short pass write path
 83 00009E24 E2EE
                                                 loc_check_dir_name_ext
 84 00009E26 47
                                                 edi
                                <1> pass_write_dirname2:
                             86 00009E27 FECC
                                <1> dec ah
                                                 short pass_write_path
 87 00009E29 740B
 88 00009E2B 83C310
 89 00009E2E 89DE
 90 00009E30 C6072F
 91 00009E33 47
 92 00009E34 EBB3
 94 00009E36 C60700
 95 00009E39 47
 96 00009E3A 89F9
                                                 ecx, edi
                                <1>
                                           mov
 97 00009E3C 5F
                                <1>
                                           pop
                                                 edi
                                                ecx, edi
 98 00009E3D 29F9
                                <1>
                                           sub
                                          ; ECX = Current Directory String Length
 99
                                <1>
100 00009E3F C3
                                 <1>
                                          retn
101
                                 <1>
102
                                 <1> get current directory:
                                        ; 15/\overline{1}0/2016
103
                                 <1>
104
                                 <1>
                                           ; 14/02/2016
105
                                          ; 24/01/2016 (TRDOS 386 = TRDOS v2.0)
                                 <1>
106
                                 <1>
                                           ; 27/03/2011
107
                                 <1>
                                           ; INPUT-> ESI = Current Directory Buffer
108
                                 <1>
109
                                 <1>
                                                     DL = TRDOS Logical Dos Drive Number + 1
110
                                 <1>
                                                          (0= Default/Current Drive)
111
                                 <1>
112
                                 <1>
                                              Note: Required dir buffer length may be <= 92 bytes
                                 <1>
                                                    for TRDOS (7*12 \text{ name chars} + 7 \text{ slash} + 0)
113
                                           ; OUTPUT -> ESI = Current Directory Buffer
114
                                 <1>
115
                                 <1>
                                                        EAX, EBX, ECX, EDX, EDI will be changed
116
                                 <1>
                                                        CX/CL = Current Directory String Length
117
                                 <1>
                                                      DL = Drive Number (0 based)
                                                       (If input is 0, output is current drv number)
118
                                 <1>
                                                        DH = same with input
119
                                 <1>
                                             cf = 0 \rightarrow AL = 0
120
                                 <1>
                                           ; cf = 1 -> error code in AL
121
                                 <1>
122
                                 <1>
123
                                 <1> loc_get_current_drive_0:
124 00009E40 80FA00
                                          cmp dl, 0
                                 <1>
125 00009E43 7708
                                                 short loc_get_current_drive_1
                                 <1>
                                           jа
126 00009E45 8A15[86590100]
                                                 dl, [Current Drv]
                                 <1>
                                           mov
127 00009E4B EB17
                                 <1>
                                           jmp
                                                short loc get current drive 2
                                 <1> loc get current drive 1:
                                           dec dl
129 00009E4D FECA
                                 <1>
130 00009E4F 3A15[610D0100]
                                 <1>
                                                 dl, [Last DOS DiskNo]
                                           cmp
```

```
131 00009E55 760D
                              <1>
                                      jna short loc_get_current_drive_2
132 00009E57 B80F000000
                              <1>
                                        mov
                                              eax, OFh; Invalid drive (Drive not ready!)
133 00009E5C F5
                              <1>
                                       cmc
                                             ; stc
134 00009E5D C3
                              <1>
135
                              <1>
                              <1> loc_get_current_drive_not_ready_retn:
136
                                    pop esi
137 00009E5E 5E
                              <1>
                                        ;mov eax, 15
138
                              <1>
139 00009E5F 66B80F00
                              <1>
                                        mov ax, 15; Drive not ready
140 00009E63 C3
                              <1>
                                       retn
141
                              <1>
                              <1> loc_get_current_drive_2:
142
                           <1> xor eax, eax
<1> mov ah, dl
143 00009E64 31C0
144 00009E66 88D4
                       cl> push esi
<l> mov esi, Logi
<l> add esi, eax

145 00009E68 56
146 00009E69 BE00010900
                                             esi, Logical_DOSDisks
147 00009E6E 01C6
                              <1>
148 00009E70 8A06
                                        mov al, [esi+LD_Name]
149 00009E72 3C41
                              <1>
                                        cmp
                                              al, 'A'
                                       jb
                              <1>
150 00009E74 72E8
                                              short loc_get_current_drive_not_ready_retn
                              <1>
151
                                     mov
152 00009E76 8A667F
                              <1>
                                              ah, [esi+LD_CDirLevel]
153 00009E79 08E4
                              <1>
                                       or
                                              ah, ah
154 00009E7B 7506
                              <1>
                                      jnz
                                              short loc_get_current_drive_3
155
                              <1>
156
                              <1>
                                        ;xor ah, ah; mov ah, 0
157 00009E7D 8826
                              <1>
                                        mov [esi], ah
158 00009E7F 31C9
                              <1>
                                        xor ecx, ecx
159 00009E81 EB1C
                              <1>
                                        jmp short loc_get_current_drive_4
160
                              <1>
161
                              <1> loc_get_current_drive_3:
                                    mov edi, PATH Array
162 00009E83 BF[E3600100]
                              <1>
                                        push edi
163 00009E88 57
                              <1>
164 00009E89 81C68000000
                              <1>
                                       add esi, LD_CurrentDirectory
165 00009E8F B920000000
                              <1>
                                       mov ecx, 32
166 00009E94 F3A5
                              <1>
                                        rep
                                              movsd
                                     pop
167 00009E96 5E
                              <1>
                                              esi ; Path Array Address
168 00009E97 5F
                              <1>
                                              edi ; pushed esi (current dir buffer offset)
                                    pop
169
                              <1>
170 00009E98 E841FFFFF
                              <1>
                                        call
                                             set_current_directory_string
171 00009E9D 89FE
                              <1>
                                      mov
                                             esi, edi
                              <1>
                              <1> loc_get_current_drive_4:
173
174 00009E9F 30C0
                              <1>
                                      xor al, al
175 00009EA1 C3
                              <1>
                                        retn
176
                               <1>
177
                               <1> change_current_directory:
                                    ; 19/02/2016
178
                               <1>
179
                               <1>
                                        ; 11/02/2016
                               <1>
                                       ; 10/02/2016
180
                                      ; 08/02/2016
181
                               <1>
                                      ; 06/02/2016 (TRDOS 386 = TRDOS v2.0)
; 18/09/2011 (DIR.ASM, 09/10/2011)
182
                               <1>
183
                               <1>
                                      ; 04/10/2009
184
                               <1>
                                      ; 2005
185
                               <1>
                                      ; INPUT ->
; ESI
186
                               <1>
187
                               <1>
                                              ESI = Directory string
                               <1>
                                             ah = CD command (CDh = save current dir string)
188
189
                               <1>
                                        ; OUTPUT ->
                               <1>
                                        ; EDI = DOS Drive Description Table
190
191
                               <1>
                                              cf = 1 \rightarrow error
192
                               <1>
                                               EAX = Error code
                                        ;
                                              cf = 0 -> successful
                               <1>
193
                                        ;
                                               ESI = PATH_Array
194
                               <1>
195
                               <1>
                                                 EAX = Current Directory First Cluster
                                        ;
196
                               <1>
197
                               <1>
                                        ; (EAX, EBX, ECX, EDX, ESI, EDI will be changed)
198
                               <1>
199 00009EA2 8825[71610100]
                              <1>
                                             [CD_COMMAND], ah
200 00009EA8 803E2F
                              <1>
                                       cmp byte [esi], '/'
                                        jne short loc_ccd_cdir_level
201 00009EAB 7505
                              <1>
202 00009EAD 46
                              <1>
                                        inc
                                              esi
203 00009EAE 30C0
                                      xor al, al
jmp short loc_ccd_parse_path_name
                              <1>
204 00009EB0 EB05
                              <1>
205
                              <1> loc_ccd_cdir_level:
                              206 00009EB2 A0[84590100]
                              <1> loc_ccd_parse_path_name:
208 00009EB7 88C4
                                     mov ah, al
                              <1>
209 00009EB9 BF[E3600100]
                               <1>
                                       mov edi, PATH_Array
210
                               <1>
                               <1>; Reset directory levels > cdir level
211
212
                                   ; is this required !?
213
                               <1>
214
                               <1>
                                        ; Relations:
215
                               <1>
                                        ; MAINPROG.ASM (pass_ccdrv_reset_cdir_FAT_fcluster)
216
                               <1>
                                        ; proc_parse_dir_name,
                                        ; proc_change_current_directory (this procedure)
217
                               <1>
218
                               <1>
                                        ; proc_change_prompt_dir_string
219
                               <1>
220 00009EBE 0FB6C8
                               <1>
                                        movzx ecx, al
221 00009EC1 FEC1
                               <1>
                                        inc cl
222 00009EC3 C0E104
                               <1>
                                        shl
                                              cl, 4
223 00009EC6 01CF
                              <1>
                                        add
                                              edi, ecx
224 00009EC8 B107
                                              cl, 7
                              <1>
225 00009ECA 28C1
                              <1>
                                        sub
                                              cl, al
226 00009ECC C0E102
                              <1>
                                        shl
                                              cl, 2
227 00009ECF 89C3
                              <1>
                                        mov
                                              ebx, eax
228 00009ED1 31C0
                              <1>
                                        xor
                                              eax, eax; 0
229 00009ED3 F3AB
                              <1>
                                        rep
                                              stosd
230 00009ED5 89D8
                              <1>
                                        mov
                                              eax, ebx
231
                               <1>
232 00009ED7 BF[E3600100]
                               <1>
                                              edi, PATH Array
                                        mov
233
                               <1>
234 00009EDC 803E20
                               <1>
                                              byte [esi], 20h
235 00009EDF F5
                               <1>
                                        cmc
```

```
236 00009EE0 7305
                                <1>
                                          jnc short pass_ccd_parse_dir_name
237
                                <1>
238
                                <1>
                                               ; ESI = Path name
                                                ; AL = CCD Level
239
                                <1>
240 00009EE2 E872010000
                                <1>
                                            call parse_dir_name
241
                                <1>
                                               ; AL = CCD Level
                                                ; AH = Last Dir Level
242
                                <1>
                                <1>
243
                                               ; (EDI = PATH_Array)
244
                                <1>
                                <1> pass_ccd_parse_dir_name:
245
246 00009EE7 9C
                                <1>
                                        pushf
247
                                <1>
                                         ;mov [CCD Level], al
248
                                <1>
249
                                <1>
                                          ;mov [Last_Dir_Level], ah
250 00009EE8 66A3[67610100]
                                <1>
                                         mov [CCD_Level], ax
251
                                <1>
252 00009EEE 31DB
                                <1>
                                         xor
                                                ebx, ebx
253 00009EF0 8A3D[86590100]
                                                bh, [Current_Drv]
                                <1>
                                         mov
                                                esi, Logical DOSDisks
254 00009EF6 BE00010900
                                <1>
                                          mov
255 00009EFB 01DE
                                <1>
                                         add
                                                esi, ebx
256
                                <1>
257 00009EFD 9D
                                <1>
                                         popf
258 00009EFE 720A
                                <1>
                                                short loc_ccd_bad_path_name_retn
                                         jс
259
                                <1>
260 00009F00 8935[63610100]
                                <1>
                                                [CCD_DriveDT], esi
                                         mov
261
                                <1>
262 00009F06 3C07
                                <1>
                                                al, 7
                                          cmp
263 00009F08 7209
                                <1>
                                          jb
                                                short loc_ccd_load_child_dir
                                <1>
264
265
                                <1> loc_ccd_bad_path_name_retn:
266 00009F0A 87F7
                                <1>
                                         xchg esi, edi
267 00009F0C B813000000
                                <1>
                                          mov
                                                eax, 19; Bad directory/path name
268 00009F11 F9
                                <1>
                                         stc
269
                                <1> loc ccd retn p:
270 00009F12 C3
                                <1>
                                         retn
271
                                <1>
272
                                <1> loc ccd load child dir:
                                       ; AL = CCD_Level
273
                                <1>
274 00009F13 08C0
                                <1>
                                               al, al
275 00009F15 7468
                               <1>
                                                short loc_ccd_load_root_dir
                                         jΖ
276
                               <1>
                                               cx, ax
277 00009F17 6689C1
                               <1>
                                         mov
                                         shl al, 4
278 00009F1A C0E004
                               <1>
                                         movzx esi, al
279 00009F1D 0FB6F0
                               <1>
280 00009F20 01FE
                               <1>
                                         add esi, edi ; offset PATH_Array
281
                               <1>
282 00009F22 8B460C
                               <1>
                                         mov
                                               eax, [esi+12]
283 00009F25 38E9
                                <1>
                                                cl, ch
                                         cmp
284 00009F27 0F84FA000000
                                <1>
                                          jе
                                                loc_ccd_load_sub_directory
285 00009F2D A3[80590100]
                                <1>
                                                [Current Dir FCluster], eax
                                         mov
286
                                <1>
287
                                <1> loc ccd load child dir next:
288 00009F32 83C610
                                <1>
                                         add esi, 16; DOS DirEntry Format FileName Address
289
                                <1>
290
                                <1>
                                         ; Directory attribute : 10h
291 00009F35 B010
                                <1>
                                         mov al, 00010000b; 10h (Attrib AND mask)
                                         ;mov ah, 11001000b; C8h
                                <1>
293
                                <1>
                                         ; Volume name attribute: 8h
294 00009F37 B408
                                <1>
                                               ah, 00001000b; 08h (Attrib NAND, AND --> zero mask)
295
                                <1>
296 00009F39 6631C9
                                <1>
                                         xor cx, cx
297 00009F3C E8B5010000
                                <1>
                                         call locate_current_dir_file
298 00009F41 7353
                                <1>
                                         jnc short loc_ccd_set_dir_cluster_ptr
299
                                <1>
                                         ; 19/02/2016
300
                                <1>
301
                                <1>
                                         ;mov edi, [CCD_DriveDT]
302 00009F43 8A25[67610100]
                                <1>
                                         mov
                                               ah, [CCD Level]
                                         cmp byte [CD_COMMAND], OCDh ;'CD' command or another
303 00009F49 803D[71610100]CD
                                <1>
304 00009F50 7509
                                <1>
                                               short loc_ccd_load_child_dir_err
305
                                <1>
                                         ; It is better to save recent successful part
306
                                <1>
                                         ; of the (requested) path as current directory.
307
                                <1>
                                         ; (Otherwise the path would be reset to back
                                         ; on the next 'CD' command.)
308
                                <1>
309 00009F52 88E1
                                         mov cl, ah
                                <1>
                                         push eax
call loc_ccd_save_current_dir
310 00009F54 50
                                <1>
311 00009F55 E8E3000000
                                <1>
312 00009F5A 58
                                <1>
                                         pop
                                               eax
313
                                <1> loc_ccd_load_child_dir_err:
                                          cmp al, 3; AL = 2 \Rightarrow File not found error
314 00009F5B 3C03
                                <1>
315 00009F5D 7202
                                                \verb|shortloc_ccd_path_not_found_retn|\\
                                <1>
                                          jЬ
316 00009F5F F9
                                <1>
                                          stc
317 00009F60 C3
                                <1>
                                          retn
318
                                <1>
                                <1> loc_ccd_path_not_found_retn:
319
320 00009F61 B003
                                        mov al, 3; Path not found
                                <1>
321 00009F63 C3
                                <1>
                                         retn
                                <1>
323
                                <1> loc_ccd_load_FAT_root_dir:
                                         cmp byte [Current FATType], 2
324 00009F64 803D[85590100]02
                                <1>
                                                short loc ccd load FAT32 root dir
325 00009F6B 776B
                                <1>
                                          jа
326
                                <1>
327
                                <1>
                                         ;mov esi, [CCD_DriveDT]
328
                                <1>
                                         ;push esi
329 00009F6D E8B51D0000
                                <1>
                                         call load_FAT_root_directory
                                         ;pop edi ; Dos Drv Description Table
330
                                <1>
331
                                <1>
332 00009F72 89F7
                               <1>
                                         mov
                                                edi, esi
333 00009F74 BE[E3600100]
                                                esi, PATH_Array
                               <1>
                                         mov
334 00009F79 7297
                                <1>
                                         jс
                                                short loc_ccd_retn_p
335
                               <1>
336 00009F7B 31C0
                               <1>
                                         xor eax, eax
                                        jmp short loc_ccd_set_cdfc
337 00009F7D EB78
                                <1>
338
                                <1>
                                <1> loc ccd load root dir:
339
340 00009F7F 803D[85590100]01
                               <1>
                                        cmp byte [Current_FATType], 1
```

```
341 00009F86 73DC
                               <1>
                                         jnb short loc_ccd_load_FAT_root_dir
                               <1>
343
                               <1> loc_ccd_load_FS_root_dir:
344 00009F88 E8611E0000
                              <1>
                                       call load_FS_root_directory
345 00009F8D EB5C
                               <1>
                                         jmp short pass_ccd_load_FAT_sub_directory
346
                               <1>
                              <1> loc ccd load_FS_sub_directory_next:
348 00009F8F E85B1E0000
                                      call load_FS_sub_directory
                              <1>
349 00009F94 EB1F
                               <1>
                                         jmp short pass_ccd_set_dir_cluster_ptr
350
                              <1>
351
                              <1> loc_ccd_set_dir_cluster_ptr:
                                     ; EDI = Directory Entry
352
                               <1>
353 00009F96 668B4714
                              <1>
                                        mov ax, [edi+20]; First Cluster High Word
354 00009F9A C1E010
                              <1>
                                        shl eax, 16
355 00009F9D 668B471A
                                      mov ax, [edi+26]; First Cluster Low Word
                              <1>
356
                               <1>
                                    mov esi, [CCD_DriveDT]
cmp byte [Current_FATType], 1
357 00009FA1 8B35[63610100]
                              <1>
358 00009FA7 803D[85590100]01 <1>
                                               short loc ccd load FS sub directory next
359 00009FAE 72DF
                               <1>
                                         jb
360
                               <1>
                                        ;push esi
361 00009FB0 E8FD1D0000
                                         call load_FAT_sub_directory
                               <1>
362
                               <1>
                                        ;pop edi ; Dos Drv Description Table
363
                               <1>
                               <1> pass_ccd_set_dir_cluster_ptr:
                                     ;mov edi, esi
365
                               <1>
366 00009FB5 BE[E3600100]
                               <1>
                                         mov
                                              esi, PATH_Array
367 00009FBA 7264
                               <1>
                                               short loc_ccd_retn_c
                                        jс
368
                               <1>
369 00009FBC A1[B1600100]
                               <1>
                                              eax, [DirBuff_Cluster]
                                        mov
370
                               <1>
                                    inc byte [CCD_Level]
371 00009FC1 FE05[67610100]
                               <1>
372 00009FC7 0FB61D[67610100]
                              <1>
                                        movzx ebx, byte [CCD Level]
                                        shl bl, 4; * 16 (<= 128)
373 00009FCE C0E304
                               <1>
374 00009FD1 01DE
                               <1>
                                        add esi, ebx; 19/02/2016
                                        mov [esi+12], eax
jmp short loc_ccd_set_cdfc
375 00009FD3 89460C
                               <1>
376 00009FD6 EB1F
                              <1>
                                        jmp
377
                               <1>
378
                               <1> loc_ccd_load_FAT32_root_dir:
                                    mov esi, PATH Array
379 00009FD8 BE[E3600100]
                               <1>
380 00009FDD 8B460C
                                              eax. [esi+12]
                               <1>
                                        mov
381 00009FE0 8B35[63610100]
                               <1>
                                        mov esi, [CCD_DriveDT]
                               <1>
                               <1> loc_ccd_load_FAT_sub_directory:
383
                                    ;push esi
                               <1>
385 00009FE6 E8C71D0000
                               <1>
                                         call load_FAT_sub_directory
                                        ;pop edi ; Dos Drv Description Table
386
                               <1>
387
                               <1>
                               <1> pass_ccd_load_FAT_sub_directory:
388
389
                               <1>
                                        ;mov edi, esi
390 00009FEB BE[E3600100]
                               <1>
                                              esi, PATH Array
                                         mov
391 00009FF0 722E
                                              short loc_ccd_retn_c
                               <1>
392
                               <1>
393 00009FF2 A1[B1600100]
                               <1>
                                        mov eax, [DirBuff_Cluster]
394
                               <1>
395
                               <1> loc_ccd_set_cdfc:
396 00009FF7 8A0D[67610100]
                               <1>
                                              cl, [CCD_Level]
                                        mov
397 00009FFD 880D[84590100]
                               <1>
                                               [Current Dir Level], cl
                                        mov
398 0000A003 A3[80590100]
                               <1>
                                              [Current_Dir_FCluster], eax
                                        mov
399
                               <1>
400 0000A008 8A2D[68610100]
                               <1>
                                        mov
                                               ch, [Last_Dir_Level]
401 0000A00E 38E9
                               <1>
                                        cmp
                                              cl, ch
402 0000A010 0F821CFFFFFF
                               <1>
                                        jb
                                               loc_ccd_load_child_dir_next
                               <1>
404 0000A016 803D[71610100]CD
                               <1>
                                              byte [CD_COMMAND], OCDh ; 'CD' command or another
405 0000A01D 741E
                               <1>
                                              short loc_ccd_save_current_dir
                                       je
406
                               <1>
407
                               <1>
                                          ; jne -> don't save, restore (the previous cdir) later !
408
                               <1>
                                         ; (saving the cdir would prevent previous cdir restoration!)
409
                               <1>
410 0000A01F F8
                               <1>
                                         clc
411
                               <1>
                               <1> loc ccd retn c:
412
413 0000A020 8B3D[63610100]
                                        mov edi, [CCD_DriveDT]
                               <1>
414 0000A026 C3
                               <1>
415
                               <1>
416
                               <1> loc_ccd_load_sub_directory:
417 0000A027 8B35[63610100]
                               <1> mov esi, [CCD DriveDT]
418 0000A02D 803D[85590100]01
                                         cmp
                              <1>
                                              byte [Current_FATType], 1
419 0000A034 73B0
                                              short loc ccd load FAT sub directory
                               <1>
                                         jnb
                                         call load_FS_sub_directory
420 0000A036 E8B41D0000
                               <1>
421 0000A03B EBAE
                               <1>
                                        jmp short pass_ccd_load_FAT_sub_directory
                               <1>
                               <1> loc_ccd_save_current_dir:
424 0000A03D BE[E3600100]
                                        mov esi, PATH_Array ; 19/02/2016
                               <1>
425 0000A042 8B3D[63610100]
                                              edi, [CCD_DriveDT]
                               <1>
                                        mov
                                       push edi
426 0000A048 57
                              <1>
                                        add edi, LD CDirLevel
427 0000A049 83C77F
                              <1>
                                       mov [edi], cl
428 0000A04C 880F
                              <1>
429 0000A04E 47
                               <1>
                                               edi ; LD_CurrentDirectory
                                        inc
                                        push esi
430 0000A04F 56
                              <1>
                                        ;mov ecx, 32 ; always < 65536 (in this procedure)
431
                               <1>
                                        mov
rep
432 0000A050 66B92000
                               <1>
                                              cx, 32
433 0000A054 F3A5
                               <1>
                                              movsd
                                        ; Current directory has been saved to
                               <1>
434
435
                               <1>
                                        ; the DOS drive description table, cdir area!
436 0000A056 5E
                               <1>
                                        pop esi ; PATH_Array
                                        pop edi ; Dos Drv Description Table
437 0000A057 5F
                               <1>
                               <1>
438
439 0000A058 C3
                               <1>
                               <1>
440
                               <1> parse dir name:
441
                                     ; 11/02/2016
442
                               <1>
                                         ; 10/02/2016
443
                               <1>
4\,4\,4
                               <1>
                                        ; 07/02/2016 (TRDOS 386 = TRDOS v2.0)
445
                               <1>
                                        ; 18/09/2011
```

```
; 17/10/2009
                                      ; INPUT ->
447
                               <1>
                                              ESI = ASCIIZ Directory String Address
448
                               <1>
                                        ;
449
                               <1>
                                              AL = Current Directory Level
                               <1>
                                            EDI = Destination Adress
450
451
                               <1>
                                                   (8 levels, each one 12+4 byte)
                                        ; OUTPUT ->
452
                                        ; EDI = Dir Entry Formatted Array
                               <1>
453
454
                               <1>
                                                with zero cluster pointer at the last level
                                              AH = Last Dir Level
455
                               <1>
                                        ;
456
                               <1>
                                              AL = Current Dir Level
457
                               <1>
458
                               <1>
                                        ; (esi, ebx, ecx will be changed)
459
                               <1>
                               <1>
460
                                        ;mov [PATH_Array_Ptr], edi
461 0000A059 88C4
                               <1>
                                        mov
                                              ah, al
462 0000A05B 66A3[08620100]
                              <1>
                                        mov
                                             [PATH CDLevel], ax
                              <1> repeat ppdn check slash:
463
464 0000A061 AC
                               <1>
                                       lodsb
                                        cmp al, '/'
465 0000A062 3C2F
                              <1>
                                              short repeat_ppdn_check_slash
466 0000A064 74FB
                              <1>
                                              al, 21h
467 0000A066 3C21
                              <1>
                                        cmp
                                        jb
468 0000A068 7219
                              <1>
                                              short loc_ppdn_retn
                                    push edi
469 0000A06A 57
                              <1>
470
                              <1> loc_ppdn_get_dir_name:
                                     mov ecx, 12
mov edi, Dir_File_Name
471 0000A06B B90C00000
                              <1>
472 0000A070 BF[0A620100]
                              <1>
                              <1> repeat_ppdn_get_dir_name:
473
474 0000A075 AA
                              <1>
                                        stosb
475 0000A076 AC
                              <1>
                                        lodsb
                                        cmp al, '/'
476 0000A077 3C2F
                              <1>
477 0000A079 740A
                              <1>
                                              short loc_check_level_dot_conv_dir_name
                                        jе
478 0000A07B 3C20
                              <1>
                                              al, 20h
                                        cmp
479 0000A07D 7605
                              <1>
                                             short loc_ppdn_end_of_path_scan
                                        jna
480 0000A07F E2F4
                              <1>
                                        loop repeat_ppdn_get_dir_name
                                        pop
481 0000A081 5F
                              <1>
                                               edi
482 0000A082 F9
                              <1>
                                        stc
483
                              <1> loc_ppdn_retn:
484 0000A083 C3
                               <1>
                              <1>
485
486
                              <1> loc_ppdn_end_of_path_scan:
487 0000A084 4E
                              <1>
                                        dec esi
                              <1> loc_check_level_dot_conv_dir_name:
488
489 0000A085 31C0
                              <1>
                                       xor eax, eax
490 0000A087 AA
                              <1>
                                        stosb
                                        mov ebx, esi
mov esi, Dir_File_Name
491 0000A088 89F3
                              <1>
492 0000A08A BE[0A620100]
                              <1>
493 0000A08F AC
                              <1>
                                        lodsb
                              <1> repeat_ppdn_name_check_dot:
494
                                       cmp -al, '.'
495 0000A090 3C2E
                              <1>
496 0000A092 7509
                                             short loc_ppdn_convert_sub_dir_name
                              <1>
                                        jne
                              <1> repeat_ppdn_name_dot_dot:
498 0000A094 AC
                              <1>
                                    lodsb
                                        cmp al, '.'
499 0000A095 3C2E
                              <1>
500 0000A097 743E
                              <1>
                                              short loc_ppdn_dot_dot
                                        jе
501 0000A099 3C21
                              <1>
                                        cmp
502 0000A09B 7226
                              <1>
                                      jb
                                              short pass_ppdn_convert_sub_dir_name
503
                              <1> loc_ppdn_convert_sub_dir_name:
                                    mov
                                              ah, [PATH_Level]
504 0000A09D 8A25[09620100]
                              <1>
                                              ah, 7
505 0000A0A3 80FC07
                              <1>
                                        cmp
506 0000A0A6 731B
                              <1>
                                        jnb
                                              short pass_ppdn_convert_sub_dir_name
507 0000A0A8 FEC4
                               <1>
                                        inc
                                              ah
508 0000A0AA 8825[09620100]
                              <1>
                                              [PATH_Level], ah
                                        mov
                                    mov
509 0000A0B0 BE[0A620100]
                              <1>
                                              esi, Dir_File_Name
510
                              <1>
                                      ;mov edi, [PATH_Array_Ptr]
511 0000A0B5 B010
                              <1>
                                        mov
                                              al, 16
512 0000A0B7 F6E4
                              <1>
                                      mul
                                              ah
513 0000A0B9 8B3C24
                              <1>
                                      mov edi, [esp]
                               <1>
                                        ;push edi
515 0000A0BC 01C7
                              <1>
                                        add edi, eax
516 0000A0BE E82A030000
                              <1>
                                        call convert_file_name
                                        ;pop edi
517
                               <1>
                               <1> pass_ppdn_convert_sub_dir_name:
518
519 0000A0C3 89DE
                              <1>
                                       mov
                                             esi, ebx
                               <1> repeat_ppdn_check_last_slash:
520
521 0000A0C5 AC
                              <1>
                                        lodsb
                                        cmp al, '/'
522 0000A0C6 3C2F
                              <1>
                                              short repeat_ppdn_check_last_slash
523 0000A0C8 74FB
                              <1>
                                        je
524 0000A0CA 3C21
                               <1>
                                              al, 21h
                                        jnb short loc_ppdn_get_dir_name
525 0000A0CC 739D
                               <1>
526
                               <1> end_of_parse_dir_name:
527 0000A0CE 5F
                                        pop edi
                               <1>
528 0000A0CF F5
                               <1>
                                        CMC
529
                               <1>
                                              al, [PATH CDLevel]
                                        ; mov
                                        ;mov ah, [PATH_Level]
                               <1>
530
531 0000A0D0 66A1[08620100]
                                               ax, [PATH CDLevel]
                               <1>
                                        mov
532 0000A0D6 C3
                              <1>
                                        retn
                               <1>
533
534
                               <1> loc_ppdn_dot_dot:
535 0000A0D7 AC
                              <1>
                                        lodsb
536 0000A0D8 3C21
                              <1>
                                        cmp al, 21h
537 0000A0DA 73F2
                               <1>
                                        jnb
                                              short end_of_parse_dir_name
                              <1> loc ppdn dot dot prev level:
538
539 0000A0DC 66A1[08620100]
                                    mov ax, [PATH CDLevel]
                              <1>
                                              ah, 1
540 0000A0E2 80EC01
                              <1>
                                        sub
541 0000A0E5 80D400
                              <1>
                                        adc
                                              ah, 0
542 0000A0E8 38E0
                              <1>
                                        cmp
                                              al, ah
543 0000A0EA 7602
                              <1>
                                              short pass_ppdn_set_al_to_ah
                                        jna
544 0000A0EC 88E0
                               <1>
                                        mov
                                              al, ah
                               <1> pass_ppdn_set_al_to_ah:
545
546 0000A0EE 66A3[08620100]
                                        mov [PATH CDLevel], ax
                              <1>
547 0000A0F4 EBCD
                               <1>
                                        jmp
                                             short pass_ppdn_convert_sub_dir_name
548
                               <1>
549
                               <1> locate_current_dir_file:
550
                                    ; 20/11/2017
                               <1>
```

<1>

446

```
; 14/02/2016
551
                                <1>
552
                                <1>
                                          ; 13/02/2016
553
                                          ; 10/02/2016
                                <1>
554
                                <1>
                                         ; 06/02/2016 (TRDOS 386 = TRDOS v2.0)
555
                                <1>
                                          ; 14/08/2010
556
                                <1>
                                          ; 19/09/2009
557
                                <1>
                                           ; 2005
                                          ; INPUT ->
                                <1>
558
559
                                <1>
                                                ESI = DOS DirEntry Format FileName Address
560
                                <1>
                                                AL = Attributes Mask
561
                                <1>
                                                (<AL AND EntryAttrib> must be equal to AL)
562
                                <1>
                                                AH = Negative Attributes Mask (If AH>0)
563
                                <1>
                                                (<AH AND EntryAttrib> must be ZERO)
564
                                <1>
                                                CH > 0 Find First Free Dir Entry or Deleted Entry
                                                CL = 0 -> Return the First Free Dir Entry
                                <1>
565
566
                                <1>
                                                CL = E5h -> Return the 1st deleted entry
                                                CL = FFh -> Return the 1st deleted or free entry
567
                                <1>
                                                {\tt CL} > 0 and {\tt CL} <> E5h and {\tt CL} <> FFh -> Return the first
568
                                                     proper entry (which fits with Atributes Masks)
569
                                <1>
                                                CX = 0 Find Valid File/Directory/VolumeName
570
                                <1>
                                                ? = Any One Char
571
                                <1>
                                                * = Every Chars
572
                                <1>
                                          ; OUTPUT ->
573
                                <1>
574
                                <1>
                                                EDI = Directory Entry Address (in Directory Buffer)
575
                                                ESI = DOS DirEntry Format FileName Address
                                <1>
576
                                <1>
                                                CF = 0 -> No Error, Proper Entry,
577
                                <1>
                                                DL = Attributes
578
                                <1>
                                                DH = Previous Entry Attr (LongName Check)
579
                                <1>
                                                AL > 0 -> Ambiguous filename wildcard "?" used
                                                AH > 0 -> Ambiguous filename wildcard "*" used
                                <1>
580
581
                                <1>
                                                AX = 0 -> Filename full fits with directory entry
582
                                <1>
                                                CH = The 1st Name Char of Current Dir Entry
                                <1>
                                                CF = 1 -> Proper entry not found, Error Code in EAX/AL
583
584
                                <1>
                                                CL = 0 and CH = 0 -> Free Entry (End Of Dir)
                                                CL = 0 and CH = E5h -> Deleted Entry fits with filters
585
                                <1>
586
                                <1>
                                                CL > 0 -> Entry not found, CH invalid
                                                CF = 0 \rightarrow
587
                                <1>
                                <1>
                                                EBX = Current Directory Entry Index/Number (BX)
588
589
                                <1>
590
                                <1>
                                               word [DirBuff_EntryCounter], 0 ; Zero Based
                                          ; mov
591
                                <1>
592 0000A0F6 8935[6B610100]
                                <1>
                                          mov
                                                 [CDLF FNAddress], esi
593 0000A0FC 66A3[69610100]
                                                 [CDLF_AttributesMask], ax
                                <1>
                                          mov
594 0000A102 66890D[6F610100]
                                <1>
                                                 [CDLF_DEType], cx
595
                                <1>
596 0000A109 31DB
                                <1>
                                                 ebx, ebx
597 0000A10B 881D[80610100]
                                <1>
                                                 [PreviousAttr], bl ; 0 ; 13/02/2016
                                          mov
                                <1>
                                                bh, [Current Drv]
599 0000A111 8A3D[86590100]
                                <1>
                                <1>
<1>
<1>
<1>
600 0000A117 381D[AC600100]
                                                byte [DirBuff ValidData], bl; 0
                                          cmp
601 0000A11D 761D
                                                short loc_lcdf_reload_current_dir2
602 0000A11F 8A1D[AA600100]
                                          mov
                                                 bl, [DirBuff_DRV]
                                <1>
603 0000A125 80EB41
                                         sub bl, 'A'
                               <1> cmp <1> jne <1> mov <1> cmp
                                         cmp bh, bl
604 0000A128 38DF
605 0000A12A 750E
                                                short loc_lcdf_reload_current_dir1
606 0000A12C 8B15[B1600100]
                                                edx, [DirBuff_Cluster]
607 0000A132 3B15[80590100]
                                                edx, [Current Dir FCluster]
608 0000A138 7412
                                <1>
                                                short loc_cdir_locatefile_search
                                        jе
609
                                <1>
                                <1> loc_lcdf_reload_current_dir1:
610
611 0000A13A 30DB
                                <1>
                                         xor bl, bl
                                <1> loc lcdf reload current dir2:
613 0000A13C 89DE
                                         mov esi, ebx
                                <1>
614 0000A13E 81C600010900
                                <1>
                                          add esi, Logical_DOSDisks
                                          call reload_current_directory
615 0000A144 E874000000
                                <1>
616 0000A149 735D
                                <1>
                                          jnc
                                                short loc_locatefile_search_again
617 0000A14B C3
                                <1>
                                          retn
618
                                <1>
                                <1> loc_cdir_locatefile_search:
619
                                <1> xor ebx, ebx
620 0000A14C 31DB
621 0000A14E 55
                                <1>
                                          push ebp; 20/11/2017
622 0000A14F E8A6000000
                                <1>
                                          call find_directory_entry
623 0000A154 5D
                                                ebp ; 20/11/2017
                                <1>
                                          pop
624 0000A155 7349
                                         jnc short loc cdir locate file retn
                                <1>
                                <1>
625
                                <1> loc_locatefile_check_stc_reason:
626
627 0000A157 08ED
                                <1>
                                         or ch, ch
628 0000A159 7444
                                <1>
                                          jz
                                                 short loc_cdir_locate_file_stc_retn
629
                                <1>
630
                                <1> loc_locatefile_check_next_entryblock:
631 0000A15B 8A3D[86590100]
                                <1>
                                         mov bh, [Current_Drv]
                                                bl, bl
632 0000A161 28DB
                                <1>
                                          sub
633 0000A163 0FB7F3
                                <1>
                                          movzx esi, bx
                                                   esi, Logical DOSDisks
634 0000A166 81C600010900
                                <1>
                                <1>
636 0000A16C 803D[84590100]00
                                <1>
                                          cmp
                                                byte [Current_Dir_Level], 0
                                        jna short loc locatefile check FAT type
637 0000A173 760A
                                <1>
638
                                <1>
639 0000A175 803D[85590100]01
                                <1>
                                                 byte [Current_FATType], 1
                                         cmp
640 0000A17C 730A
                                                short loc locatefile load subdir cluster
                                <1>
                                          inb
641 0000A17E C3
                                <1>
642
                                <1>
                                <1> loc_locatefile_check_FAT_type:
643
644 0000A17F 803D[85590100]03
                                         cmp byte [Current FATType], 3
                               <1>
645 0000A186 7218
                                <1>
                                                short loc_cdir_locate_file_retn
646
                                <1>
                                <1> loc locatefile load subdir cluster:
648 0000A188 A1[B1600100]
                               <1>
                                        mov eax, [DirBuff_Cluster]
649 0000A18D E83A1A0000
                                          call get_next_cluster
                                <1>
                                          jnc short loc_locatefile_next_cluster
650 0000A192 730D
                               <1>
                                     jnz short loc_locatefile_drive_not_ready_read_err
                               <1>
651 0000A194 09C0
652 0000A196 7507
                               <1>
653 0000A198 F9
                               <1>
                               <1> loc_locatefile_file_notfound:
655 0000A199 B802000000
                                        mov eax, 2 ; File/Directory/VolName not found
                                <1>
```

```
656 0000A19E C3
                                 <1>
                                           retn
                                 <1>
                                 <1> loc_locatefile_drive_not_ready_read_err:
658
659
                                 <1>
                                       ;mov eax, 17 ;Drive not ready or read error
                                 <1> loc_cdir_locate_file_stc_retn:
660
661 0000A19F F5
                                 <1>
                                          cmc ;stc
                                 <1> loc cdir locate file retn:
662
663 0000A1A0 C3
                                 <1>
                                         retn
664
                                 <1>
665
                                 <1> loc locatefile next cluster:
666 0000A1A1 E80C1C0000
                                 <1>
                                         call load_FAT_sub_directory
                                           ;jc short loc_locatefile_drive_not_ready_read_err
jc short loc_cdir_locate_file_retn
667
                                 <1>
668 0000A1A6 72F8
                                 <1>
669
                                 <1>
                                 <1> loc_locatefile_search_again:
670
                                      mov esi, [CDLF_FNAddress]
671 0000A1A8 8B35[6B610100]
                                 <1>
                                                 ax, [CDLF_AttributesMask]
672 0000A1AE 66A1[69610100]
                                 <1>
                                           mov
673 0000A1B4 668B0D[6F610100]
                                 <1>
                                           mov
                                                 cx, [CDLF_DEType]
674 0000A1BB EB8F
                                 <1>
                                                 short loc_cdir_locatefile_search
                                           jmp
675
                                 <1>
                                 <1> reload_current_directory:
676
                                        \frac{1}{3} 06/02/\frac{1}{2}016 (TRDOS 386 = TRDOS v2.0)
677
                                 <1>
                                           ; 13/06/2010
678
                                 <1>
679
                                 <1>
                                           ; 22/09/2009
                                 <1>
680
681
                                 <1>
                                           ; INPUT ->
682
                                 <1>
                                                  ESI = Dos drive description table address
683
                                 <1>
                                 <1>
                                           ;mov al, [esi+LD_FATType]
685 0000A1BD A0[85590100]
                                 <1>
                                                 al, [Current_FATType]
                                           mov
686 0000A1C2 3C02
                                 <1>
                                           cmp
                                                 al, 2
687 0000A1C4 7729
                                 <1>
                                                  short loc reload FAT sub directory
                                           jа
688 0000A1C6 8A25[84590100]
                                                  ah, [Current_Dir_Level]
                                 <1>
                                           mov
689 0000A1CC 08C0
                                <1>
                                                 al, al
                                         jz
                                                  short loc_reload_FS_directory
690 0000A1CE 740A
                                 <1>
                                <1> or ah, ah
<1> jnz short loc_reload_FAT_sub_directory
691 0000A1D0 08E4
692 0000A1D2 751B
                                <1> loc_reload_FAT_12_16_root_directory:
693
                                       call load_FAT_root_directory
694 0000A1D4 E84E1B0000
                                 <1>
695 0000A1D9 C3
                                <1>
                                           retn
                                <1> loc_reload_FS_directory:
696
                                       and ah, ah
jnz short loc_reload_FS_sub_directory
697 0000A1DA 20E4
                                <1>
698 0000A1DC 7506
                                <1>
699
                                <1> loc_reload_FS_root_directory:
700 0000A1DE E80B1C0000
                                <1> call load_FS_root_directory
701 0000A1E3 C3
                                <1>
                                           retn
                                <1> loc_reload_FS_sub_directory:
                                      mov eax, [Current_Dir_FCluster]
703 0000A1E4 A1[80590100]
                                <1>
                                           call load FS sub directory
704 0000A1E9 E8011C0000
                                <1>
                                         retn
705 0000A1EE C3
                                 <1>
706
                                 <1> loc_reload_FAT_sub_directory:
                                       mov eax, [Current_Dir_FCluster] call load_FAT_sub_directory
707 0000A1EF A1[80590100]
                                 <1>
708 0000A1F4 E8B91B0000
                                 <1>
709 0000A1F9 C3
                                 <1>
710
                                 <1>
711
                                 <1> find_directory_entry:
                                 712
                                          ; 13/02/2016
713
                                 <1>
714
                                 <1>
                                           ; 10/02/2016
715
                                           ; 06/02/2016 (TRDOS 386 = TRDOS v2.0)
                                 <1>
716
                                 <1>
                                          ; 14/08/2010 (DIR.ASM, "proc_find_direntry")
                                          ; 19/09/2009
717
                                 <1>
                                           ; 2005
718
                                 <1>
719
                                 <1>
                                           ; INPUT ->
                                                 ESI = Sub Dir or File Name Address
720
                                 <1>
721
                                 <1>
                                                  AL = Attributes Mask
722
                                 <1>
                                                  (<AL AND EntryAttrib> must be equal to AL)
                                                  AH = Negative Attributes Mask (If AH>0)
723
                                 <1>
724
                                 <1>
                                                  (<AH AND EntryAttrib> must be ZERO)
                                                  CH > 0 Find First Free Dir Entry or Deleted Entry
725
                                 <1>
726
                                 <1>
                                                  CL = 0 -> Return the First Free Dir Entry
727
                                 <1>
                                                  CL = E5h -> Return the 1st deleted entry
                                                  {\tt CL} = FFh -> Return the 1st deleted or free entry
                                 <1>
728
729
                                 <1>
                                                  {\tt CL} > 0 and {\tt CL} <> E5h and {\tt CL} <> FFh -> Return the first
                                                         proper entry (which fits with Atributes Masks)
730
                                 <1>
                                                  CX = 0 -> Find Valid File/Directory/VolumeName
731
                                 <1>
732
                                 <1>
                                                  ? = Any One Char
                                                  * = Every Chars
733
                                 <1>
734
                                 <1>
                                                  EBX = Current Dir Entry (BX)
735
                                 <1>
                                           ;
                                           ; OUTPUT ->
736
                                 <1>
737
                                  <1>
                                                  EDI = Directory Entry Address (in DirectoryBuffer)
                                                  ESI = Sub Dir or File Name Address
738
                                 <1>
739
                                 <1>
                                                  CF = 0 -> No Error, Proper Entry,
740
                                 <1>
                                                  DL = Attributes
741
                                 <1>
                                                  DH = Previous Entry Attr (LongName Check)
                                                  AL > 0 -> Ambiguous filename wildcard "?" used
742
                                 <1>
                                                  \rm AH \, > \, 0 -> Ambiguous filename wildcard "*" used
743
                                 <1>
744
                                 <1>
                                                  AX = 0 -> Filename full fits with directory entry
745
                                 <1>
                                                  EBX = CurrentDirEntry (BX)
746
                                 <1>
                                                  CH = The 1st Name Char of Current Dir Entry
                                                  CF = 1 -> Proper entry not found, Error Code in AX/AL
747
                                 <1>
748
                                                  CL = 0 and CH = 0 \rightarrow Free Entry (End Of Dir)
                                 <1>
749
                                 <1>
                                                  CL = 0 and CH = E5h \rightarrow Deleted Entry fits with filters
750
                                 <1>
                                                  CL > 0 -> Entry not found, CH invalid
751
                                 <1>
752
                                  <1>
                                           ; (EAX, EBX, ECX, EDX, EDI, EBP will be changed)
753
                                 <1>
754 0000A1FA 663B1D[AF600100]
                                 <1>
                                           cmp
                                                  bx, [DirBuff_LastEntry]
755 0000A201 0F8739010000
                                 <1>
                                                     loc_ffde_stc_retn_255
                                             jа
756
                                 <1>
757
                                  <1>
                                                    [DirBuff_CurrentEntry], bx
                                           ; mov
758
                                 <1>
759 0000A207 BF00000800
                                                  edi, Directory_Buffer
                                 <1>
760 0000A20C 66A3[7C610100]
                                 <1>
                                                  [FDE_AttrMask], ax
                                           mov
```

```
761
                               <1>
762 0000A212 29C0
                               <1>
                                        sub
                                             eax, eax
763
                               <1>
764
                              <1>
                                        ;;mov [PreviousAttr], al ; 0 ;; 13/02/2016
765 0000A214 66A3[7E610100]
                                              [AmbiguousFileName], ax; 0
                              <1>
766
                              <1>
767 0000A21A 6689D8
                              <1>
                                        mov
                                              ax, bx
768 0000A21D 66C1E005
                                              ax, 5 ; * 32 ; Directory entry size
                              <1>
                                        shl
769 0000A221 01C7
                              <1>
                                        add
                                              edi, eax
                              <1>
771 0000A223 08ED
                              <1>
                                        or ch, ch
                                        jnz loc_find_free_deleted_entry 0
772 0000A225 0F852C010000
                              <1>
773
                              <1>
774 0000A22B 08C9
                              <1>
                                        or
                                              cl, cl
775 0000A22D 0F850D010000
                              <1>
                                        jnz loc_ffde_stc_retn_255
776
                              <1>
777
                              <1> check find dir entry:
778 0000A233 66A1[7C610100]
                              <1>
                                        mov ax, [FDE_AttrMask]
779 0000A239 8A2F
                              <1>
                                              ch, [edi]
                                        mov
780 0000A23B 80FD00
                              <1>
                                              ch, 0 ; Is it never used entry?
                                        cmp
// 1>
<1>
<1>
<1>
<1>
<1>

                                        jna loc_find_direntry_stc_retn
                                       push esi
mov dl, [edi+0Bh] ; File attributes
782 0000A244 56
783 0000A245 8A570B
784 0000A248 80FDE5
                                       cmp
                                             ch, 0E5h; Is it a deleted file?
785 0000A24B 746D
                                      je
                                              short loc_find_dir_next_entry_prevdeleted
786
                              <1>
787 0000A24D 80FA0F
                              <1>
                                        cmp
                                               dl, OFh ; longname sub component check
788 0000A250 7505
                              <1>
                                        jne
                                               short loc_check_attributes_mask
789 0000A252 E8ED010000
                              <1>
                                        call save_longname_sub_component
790
                              <1>
791
                              <1> loc_check_attributes_mask:
                                    mov dh, al and dh, dl
792 0000A257 88C6
                              <1>
                              <1>
793 0000A259 20D6
794 0000A25B 38F0
                              <1>
                                        cmp al, dh
                              <1>
                                        jne loc_find_dir_next_entry
795 0000A25D 0F85BA000000
796 0000A263 20D4
                              <1>
                                        and ah, dl
                         <1>
<1>
797 0000A265 0F85B2000000
                                        jnz loc_find_dir_next_entry
                                        cmp dl, 0Fh
798 0000A26B 80FA0F
                                        jne short pass_direntry_attr_check
799 0000A26E 751A
                              <1>
800
                              <1>
801 0000A270 3C0F
                              <1>
                                        cmp al, 0Fh; AL = 0Fh -> find long name
                                       jne loc_find_dir_next_entry
802 0000A272 0F85A5000000
                              <1>
                              <1>
803
                                        pop esi
804 0000A278 5E
                              <1>
                                        xor
805 0000A279 6631C0
                              <1>
                                             ax, ax
806 0000A27C 8A35[80610100]
                               <1>
                                        mov
                                              dh, [PreviousAttr]
                                        mov [DirBuff_CurrentEntry], bx
807 0000A282 66891D[AD600100] <1>
808 0000A289 C3
                              <1>
                                        retn
809
                              <1>
810
                              <1> pass_direntry_attr_check:
811 0000A28A 89FD
                              <1>
                                        mov ebp, edi ; 14/02/2016
812 0000A28C B908000000
                              <1>
                                        mov
                                              ecx, 8
                              <1> loc_lodsb_find_dir:
813
814 0000A291 AC
                              <1>
                                        cmp al, '*'
jne short pass_fde_ambiguous1_check
815 0000A292 3C2A
                              <1>
816 0000A294 7508
                              <1>
                           <1>
817 0000A296 FE05[7F610100]
                                        inc byte [AmbiguousFileName+1]
818 0000A29C EB28
                              <1>
                                      jmp short loc_check_direntry_extension
819
                              <1>
                              <1> pass_fde_ambiguous1_check:
820
                                    cmp al, '?'
821 0000A29E 3C3F
                              <1>
822 0000A2A0 750D
                              <1>
                                              short pass_fde_ambiguous2_check
                                        jne
823 0000A2A2 FE05[7E610100]
                                              byte [AmbiguousFileName]
                              <1>
                                        inc
824 0000A2A8 803F20
                              <1>
                                             byte [edi], 20h
825 0000A2AB 764E
                              <1>
                                        jna
                                             short loc_find_dir_next_entry_ebp
826 0000A2AD EB14
                              <1>
                                        jmp
                                              short loc_scasb_find_dir_inc_di
827
                              <1>
                              <1> pass_fde_ambiguous2_check:
828
                              <1> cmp
829 0000A2AF 3C20
                                             al, 20h
830 0000A2B1 750C
                              <1>
                                              short loc_scasb_find_dir
                                        jne
831 0000A2B3 803F20
                              <1>
                                              byte [edi], 20h
                                              short loc_find_dir_next_entry ebp
832 0000A2B6 7543
                              <1>
                                        jne
833 0000A2B8 EB0C
                              <1>
                                             short loc_check_direntry_extension
                                        jmp
834
                              <1>
                              <1> loc_find_dir_next_entry_prevdeleted:
835
                                     or dl, 80h ; Bit 7 -> deleted entry sign
836 0000A2BA 80CA80
                              <1>
837 0000A2BD EB5E
                              <1>
                                        jmp short loc find dir next entry
838
                              <1>
839
                               <1> loc_scasb_find_dir:
                              <1> cmp al, [edi]
<1> jne short loc_find_dir_next_entry_ebp
840 0000A2BF 3A07
841 0000A2C1 7538
                               <1> loc_scasb_find_dir_inc_di:
843 0000A2C3 47
                               <1>
                                        inc
                                             edi
                               <1>
                                        loop loc_lodsb_find_dir
844 0000A2C4 E2CB
845
                               <1>
846
                               <1> loc_check_direntry_extension:
                                       mov esi, 8
847 0000A2C6 BE08000000
                              <1>
848 0000A2CB 89F7
                                              edi, esi ; 8
                              <1>
                                        mov
849 0000A2CD 033424
                              <1>
                                              esi, [esp] ; Sub Dir or File Name Address
                                        add
850 0000A2D0 01EF
                                              edi, ebp
                              <1>
                                        add
851 0000A2D2 B103
                              <1>
                                        mov
                                              cl, 3
                               <1> loc lodsb find dir ext:
852
853 0000A2D4 AC
                              <1>
                                        lodsb
                                              al, '*'
854 0000A2D5 3C2A
                              <1>
                                        cmp
855 0000A2D7 7508
                               <1>
                                              short pass_fde_ambiguous3_check
                                        ine
856 0000A2D9 FE05[7F610100]
                              <1>
                                        inc
                                              byte [AmbiguousFileName+1]
857 0000A2DF EB1E
                               <1>
                                              short loc_find_dir_proper_direntry
                                        jmp
858
                               <1>
                               <1> pass fde ambiguous3 check:
859
                                       cmp al, '?'
860 0000A2E1 3C3F
                               <1>
861 0000A2E3 750D
                              <1>
                                        jne
                                              short pass_fde_ambiguous4_check
862 0000A2E5 FE05[7E610100]
                               <1>
                                              byte [AmbiguousFileName]
                                        inc
863 0000A2EB 803F20
                               <1>
                                        cmp
                                              byte [edi], 20h
864 0000A2EE 760B
                               <1>
                                        jna short loc_find_dir_next_entry_ebp
865 0000A2F0 EB49
                               <1>
                                             short loc_scasb_find_dir_ext_inc_di
                                        jmp
```

```
866
                               <1>
                               <1> pass fde ambiguous4 check:
868 0000A2F2 3C20
                                      cmp
                               <1>
                                              al, 20h
869 0000A2F4 7541
                              <1>
                                         jne short loc_scasb_find_dir_ext
870 0000A2F6 803F20
                              <1>
                                         cmp
                                              byte [edi], 20h
871 0000A2F9 7404
                              <1>
                                        jе
                                               short loc_find_dir_proper_direntry
                              <1>
                               <1> loc_find_dir_next_entry_ebp:
873
874 0000A2FB 89EF
                               <1>
                                        mov edi, ebp ; 14/02/2016
875 0000A2FD EB1E
                               <1>
                                         jmp
                                              short loc_find_dir_next_entry
876
                               <1>
877
                               <1> loc_find_dir_proper_direntry:
878 0000A2FF 30C9
                                       xor cl, cl
                               <1>
879
                               <1> loc_find_dir_proper_direntry_1:
                                     pop esi
                               <1>
880 0000A301 5E
881 0000A302 89EF
                               <1>
                                         mov edi, ebp
                                        mov ch, [edi]
882 0000A304 8A2F
                               <1>
                                               dl, [edi+0Bh] ; Dir entry attributes
                                      mov
883 0000A306 8A570B
                               <1>
                                               ax, [AmbiguousFileName]
884 0000A309 66A1[7E610100]
                               <1>
                                         mov
                               <1> loc_find_dir_proper_direntry_2:
                                     mov
886 0000A30F 8A35[80610100]
                                               dh, [PreviousAttr]
                               <1>
887 0000A315 66891D[AD600100]
                               <1>
                                        mov
                                               [DirBuff_CurrentEntry], bx
888 0000A31C C3
                               <1>
                                        retn
                               <1>
890
                               <1> loc_find_dir_next_entry:
891 0000A31D 8815[80610100]
                               <1>
                                        mov byte [PreviousAttr], dl; LongName check
                               <1> loc find_dir_next_entry_1:
893 0000A323 5E
                                      pop esi
                               <1>
894 0000A324 83C720
                               <1>
                                         add
                                               edi, 32
895
                               <1>
                                         ;inc word [DirBuff_EntryCounter]
896 0000A327 6643
                               <1>
                                         inc
897 0000A329 663B1D[AF600100]
                               <1>
                                         cmp
                                               bx, [DirBuff LastEntry]
                                        ja
898 0000A330 770E
                                               short loc_ffde_stc_retn_255
                               <1>
899 0000A332 E9FCFEFFFF
                               <1>
                                         jmp
                                               check_find_dir_entry
900
                               <1>
                               <1> loc_scasb_find_dir_ext:
901
902 0000A337 3A07
                               <1>
                                    cmp al, [edi]
903 0000A339 75C0
                               <1>
                                        jne
                                              short loc_find_dir_next_entry_ebp
904
                               <1> loc_scasb_find_dir_ext_inc_di:
905 0000A33B 47
                                    inc edi
                               <1>
906 0000A33C E296
                               <1>
                                         loop
                                               loc_lodsb_find_dir_ext
907 0000A33E EBC1
                                               short loc find dir proper direntry 1
                               <1>
                                         jmp
908
                               <1>
909
                               <1> loc_ffde_stc_retn_255:
                                     ;mov cx, OFFFFh
910
                               <1>
911 0000A340 31C9
                               <1>
912 0000A342 49
                                         dec ecx; OFFFFFFFh
                               <1>
                                        ;xor eax, eax
913
                               <1>
914
                               <1> loc_find_direntry_stc_retn:
915
                               <1> loc_check_ffde_retn_1:
916
                               <1>
                                      ;mov ax, 2
917 0000A343 B802000000
                               <1>
                                        mov
                                               eax, 2 ; File Not Found
918 0000A348 8A35[80610100]
                               <1>
                                        mov
                                               dh, [PreviousAttr]
919 0000A34E 66891D[AD600100]
                               <1>
                                               [DirBuff_CurrentEntry], bx
                                        mov
920 0000A355 F9
                               <1>
                                        stc
921 0000A356 C3
                               <1>
                                         retn
                               <1>
923
                               <1> loc_find_free_deleted_entry_0:
                                     mov ax, [FDE_AttrMask]
924 0000A357 66A1[7C610100]
                               <1>
925 0000A35D 8A2F
                                        mov
                               <1>
                                              ch, [edi]
926 0000A35F 8A570B
                               <1>
                                         mov dl, [edi+0Bh] ; File attributes
927 0000A362 08C9
                               <1>
                                        or
                                               cl, cl
                                        j z
928 0000A364 7407
                               <1>
                                               short loc_check_ffde_0_repeat
                                        ;cmp cl, 0E5h
929
                               <1>
930
                               <1>
                                               short pass_loc_check_ffde_0_err
                                         ;je
931 0000A366 80F9FF
                               <1>
                                         cmp
                                               cl, OFFh
                                               short loc find free deleted entry 1
932 0000A369 7432
                               <1>
                                         jе
933 0000A36B EB4D
                               <1>
                                         jmp
                                               short pass_loc_check_ffde_0_err
934
                               <1>
935
                               <1> loc_check_ffde_0_repeat:
936 0000A36D 08ED
                               <1>
                                        or
                                               ch, ch
                                               short loc check ffde 0 next
937 0000A36F 7511
                               <1>
                                         jnz
938
                               <1>
939
                               <1> loc_check_ffde_retn_2:
                                      sub ax, ax
940 0000A371 6629C0
                               <1>
941 0000A374 8A35[80610100]
                                               dh, [PreviousAttr]
                               <1>
                                         mov
942 0000A37A 66891D[AD600100]
                               <1>
                                              [DirBuff_CurrentEntry], bx
                                         mov
943 0000A381 C3
                               <1>
                                        retn
944
                               <1>
945
                               <1> loc_check_ffde_0_next:
946 0000A382 6643
                               <1>
                                      inc bx
947 0000A384 83C720
                                <1>
                                         add
                                               edi, 32
                                         ;inc word [DirBuff_EntryCounter]
948
                                <1>
                                <1>
949
950 0000A387 663B1D[AF600100]
                               <1>
                                           cmp bx, [DirBuff LastEntry]
951 0000A38E 77B0
                               <1>
                                         jа
                                               short loc_ffde_stc_retn_255
952 0000A390 8815[80610100]
                               <1>
                                               [PreviousAttr], dl
                                         mov
953 0000A396 8A2F
                               <1>
                                         mov
                                               ch, [edi]
954 0000A398 8A570B
                               <1>
                                               dl, [edi+0Bh] ; file attributes
955 0000A39B EBD0
                               <1>
                                               short loc_check_ffde_0_repeat
                                         jmp
956
                               <1>
                               <1> loc find_free_deleted_entry_1:
957
958 0000A39D 28D2
                               <1>
                                         sub
                                              dl, dl
                               <1> loc_find_free_deleted_entry_2:
959
960 0000A39F 20ED
                               <1>
                                         and
                                               ch, ch
                                               short loc_check_ffde_retn_2
961 0000A3A1 74CE
                               <1>
                                         jz
                                               ch, 0E5h
962 0000A3A3 80FDE5
                               <1>
                                         cmp
963 0000A3A6 74C9
                               <1>
                                         jе
                                               short loc_check_ffde_retn_2
964 0000A3A8 6643
                               <1>
                                         inc
965 0000A3AA 83C720
                                               edi, 32
                               <1>
                                         add
966 0000A3AD 663B1D[AF600100]
                               <1>
                                         cmp
                                               bx, [DirBuff_LastEntry]
967 0000A3B4 778A
                               <1>
                                               short loc_ffde_stc_retn_255
                                         jа
968 0000A3B6 8A2F
                               <1>
                                         mov
                                               ch, [edi]
969 0000A3B8 EBE5
                               <1>
                                               short loc_find_free_deleted_entry_2
                                         jmp
                               <1>
970
```

```
971
                                 <1> pass_loc_check_ffde_0_err:
 972 0000A3BA 38CD
                                 <1>
                                           cmp ch, cl
                                                 short loc_check_ffde_attrib
973 0000A3BC 741F
                                 <1>
                                           jе
974
                                 <1>
 975 0000A3BE 6643
                                 <1>
                                           inc
                                                 bx
976 0000A3C0 83C720
                                 <1>
                                           add
                                                 edi, 32
977 0000A3C3 663B1D[AF600100] <1> cmp bx, [DirBuff_LastEntry]
978 0000A3CA 0F8770FFFFFF <1> ja loc_ffde_stc_retn_25
979 0000A3D0 8815[80610100] <1> mov [PreviousAttr], dl
980 0000A3D6 8A2F <1> mov ch, [edi]
                                          ja loc_ffde_stc_retn_255
 981 0000A3D8 8A570B
                                           mov dl, [edi+0Bh]
                                 <1>
 982 0000A3DB EBDD
                                 <1>
                                           jmp short pass_loc_check_ffde_0_err
983
                                 <1>
                                 <1> loc check ffde attrib:
984
 985 0000A3DD 88C6
                                       mov dh, al
                                 <1>
986 0000A3DF 20D6
                                 <1>
                                           and
                                                 dh, dl
 987 0000A3E1 38F0
                                                al, dh
                                <1>
                                           cmp
 988 0000A3E3 759D
                                        jne short loc_check_ffde_0_next
                                <1>
 989 0000A3E5 20D4
                                 <1>
                                           and
                                                 ah, dl
                                           jnz short loc_check_ffde_0_next
990 0000A3E7 7599
                                <1>
991 0000A3E9 30C9
                                <1>
                                         xor cl, cl
                                           jmp loc_check_ffde_retn 2
                                 <1>
 992 0000A3EB EB84
993
                                 <1>
 994
                                 <1> convert_file_name:
                                 <1>; 06/03/2016
 995
996
                                 <1>
                                           ; 11/02/2016
                                         ; 07/02/2016 (TRDOS 386 = TRDOS v2.0)
 997
                                  <1>
                                         ; 06/10/2009
 998
                                 <1>
                                          ; 2005
999
                                  <1>
1000
                                 <1>
                                         ; INPUT ->
1001
                                  <1>
                                         ; ESI = Dot File Name Location
1002
                                  <1>
                                                 EDI = Dir Entry Format File Name Location
1003
                                 <1>
1004
                                 <1>
                                         ; OUTPUT ->
                                         ; EDI = Dir Entry Format File Name Location
1005
                                 <1>
1006
                                 <1>
                                                 ESI = Dot File Name Location (capitalized)
1007
                                  <1>
1008
                                 <1>
                                           ; (ECX, AL will be changed)
1009
                                 <1>
1010 0000A3ED 56
                                           push esi
                                 <1>
1011 0000A3EE 57
                                           push edi
                                 <1>
1012
                                 <1>
1013 0000A3EF B90B000000
                                 <1>
                                                  ecx, 11
                                           mov
                                                 al, 20h
1014 0000A3F4 B020
                                 <1>
                                           mov
1015 0000A3F6 F3AA
                                 <1>
                                                 stosb
                                           rep
1016
                                 <1>
1017 0000A3F8 8B3C24
                                 <1>
                                           mov
                                                  edi, [esp]
1018
                                 <1>
                                                 cl, 12; file name length (max.)
1019 0000A3FB B10C
                                 <1>
                                           mov
                                           ; 06/03/2016
1020
                                 <1>
                                           ; Directory entry name limit (11 bytes) check for
1021
                                 <1>
1022
                                  <1>
                                           ; 'rename_directory_entry' procedure.
1023
                                 <1>
                                           ; (EDI points to Directory Entry)
1024
                                 <1>
                                           ; (If the file name would not contain a dot
1025
                                 <1>
                                           ; and file name length would be 12, this would cause to
1026
                                 <1>
                                           ; overwrite the attributes byte of the directory entry.)
1027
                                 <1>
1028 0000A3FD B50B
                                 <1>
                                           mov ch, 11; directory entry's name length
1029
                                 <1> loc_check_first_dot:
                                 <1> mov al, [esi]
1030 0000A3FF 8A06
1031 0000A401 3C2E
                                 <1>
                                           cmp al, 2Eh
1032 0000A403 750C
                                           jne short pass_check_first_dot
mov [edi], al
                                 <1>
1033 0000A405 8807
                                <1>
1034 0000A407 47
                                 <1>
                                           inc edi
                                           inc
1035 0000A408 46
                                 <1>
                                                 esi
1036 0000A409 FEC9
                                 <1>
                                           dec
                                                 cl
1037 0000A40B 75F2
                                           jnz short loc_check_first_dot
                                 <1>
1038
                                         ;; (ecx <= 12)
                                 <1>
1039
                                 <1>
                                           ;;loop loc_check_first_dot
1040 0000A40D EB30
                                 <1>
                                           jmp short stop_convert_file
1041
                                 <1>
1042
                                 <1> loc_get_fchar:
1043 0000A40F 8A06
                                 <1>
                                         mov al, [esi]
1044
                                 <1> pass_check_first_dot:
                                 1045 0000A411 3C61
1046 0000A413 7208
                                 <1>
                                           jb
                                                 short pass_name_capitalize
                                      cmp al, 7Ah; 'z'
ja short pass_name_capitalize
and al, 0DFh
mov [esi], al
1047 0000A415 3C7A
                                 <1>
1048 0000A417 7704
                                 <1>
1049 0000A419 24DF
                                 <1>
1050 0000A41B 8806
                                 <1>
1051
                                 <1> pass_name_capitalize:
1052 0000A41D 3C21
                                  <1>
                                           cmp al, 21h
                                                 short stop_convert_file
al, 2Eh; '.'
1053 0000A41F 721E
                                 <1>
                                           jb
1054 0000A421 3C2E
                                 <1>
1055 0000A423 750C
                                 <1>
                                           ine
                                                 short pass_dot_space
1056
                                 <1> add_dot_space:
1057 0000A425 80F904
                                 <1>
                                                 cl, 4
                                           cmp
1058 0000A428 760E
                                 <1>
                                                  short inc_and_loop
                                           jna
1059 0000A42A 47
                                 <1>
                                                  edi
                                           inc
1060 0000A42B FECD
                                                 ch ; 06/03/2016
                                 <1>
                                           dec
1061 0000A42D FEC9
                                 <1>
                                           dec
                                                  cl
                                  <1>
                                           jmp
1062 0000A42F EBF4
                                                  short add_dot_space
1063
                                 <1>
1064
                                  <1>
                                                 al, 4
1065
                                  <1>
                                           ;cmp
                                                 cl, al
1066
                                  <1>
                                           ;jna
                                                 short inc_and_loop
                                           ; sub cl, al
1067
                                  <1>
1068
                                  <1>
                                           ;add edi, ecx
1069
                                  <1>
                                           ;mov
                                                 cl, al
1070
                                  <1>
                                           ;jmp short inc_and_loop
1071
                                 <1>
1072
                                  <1> pass_dot_space:
1073 0000A431 8807
                                 <1>
                                       mov [edi], al
                                  <1> loc after double dot:
1074
1075
                                  <1> ; 06/03/2016
```

```
1076 0000A433 FECD
                                <1>
                                                ch ; count down for 11 bytes dir entry limit
                                          dec
1077 0000A435 740A
                                <1>
                                          jz
                                                short stop_convert_file_x
1078 0000A437 47
                                <1>
                                          inc
                                                edi
1079
                                <1> inc_and_loop:
1080 0000A438 FEC9
                                <1>
                                          dec cl; count down for 12 bytes filename limit
1081 0000A43A 7403
                                <1>
                                          jг
                                                short stop_convert_file
1082 0000A43C 46
                                <1>
                                          inc esi
                                       ;; (ecx <= 12)
1083
                                <1>
                                          ;;loop loc_get_fchar
1084
                                <1>
1085 0000A43D EBD0
                                <1>
                                          jmp short loc_get_fchar
1086
                                <1>
1087
                                <1> stop_convert_file:
                                       ; 06/03/2016
1088
                                <1>
1089 0000A43F 30ED
                                <1>
                                          xor ch, ch
                                <1>
                                          ; ECX < 256 ; 'find_first_file' -> xor cl, cl
1090
1091
                                <1> stop_convert_file_x:
1092 0000A441 5F
                                <1>
                                          pop edi
1093 0000A442 5E
                                <1>
                                          pop
                                                esi
1094 0000A443 C3
                                <1>
                                          retn
1095
                                <1>
1096
                                <1> save_longname_sub_component:
                                      ; 13/02/2016
1097
                                <1>
                                          ; 06/02/2016 (TRDOS 386 = TRDOS v2.0)
1098
                                <1>
                                        ; 28/02/2010
1099
                                 <1>
                                        ; 17/10/2009
                                 <1>
1100
1101
                                 <1>
                                          ; INPUT ->
1102
                                 <1>
                                                EDI = Directory Entry
                                <1>
1103
                                                // This procedure is called
1104
                                 <1>
                                                // from 'find_directory_entry' procedure.
                                          ;
                                                // If the last entry returns with
1105
                                 <1>
                                          ;
1106
                                 <1>
                                                // a non-zero LongnameFound value and
                                                // if LFN CheckSum value is equal to
1107
                                 <1>
                                                // the next shortname checksum,
                                <1>
1108
1109
                                <1>
                                                // long name is valid.
                                                // If a longname is longer than 65 bytes,
1110
                                <1>
                                          ;
                                                // it is invalid for trdos. (>45h)
1111
                                <1>
1112
                                <1>
1113 0000A444 57
                                <1>
                                          push edi
1114 0000A445 56
                                <1>
                                          push
                                                esi
1115
                                <1>
                                          ;push ebx
1116
                                <1>
                                          ;push ecx
1117
                                 <1>
                                          ;push edx
1118 0000A446 50
                                <1>
                                          push eax
1119
                                <1>
1120 0000A447 29C9
                                <1>
                                          sub
                                                 ecx, ecx
                                                eax, eax
1121
                                <1>
                                          ;sub
1122 0000A449 B11A
                                <1>
                                                cl, 26
                                          mov
                                <1>
1123
                                          movzx eax, byte [edi] ; LDIR Order
1124 0000A44B 0FB607
                                <1>
1125 0000A44E 3C41
                                <1>
                                          cmp al, 41h; 40h (last long entry sign) + 1
1126 0000A450 722B
                                <1>
                                                 short pass_pslnsc_last_long_entry
1127
                                <1>
1128 0000A452 88C4
                                <1>
                                          mov
                                                 ah, al
1129 0000A454 80EC40
                                <1>
                                          sub
                                                 ah, 40h
1130 0000A457 8825[82610100]
                                <1>
                                                [LFN_EntryLength], ah
                                          mov
1131
                                <1>
1132 0000A45D 3C45
                                <1>
                                                 al, 45h; 40h (last long entry sign) + 5
                                          cmp
                                                ; {\tt Max} 130 byte length is usable in TRDOS
1133
                                <1>
1134
                                 <1> ; 26*5 = 130
1135 0000A45F 7753
                                <1>
                                          jа
                                                short loc_pslnsc_retn
1136
                                <1>
1137 0000A461 2407
                                 <1>
                                          and
                                                 al, 07h ; 0Fh
1138 0000A463 A2[81610100]
                                <1>
                                                 [LongNameFound], al
                                          mov
1139
                                <1>
1140 0000A468 FEC8
                                <1>
                                          dec
                                                al
1141
                                <1>
                                          ;mov
                                                cl, 26
1142 0000A46A F6E1
                                <1>
                                          mul
                                                cl
1143
                                <1>
1144 0000A46C 89C6
                                <1>
                                                esi, eax
1145 0000A46E 01CE
                                <1>
                                          add
                                                esi, ecx
1146
                                <1>
                                                ; to make is an ASCIIZ string
1147
                                <1>
                                                ; with ax+26 bytes length
                                          add
1148 0000A470 81C6[84610100]
                                               esi, LongFileName
                                <1>
1149 0000A476 66C7060000
                                <1>
                                                word [esi], 0
                                                short loc_pslsc_move_ldir_name2
1150 0000A47B EB16
                                <1>
                                          jmp
1151
                                <1>
1152
                                <1> pass_pslnsc_last_long_entry:
                                          cmp al, 04h
1153 0000A47D 3C04
                                <1>
1154 0000A47F 7733
                                                 short loc pslnsc retn
                                <1>
                                          jа
1155 0000A481 FE0D[81610100]
                                <1>
                                                byte [LongNameFound]
                                          dec
1156 0000A487 3A05[81610100]
                                <1>
                                          cmp al, [LongNameFound]
                                                short loc pslnsc retn
1157 0000A48D 7525
                                 <1>
                                          jne
1158
                                <1>
1159
                                 <1> loc_pslsc_move_ldir_name1:
                                      _ dec al
1160 0000A48F FEC8
                                <1>
1161
                                <1>
                                          ;mov cl, 26
1162 0000A491 F6E1
                                <1>
                                          mul cl
                                <1>
1163
1164
                                <1> loc_pslsc_move_ldir_name2:
1165 0000A493 8A4F0D
                               <1> mov cl, [edi+ODh]; long name checksum
1166 0000A496 880D[83610100] <1>
                                          mov
                                               [LFN_CheckSum], cl
1167 0000A49C 89FE
                                                esi, edi ; LDIR Order
                                <1>
                                          mov
                                          mov
1168 0000A49E BF[84610100]
                                                edi, LongFileName
                                <1>
1169 0000A4A3 01C7
                                                edi, eax
                                <1>
                                        add
1170 0000A4A5 46
                                <1>
                                          inc
                                                esi
1171 0000A4A6 B105
                                                cl, 5 ; chars 1 to 5
                                <1>
                                          mov
1172 0000A4A8 F366A5
                               <1>
                                          rep
                                                movsw
1173 0000A4AB 83C603
                                <1>
                                               esi, 3
                                          add
1174 0000A4AE A5
                                          movsd ; char 6 & 7
                                <1>
1175 0000A4AF A5
                                <1>
                                         movsd ; char 8 & 9
1176 0000A4B0 A5
                                <1>
                                          movsd ; char 10 & 11
                                          inc esi
inc esi
1177 0000A4B1 46
                                <1>
1178 0000A4B2 46
                                <1>
1179 0000A4B3 A5
                                          movsd ; char 12 & 13
                                <1>
1180
                                <1>
```

```
1181
                                <1> loc_pslnsc_retn:
                                <1> pop eax ;pop edx
1182 0000A4B4 58
1183
                               <1>
1184
                               <1>
1185
                               <1>
                                         ;pop ebx
1186 0000A4B5 5E
                               <1>
                                               esi
                                         pop
1187 0000A4B6 5F
                               <1>
                                               edi
                                         pop
1188
                               <1>
1189 0000A4B7 C3
                               <1>
                                         retn
1190
                                <1>
1191
                                <1> parse_path_name:
                                      ; 10/02/2016
1192
                                <1>
                                         ; 08/02/2016 (TRDOS 386 = TRDOS v2.0)
1193
                                <1>
                                       ; 10/009/2011 ('proc parse pathname')
1194
                                <1>
                                       ; 27/11/2009
; 05/12/2004
1195
                                <1>
1196
                                <1>
1197
                                <1>
                                      ; INPUT ->
                                <1>
1198
1199
                                <1>
                                         ; ESI = Beginning of ASCIIZ pathname string
1200
                                <1>
                                               EDI = Destination Address
                                                    (which is TR-DOS FindFile data buffer)
1201
                                <1>
                                         ; OUTPUT ->
1202
                                <1>
                                         ; CF = 1 \rightarrow Error
1203
                                <1>
1204
                                <1>
                                                  EAX = Error Code (AL)
                                <1>
1205
1206
                                <1>
                                         ; (Modified registers: eax, ecx, esi, edi)
1207
                                <1>
                                         ; Clear the pathname bytes in TR-DOS Findfile data buffer
1208
                               <1>
1209 0000A4B8 57
                                <1>
                                         push edi
1210 0000A4B9 B914000000
                               <1>
                                               ecx, 20 ; 80 bytes
                                         mov
1211 0000A4BE 31C0
                               <1>
                                         xor
                                               eax, eax
1212 0000A4C0 F3AB
                               <1>
                                         rep
                                               stosd
1213 0000A4C2 5F
                               <1>
                                         pop
                                               edi
1214
                               <1>
1215 0000A4C3 668B06
                               <1>
                                        mov
                                              ax, [esi]
                                               ah, ':'
1216 0000A4C6 80FC3A
                               <1>
                                         cmp
1217 0000A4C9 741C
                               <1>
                                         jе
                                               short loc_ppn_change_drive
1218 0000A4CB A0[86590100]
                                              al, [Current_Drv]
                               <1>
                                         mov
1219 0000A4D0 EB33
                               <1>
                                              short pass_ppn_change_drive
                                         jmp
1220
                               <1>
1221
                               <1> pass_ppn_cdir:
1222 0000A4D2 8B35[A6620100]
                               <1>
                                        mov esi, [First_Path_Pos]
1223 0000A4D8 AC
                               <1>
                                         lodsb
1224
                               <1> loc_ppn_get_filename:
                                     add edi, 65; FindFile_Name location
1225 0000A4D9 83C741
                               <1>
                                         ; TRDOS Filename length must not be more than 12 bytes
1226
                               <1>
                                      mov cl, 12
1227
                               <1>
                                         ;mov ecx, 12
1228 0000A4DC B10C
                               <1>
1229
                               <1> loc_ppn_get_fnchar_next:
1230 0000A4DE AA
                               <1> stosb
1231 0000A4DF AC
                               <1>
                                         lodsb
                              <1> cmp al, 21h
<1> jb short loc_ppn_clc_return
<1> loop loc_ppn_get_fnchar_next
1232 0000A4E0 3C21
1233 0000A4E2 7274
1234 0000A4E4 E2F8
1235
                               <1> loc_ppn_return:
1236 0000A4E6 C3
                               <1>
                                         retn
1237
                               <1>
1238
                               <1> loc_ppn_change_drive:
                                     and al, ODFh
1239 0000A4E7 24DF
                               <1>
                               <1>
                                         sub al, 'A'; A:
1240 0000A4E9 2C41
                             <1>
1241 0000A4EB 726F
                                         jc short loc_ppn_invalid_drive
                                        cmp [Last_DOS_DiskNo], al
jb short loc_ppn invalid
1242 0000A4ED 3805[610D0100]
                               <1>
1243 0000A4F3 7267
                               <1>
                                               short loc_ppn_invalid_drive
1244
                               <1>
1245 0000A4F5 46
                               <1>
                                         inc
                                              esi
1246 0000A4F6 46
                               <1>
                                         inc
                                               esi
1247 0000A4F7 8A26
                               <1>
                                         mov
                                               ah, [esi]
1248 0000A4F9 80FC21
                               <1>
                                         cmp
                                               ah, 21h
1249 0000A4FC 7307
                               <1>
                                         jnb
                                              short pass_ppn_change_drive
1250
                               <1>
1251
                               <1> loc_ppn_cmd_failed:
                               <1> ; File or directory name is not existing
1252
1253 0000A4FE 8807
                               <1>
                                         mov [edi], al ; Drv
1254 0000A500 66B80100
                               <1>
                                       mov ax, 1 ; eax = 1
                                       ; TR-DOS Error Code 01h = Bad Command Argument
1255
                                <1>
1256
                                         ; MS-DOS Error Code 01h : Invalid Function Number
                                <1>
1257
                                <1>
1258
                                <1>
                                         ; (MainProg ErrMsg: "Bad command or file name!")
1259 0000A504 C3
                                <1>
1260
                                <1>
                                <1> pass_ppn_change_drive:
1261
1262 0000A505 8935[A6620100]
                                <1>
                                      mov [First_Path_Pos], esi
1263 0000A50B C705[AA620100]0000- <1>
                                              dword [Last_Slash_Pos], 0
                                         mov
1263 0000A513 0000
1264 0000A515 AA
                               <1>
                                         stosb
1265 0000A516 8A06
                               <1>
                                         mov al, [esi]
                               <1> loc_scan_ppn_dslash:
1267 0000A518 3C2F
                               <1>
                                         cmp al, '/'
1268 0000A51A 7506
                               <1>
                                               short loc_scan_next_slash_pos
                                         jne
                                              [Last Slash Pos], esi
1269 0000A51C 8935[AA620100]
                            <1>
                                        mov
                               <1> loc_scan_next_slash_pos:
1271 0000A522 46
                               <1> inc esi
                                               al, [esi]
1272 0000A523 8A06
                               <1>
                                         mov
1273 0000A525 3C20
                                    cmp al, 20h
                               <1>
short loc_scan_ppn_dslash
                                               dword [Last_Slash_Pos], 0
                                              short pass_ppn_cdir
1277
                               <1>
1278 0000A532 8B0D[AA620100]
                               <1>
                                               ecx, [Last_Slash_Pos]
                                      mov
1279 0000A538 8B35[A6620100]
                               <1>
                                               esi, [First_Path_Pos]
1280 0000A53E 29F1
                               <1>
                                       sub
                                               ecx, esi
1281 0000A540 41
                               <1>
                                        inc
                                               ecx
                               <1>
1282
                                              ecx, 64
                                         ;cmp
                               <1> cmp
1283 0000A541 80F940
                                               cl, 64
1284 0000A544 7715
                               <1>
                                               short loc ppn invalid drive stc
                                       ja
```

```
1285
1286 0000A546 89F8
                                 <1>
                                          mov
                                                eax, edi ; Dest Dir String Location (65 byte)
1287 0000A548 F3A4
                                 <1>
                                          rep
                                                 movsb
1288
                                 <1>
                                          ;mov [edi], cl ; 0, End of Dir String
1289 0000A54A 8B35[AA620100]
                                 <1>
                                                esi, [Last_Slash_Pos]
                                          mov
1290 0000A550 46
                                 <1>
                                          inc
                                                 esi
1291 0000A551 89C7
                                <1>
                                          mov edi, eax
1292 0000A553 AC
                                <1>
                                          lodsb
1293 0000A554 3C21
                                <1>
                                           cmp
                                                al, 21h
                                        jnb
                                                short loc_ppn_get_filename
1294 0000A556 7381
                                <1>
1295
                                <1> loc_ppn_clc_return:
1296
                                        ;clc
                                 <1>
1297 0000A558 31C0
                                <1>
                                           xor
                                                 eax, eax
1298 0000A55A C3
                                <1>
                                           retn
1299
                                 <1>
                                <1> loc_ppn_invalid_drive_stc:
1300
1301 0000A55B F5
                                         cmc ; stc
                                 <1>
1302
                                 <1> loc_ppn_invalid_drive:
                                        ; cf = 1
1303
                                 <1>
1304
                                 <1>
                                          ; The Drive Letter/Char < "A" or > "Z"
1305 0000A55C 66B80F00
                                 <1>
                                        mov ax, 0Fh
                                        ; MS-DOS Error Code OFh = Disk Drive Invalid
1306
                                 <1>
1307
                                 <1>
                                          ; (MainProg ErrMsg: "Drive not ready or read error!")
1308 0000A560 C3
                                 <1>
1309
                                 <1>
                                 <1> find_longname:
1310
                                      ; 13/02/2016 (TRDOS 386 = TRDOS v2.0)
1311
                                 <1>
1312
                                 <1>
                                          ; 24/01/2010 (DIR.ASM, 'proc_find_longname')
1313
                                 <1>
                                          ; 17/10/2009
1314
                                 <1>
                                        ; INPUT ->
1315
                                 <1>
1316
                                 <1>
                                                 ESI = DOS short file name address
                                          ;
                                                 for example: "filename.ext"
1317
                                 <1>
                                           ;
1318
                                 <1>
                                         ; OUTPUT ->
1319
                                 <1>
1320
                                 <1>
                                               ESI = ASCIIZ longname address (cf = 0)
                                                 cf = 1 -> error number returns in EAX (AL)
1321
                                 <1>
                                           ;
1322
                                 <1>
                                                AL = 0 \& CF=1 \rightarrow longname not found
                                          ;
                                                     the file/directory has no longname
1323
                                 <1>
                                                cf = 0 -> AL = FAT Type
1324
                                 <1>
1325
                                 <1>
                                          ; 17/10/2009
1326
                                 <1>
                                          ; ASCIIZ string will be returned
1327
                                 <1>
                                          ; as LongFileName
1328
                                 <1>
1329
                                 <1>
                                          ; clearing/reset is not needed
1330
                                 <1>
                                          ;mov ecx, 33
                                          ;mov edi, LongFileName
1331
                                 <1>
                                          ; sub ax, ax ; 0
1332
                                 <1>
                                          ;rep stosw
1333
                                 <1>
1334
                                 <1>
1335
                                 <1>
                                          ;mov byte [LongNameFound], 0
1336
                                 <1>
                                          ; ESI = ASCIIZ file/directory name address
1337
                                 <1>
                                          ; AL = Attributes AND mask
1338
                                 <1>
1339
                                 <1>
                                               (Result of AND must be equal to AL)
                                         ;
                                          ; AH = Negative attributes mask
1340
                                 <1>
                                          ; (Result of AND must be ZERO)
1341
                                 <1>
                                          mov ax, 0800h
1342 0000A561 66B80008
                                 <1>
1343
                                 <1>
                                                 ; it must not be volume name or longname
1344 0000A565 E87DDDFFFF
                                 <1>
                                           call find_first_file
1345 0000A56A 7216
                                 <1>
                                                short loc_fln_retn
                                 <1>
1347
                                 <1> loc_fln_check_FAT_Type:
1348 0000A56C 803D[85590100]01 <1>
                                          cmp byte [Current_FATType], 1
1349 0000A573 7306
                                 <1>
                                                short loc_fln_check_longname_yes_sign
                                           jnb
1350
                                 <1>
1351 0000A575 E839000000
                                <1>
                                          call get_fs_longname
1352 0000A57A C3
                                <1>
                                          retn
1353
                                 <1>
                                <1> loc_fln_check_longname_yes_sign:
1354
1355 0000A57B 08FF
                                <1>
                                           or bh, bh
1356 0000A57D 7504
                                 <1>
                                           jnz short loc fln check longnamefound number
1357
                                 <1> loc_fln_longname_not_found_retn:
1358 0000A57F 31C0
                                <1>
                                         xor eax, eax
1359
                                 <1>
                                          ; cf = 1 \& al = 0 \rightarrow longname not found
1360 0000A581 F9
                                 <1>
                                          stc
1361
                                 <1> loc fln retn:
1362 0000A582 C3
                                 <1>
1363
                                 <1>
                                 <1> loc_fln_check_longnamefound_number:
1364
                                       ; 'LongNameFound' is \overline{\text{set}} by
1365
                                 <1>
                                             ; by 'save_longname_sub_component'
1366
                                 <1>
1367
                                 <1>
                                           ; which is called from
1368
                                           ; 'find_directory_entry'
                                 <1>
1369
                                 <1>
                                          ; which is called from
1370
                                 <1>
                                           ; 'find_first_file'
                                          ; It must 1 if the longname is valid
                                 <1>
1372 0000A583 803D[81610100]01
                                           cmp byte [LongNameFound], 1
                                 <1>
1373 0000A58A 75F3
                                 <1>
                                           jne short loc_fln_longname_not_found_retn
1374
                                 <1>
1375
                                 <1> loc fln calculate checksum:
1376 0000A58C E813000000
                                 <1>
                                          call calculate_checksum
1377
                                 <1>
                                           ; AL = shortname checksum
1378
                                 <1>
1379
                                 <1> loc_fln_longname_validation:
1380
                                           ; 'LFN CheckSum' has been set already
                                 <1>
                                           ; by 'save longname sub component'
1381
                                 <1>
                                           ; which is called from
1382
                                 <1>
                                           ; 'find directory entry'
1383
                                 <1>
1384
                                           ; which is called from
                                 <1>
1385
                                 <1>
                                           ; 'find_first_file'
1386 0000A591 3805[83610100]
                                 <1>
                                                 [LFN CheckSum], al
                                           cmp
1387 0000A597 75E6
                                 <1>
                                           jne
                                                 short loc fln longname not found retn
                                 <1>
1389 0000A599 BE[84610100]
                                 <1>
                                                 esi, LongFileName
                                          mov
```

<1>

```
al, [Current_FATType]
1391 0000A5A3 C3
                                  <1>
                                            retn
1392
                                  <1>
1393
                                  <1> calculate checksum:
1394
                                  <1>
                                           ; 13/02/2016 (TRDOS 386 = TRDOS v2.0)
1395
                                  <1>
                                            ; 17/10/2009 (DIR.ASM, 'proc_calculate_checksum')
1396
                                  <1>
                                           ; INPUT ->
1397
                                  <1>
1398
                                  <1>
                                                  ESI = 11 byte DOS File Name location
1399
                                  <1>
                                                  (in DOS Directory Entry Format)
                                            ;
                                            ; OUTPUT ->
1400
                                  <1>
1401
                                  <1>
                                                   AL = 8 bit checksum (CRC) value
                                            ;
1402
                                  <1>
                                           ; (Modified registers: EAX, ECX, ESI)
1403
                                  <1>
1404
                                  <1>
1405
                                  <1>
                                            ; Erdogan Tan [ 17-10-2009 ]
                                            ; 'ror al, 1' instruction
1406
                                  <1>
                                  <1>
1407
1408
                                  <1>
                                            ; Erdogan Tan [ 20-06-2004 ]
1409
                                  <1>
                                            ; This 8086 assembly code is an original code
                                            ; which is adapted from C code in
1410
                                  <1>
                                            ; Microsoft FAT32 File System Specification
1411
                                  <1>
                                  <1>
                                            ; Version 1.03, December 6, 2000
1412
1413
                                  <1>
                                            ; Page 28
                                  <1>
1414
1415 0000A5A4 30C0
                                  <1>
                                            xor
                                                 al, al
1416 0000A5A6 B90B000000
                                  <1>
                                           mov ecx, 11
1417
                                  <1> loc_next_sum:
1418
                                  <1>
                                           ;xor ah, ah
1419
                                  <1>
                                            ;test al, 1
1420
                                  <1>
                                            ;jz short pass_ah_80h
1421
                                  <1>
                                            ;mov ah, 80h
1422
                                  <1> ;pass_ah_80h:
1423
                                  <1>
                                       ;shr al, 1
                                            ror al, 1 ; 17/10/2009
1424 0000A5AB D0C8
                                  <1>
1425 0000A5AD 0206
                                  <1>
                                            add
                                                  al, [esi]
1426 0000A5AF 46
                                 <1>
                                            inc
                                                  esi
1427
                                  <1>
                                            ;add al, ah
1428 0000A5B0 E2F9
                                  <1>
                                            loop
                                                  loc_next_sum
1429 0000A5B2 C3
                                 <1>
                                            retn
1430
                                  <1>
1431
                                  <1> get_fs_longname:
1432
                                           ; temporary (13/02/2016)
                                  <1>
1433 0000A5B3 31C0
                                  <1>
                                            xor eax, eax
1434 0000A5B5 F9
                                  <1>
                                            stc
1435 0000A5B6 C3
                                  <1>
                                            retn
1436
                                  <1>
                                  <1> make_sub_directory:
1437
                                        ; 16/10/2016
1438
                                  <1>
1439
                                  <1>
                                            ; 02/03/2016, 03/03/2016
1440
                                  <1>
                                          ; 26/02/2016, 27/02/2016
                                          ; 21/02/2016 (TRDOS 386 = TRDOS v2.0)
; 01/08/2011 (DIR.ASM, 'proc_make_directory')
1441
                                  <1>
                                  <1>
1442
                                          ; 10/07/2010
1443
                                  <1>
                                          ; INPUT ->
1444
                                  <1>
1445
                                  <1>
                                                  ESI = ASCIIZ Directory Name
                                                  CL = Directory Attributes
1446
                                            ;
                                            ; OUTPUT ->
                                  <1>
1447
1448
                                  <1>
                                                  EAX = New sub dir's first cluster
1449
                                  <1>
                                                  ESI = Logical Dos Drv Descr. Table Addr.
                                                  CF = 1 -> error code in AL (EAX)
1450
                                  <1>
1451
                                  <1>
1452
                                  <1>
                                            ;test cl, 10h ; directory
1453
                                  <1>
                                            ;jz short loc_make_directory_access_denied
1454
                                  <1>
                                            ;test cl, 08h ; volume name
1455
                                  <1>
                                            ;jnz short loc_make_directory_access_denied
                                  <1>
1457 0000A5B7 80E107
                                            and
                                                   cl, 07h
                                  <1>
1458 0000A5BA 880D[00630100]
                                  <1>
                                                  byte [mkdir_attrib], cl
                                            mov
1459
                                  <1>
1460 0000A5C0 56
                                  <1>
                                            push esi
1461 0000A5C1 31DB
                                  <1>
                                            xor
                                                  ebx, ebx
1462 0000A5C3 8A3D[86590100]
                                                  bh, [Current_Drv]
                                  <1>
                                            mov
1463 0000A5C9 BE00010900
                                                  esi, Logical DOSDisks
                                  <1>
1464 0000A5CE 01DE
                                  <1>
                                            add
                                                  esi, ebx
1465 0000A5D0 5B
                                  <1>
                                            pop
                                                  ebx
1466
                                  <1>
                                            ; 10/07/2010 \rightarrow 1st writable disk check for trdos
1467
                                  <1>
1468
                                  <1>
                                            ; LD_DiskType = 0 for write protection (read only)
                                            cmp byte [esi+LD_DiskType], 1; 0 = Invalid
jnb short loc_mkdir_check_file_sytem
1469 0000A5D1 807E0101
                                  <1>
1470 0000A5D5 730B
                                  <1>
                                  <1>
                                            ; 16/10/2016 (13h -> 30)
1472 0000A5D7 B81E000000
                                                  eax, 30 ; 'Disk write-protected' error
                                 <1>
                                            mov
1473 0000A5DC BA00000000
                                  <1>
                                            mov edx, 0
                                            ; err retn: EDX = 0, EBX = Dir name offset
1474
                                  <1>
1475
                                  <1>
                                            ;ESI = Logical DOS drive description table address
1476 0000A5E1 C3
                                  <1>
1477
                                  <1>
                                  <1> ;loc make directory access denied:
1478
1479
                                  <1>
                                           ;mov ax, 05h; access denied (invalid attributes input)
1480
                                  <1>
                                            ;stc
1481
                                  <1>
                                            ;retn
1482
                                  <1>
1483
                                  <1> loc_mkdir_check_file_sytem:
                                            cmp byte [esi+LD_FATType], 1
jnb short loc_mkdir_check_free_sectors
1484 0000A5E2 807E0301
                                 <1>
1485 0000A5E6 730B
                                 <1>
                                            jnb
                                  <1>
1487
                                 <1> loc_make_fs_directory:
1488 0000A5E8 A1[80590100]
                                 <1>
                                           mov eax, [Current_Dir_FCluster]
                                            ; EAX = Parent directory DDT Address
1489
                                 <1>
1490
                                 <1>
                                           ; ESI = Logical DOS Drive DT Address
1491
                                 <1>
                                            ; EBX = Directory name offset (as ASCIIZ name)
                                            call make_fs_directory
1492 0000A5ED E8D5150000
                                 <1>
1493 0000A5F2 C3
                                  <1>
1494
                                  <1>
```

1390 0000A59E A0[85590100]

<1>

mov

```
<1> loc_mkdir_check_free_sectors:
1495
                                <1> movzx eax, byte [esi+LD_BPB+SecPerClust]
1496 0000A5F3 0FB64613
                                          mov ecx, [esi+LD_FreeSectors]
1497 0000A5F7 8B4E74
                                <1>
1498 0000A5FA 39C1
                                <1>
                                          cmp ecx, eax
1499 0000A5FC 7255
                                <1>
                                          jb short loc_mkdir_insufficient_disk_space
1500
                                <1>
                                <1> loc make fat directory:
1501
                                       mov [mkdir_DirName_Offset], ebx
1502 0000A5FE 891D[F0620100]
                                <1>
1503 0000A604 890D[FC620100]
                                <1>
                                                [mkdir_FreeSectors], ecx
                                          mov
                                 <1>
1505
                                 <1>
                                          ;mov al, [esi+LD_BPB+SecPerClust]
1506 0000A60A A2[02630100]
                                          mov byte [mkdir SecPerClust], al
                                 <1>
1507
                                 <1>
1508
                                 <1> loc_mkdir_gffc_1:
1509 0000A60F E80F180000
                                 <1>
                                       call get_first_free_cluster
1510 0000A614 722A
                                                 short loc mkdir gffc retn
                                 <1>
                                          jс
                                 <1>
                                 <1> ;loc_mkdir_gffc_1_cont:
1512
                                          ;cmp eax, 2
;jb short loc_mkdir_gffc_insufficient_disk_space
1513
                                 <1>
1514
                                 <1>
1515
                                 <1>
1516
                                 <1> ;loc_mkdir_gffc_1_save_fcluster:
1517 0000A616 A3[F4620100]
                                 <1>
                                         mov [mkdir_FFCluster], eax
1518
                                 <1>
                                 <1> loc mkdir_locate_ffe:
1519
1520
                                 <1>
                                         ; Current directory fcluster <> Directory buffer cluster
1521
                                 <1>
                                          ; Current directory will be reloaded by
                                 <1>
1522
                                         ; 'locate_current_dir_file' procedure
1523
                                 <1>
1524
                                 <1>
                                          ; ESI = Logical DOS Drive Description Table Address
1525
                                 <1>
                                          ;push esi ; 27/02/2016
1526 0000A61B 31C0
                                 <1>
                                          xor eax, eax
1527 0000A61D 89C1
                                <1>
                                          mov ecx, eax
1528 0000A61F 6649
                                <1>
                                          dec cx ; FFFFh
                                        ; CX = FFFFh -> find first deleted or free entry
1529
                                 <1>
1530
                                <1>
                                          ; ESI would be ASCIIZ filename address if the call
                                 <1>
                                          ; would not be for first free or deleted dir entry
1532 0000A621 E8D0FAFFFF
                                <1>
                                          call locate_current_dir_file
1533 0000A626 734C
                                <1>
                                                short loc_mkdir_set_ff_dir_entry_1
                                          jnc
                                          ;pop esi
1534
                                <1>
                                          ; ESI = Logical DOS Drive Description Table Address
1535
                                <1>
1536 0000A628 83F802
                                 <1>
                                          cmp eax, 2 ; cmp al, 2 ; File/Dir not found !
1537 0000A62B 752B
                                <1>
                                                short loc_mkdir_stc_return
                                          jne
1538
                                 <1>
                                 <1> loc_mkdir_add_new_cluster:
1539
                                       cmp byte [Current_FATType], al ; 2
1540 0000A62D 3805[85590100]
                                <1>
                                          ;cmp byte ptr [esi+LD FATType], 2
                                <1>
1542 0000A633 770C
                                                short loc_mkdir_add_new_cluster_check_fsc
                                 <1>
                                          jа
1543 0000A635 803D[84590100]01
                                 <1>
                                          cmp
                                                 byte [Current_Dir_Level], 1
1544
                                 <1>
                                          ;cmp byte [esi+LD_CDirLevel], 1
                                                short loc_mkdir_add_new_cluster_check_fsc
1545 0000A63C 7303
                                 <1>
1546
                                 <1>
1547 0000A63E B00C
                                 <1>
                                          mov
                                                al, 12 ; No more files
1548
                                 <1> loc_mkdir_gffc_retn:
1549 0000A640 C3
                                 <1>
                                          retn
1550
                                 <1>
                                 <1> loc mkdir add new cluster check fsc:
1552 0000A641 8B0D[FC620100]
                                       mov ecx, [mkdir_FreeSectors]
                                 <1>
1553
                                 <1>
                                          ;movzx eax, byte [mkdir_SecPerClust]
                                          mov al, [mkdir SecPerClust]
1554 0000A647 A0[02630100]
                                <1>
1555 0000A64C 66D1E0
                                <1>
                                          shl
                                                ax, 1 ; AX = 2 * AX
1556 0000A64F 39C1
                                 <1>
                                          cmp
                                                ecx, eax
1557 0000A651 7350
                                <1>
                                                short loc_mkdir_add_new_subdir_cluster
                                          jnb
1558
                                 <1>
                                 <1> loc_mkdir_insufficient_disk_space:
1559
1560
                                 <1>
                                         ;mov edx, ecx
1561
                                 <1> ;loc_mkdir_gffc_insufficient_disk_space:
                                       mov ax, 27h; MSDOS err => insufficient disk space
1562 0000A653 66B82700
                                 <1>
1563
                                 <1>
                                          ; err retn: EDX = Free sectors, EBX = Dir name offset
                                           ; ESI -> Dos drive description table address
1564
                                 <1>
1565
                                 <1>
                                          ;; ecx = edx
1566
                                 <1>
1567 0000A657 C3
                                 <1>
                                          retn
1568
                                <1>
                                 <1> loc_mkdir_stc_return:
1569
1570 0000A658 F9
                                <1>
                                          stc
1571 0000A659 C3
                                 <1>
                                          retn
1572
                                 <1>
1573
                                 <1> loc_mkdir_gffc_2:
1574 0000A65A E8C4170000
                                      call get_first_free_cluster
                                <1>
1575 0000A65F 72DF
                                 <1>
                                           jc short loc_mkdir_gffc_retn
                                 <1>
1577
                                 <1> ;loc_mkdir_gffc_1_cont:
1578
                                          ; cmp eax, 2
1579
                                          ;jb short loc mkdir gffc insufficient disk space
                                 <1>
1580
                                 <1>
                                 <1> ;loc mkdir gffc 2 save fcluster:
1581
1582 0000A661 A3[F4620100]
                                          mov [mkdir FFCluster], eax
                                 <1>
1583
                                 <1>
1584 0000A666 A1[F8620100]
                                 <1>
                                                eax, [mkdir_LastDirCluster]
                                          mov
1585
                                 <1>
1586 0000A66B E842170000
                                 <1>
                                          call load_FAT_sub_directory
1587 0000A670 72CE
                                                 short loc mkdir gffc retn
                                 <1>
                                          jс
                                 <1>
1589 0000A672 31FF
                                 <1>
                                          xor edi, edi
                                 <1> loc mkdir set ff dir entry 1:
1590
1591
                                 <1>
                                          ; 27/02/2016
1592 0000A674 56
                                 <1>
                                          push esi; Logical DOS Drv Desc. Tbl. address
1593
                                 <1>
                                          ; EDI = Directory Entry Address
1594 0000A675 8B35[F0620100]
                                 <1>
                                          mov esi, [mkdir_DirName_Offset]
1595 0000A67B A1[F4620100]
                                 <1>
                                                eax, [mkdir_FFCluster]
                                          mov
                                 <1>
                                                 cx, 10h ; CL = Directory attribute
1597 0000A680 66B91000
                                 <1>
                                          mov
                                 <1>
                                                      ; CH = 0 \rightarrow File size is 0
1599 0000A684 0A0D[00630100]
                                                 cl, [mkdir attrib]; S, H, R
                                 <1>
                                          or
```

```
1600 0000A68A E8B0010000
                                <1>
                                          call make_directory_entry
                                 <1>
1602 0000A68F 5E
                                <1>
                                          pop
1603
                                <1>
                                          mov byte [DirBuff_ValidData], 2
call save_directory_buffer
1604 0000A690 C605[AC600100]02
                                <1>
                                          mov
1605 0000A697 E880020000
                                <1>
1606 0000A69C 0F83DA000000
                                          jnc loc_mkdir_set_ff_dir_entry_2
                                <1>
1607
                                <1>
1608
                                <1> loc_mkdir_return:
1609 0000A6A2 C3
                                <1>
                                          retn
1610
                                <1>
1611
                                <1> loc_mkdir_add_new_subdir_cluster:
                                      mov edx, [DirBuff Cluster]
1612 0000A6A3 8B15[B1600100]
                                <1>
1613 0000A6A9 8915[F8620100]
                                <1>
                                          mov [mkdir_LastDirCluster], edx
1614
                                <1>
1615 0000A6AF A1[F4620100]
                                <1>
                                         mov
                                                eax, [mkdir_FFCluster]
                                      call load FAT sub directory
1616 0000A6B4 E8F9160000
                                <1>
                                       jc short loc_mkdir_return
1617 0000A6B9 72E7
                                <1>
1618
                                <1>
                                          ; eax = 0
1619
                                <1>
                                         ; ecx = directory buffer sector count (<= 128)
1620
                                <1>
                                <1> pass mkdir add new subdir cluster:
1621
1622 0000A6BB 29FF
                                      sub edi, edi ; 0
                                <1>
                                <1>
                                          ;mov al, 128; double word
1624
                                <1>
                                          ;mul ecx ; ecx = directory buffer sector count
1625
                                <1>
                                          ;mov ecx, eax
1626
                                <1>
                                         ;shl cx, 7; 128 * sector count
                          1627 0000A6BD 668B4611
1628 0000A6C1 66C1E802
                                                ax, 2; 'byte count / 4' for 'stosd'
                                                cx ; max = 128*(512/4) \rightarrow 16384 (stosd)
                                         mul
1629 0000A6C5 66F7E1
                                     mov cx, ax
sub ax, ax; 0
rep stosd; clear directory buffer
1630 0000A6C8 6689C1
                               <1>
1631 0000A6CB 6629C0
                                <1>
1632 0000A6CE F3AB
                                <1>
1633
                                <1>
                                     mov byte [DirBuff_ValidData], 2
call save_directory_buffer
jc short loc_mkdir_return
1634 0000A6D0 C605[AC600100]02 <1>
1635 0000A6D7 E840020000
                                <1>
1636 0000A6DC 72C4
                                <1>
1637
                                <1>
1638
                                <1> loc_mkdir_save_added_cluster:
1639 0000A6DE A1[F8620100]
                               <1> mov eax, [mkdir_LastDirCluster]
<1> mov ecx, [mkdir_FFCluster]
1640 0000A6E3 8B0D[F4620100]
1647 0000A6FA 7518
                                        jnz short loc_mkdir_save_fat_buffer_stc_retn
                                <1>
1648
                                <1>
1649
                                <1> loc_mkdir_save_fat_buffer_0:
                                         mov eax, [mkdir FFCluster]
1650 0000A6FC A1[F4620100]
                               <1>
1651 0000A701 A3[F8620100]
                                <1>
                                          mov
                                                [mkdir_LastDirCluster], eax
1652
                                <1>
1653 0000A706 31C9
                                <1>
                                     dec ecx; FFFFFFFFh; ESI = Logical DOS Drive Description Tacall update_cluster
jnc short loc_mkdir_save_fat_buffer_1
or eax. eax
                                        xor ecx, ecx
1654 0000A708 49
                                <1>
1655
                                <1>
                                          ; ESI = Logical DOS Drive Description Table address
                           <1>
1656 0000A709 E8E8170000
1657 0000A70E 731A
                                <1>
1658 0000A710 09C0
                                <1>
                                         or
                                                eax, eax
                                         jz
                                                short loc_mkdir_save_fat_buffer_1
1659 0000A712 7416
                                <1>
1660
                                <1>
                                <1> loc mkdir save_fat_buffer_stc_retn:
1661
                                          ; 01/03/2016
1662
                                <1>
1663 0000A714 803D[A2600100]01 <1>
                                          cmp byte [FAT_ClusterCounter], 1
1664 0000A71B 720C
                                <1>
                                          jb short loc_mkdir_save_fat_buffer_retn
1665
                                <1>
1666 0000A71D 66BB00FF
                                <1>
                                                bx, 0FF00h ; recalculate free space (BL = 0)
                                <1>
                                                         ; (BH = FFh -> Use ESI as Drv Param. Tbl.)
1667
1668 0000A721 50
                                <1>
                                          push eax
                                          call calculate_fat_freespace
1669 0000A722 E8211B0000
                                <1>
1670 0000A727 58
                                <1>
1671 0000A728 F9
                                <1>
                                          stc
1672
                                <1> loc_mkdir_save_fat_buffer_retn:
1673 0000A729 C3
                                <1>
1674
                                <1>
                                <1> loc mkdir save fat buffer 1:
1675
                                <1> ; byte [FAT_BuffValidData] = 2
1677 0000A72A E8841A0000
                                <1>
                                         call save fat buffer
1678 0000A72F 72E3
                                <1>
                                        jc short loc_mkdir_save_fat_buffer_stc_retn
1679
                                <1>
                                          ; 01/03/2016
1680
                                <1>
1681 0000A731 803D[A2600100]01
                                          cmp byte [FAT_ClusterCounter], 1
                                 <1>
                                                 short loc_mkdir_save_fat_buffer_2
1682 0000A738 721B
                                 <1>
                                 <1>
                                 <1>
1684
                                          ; ESI = Logical DOS Drive Description Table address
                                                 eax, [FAT ClusterCounter]
1685 0000A73A A1[A2600100]
                                 <1>
                                          mov
                                                bx, OFF01h; add free clusters
1686 0000A73F 66BB01FF
                                 <1>
                                          mov
1687 0000A743 E8001B0000
                                <1>
                                          call calculate fat freespace
1688
                                 <1>
                                          ;inc eax; OFFFFFFFF -> 0; recalculation is needed!
1689
                                 <1>
1690
                                 <1>
                                          ;jnz short loc_mkdir_save_fat_buffer_2
1691
                                 <1>
                                          ; ecx > 0 -> Recalculation is needed
1692
                                 <1>
1693 0000A748 09C9
                                 <1>
                                                 ecx, ecx
1694 0000A74A 7409
                                 <1>
                                          jΖ
                                                short loc_mkdir_save_fat_buffer_2
1695
                                 <1>
1696 0000A74C 66BB00FF
                                 <1>
                                                bx, OFFOOh; ; recalculate free space
1697 0000A750 E8F31A0000
                                 <1>
                                          call calculate_fat_freespace
1698
                                 <1>
                                 <1> loc mkdir save fat buffer 2:
1699
1700 0000A755 C605[03630100]01
                                          mov byte [mkdir add new cluster], 1
                                <1>
1701 0000A75C E9C4000000
                                 <1>
                                                loc mkdir upd parent dir lmdt
1702
                                 <1>
1703
                                 <1> loc mkdir update sub dir cluster:
1704 0000A761 A1[F4620100]
                                         mov eax, [mkdir FFCluster]
                                 <1>
```

```
1705 0000A766 29C9
                                 <1>
                                          sub ecx, ecx; 0
                                 <1>
                                          ; 01/03/2016
                                          mov [FAT_ClusterCounter], ecx; 0; Reset
1707 0000A768 890D[A2600100]
                                 <1>
1708 0000A76E 49
                                 <1>
                                           dec ecx; OFFFFFFFh
1709
                                 <1>
1710
                                 <1>
                                           ; ESI = Logical DOS Drive Descisption Table address
1711 0000A76F E882170000
                                 <1>
                                          call update cluster
                                         jnc
1712 0000A774 7379
                                 <1>
                                                 short loc_mkdir_save_fat_buffer_3
1713 0000A776 09C0
                                 <1>
                                                 eax, eax ; EAX = 0 \rightarrow cluster value is 0 or eocc
                                           or
                                          jz
1714 0000A778 7475
                                                 short loc_mkdir_save_fat_buffer_3
                                <1>
                                           ; 01/03/2016
1715
                                 <1>
1716 0000A77A EB98
                                 <1>
                                           jmp short loc_mkdir_save_fat_buffer_stc_retn
1717
                                 <1>
1718
                                 <1> loc_mkdir_set_ff_dir_entry_2:
1719
                                          ; ESI = Logical DOS Drive Description Table address
                                 <1>
                                           mov eax, [mkdir FFCluster]
1720 0000A77C A1[F4620100]
                                 <1>
                                 <1>
                                           ; Load disk sectors as a directory cluster
1722 0000A781 E82C160000
                                 <1>
                                           call load_FAT_sub_directory
1723 0000A786 7266
                                 <1>
                                           jc short retn_make_fat_directory
1724
                                 <1>
1725
                                 <1>
                                           ; eax = 0
1726
                                 <1>
                                           ; ecx = directory buffer sector count (<= 128)
1727
                                 <1>
1728 0000A788 BF40000800
                                 <1>
                                           mov edi, Directory_Buffer + 64; 26/02/2016
1729
                                 <1>
1730
                                 <1>
                                           ; 02/03/2016
                                           mov ax, [esi+LD_BPB+BytesPerSec] ; 512
1731 0000A78D 668B4611
                                <1>
                                                 ax, 2; 'byte count / 4' for 'stosd'
1732 0000A791 66C1E802
                                 <1>
                                           shr
1733 0000A795 F7E1
                                 <1>
                                           mul
1734 0000A797 89C1
                                <1>
                                           mov
                                                 ecx, eax
1735 0000A799 6629C0
                                <1>
                                           sub
                                                 ax, ax
1736 0000A79C F3AB
                                 <1>
                                          rep
                                                 stosd
1737
                                 <1>
1738
                                 <1>
                                          ;; mov al, 128; double word
                                           ;; mul ecx ; ecx = directory buffer sector count
1739
                                 <1>
1740
                                 <1>
                                           ;;mov ecx, eax
1741
                                 <1>
                                           ;shl cx, 7; 128 * sector count
1742
                                 <1>
                                           ;;sub eax, eax
                                           ;;sub al, al ; 0
1743
                                 <1>
                                           ;rep stosd ; clear directory buffer
1744
                                 <1>
1745
                                 <1>
1746 0000A79E BF00000800
                                                 edi, Directory Buffer; 26/02/2016
                                 <1>
                                           mov
1747
                                 <1>
1748 0000A7A3 56
                                 <1>
                                           push
                                                 esi
1749
                                 <1>
1750 0000A7A4 BE[04630100]
                                 <1>
                                           mov
                                                  esi, mkdir_Name
1751 0000A7A9 66C7062E00
                                                 word [esi], 2Eh; db '.', '0'
                                 <1>
                                           mov
1752
                                 <1>
1753 0000A7AE A1[F4620100]
                                 <1>
                                                  eax, [mkdir_FFCluster]
                                                 cx, 10h; \overline{CL} = Directory attribute
1754 0000A7B3 66B91000
                                 <1>
                                           mov
1755
                                 <1>
                                                        ; CH = 0 \rightarrow File size is 0
1756 0000A7B7 E883000000
                                 <1>
                                           call
                                                make_directory_entry
1757
                                 <1>
1758 0000A7BC BF20000800
                                 <1>
                                                  edi, Directory_Buffer + 32 ; 26/02/2016
1759
                                 <1>
1760
                                 <1>
                                           ; 03/03/2016
1761
                                 <1>
                                           ; Following modification has been done according to
1762
                                           ; 'Microsoft Extensible Firmware Initiative
                                 <1>
1763
                                 <1>
                                           ; FAT32 File System Specification' document,
1764
                                          ; 'FAT: General Overview of On-Disk Format-Page 25'.
                                 <1>
                                          ; "Finally, you set DIR_FstClusLO and DIR_FstClusHI
1765
                                 <1>
1766
                                 <1>
                                           ; for the dotdot entry (the second entry) to the
1767
                                 <1>
                                           ; first cluster number of the directory in which you
1768
                                 <1>
                                           ; just created the directory (value is 0 if this directory
1769
                                 <1>
                                           ; is the root directory even for FAT32 volumes)."
1770
                                 <1>
                                           ; (Correctness of this modification has been verified
                                           ; by using Windows 98 'scandisk.exe'.)
1771
                                 <1>
1772
                                 <1>
1773 0000A7C1 29C0
                                 <1>
                                           sub
                                                 eax, eax
1774 0000A7C3 3805[84590100]
                                 <1>
                                                 byte [Current_Dir_Level], al ; 0
                                           cmp
                                                 short loc mkdir set ff dir entry 3
1775 0000A7C9 7605
                                 <1>
1776 0000A7CB A1[80590100]
                                 <1>
                                           mov
                                                 eax, [Current_Dir_FCluster] ; parent dir
                                 <1> loc_mkdir_set_ff_dir_entry_3:
1777
1778 0000A7D0 66C746012E00
                                 <1>
                                          mov word [esi+1], 2Eh; db'.', '0'
1779
                                 <1>
                                           ;mov cx, 10h
1780
                                 <1>
1781 0000A7D6 E864000000
                                 <1>
                                           call make_directory_entry
1782
                                 <1>
1783 0000A7DB 5E
                                 <1>
                                           pop
1784
                                 <1>
                                                 byte [DirBuff_ValidData], 2
1785 0000A7DC C605[AC600100]02
                                 <1>
                                           mov
                                                 save_directory_buffer
1786 0000A7E3 E834010000
                                 <1>
                                           call
1787 0000A7E8 0F8373FFFFFF
                                 <1>
                                             jnc
                                                     loc_mkdir_update_sub_dir_cluster
1788
                                 <1>
1789
                                 <1> retn_make_fat_directory:
1790 0000A7EE C3
                                 <1>
                                           retn
1791
                                 <1>
1792
                                 <1> loc_mkdir_save_fat_buffer_3:
1793
                                 <1>
                                           ; 01/03/2016
1794
                                           ; byte [FAT BuffValidData] = 2
                                 <1>
1795 0000A7EF E8BF190000
                                           call save fat buffer
                                 <1>
                                           jс
                                                    loc mkdir save fat buffer stc retn
1796 0000A7F4 0F821AFFFFFF
                                 <1>
1797
                                 <1>
                                                 byte [FAT ClusterCounter], 1
1798 0000A7FA 803D[A2600100]01
                                <1>
1799 0000A801 721B
                                 <1>
                                           jb
                                                 short loc_mkdir_save_fat_buffer_4
1800
                                 <1>
1801
                                 <1>
                                           ; ESI = Logical DOS Drive Description Table address
1802 0000A803 A1[A2600100]
                                 <1>
                                           mov
                                                  eax, [FAT_ClusterCounter]
1803 0000A808 66BB01FF
                                 <1>
                                                 bx, OFF01h; add free clusters
                                           call calculate_fat_freespace
1804 0000A80C E8371A0000
                                 <1>
1805
                                 <1>
1806
                                 <1>
                                          ;inc eax; OFFFFFFFF -> 0; recalculation is needed!
1807
                                 <1>
                                           ;jnz short loc_mkdir_save_fat_buffer_4
1808
                                 <1>
1809
                                 <1>
                                           ; ecx > 0 -> Recalculation is needed
```

```
or ecx, ecx
1810 0000A811 09C9
                              <1>
                                     jz short loc_mkdir_save_fat_buffer_4
1811 0000A813 7409
                               <1>
1812
                               <1>
1813 0000A815 66BB00FF
                               <1>
                                       mov bx, 0FF00h; recalculate free space
1814 0000A819 E82A1A0000
                               <1>
                                        call calculate_fat_freespace
1815
                               <1>
                                <1> loc mkdir save fat buffer 4:
1817 0000A81E C605[03630100]00
                                        mov byte [mkdir_add_new_cluster], 0
                               <1>
1818
                                <1>
1819
                                <1> loc_mkdir_upd_parent_dir_lmdt:
1820 0000A825 E88D010000
                               <1>
                                        call update_parent_dir_lmdt
1821
                                <1>
                                         ; 01/03/2016
1822
                                <1>
1823 0000A82A 803D[03630100]00 <1>
                                      cmp byte [mkdir_add_new_cluster], 0
                                        ja loc_mkdir_gffc 2
1824 0000A831 0F8723FEFFFF
                               <1>
1825
                               <1>
                               <1> loc mkdir retn new dir cluster:
                               1827 0000A837 A1[F4620100]
1828 0000A83C 31D2
                               <1>
                                         xor
                                              edx, edx
                               <1> loc mkdir retn:
1829
1830 0000A83E C3
                                        retn
                               <1>
1831
                                <1>
1832
                                <1> make_directory_entry:
                                    ; 02/03/2016
1833
                                <1>
                                       ; 21/02/2016 (TRDOS 386 = TRDOS v2.0)
1834
                                <1>
                                        ; 09/08/2010 (DIR.ASM, 'proc_make_directory_entry')
1835
                                <1>
                                       ; 17/07/2010
1836
                                <1>
                                       ; INPUT ->
1837
                                <1>
1838
                                <1>
                                               EDI = Directory Entry Address
                                        ;
1839
                                <1>
                                               ESI = Dot File Name Location
                                         ;
1840
                                <1>
                                               EAX = First Cluster
                                <1>
                                               File Size = 0 (Must be set later)
1841
                                         ;
                                               CL = Attributes
1842
                                <1>
                                         ;
1843
                                <1>
                                               CH = 0 (File size = 0)
                                       ;
1844
                                <1>
                                               (If CH>0, File size is in dword [EBX]) (*)
                                         ; OUTPUT ->
1845
                                <1>
                                              EDI = Directory Entry Address
1846
                                <1>
                                         ;
                                               ESI = Dot File Name Location (Capitalized)
1847
                                <1>
                                         ;
1848
                                <1>
                                               If CH input = 0, File Size = 0
1849
                                <1>
                                               Otherwise file size is as dword [EBX] (*)
                                         ;
1850
                                <1>
                                               DX = Date, AX = Time in DOS Dir Entry format
                                <1>
                                               EBX = same
1851
                                         ;
1852
                                <1>
                                               ECX = same
                                         ;
1853
                                <1>
1854 0000A83F 51
                                <1>
                                        push ecx
1855
                               <1>
1856 0000A840 884F0B
                               <1>
                                         mov
                                               [edi+11], cl ; Attributes
1857 0000A843 6689471A
                               <1>
                                         mov
                                               [edi+26], ax ; FClusterLw, 26
                                               eax, 16
1858 0000A847 C1E810
                               <1>
                                         shr
1859 0000A84A 66894714
                               <1>
                                               [edi+20], ax ; FClusterHw, 20
                                         mov
1860 0000A84E 6631C0
                               <1>
                                         xor
1861 0000A851 6689470C
                               <1>
                                         mov
                                               [edi+12], ax; NTReserved, 12
1862
                               <1>
                                                          ; CrtTimeTenth, 13
1863 0000A855 08ED
                               <1>
                                         or
                                               ch, ch
1864 0000A857 7402
                                <1>
                                               short loc_make_direntry_set_filesize
                                         jz
1865
                               <1>
1866 0000A859 8B03
                                <1>
                                         mov
                                               eax, [ebx]
1867
                                <1>
1868
                                <1> loc_make_direntry_set_filesize:
1869 0000A85B 89471C
                                <1>
                                       mov [edi+28], eax; FileSize, 28
1870
                                <1>
1871 0000A85E E88AFBFFFF
                                <1>
                                         call convert file name
1872
                               <1>
                                         ;EDI = Dir Entry Format File Name Location
1873
                                <1>
                                        ;ESI = Dot File Name Location (capitalized)
1874
                                <1>
1875 0000A863 E816000000
                               <1>
                                         call convert current date time
                                         ; OUTPUT -> DX = Date in dos dir entry format
1876
                               <1>
                                               AX = Time in dos dir entry format
1877
                               <1>
                                         ;
1878 0000A868 6689470E
                               <1>
                                               [edi+14], ax ; CrtTime, 14
                                         mov
1879 0000A86C 66895710
                               <1>
                                              [edi+16], dx ; CrtDate, 16
                                         mov
1880 0000A870 66895712
                               <1>
                                         mov
                                              [edi+18], dx ; LastAccDate, 18
1881 0000A874 66894716
                               <1>
                                               [edi+22], ax; WrtTime, 14
                                         mov
                                               [edi+24], dx; WrtDate, 16
1882 0000A878 66895718
                               <1>
                                         mov
1883 0000A87C 59
                               <1>
                                        pop
1884
                                <1>
1885 0000A87D C3
                                <1>
                                         retn
1886
                                <1>
1887
                                <1> convert_current_date_time:
1888
                                <1>
                                     ; 21/02/2016 (TRDOS 386 = TRDOS v2.0)
1889
                                         ; 13/06/2010 (DIR.ASM, 'proc convert current date time')
                                <1>
1890
                                <1>
                                         ; converts date&time to dos dir entry format
                                         ; INPUT -> none
1891
                                <1>
                                         ; OUTPUT -> DX = Date in dos dir entry format
1892
                                <1>
1893
                                <1>
                                                   AX = Time in dos dir entry format
1894
                                <1>
1895 0000A87E B404
                                               ah, 04h ; Return Current Date
                               <1>
                                         mov
1896 0000A880 E852B1FFFF
                                         call int1Ah
                               <1>
1897
                               <1>
1898 0000A885 88E8
                               <1>
                                               al, ch ; <- century BCD
1899 0000A887 240F
                               <1>
                                               al, OFh
                                         and
1900 0000A889 88EC
                               <1>
                                         mov
                                               ah, ch
1901 0000A88B C0EC04
                               <1>
                                         shr
                                               ah, 4
1902 0000A88E D50A
                               <1>
                                         aad
1903 0000A890 88C5
                               <1>
                                               ch, al ; -> century
1904
                               <1>
1905 0000A892 88C8
                               <1>
                                         mov
                                               al, cl ; <- year BCD
1906 0000A894 240F
                               <1>
                                         and
                                               al, OFh
1907 0000A896 88CC
                               <1>
                                         mov
                                               ah, cl
1908 0000A898 C0EC04
                               <1>
                                         shr
                                               ah, 4
1909 0000A89B D50A
                               <1>
                                         aad
1910 0000A89D 88C1
                                               cl, al ; -> year
                               <1>
                                         mov
1911
                               <1>
1912 0000A89F 88E8
                               <1>
                                         mov
                                               al, ch
1913 0000A8A1 B464
                               <1>
                                               ah, 100
                                         mov
1914 0000A8A3 F6E4
                               <1>
                                         mul
                                               ah
```

```
1915 0000A8A5 30ED
                                <1>
                                                ch, ch
                                          xor
1916 0000A8A7 6601C8
                                <1>
                                           add
                                                 ax, cx
1917 0000A8AA 662DBC07
                                <1>
                                                ax, 1980 ; ms-dos epoch
                                           sub
1918 0000A8AE 6689C1
                                <1>
                                                 cx, ax
1919
                                <1>
1920 0000A8B1 88F0
                                <1>
                                           mov
                                                 al, dh; <- month in bcd
1921 0000A8B3 240F
                                                 al, OFh
                                <1>
                                          and
1922 0000A8B5 88F4
                                          mov
                                <1>
                                                 ah, dh
1923 0000A8B7 C0EC04
                                <1>
                                           shr
                                                 ah, 4
1924 0000A8BA D50A
                                <1>
                                           aad
1925 0000A8BC 88C6
                                <1>
                                                 dh, al ; -> month
1926
                                 <1>
1927 0000A8BE 88D0
                                <1>
                                                 al, dl ; <- day BCD
                                           mov
1928 0000A8C0 240F
                                <1>
                                           and
                                                 al, OFh
1929 0000A8C2 88D4
                                <1>
                                           mov
                                                 ah, dl
1930 0000A8C4 C0EC04
                                <1>
                                           shr
                                                 ah, 4
1931 0000A8C7 D50A
                                <1>
                                           aad
                                                 dl, al ; -> day
1932 0000A8C9 88C2
                                <1>
                                          mov
1933
                                <1>
1934 0000A8CB 88C8
                                <1>
                                                 al, cl ; count of years from 1980
                                           mov
1935 0000A8CD 66C1E004
                                                 ax, 4
                                <1>
                                           shl
                                                 al, dh; month of year, 1 to 12
1936 0000A8D1 08F0
                                 <1>
                                           or
1937 0000A8D3 66C1E005
                                <1>
                                           shl
                                                 ax, 5
1938 0000A8D7 08D0
                                <1>
                                                 al, dl; day of year, 1 to 31
1939
                                 <1>
1940 0000A8D9 6650
                                 <1>
                                          push ax ; push date
1941
                                <1>
                                                 ah, 02h ; Return Current Time
1942 0000A8DB B402
                                <1>
                                           mov
1943 0000A8DD E8F5B0FFFF
                                <1>
                                           call
                                                 int1Ah
1944
                                <1>
1945 0000A8E2 88E8
                                <1>
                                           mov
                                                 al, ch ; <- hours BCD
1946 0000A8E4 240F
                                 <1>
                                           and
                                                 al, OFh
1947 0000A8E6 88EC
                                <1>
                                           mov
                                                 ah, ch
1948 0000A8E8 C0EC04
                                <1>
                                                 ah, 4
1949 0000A8EB D50A
                                <1>
                                           aad
                                                 ch, al ; -> hours
1950 0000A8ED 88C5
                                <1>
                                           mov
1951
                                <1>
1952 0000A8EF 88C8
                                <1>
                                                 al, cl ; <- minutes BCD
                                           mov
1953 0000A8F1 240F
                                <1>
                                                 al, OFh
                                           and
1954 0000A8F3 88CC
                                <1>
                                                 ah, cl
                                           mov
1955 0000A8F5 C0EC04
                                <1>
                                           shr
                                                 ah, 4
1956 0000A8F8 D50A
                                 <1>
                                           aad
1957 0000A8FA 88C1
                                <1>
                                                 cl, al ; -> minutes
                                           mov
1958
                                <1>
                                                 al, dh ; <- seconds BCD
1959 0000A8FC 88F0
                                <1>
                                          mov
                                                 al, OFh
1960 0000A8FE 240F
                                <1>
                                           and
1961 0000A900 88F4
                                <1>
                                                 ah, dh
                                           mov
1962 0000A902 C0EC04
                                <1>
                                           shr
                                                 ah, 4
1963 0000A905 D50A
                                <1>
                                           aad
1964 0000A907 88C6
                                <1>
                                                 dh, al ; -> seconds
                                          mov
1965
                                <1>
1966 0000A909 88E8
                                <1>
                                                 al, ch ; hours
                                          mov
1967 0000A90B 66C1E006
                                <1>
                                          shl
                                                ax, 6
1968 0000A90F 08C8
                                <1>
                                                al, cl ; minutes
1969 0000A911 66C1E005
                                <1>
                                          shl
                                                ax, 5
1970 0000A915 DOEE
                                <1>
                                          shr
                                                dh, 1 ; 2 seconds
1971
                                <1>
                                          ; There is a bug in TRDOS v1 here !
1972
                                <1>
                                          ; it was 'or al, dl' !
1973 0000A917 08F0
                                 <1>
                                          or al, dh; seconds
1974
                                <1>
1975 0000A919 665A
                                 <1>
                                          pop dx ; pop date
1976
                                 <1>
1977 0000A91B C3
                                 <1>
                                           retn
1978
                                 <1>
1979
                                 <1> save_directory_buffer:
                                        ; 15/10/<del>2</del>016
1980
                                 <1>
1981
                                 <1>
                                          ; 23/03/2016
                                        ; 26/02/2016
; 22/02/2016 (TRDOS 386 = TRDOS v2.0)
; 01/08/2011
1982
                                 <1>
1983
                                 <1>
1984
                                 <1>
                                        ; 14/03/2010
1985
                                 <1>
                                        ; INPUT ->
1986
                                 <1>
1987
                                 <1>
                                                 none
                                         ; OUTPUT ->
1988
                                 <1>
                                         ; cf = 0 \rightarrow write OK...
1989
                                 <1>
                                          ; cf = 1 -> error code in AL (EAX)
1990
                                 <1>
1991
                                 <1>
                                          ; cf = 1 \& AL = 0Dh \Rightarrow CH \& CL = FS \& FAT type
1992
                                 <1>
                                         ; EBX = Directory Buffer Address
1993
                                 <1>
1994
                                 <1>
                                           ; (EAX, ECX, EDX will be modified)
1995
                                 <1>
1996 0000A91C BB00000800
                                 <1>
                                           mov
                                                 ebx, Directory_Buffer
1997 0000A921 803D[AC600100]02
                                <1>
                                           cmp
                                                byte [DirBuff_ValidData], 2
1998 0000A928 7403
                                 <1>
                                                 short loc_save_dir_buffer
                                           jе
1999 0000A92A 31C0
                                 <1>
                                           xor
                                                 eax, eax
2000 0000A92C C3
                                 <1>
                                           retn
                                <1>
2002
                                <1> loc_save_dir_buffer:
2003 0000A92D 56
                                <1>
                                          push esi
2004 0000A92E 31DB
2005 0000A930 8A3D[AA600100] <1>
2006 0000A936 80EF41 <1>
2006 0000A936 80EF41 <1>
                                           xor ebx, ebx
                                           mov bh, [DirBuff_DRV]
                                          sub bh, 'A'
                                          mov esi, Logical DOSDisks
                         add esi, ebx
2008 0000A93E 01DE
                                      mov cx, [esi+LD_FATType]
; CH = FS Type (A1h for FS)
; CL = FAT Type (0 for FS)
or cl, cl
2009 0000A940 668B4E03
2010
                               <1>
2012 0000A944 08C9
                             <1>
<1>
                                        jz
2013 0000A946 7433
                                               short loc_save_dir_buff_stc_retn
2014
                                <1>
                           2015
2016 0000A948 A1[B1600100]
2017 0000A94D 28FF
                                                bh, bh; ebx = 0
2018 0000A94F 09C0
                                <1>
                                                 eax, eax
                                           or
2019 0000A951 7540
                                <1>
                                          jnz short loc_save_sub_dir_buffer
```

```
2020 0000A953 8A25[AB600100] <1>
                                            mov ah, [DirBuff_FATType]
2021 0000A959 FEC3
                             <1>
                                             inc
                                                    bl ; bl = 1
                                                    ah, bl
2022 0000A95B 38DC
                                             cmp
2023 0000A95D 721D
                                              jb
                                                     short loc_save_dir_buff_inv_data_retn
                                  <1>
<1>
2024 0000A95F FEC3
                                             inc
                                                     bl ; bl = 2
2025 0000A961 38E3
                                              cmp
                                                    bl, ah
2026 0000A963 7217
                                  <1>
                                                     short loc_save_dir_buff_inv_data_retn
                                   <1>
2027
2028
                                   <1> loc_save_root_dir_buffer:

2029 0000A965 668B5E17
2030 0000A969 6683C30F
2031 0000A96D 66C1EB04
                                                    bx, 4 ; 16 dir entries per sector
2032 0000A971 6609DB
2033 0000A974 7405
2034
2035 0000A976 8B4664
                                                    eax, [esi+LD_ROOTBegin] ; 26/02/2016
2036 0000A979 EB23
                                  <1>
                                             jmp short loc_write_directory_to_disk
2037
                                   <1>
2038
                                   <1> loc_save_dir_buff_stc_retn:
2039 0000A97B F9
                                   <1> stc
                                   <1> loc_save_dir_buff_inv_data_retn:
2040
                                   <1> ; \( \bar{15}/10/2016 \) (0Dh -> \( 29 \)

2041
                                             mov al, 29; Invalid data!
2042 0000A97C B01D
                                   <1>
                                         mov byte [DirBuff_ValidData], 0
jmp short loc_save_dir_buff_retn
2043 0000A97E C605[AC600100]00 <1>
2044 0000A985 EB05
                                   <1>
2045
                                   <1>
2046
                                   <1> loc write directory to disk err:
2047
                                   <1> ; 15/10/2016 (disk write error code, 1Dh -> 18)
2048 0000A987 B812000000
                                  <1>
                                                    eax, 18 ; Drive not ready or write error
                                             mov
2049
                                   <1>
2050
                                   <1> loc_save_dir_buff_retn:
2051 0000A98C BB00000800
                                   <1> mov ebx, Directory_Buffer
2052 0000A991 5E
                                  <1>
                                              pop
                                                    esi
2053 0000A992 C3
                                  <1>
                                             retn
2054
                                   <1>
                         <1> loc_save_sub_dir_buffer:
<1> ; ebx = 0
<1> sub eax, 2
<1> mov bl, [esi+LD_BPB+SecPerClust
<1> mul ebx
<1> add eax, [esi+LD_DATABegin]
2055
2057 0000A993 83E802
2058 0000A996 8A5E13
                                                    bl, [esi+LD_BPB+SecPerClust]
2059 0000A999 F7E3
2060 0000A99B 034668
                                            ;mov ecx, ebx
2061
                                   <1>
2062
                                   <1>
                           vic vice write directory to disk:
<1> mov ecx, ebx
<1> mov ebx, Directory Buffer
<1> call disk write
2063
2064 0000A99E 89D9
2065 0000A9A0 BB00000800
2066 0000A9A5 E8A34E0000
2067 0000A9AA 72DB
                                   <1>
                                            jc short loc_write_directory_to_disk_err
2068
                                   <1>
                                  <1> loc_save_dir_buff_validate_retn:
2070 0000A9AC C605[AC600100]01 <1> mov byte [DirBuff_ValidData], 1
2071 0000A9B3 31C0
                                   <1>
                                             xor
                                                    eax, eax
                                              ; 26/02/2016
2072
                                   <1>
2073 0000A9B5 EBD5
                                   <1>
                                                    short loc_save_dir_buff_retn
                                            jmp
2074
                                   <1>
2075
                                   <1> update_parent_dir_lmdt:
                                    <1> ; 29/12/2017
2076
                                            ; 22/02/2016 (TRDOS 386 = TRDOS v2.0)
2077
                                    <1>
                                             ; 01/08/2011
2078
                                    <1>
                                           ; 16/10/2010
2079
                                   <1>
2080
                                   <1>
                                           ; INPUT -> ; none
2081
                                    <1>
2082
                                    <1>
                                            ; OUTPUT ->
2083
                                    <1>
                                           ; (last modification date & time of the parent dir
2084
                                    <1>
2085
                                    <1>
                                                     will be changed/updated)
2086
                                    <1>
2087
                                            ; (EAX, EBX, ECX, EDX, EDI will be changed)
                                   <1>
2088
                                   <1>
2089 0000A9B7 29C0
                                   <1>
                                              sub eax, eax
                                              mov ah, [Current Dir Level]
2090 0000A9B9 8A25[84590100]
                                   <1>
2091 0000A9BF A0[85590100]
                                   <1>
                                              mov
                                                     al, [Current_FATType]
2092 0000A9C4 3C01
                                                    al, 1
                                   <1>
                                              cmp
2093 0000A9C6 723A
                                   <1>
                                                    short loc_UPDLMDT_proc_retn
2094
                                   <1>
                                   <1> loc_update_parent_dir_lm_date_time:
2095
2096 0000A9C8 08E4
                                  <1> or ah, ah
2097 0000A9CA 7436
                                   <1>
                                                     short loc_UPDLMDT_proc_retn
                                              jz
2098
                                   <1>
2099 0000A9CC 56
                                   <1>
                                              push esi; *
2100 0000A9CD 8825[24630100]
                                <1>
                                              mov [UPDLMDT CDirLevel], ah
2101 0000A9D3 8B15[80590100]
                                   <1>
                                              mov
                                                     edx, [Current_Dir_FCluster]
2102 0000A9D9 8915[25630100]
                                                     [UPDLMDT_CDirFCluster], edx
                                   <1>
                                              mov
2103
                                    <1>
2104 0000A9DF FECC
                                    <1>
                                              dec
                                                     ah
2105 0000A9E1 B90C000000
                                    <1>
                                              mov
                                                     ecx, 12
2106 0000A9E6 BE[E3600100]
                                   <1>
                                                      esi, PATH_Array
                                              mov
2107
                                   <1>
2108 0000A9EB 8825[84590100]
                                   <1>
                                                     [Current_Dir_Level], ah
2109 0000A9F1 08E4
                                    <1>
                                             or
                                                     ah, ah
2110 0000A9F3 750E
                                                     short loc_update_parent_dir_lmdt_load_sub_dir_1
                                   <1>
                                            jnz
2111 0000A9F5 803D[85590100]02
                                   <1>
                                              cmp
                                                     byte [Current FATType], 2
2112 0000A9FC 770B
                                   <1>
                                                     short loc_update_parent_dir_lmdt_load_sub_dir_2
                                              jа
                                                     al, al; eax = 0
2113 0000A9FE 28C0
                                   <1>
2114 0000AA00 EB0A
                                   <1>
                                                    short loc_update_parent_dir_lmdt_load_sub_dir_3
2115
                                   <1>
                                   <1> loc UPDLMDT proc retn:
2116
2117 0000AA02 C3
                                   <1>
                                              retn
2118
                                   <1>
2119
                                   <1> loc_update_parent_dir_lmdt_load_sub_dir_1:
2120 0000AA03 B010
                                   <1>
                                              mov al, 16
2121 0000AA05 F6E4
                                   <1>
                                              mul
                                                     ah
2122 0000AA07 01C6
                                   <1>
                                              add
                                                    esi, eax
2123
                                    <1>
2124
                                    <1> loc_update_parent_dir_lmdt_load_sub_dir_2:
```

```
2125 0000AA09 8B460C
                                <1>
                                          mov eax, [esi+12] ; Parent Dir First Cluster
2126
                                <1>
2127
                                <1> loc_update_parent_dir_lmdt_load_sub_dir_3:
2128 0000AA0C A3[80590100]
                                <1>
                                         mov [Current_Dir_FCluster], eax
                                <1>
2129
2130 0000AA11 83C610
                                <1>
                                          add
                                                esi, 16
                                                di, Dir_File_Name
2131 0000AA14 66BF[0A62]
                               <1>
                                          mov
2132 0000AA18 F3A4
                                <1>
                                               movsb
                                         rep
2133
                                <1>
                                     mov esi, Logi
sub ebx, ebx
                                                esi, Logical_DOSDisks
2134 0000AA1A BE00010900
                               <1>
2135 0000AA1F 29DB
                               <1>
2136 0000AA21 8A3D[86590100]
                                <1>
                                         mov
                                                bh, [Current_Drv]
                                          add esi, ebx
2137 0000AA27 01DE
                               <1>
2138 0000AA29 E88FF7FFF
                                          call reload current directory
                               <1>
2139 0000AA2E 7230
                                <1>
                                        jc short loc_update_parent_dir_lmdt_restore_cdirlevel
2140
                                <1>
                               <1> loc_update_parent_dir_lmdt_locate_dir:
2141
                                         mov esi, Dir_File_Name xor cx, cx
2142 0000AA30 BE[0A620100]
                               <1>
2143 0000AA35 6631C9
                                <1>
                                          mov ax, 0810h; Only directories
2144 0000AA38 66B81008
                               <1>
2145 0000AA3C E8B5F6FFFF
                                         call locate_current_dir_file
                               <1>
                                          ; EDI = DirBuff Directory Entry Address
                                <1>
2147 0000AA41 721D
                               <1>
                                         jc short loc_update_parent_dir_lmdt_restore_cdirlevel
2148
                                <1>
2149 0000AA43 E836FEFFFF
                                <1>
                                         call convert_current_date_time
                                          mov
                                                 [edi+18], dx ; Last Access Date
2150 0000AA48 66895712
                                <1>
2151 0000AA4C 66895718
                                <1>
                                         mov [edi+24], dx ; Last Write Date
                                        mov [edi+22], ax ; Last Write Time
2152 0000AA50 66894716
                                <1>
2153
                                <1>
                                     mov byte [DirBuff_ValidData], 2
call save_directory_buffer
2154 0000AA54 C605[AC600100]02 <1>
2155 0000AA5B E8BCFEFFFF
                                <1>
2156
                                <1>
                                         ; 29/12/2017
2157
                                <1>
                                          ;jc short loc_update_parent_dir_lmdt_restore_cdirlevel
2158
                                <1>
                                         ;xor al, al
2159
                                <1> loc_update_parent_dir_lmdt_restore_cdirlevel:
2160
                                <1>
                                         ; current directory level restoration
2161 0000AA60 8A25[24630100]
                                         mov ah, [UPDLMDT CDirLevel]
                                <1>
                                      mov [Current_Dir_Level], ah
2162 0000AA66 8825[84590100]
                                <1>
                                          mov edx, [UPDLMDT CDirFCluster]
2163 0000AA6C 8B15[25630100]
                                <1>
2164 0000AA72 8915[80590100]
                                         mov [Current_Dir_FCluster], edx
                                <1>
2165
                                <1>
2166 0000AA78 5E
                                          pop esi; *
                                <1>
2167 0000AA79 C3
                                <1>
                                          retn
2168
                                <1>
                                <1> delete_longname:
2169
                                      ; 27/02/2016 (TRDOS 386 = TRDOS v2.0)
2170
                                <1>
2171
                                <1>
                                          ; 01/08/2011 (DIR.ASM, 'proc_delete_longname')
                                        ; 14/03/2010
2172
                                <1>
                                        ; INPUT ->
; EAX = Directory Entry (Index) Number (< 65536)
2173
                                <1>
2174
                                <1>
                                        ; OUTPUT ->
2175
                                 <1>
                                        ; cf = 0 \rightarrow OK (EAX = 0)
2176
                                 <1>
                                                cf = 1 -> error code in EAX (AL)
2177
                                <1>
2178
                                <1>
2179
                                 <1>
                                        ; (Modified registers: EAX, EDX, ECX, EBX, EDI)
2180
                                <1>
2181 0000AA7A 66A3[54630100]
                                <1>
                                        mov [DLN EntryNumber], ax
2182 0000AA80 C605[56630100]40
                                                 byte [DLN_40h], 40h
                                <1>
                                         mov
2183
                                <1>
2184 0000AA87 E858000000
                                <1>
                                          call locate_current_dir_entry
2185 0000AA8C 7308
                                <1>
                                          jnc short loc_dln_check_attributes
2186 0000AA8E C3
                                <1>
2187
                                <1>
2188
                                <1> loc_dln_longname_not_found:
                                      mov
2189 0000AA8F B802000000
                                <1>
                                               eax, 2
2190 0000AA94 F9
                                <1>
                                          stc
2191 0000AA95 C3
                                <1>
                                          retn
2192
                                <1>
2193
                                <1> loc_dln_check_attributes:
2194 0000AA96 B00F
                               <1> mov al, OFh ; long name
                               <1>
2195 0000AA98 8A670B
                                                ah, [edi+0Bh] ; dir entry attributes
2196 0000AA9B 38C4
                                <1>
                                          cmp
                                                ah, al
2197 0000AA9D 75F0
                               <1>
                                                short loc_dln_longname_not_found
                                          jne
2198 0000AA9F 8A27
                               <1>
                                                ah, [edi]
2199 0000AAA1 2A25[56630100]
2200 0000AAA7 76E6
                               <1>
                                         sub
                                                ah, [DLN 40h]
                                                short loc_dln_longname_not_found
                                <1>
                                          jna
2201 0000AAA9 80FC14
                                <1>
                                                ah, 14h; 84-64=20 \rightarrow 20*13=260 bytes
                                         cmp
2202 0000AAAC 77E1
                                                short loc dln longname not found
                                <1>
                                        ja
2203
                                <1>
2204 0000AAAE C607E5
                                                byte [edi], OE5h ; deleted sign
                                <1>
                                          mov
2205 0000AAB1 C605[AC600100]02
                               <1>
                                          mov
                                                byte [DirBuff_ValidData], 2 ; changed/write sign
2206 0000AAB8 C605[56630100]00
                                                byte [DLN_40h], 0 ; 40h -> 0
                                 <1>
                                          mov
2207
                                 <1>
2208
                                 <1> loc_dln_delete_next_ln_entry:
2209 0000AABF 80FC01
                                         cmp ah, 1
                                 <1>
                                                short loc_dln_longname_retn
2210 0000AAC2 7616
                                 <1>
                                          jna
                                 <1> loc dln delete next ln entry 0:
2212 0000AAC4 66FF05[54630100]
                                         inc word [DLN_EntryNumber]
                                 <1>
2213 0000AACB 0FB705[54630100]
                                 <1>
                                          movzx eax, word [DLN_EntryNumber]
2214 0000AAD2 E80D000000
                                          call locate current dir entry
                                 <1>
2215 0000AAD7 73BD
                                 <1>
                                          jnc short loc_dln_check_attributes
2216
                                 <1>
                                 <1> loc_dln_longname_stc_retn:
2217
2218 0000AAD9 C3
                                 <1>
2219
                                 <1>
2220
                                 <1> loc_dln_longname_retn:
2221
                                 <1>
                                         ;cmp byte [DirBuff ValidData], 2
                                          ;jne short loc_dln_longname_retn_xor_eax
2222
                                <1>
2223 0000AADA E83DFEFFFF
                                 <1>
                                          call save_directory_buffer
2224 0000AADF 72F8
                                          jс
                                                short loc_dln_longname_stc_retn
                                <1>
2225
                                <1>
2226
                                <1> loc_dln_longname_retn_xor_eax:
2227 0000AAE1 31C0
                                <1>
                                          xor
                                                eax, eax
2228 0000AAE3 C3
                                 <1>
                                 <1>
2229
```

```
2230
                                               <1> locate_current_dir_entry:
                                                       ; 16/10/2016
2231
2232
                                                            ; 15/10/2016
                                               <1>
2233
                                               <1>
                                                           ; 23/03/2016
                                                          ; 27/02/2016 (TRDOS 386 = TRDOS v2.0)
2234
                                               <1>
2235
                                               <1>
                                                            ; 01/08/2011 (DIR.ASM, 'proc_locate_current_dir_entry')
2236
                                                            ; 07/03/2010
                                                          ; INPUT ->
                                               <1>
2237
                                                          ;
2238
                                               <1>
                                                                     EAX = Directory Entry (Index) Number (< 65536)
                                                            ; OUTPUT ->
2239
                                               <1>
2240
                                               <1>
                                                                     EDI = Directory Entry Address
2241
                                               <1>
                                                                      EAX = Cluster Number of Directory Buffer
2242
                                               <1>
                                                                     EBX = Directory Buffer Entry Offset
2243
                                               <1>
                                                                    ECX = DirBuff Valid Data identifier (CL)
2244
                                               <1>
                                                                     If CF = 0 and CL = 2 then
                                                                     directory buffer modified and must be written to disk.
2245
                                               <1>
2246
                                               <1>
                                                                     If CF = 0 and CL = 1 then
2247
2248
                                               <1>
                                                                         dir buffer has been written to disk, already.
2249
                                               <1>
                                                                      CF = 1 -> Error code in EAX (AL)
                                                             ;
2250
                                               <1>
2251
                                               <1>
                                                            ; (Modified registers: EAX, EDX, ECX, EBX, EDI)
2252
                                               <1>
                                               <1> loc_locate_current_dir_entry:
                                                         push esi
2254 0000AAE4 56
                                              <1>
| Coa, eax | mov edx, 32 | cl> mul edx | mov edx, 32 | cl> mul edx | mov ebx, ebx | cl> mov | [LCDE_ByteOffset] | cl> mov | [LCDE_ByteOffset] | cl> mov ebx, ebx | cl
                                                                     [LCDE_ByteOffset], eax
                                                            mov al, [DirBuff DRV]
                                                            mov esi, Logical_DOSDisks
<1> loc_lcde_cdl_check:
2267
                                             <1> cmp byte [Current_Dir_Level], 0
<1> ja short loc_lcde_calc_dirbuff_cluster_offset
2268 0000AB11 803D[84590100]00
2269 0000AB18 772A
                                                        ; 27/02/2016
2270
                                              <1>
                                                         ; TRDOS v1 has bug here for FAT32 fs !
2271
                                              <1>
2272
                                              <1>
                                                            ; (Root Directory Entries for FAT32 = 0)
                                                         cmp byte [esi+LD_FATType], 3 ; FAT32
2273 0000AB1A 807E0303
                                              <1>
2274 0000AB1E 7324
                                              <1>
                                                          jnb short loc_lcde_calc_dirbuff_cluster_offset
2275
                                              <1>
2276
                                              <1> loc_lcde_cdl_check_FAT12_16:
                                          <1>
2277 0000AB20 668B4617
                                                      mov ax, [esi+LD_BPB+RootDirEnts]
2278 0000AB24 6648
                                              <1>
                                                             dec
                                                                      ax
                                             <1>
                                                            ;xor dx, dx
                                 <1>
                                                             cmp ax, cx; cx = Directory Entry (Index) Number
2280 0000AB26 6639C8
2281 0000AB29 720E
                                              <1>
                                                            jb
                                                                     short loc_lcde_stc_12h_retn
2282 0000AB2B 66890D[58630100] <1>
                                                            mov
                                                                     [LCDE_EntryIndex], cx
                                                          xor eax, eax
2283 0000AB32 31C0
                                              <1>
                                                            jmp
2284 0000AB34 E993000000
                                              <1>
                                                                          loc_lcde_check_dir_buffer_cluster
2285
                                              <1>
                                            <1> loc_lcde_stc_12h_retn:
                                        <1>
<1>
2287 0000AB39 5E
                                                       pop esi
2288 0000AB3A 89CB
                                                            mov ebx, ecx mov ecx, edx
                                             <1>
2289 0000AB3C 89D1
                                                        ; 16/10/2016 (12h -> 12)
2290
                                              <1>
2291 0000AB3E B80C000000
                                              <1>
                                                            mov eax, 12; No more files
2292 0000AB43 C3
                                              <1>
                                                            retn
2293
                                              <1>
2294
                                              <1> loc_lcde_calc_dirbuff_cluster_offset:
                                              <1> mov bl, [esi+LD_BPB+SecPerClust]
2295 0000AB44 8A5E13
                                        2296 0000AB47 30FF
                                                            xor
                                                                     bh, bh
                                          <1> mov
<1> mul
<1> or
2297 0000AB49 668B4611
                                                                     ax, [esi+LD_BPB+BytesPerSec]
2298 0000AB4D 66F7E3
                                             2299 0000AB50 6609D2
                                                                     dx, dx ; If bytes per cluster > 32KB it is invalid
2300 0000AB53 755D
2301
2302 0000AB55 6689C1
                                                                     cx, ax ; BYTES PER CLUSTER
                                              <1>
                                                            mov
2303 0000AB58 A1[60630100]
                                              <1>
                                                                     eax, [LCDE_ByteOffset]
2304
                                               <1>
                                                            ; sub edx, edx
2305 0000AB5D F7F1
                                              <1>
                                                            div
                                                                      ecx
2306 0000AB5F 3DFFFF0000
                                              <1>
                                                         cmp
                                                                     eax, 65535
2307 0000AB64 774C
                                              <1>
                                                                     short loc_lcde_invalid_format
2308
                                               <1>
2309
                                               <1>
                                                            ; cluster sequence number of directory (< 65536)
2310 0000AB66 66A3[5A630100]
                                              <1>
                                                             mov [LCDE ClusterSN], ax
                                               <1>
2312 0000AB6C 6689D0
                                              <1>
                                                                      ax, dx ; byte offset in cluster (directory buffer)
2313 0000AB6F 66BB2000
                                               <1>
                                                                     bx, 32; 1 dir entry = 32 bytes
2314 0000AB73 6629D2
                                              <1>
                                                             sub dx, dx; 0
2315 0000AB76 66F7F3
                                              <1>
                                                             div bx
                                                                    [LCDE EntryIndex], ax; dir entry index/sequence number
2316 0000AB79 66A3[58630100]
                                              <1>
                                                            mov
2317
                                              <1>
                                                                                                ; (in directory buffer/cluster)
2318
                                               <1> loc_lcde_get_current_sub_dir_fcluster:
2319 0000AB7F A1[80590100]
                                              <1>
                                                            mov eax, [Current_Dir_FCluster]
2320
                                               <1>
2321
                                               <1> loc lcde get next cluster:
2322 0000AB84 66833D[5A630100]00 <1>
                                                            cmp word [LCDE_ClusterSN], 0
2323 0000AB8C 763E
                                               <1>
                                                                      short loc_lcde_check_dir_buffer_cluster
                                                            mov [LCDE_Cluster], eax call get_next_cluster
2324 0000AB8E A3[5C630100]
                                               <1>
                                                           mov
2325 0000AB93 E834100000
                                              <1>
                                              2326 0000AB98 7220
2327 0000AB9A 66FF0D[5A630100] <1>
2328 0000ABA1 EBE1
                                              <1>
                                                          jmp
                                                                    short loc_lcde_get_next_cluster
2329
                                              <1>
                                              <1> loc_lcde_reload_current_directory:
2330
2331 0000ABA3 51
                                              <1> push ecx
                                                             call reload_current_directory
                                             <1>
2332 0000ABA4 E814F6FFFF
2333 0000ABA9 59
                                              <1>
                                                             pop ecx
                                                            jnc loc_lcde_cdl_check
2334 0000ABAA 0F8361FFFFFF
                                              <1>
```

```
2335 0000ABB0 5E
                               <1>
                                         pop
                                                esi
2336 0000ABB1 C3
                                <1>
                                         retn
2337
                                <1>
2338
                                <1> loc_lcde_invalid_format:
                                       ; \frac{1}{15}/10/20\overline{16} (0Bh -> 28)
2339
                                <1>
2340 0000ABB2 B81C000000
                                <1>
                                         mov eax, 28; Invalid Format!
                               <1> loc lcde_drive_not_ready_read_err:
2342 0000ABB7 F9
                                      stc
                               <1>
2343 0000ABB8 5E
                               <1>
                                         pop
                                                esi
2344 0000ABB9 C3
                               <1>
                                         retn
2345
                                <1>
2346
                                <1> loc_lcde_check_gnc_error:
2347 0000ABBA 09C0
                               <1> or eax, eax
2348 0000ABBC 75F9
                               <1>
                                               short loc_lcde_drive_not_ready_read_err
                                     dec word [LCDE_cluster]
jnz short loc_lcde_inval
mov eax, [LCDE_Cluster]
                                         dec word [LCDE_ClusterSN]
jnz short loc_lcde_invalid_format
2349 0000ABBE 66FF0D[5A630100] <1>
2350 0000ABC5 75EB
                                <1>
2351 0000ABC7 A1[5C630100]
                               <1>
2352
                                <1>
2353
                                <1> loc_lcde_check_dir_buffer_cluster:
                               <1> cmp eax, [DirBuff_Cluster]
2354 0000ABCC 3B05[B1600100]
2355 0000ABD2 755C
                                <1>
                                               short loc_lcde_load_dir_cluster
2361 <1>
2362 0000ABE6 807E0303 <1>
2363 0000ABEA 7359 <1>
                                       jnb
                                               short loc_lcde_load_dir_cluster_0
                               <1>
2364
2365 0000ABEC 0FB74E17
                               <1>
                                      movzx ecx, word [esi+LD_BPB+RootDirEnts]
                                         add cx, 15; round up (16 entries per sector) shr cx, 4; 1 sector contains 16 dir entries
2366 0000ABF0 6683C10F
                                <1>
2367 0000ABF4 66C1E904
                               <1>
2368
                               <1>
                                                   eax, [esi+LD_ROOTBegin]
2369 0000ABF8 8B4664
                                <1>
                                          mov
2370 0000ABFB EB54
                                <1>
                                          jmp short loc_lcde_load_dir_cluster_1
                                <1>
2372
                                <1> loc_lcde_validate_dirBuff:
                                         mov byte [DirBuff ValidData], 1
2373 0000ABFD C605[AC600100]01
                               <1>
2374
                                <1>
2375
                                <1> lcde_check_dir_buffer_cluster_next:
                                         movzx ebx, word [LCDE EntryIndex]
2376 0000AC04 0FB71D[58630100]
                                <1>
2377 0000AC0B 663B1D[AF600100]
                                          cmp bx, [DirBuff_LastEntry]
                               <1>
                                                short loc lcde invalid format
2378 0000AC12 779E
                                <1>
                                          jа
2379 0000AC14 B820000000
                                <1>
                                         mov
                                               eax, 32
2380 0000AC19 F7E3
                                <1>
                                         mul
                                               ebx
                                         ; or edx, edx
                               <1>
2382
                               <1>
                                       ;jnz short loc_lcde_invalid_format
2383
                                <1>
2384 0000AC1B BF00000800
                               <1>
                                                edi, Directory_Buffer
                                         mov
2385 0000AC20 01C7
                               <1>
                                        add edi, eax ; add entry offset to buffer address
2386
                                <1>
                               <1> loc_lcde_dir_buffer_last_check:
2387
2388 0000AC22 A1[B1600100]
                               <1>
                                     mov eax, [DirBuff_Cluster]
2389 0000AC27 0FB60D[AC600100]
                              <1>
                                         movzx ecx, byte [DirBuff_ValidData]
2390
                                <1>
                                <1> loc lcde retn:
2392 0000AC2E 5E
                                <1>
                                       pop esi
2393 0000AC2F C3
                                <1>
                                          retn
2394
                                <1>
2395
                                <1> loc_lcde_load_dir_cluster:
2396
                                <1> ;cmp byte [DirBuff_ValidData], 2
2397
                               <1>
                                         ;jne short loc_lcde_load_dir_cluster_n2
2398 0000AC30 50
                               <1>
                                         call save_directory_buffer
2399 0000AC31 E8E6FCFFFF
                               <1>
                                        pop eax
jc short loc_lcde_retn
2400 0000AC36 58
                                <1>
2401 0000AC37 72F5
                                <1>
2402
                                <1>
2403
                                <1> loc_lcde_load_dir_cluster_n2:
2404 0000AC39 C605[AC600100]00 <1> mov byte [DirBuff ValidData], 0
2405 0000AC40 A3[B1600100]
                               <1>
                                         mov [DirBuff_Cluster], eax
2406
                                <1>
                               <1> loc_lcde_load_dir_cluster_0:
2407
2408 0000AC45 83E802
                             <1> sub eax, 2
                                         movzx ecx, byte [esi+LD_BPB+SecPerClust]
2409 0000AC48 0FB64E13
                               <1>
2410 0000AC4C F7E1
                               <1>
                                         mul ecx
                                         add eax, [esi+LD_DATABegin]
2411 0000AC4E 034668
                               <1>
2412
                                <1>
2413
                                <1> loc_lcde_load_dir_cluster_1:
                                <1> mov ebx, Directory_Buffer
2414 0000AC51 BB00000800
2415
                                <1>
                                          ; ecx = sector count
2416 0000AC56 E8014C0000
                                <1>
                                          call disk_read
2417 0000AC5B 73A0
                                <1>
                                               short loc_lcde_validate_dirBuff
                                <1>
2419
                                <1>
                                         : 15/10/2016
2420
                                          ; (Disk read error instead of drv not ready err)
                                <1>
2421 0000AC5D B811000000
                                <1>
                                          mov eax, 17; Drive not ready or read error!
2422 0000AC62 EBCA
                                <1>
                                          jmp short loc_lcde_retn
2423
                                <1>
2424
                                <1>
2425
                                <1> remove_file:
2426
                                <1>
                                       ; 15/10/2016
                                          ; 28/02/2016 (TRDOS 386 = TRDOS v2.0)
2427
                                <1>
                                        ; 10/04/2011 (FILE.ASM, 'proc delete file')
2428
                                <1>
                                        ; 09/08/2010
2429
                                <1>
2430
                                <1>
                                         ; INPUT ->
                                                EDI = Directory Buffer Entry Address
2431
                                <1>
                                <1>
                                                CX = Directory Buffer Entry Counter/Index
2432
                                                 BL = Longname Entry Length
2433
                                <1>
2434
                                <1>
                                                 BH = Logical DOS Drive Number
                                         ;
2435
                                <1>
2436 0000AC64 29C0
                                <1>
                                         sub
                                                eax, eax
2437 0000AC66 88FC
                                <1>
                                         mov
                                                ah, bh
2438 0000AC68 BE00010900
                                <1>
                                                esi, Logical_DOSDisks
2439 0000AC6D 01C6
                                <1>
                                                esi, eax
                                         add
```

```
2440
                                 <1>
2441 0000AC6F 807E0301
                                 <1>
                                           cmp
                                                 byte [esi+LD FATType], 1
2442 0000AC73 7312
                                 <1>
                                                 short loc_del_fat_file
                                           jnb
2443
                                 <1>
                                                 byte [esi+LD_FSType], 0A1h
2444 0000AC75 807E04A1
                                 <1>
                                           cmp
2445 0000AC79 7406
                                 <1>
                                           jе
                                                  short loc_del_fs_file
                                 <1>
                                 <1> loc_del_file_invalid_format:
2447
                                        xor ah, ah
2448 0000AC7B 30E4
                                 <1>
                                           ; 15/10/2016 (OBh -> 28)
2449
                                 <1>
2450 0000AC7D B01C
                                 <1>
                                           mov
                                                 al, 28 ; Invalid Format
2451 0000AC7F F9
                                 <1>
2452 0000AC80 C3
                                 <1>
                                           retn
2453
                                 <1>
                                 <1> loc_del_fs_file:
2454
2455 0000AC81 E83F0F0000
                                 <1>
                                          call delete_fs_file
2456 0000AC86 C3
                                 <1>
                                           retn
2457
                                 <1>
2458
                                 <1> loc_del_fat_file:
2459 0000AC87 E808000000
                                 <1>
                                         call delete directory entry
2460 0000AC8C 7205
                                 <1>
                                               short loc_del_file_err_retn
2461
                                 <1>
2462
                                 <1> loc_delfile_unlink_cluster_chain:
2463 0000AC8E E863170000
                                 <1>
                                         call truncate_cluster_chain
2464
                                 <1>
                                           ;jc short loc_del_file_err_retn
2465
                                 <1>
2466
                                 <1> loc delfile return:
2467
                                 <1> loc_del_file_err_retn:
2468 0000AC93 C3
                                 <1>
                                           retn
2469
                                 <1>
2470
                                 <1> delete_directory_entry:
                                       ; 15/10/20<del>1</del>6
2471
                                 <1>
                                           ; 28/02/2016 (TRDOS 386 = TRDOS v2.0)
2472
                                 <1>
2473
                                 <1>
                                          ; 01/08/2011 (DIR.ASM, 'proc_delete_directory_entry')
                                         ; 10/04/2011
2474
                                 <1>
2475
                                 <1>
                                           ; INPUT ->
                                                 ESI = Logical Dos Drive Descripton Table Address
2476
                                           ;
2477
                                 <1>
                                                  EDI = Directory Buffer Entry Address
2478
                                 <1>
                                                  CX = Directory Buffer Entry Counter/Index
2479
                                 <1>
                                                  BL = Longname Entry Length
                                           ;
                                           ; OUTPUT ->
2480
                                 <1>
                                                 ESI = Logical dos drive descripton table address
2481
                                 <1>
                                                  EAX = First cluster to be truncated/unlinked
2482
                                 <1>
                                           ;
                                                  CF = 1 -> Error code in EAX (AL)
2483
                                 <1>
2484
                                 <1>
                                                   CF = 0 \& BH <> 0 -> LMDT write error (BH = 1)
                                                   CF = 0 \& BL <> 0 -> Long name delete error (BL = FFh)
2485
                                 <1>
2486
                                 <1>
                                 <1>
                                           ; (EDI, EBX, ECX register contents will be changed)
2487
2488
                                 <1>
2489 0000AC94 881D[EE620100]
                                 <1>
                                                  [DelFile_LNEL], bl
                                           mov
2490 0000AC9A 66890D[EC620100]
                                 <1>
                                                  [DelFile_EntryCounter], cx
                                 <1>
2492 0000ACA1 668B4714
                                 <1>
                                           mov
                                                  ax, [edi+20] ; First Cluster High Word
2493 0000ACA5 C1E010
                                 <1>
                                           shl
2494 0000ACA8 668B471A
                                 <1>
                                                  ax, [edi+26] ; First Cluster Low Word
                                           mov
2495
                                 <1>
2496 0000ACAC A3[E8620100]
                                 <1>
                                                 [DelFile_FCluster], eax
                                           mov
2497
                                 <1>
2498
                                 <1> loc_del_short_name:
2499 0000ACB1 C607E5
                                         mov byte [edi], OE5h ; Deleted sign
                                 <1>
2500
                                 <1>
2501 0000ACB4 C605[AC600100]02
                                 <1>
                                           mov
                                                 byte [DirBuff ValidData], 2
                                           call save_directory_buffer
2502 0000ACBB E85CFCFFFF
                                 <1>
2503 0000ACC0 723D
                                 <1>
                                                  short loc_delete_direntry_err_return
2504
                                 <1>
2505
                                 <1> loc_del_long_name:
2506 0000ACC2 0FB615[EE620100]
                                 <1>
                                       movzx edx, byte [DelFile_LNEL]
2507 0000ACC9 08D2
                                           or
                                                 dl, dl
                                 <1>
2508 0000ACCB 7416
                                 <1>
                                                  short loc_del_dir_entry_update_parent_dir_lm_date
                                           jz
2509
                                 <1>
2510 0000ACCD 8835[EE620100]
                                 <1>
                                           mov
                                                 byte [DelFile_LNEL], dh ; 0
2511
                                 <1>
2512 0000ACD3 0FB705[EC620100]
                                           movzx eax, word [DelFile_EntryCounter]
                                 <1>
2513 0000ACDA 29D0
                                 <1>
                                           sub eax, edx
                                           ;jnc short loc_del_long_name_continue
2514
                                 <1>
2515 0000ACDC 7205
                                 <1>
                                                  short loc_del_dir_entry_update_parent_dir_lm_date
2516
                                 <1>
                                 <1> ;loc_del_direntry_inv_data_return: ; 15/10/2016 (ODh -> 29)
2517
                                          mov eax, 29; ODh (TRDOS 8086); Invalid data
2518
                                  <1>;
2519
                                 <1>;
                                           retn
2520
                                 <1>
2521
                                  <1> loc_del_long_name_continue:
                                 <1>
                                           ; AX = Directory Entry Number of the long name last entry
2523 0000ACDE E897FDFFFF
                                 <1>
                                           call delete_longname
                                 <1>
                                           ;jc short loc delete direntry err return
2524
2525
                                 <1>
2526
                                 <1> loc_del_dir_entry_update_parent_dir_lm_date:
2527 0000ACE3 801D[EE620100]00
                                           sbb byte [DelFile_LNEL], 0 ; 0FFh if cf = 1
                                 <1>
2528
                                 <1>
2529 0000ACEA E8C8FCFFFF
                                 <1>
                                           call update_parent_dir_lmdt
2530 0000ACEF B700
                                 <1>
                                           mov
                                                 bh, 0
2531 0000ACF1 80D700
                                 <1>
                                           adc
                                                 bh, 0
                                 <1>
2533 0000ACF4 8A1D[EE620100]
                                 <1>
                                                 bl, byte [DelFile_LNEL]
2534
                                 <1>
2535
                                 <1> loc_delete_direntry_return:
2536 0000ACFA A1[E8620100]
                                         mov eax, [DelFile FCluster]
                                 <1>
                                 <1> loc_delete_direntry_err_return:
2537
2538 0000ACFF C3
                                 <1>
2539
                                 <1>
                                 <1> rename directory entry:
2540
                                       ; 13/11/20<del>1</del>7
2541
                                 <1>
2542
                                 <1>
                                           ; 15/10/2016
2543
                                 <1>
                                           ; 06/03/2016 (TRDOS 386 = TRDOS v2.0)
                                 <1>
2544
                                           ; 01/08/2011 (DIR.ASM, 'proc_rename_directory_entry')
```

```
; 19/11/2010
2545
                                  <1>
2546
                                  <1>
                                           ; INPUT -> (Current Directory)
                                                  CX = Directory Entry Number
2547
                                  <1>
2548
                                  <1>
                                                   EAX = First Cluster number of file or directory
2549
                                  <1>
                                                   EBX = Longname Length (dir entry count) (< 256)
                                                   ESI = New file (or directory) name (no path).
2550
                                  <1>
                                                        (ASCIIZ string)
2551
                                  <1>
                                           ;
                                           ; OUTPUT ->
2552
                                  <1>
2553
                                  <1>
                                                  CF = 0 \rightarrow successfull
                                                   CF = 1 -> error code in EAX (AL)
2554
                                  <1>
                                            ;
2555
                                  <1>
2556
                                  <1>
                                            ; (EAX, EBX, ECX, EDX, ESI, EDI will be changed)
2557
                                  <1>
2558 0000AD00 803D[85590100]00
                                  <1>
                                            cmp byte [Current_FATType], 0
2559 0000AD07 7706
                                  <1>
                                                  short loc_rename_directory_entry
2560
                                  <1>
2561 0000AD09 E8B80E0000
                                  <1>
                                            call rename_fs_file_or_directory
2562 0000AD0E C3
                                  <1>
                                            retn
2563
                                  <1>
                                  <1> loc_rename_directory_entry:
2564
2565 0000AD0F 881D[EE620100]
                                 <1>
                                           mov [DelFile_LNEL], bl
2566 0000AD15 66890D[EC620100]
                                                   [DelFile EntryCounter], cx
                                  <1>
                                            mov
2567 0000AD1C A3[E8620100]
                                                  [DelFile_FCluster], eax
                                  <1>
                                            mov
                                  <1>
2569 0000AD21 0FB7C1
                                  <1>
                                           movzx eax, cx
2570 0000AD24 E8BBFDFFFF
                                  <1>
                                            call locate_current_dir_entry
2571 0000AD29 7308
                                  <1>
                                            jnc short loc rename direntry check fcluster
2572
                                  <1>
                                  <1> loc rename_direntry_pop_retn:
2573
2574 0000AD2B C3
                                  <1>
                                           retn
2575
                                  <1>
2576
                                  <1> loc_rename_direntry_pop_invd_retn:
2577 0000AD2C F9
                                  <1>
                                           stc
2578
                                  <1> loc rename direntry invd retn:
2579
                                  <1>
                                           ; 15/10/2016 (ODh -> 29)
2580 0000AD2D B81D000000
                                 <1>
                                            mov eax, 29; Invalid data
                                  <1> loc rename retn:
2582 0000AD32 C3
                                  <1>
                                           retn
2583
                                  <1>
                                  <1> loc_rename_direntry_check_fcluster:
2584
                                       mov dx, [edi+20]; First Cluster HW
2585 0000AD33 668B5714
                                 <1>
2586 0000AD37 C1E210
                                                  edx, 16 ; 13/11/2017
                                  <1>
                                            shl
                                                  dx, [edi+26] ; First Cluster LW
2587 0000AD3A 668B571A
                                  <1>
                                            mov
2588 0000AD3E 3B15[E8620100]
                                 <1>
                                           cmp edx, [DelFile_FCluster]
                                                 short loc_rename_direntry_pop_invd_retn
2589 0000AD44 75E6
                                  <1>
                                           jne
2590
                                  <1>
                                            ; ESI = New file (or directory) name. (ASCIIZ string)
                                           ; 06/03/2016
2591
                                  <1>
                                          ; TRDOS v2 - NOTE: 'convert_file_name' procedure
2592
                                  <1>
2593
                                  <1>
                                            ; has been modified for eliminating following situation.
2594
                                  <1>
2595
                                  <1>
                                           ; TRDOS v1 - NOTE: If file/dir name is more than 11 bytes
2596
                                  <1>
                                           ; without a dot, attributes (edi+11) byte will be overwritten !
2597
                                  <1>
                                            ; (Dot file name input must be proper for 11 byte dir entry
2598
                                  <1>
                                           ; type file name output.)
2599 0000AD46 E8A2F6FFFF
                                  <1>
                                           call convert_file_name
2600
                                  <1>
2601 0000AD4B C605[AC600100]02
                                  <1>
                                                    byte [DirBuff ValidData], 2
                                            mov
2602 0000AD52 E8C5FBFFFF
                                            call save_directory_buffer
                                  <1>
2603 0000AD57 72D9
                                  <1>
                                                  short loc_rename_retn
                                            jc
2604
                                  <1>
2605
                                  <1> loc_rename_direntry_del_ln:
2606 0000AD59 0FB615[EE620100]
                                  <1>
                                           movzx edx, byte [DelFile_LNEL]
2607 0000AD60 08D2
                                  <1>
                                                   dl, dl
                                            or
2608 0000AD62 7410
                                  <1>
                                                   short loc_rename_direntry_update_parent_dir_lm_date
2609
                                  <1>
2610 0000AD64 0FB705[EC620100]
                                  <1>
                                            movzx eax, word [DelFile_EntryCounter]
2611 0000AD6B 29D0
                                  <1>
                                            sub eax, edx
2612 0000AD6D 72BE
                                  <1>
                                                   short loc_rename_direntry_invd_retn
2613
                                  <1>
2614
                                  <1> loc_rename_direntry_del_ln_continue:
2615
                                  <1>
                                            ; EAX = Directory Entry Number of the long name last entry
2616 0000AD6F E806FDFFFF
                                  <1>
                                            call delete_longname
2617
                                  <1>
2618
                                  <1> loc_rename_direntry_update_parent_dir_lm_date:
2619 0000AD74 E83EFCFFFF
                                  <1>
                                            call update_parent_dir_lmdt
2620 0000AD79 31C0
                                  <1>
                                            xor
                                                   eax, eax
2621 0000AD7B C3
                                  <1>
                                            retn
2622
                                  <1>
                                  <1> move source file to destination file:
2623
2624
                                       -; 15/\overline{10}/20\overline{16}
                                  <1>
                                            ; 11/03/2016
2625
                                  <1>
                                            ; 10/03/2016 (TRDOS 386 = TRDOS v2.0)
2626
                                  <1>
                                            ; 01/08/2011 (FILE.ASM)
2627
                                  <1>
2628
                                  <1>
                                            ; 04/08/2010
2629
                                  <1>
                                                Phase 1 -> Check destination file,
2630
                                  <1>
2631
                                  <1>
                                                           'not found' is required
2632
                                  <1>
                                               Phase 2 -> Check source file
2633
                                  <1>
                                                           'found' and proper attributes is required
2634
                                  <1>
                                                Phase 3 -> Make destination directory entry,
2635
                                  <1>
                                                       add new dir cluster or section if it is required
                                                Phase 4 -> Delete source directory entry.
2636
                                  <1>
                                                 cf = 1 causes to return before the phase 4.
2637
                                  <1>
2638
                                  <1>
                                                 (source file protection against any possible errors)
2639
                                  <1>
2640
                                  <1>
                                            ; 08/05/2011 major modification
                                                         -> destination file deleting is removed
2641
                                  <1>
2642
                                  <1>
                                                         for msdos move/rename compatibility.
2643
                                  <1>
                                                         (Access denied error will return if
                                  <1>
                                                         the destination file is found...)
2644
                                            ; INPUT ->
2645
                                  <1>
2646
                                  <1>
                                                   ESI = Source File Pathname (Asciiz)
                                                    EDI = Destination File Pathname (Asciiz)
2647
                                  <1>
                                                     AL = 0 \longrightarrow Interrupt (System call)
2648
                                  <1>
2649
                                  <1>
                                                     AL > 0 --> Command Interpreter (Question)
```

```
2650
                                <1>
                                                  AL = 1 --> Question Phase
2651
                                <1>
                                                  AL = 2 \longrightarrow Progress Phase
                                         ;
                                         ; OUTPUT ->
2652
                                <1>
2653
                                <1>
                                              cf = 0 \rightarrow OK
2654
                                <1>
                                                 EAX = Destination directory first cluster
2655
                                <1>
                                                  ESI = Logical DOS drive description table
2656
                                <1>
                                                EBX = Destination file structure offset
                                                 CX = 0 (CX > 0 --> calculate free space error)
                                <1>
2657
2658
                                <1>
                                                  cf = 1 -> Error code in EAX (AL)
2659
                                <1>
2660
                                <1>
                                         ; (EDX, ECX, EBX, ESI, EDI will be changed)
2661
                                <1>
2662 0000AD7C 3C02
                                <1>
                                              al, 2
                                         cmp
2663 0000AD7E 0F847F010000
                               <1>
                                               msftdf_df2_check_directory
                                         jе
2664 0000AD84 A2[6E640100]
                                <1>
                                         mov [move_cmd_phase], al
2665
                                <1>
2666
                               <1> msftdf_parse_sf_path:
                                     ; ESI = ASCIIZ pathname (Source)
2667
                               <1>
                                         push edi
2668 0000AD89 57
                               <1>
                                       mov edi, SourceFile_Drv
2669 0000AD8A BF[6C630100]
                               <1>
                                     call parse_path_name
2670 0000AD8F E824F7FFFF
                               <1>
                                       pop esi
jc shor
2671 0000AD94 5E
                                <1>
2672 0000AD95 7211
                               <1>
                                               short msftdf_psf_retn
2673
                               <1>
                                <1> msftdf_parse_df_path:
2674
                                      ; ESI = ASCIIZ pathname (Destination)
2675
                                <1>
2676 0000AD97 BF[EC630100]
2677 0000AD9C E817F7FFFF
                                         mov edi, DestinationFile_Drv
                               <1>
                                      call parse_path_name
                               <1>
                                       jnc short msftdf_check_sf_drv
2678 0000ADA1 7306
                               <1>
2679
                               <1>
                                      cmp al, 1 ; File or directory name is not existing
2680 0000ADA3 3C01
                               <1>
                                       jna
2681 0000ADA5 7602
                                <1>
                                              short msftdf_check_sf_drv
                               <1>
2682
2683
                               <1> msftdf_stc_retn:
2684 0000ADA7 F9
                                <1> stc
                                <1> msftdf_psf_retn:
2685
2686 0000ADA8 C3
                                <1>
                                        retn
2687
                                <1>
2688
                                <1> msftdf_check_sf_drv:
2689 0000ADA9 A0[6C630100]
                                       mov al, [SourceFile_Drv]
                                <1>
2690
                                <1>
2691
                                <1> msftdf_check_df_drv:
2692 0000ADAE 8A15[EC630100]
                                         mov dl, [DestinationFile_Drv]
                               <1>
2693
                                <1>
                                <1> msftdf_compare_sf_df_drv:
2694
2695 0000ADB4 29DB
                                <1> sub ebx, ebx
                                               bh, [Current Drv]
2696 0000ADB6 8A3D[86590100]
                               <1>
                                         mov
                                       cmp dl, al
2697 0000ADBC 38C2
                               <1>
2698 0000ADBE 7409
                                <1>
                                               short msftdf_check_sf_df_drv_ok
2699
                                <1>
2700
                                <1> msftdf_not_same_drv:
2701
                                <1> ; DL = source file's drive number
                                         mov dh, al ; destination file's drive number
2702 0000ADC0 88C6
                               <1>
                                      ; 15/10/2016 (11h -> 21)
2703
                                <1>
2704 0000ADC2 B815000000
                               <1>
                                               eax, 21; Not the same drive
                                         mov
2705 0000ADC7 F9
                                <1>
                                         stc
2706 0000ADC8 C3
                                <1>
                                         retn
2707
                                <1>
2708
                                <1> msftdf_check_sf_df_drv_ok:
2709 0000ADC9 8815[6F640100]
                                     mov [msftdf_sf_df_drv], dl
                               <1>
2710
                               <1>
                                         sub eax, eax
2711 0000ADCF 29C0
                               <1>
2712 0000ADD1 88D4
                               <1>
                                         mov ah, dl
2713 0000ADD3 0500010900
                               <1>
                                     add eax, Logical_DOSDisks
2714 0000ADD8 A3[70640100]
                               <1>
                                       mov [msftdf_drv_offset], eax
2715
                                <1>
2716 0000ADDD 38FA
                               <1>
                                         cmp dl, bh; byte [Current Drv]
2717 0000ADDF 7407
                               <1>
                                               short msftdf_df_check_directory
                                       je
2718
                                <1>
2719
                               <1> msftdf_change_drv:
2720 0000ADE1 E85EC1FFFF
                                      call change current drive
                               <1>
2721 0000ADE6 726D
                                <1>
                                         jc short msftdf_df_error_retn
2722
                               <1>
                               <1> msftdf check destination file:
2723
2724
                                <1> msftdf_df_check_directory:
2725 0000ADE8 BE[ED630100]
                               <1>
                                         mov esi, DestinationFile_Directory
2726 0000ADED 803E20
                                <1>
                                         cmp byte [esi], 20h
2727 0000ADF0 760F
                                <1>
                                        jna short msftdf df find 1
2728
                                <1>
                                <1> msftdf_df_change_directory:
2729
                                     inc byte [Restore_CDIR]
2730 0000ADF2 FE05[620D0100]
                                <1>
2731 0000ADF8 30E4
                                                ah, ah; CD COMMAND sign -> 0
                                <1>
                                         xor
                                          call change_current_directory
2732 0000ADFA E8A3F0FFFF
                                <1>
2733 0000ADFF 7254
                                <1>
                                                short msftdf_df_error_retn
2734
                                <1>
2735
                                <1> ;msftdf_df_change_prompt_dir_string:
2736
                                <1> ;
                                        call change_prompt_dir_string
2737
                                <1>
                                <1> msftdf df find 1:
2738
                                         mov esi, DestinationFile_Name
2739 0000AE01 BE[2E640100]
                               <1>
2740 0000AE06 803E20
                                          cmp byte [esi], 20h
                               <1>
                                         jna short msftdf_df_copy_sf_name
2741 0000AE09 7631
                                <1>
2742
                               <1>
2743
                               <1> msftdf df find 2:
                                         xor ax, ax ; DestinationFile_AttributesMask -> any/zero
call find_first_file
2744 0000AE0B 6631C0
                                <1>
2745 0000AE0E E8D4D4FFFF
                               <1>
                                         jnc msftdf permission denied retn
2746 0000AE13 0F838D000000
                               <1>
                                <1>
                                <1> msftdf df check error code:
2748
2749
                                       ;cmp eax, 2 ; File not found error
                                <1>
2750 0000AE19 3C02
                                <1>
                                         cmp al, 2
2751 0000AE1B 7537
                                <1>
                                               short msftdf_df_stc_retn
                                         jne
2752
                                <1>
2753
                                <1> msftdf_df_check_fname:
2754
                                <1> ; 15/10/2016
```

```
<1>
                           <1> msftdf_convert_df_direntry_name:
2762
2763
                           2764 0000AE30 BF[3E640100] <1>
2765 0000AE35 E8B3F5FFFF <1>
2766 0000AE3A EB1A <1>
                                   mov edi, DestinationFile_DirEntry
                                   call convert_file_name
                                 jmp short msftdf_restore_current dir 1
2766 0000AE3A EB1A
                           <1>
                           <1>
2767
<1> msftdf_df_stc_retn:
2783
2784 0000AE54 F9
                           <1> stc
2785
                           <1> msftdf_restore_cdir_failed:
                           <1> msftdf_df_error_retn:
2786
2787 0000AE55 C3
                                   retn
                           <1>
2788
                           <1>
                           <1> msftdf_restore_current_dir_1:
2789
2792 0000AE5F 8B35[70640100] <1>
2793 0000AE65 E891C1FFFF
2794 0000AE6A 72E9
2795
                           <1>
2796
                           <1> msftdf_sf_check_directory:
2800
                           <1> msftdf_sf_change_directory:
2801 0000AE76 FE05[620D0100] <1> inc byte [Restore_CDIR]
                                  xor ah, ah ; CD_COMMAND sign -> 0
call change_current_directory
2802 0000AE7C 30E4
                           <1>
2803 0000AE7E E81FF0FFFF
                           <1>
                                 call cnange_call _
jc short msftdf_return
2804 0000AE83 7227
                           <1>
2805
                           <1>
                           2806
2807
2808
                           <1>
2809
                           <1> msftdf_sf_find:
mov ax, 1800h; Only files
                                call find_first_file jc short msftdf_return
2813 0000AE93 7217
                           <1>
2814
                           <1>
                          <1> msftdf_sf_ambgfn_check:
2815
                           <1> or dx, dx; Ambiguous filename chars used sign (DX>0)
2816 0000AE95 6609D2
                          <1>
2817 0000AE98 7407
                                        short msftdf_sf_found
                                   jΖ
                           <1>
2819
                           <1> msftdf_ambiguous_file_name_error:
2820 0000AE9A B802000000
                           <1> mov eax, 2 ; File not found error
2821 0000AE9F F9
                           <1>
2822 0000AEA0 C3
                           <1>
                                  retn
2823
                           <1>
2824
                          <1> msftdf sf found:
2825 0000AEA1 80E31F
                         <1> and bl, 1Fh; Attributes, D-V-S-H-R
                                  jz short msftdf_save_sf_structure
2826 0000AEA4 7416
                           <1>
2827
                           <1>
                           <1> msftdf_permission_denied_retn:
2828
2829 0000AEA6 B805000000
                           <1> mov eax, 05h; Access (Permission) denied!
2830 0000AEAB F9
                           <1>
                                   stc
                           <1> msftdf rest_cdir_err_retn:
                           <1> msftdf_return:
2832
2833 0000AEAC C3
                           <1>
                                   retn
                           <1>
2834
2835
                           <1> msftdf_phase_1_return:
2836 0000AEAD 31C0
                                   xor eax, eax
                            <1>
                       <1>
2837 0000AEAF A2[6E640100]
                                         [move_cmd_phase], al ; 0
                                   mov
2838 0000AEB4 FEC0
                           <1>
                                        al; moval, 1
2839 0000AEB6 BB[03AF0000]
                           <1>
                                        ebx, msftdf df2 check directory
                                   mov
                                   ;mov edx, OFFFFFFFh
2840
                           <1>
2841 0000AEBB C3
                           <1>
                                   retn
2842
                           <1>
                           <1> msftdf save sf structure:
2843
2844 0000AEBC BE[78620100]
2845 0000AEC1 BF[BE630100]
                          <1> mov esi, FindFile DirEntry
                           <1>
                                   mov
                                        edi, SourceFile_DirEntry
2846 0000AEC6 B908000000
                                        ecx, 8
                           <1>
                                   mov
2847 0000AECB F3A5
                                       movsd
                           <1>
                                   rep
2848
                           <1>
                           <1> msftdf_df_copy_sf_parameters:
2849
2850 0000AECD BE0B000000
                           <1>
                                  mov esi, 11
2851 0000AED2 89F7
                           <1>
                                   mov edi, esi
                                add
2852 0000AED4 81C6[BE630100] <1>
                                        esi, SourceFile_DirEntry
2853 0000AEDA 81C7[3E640100]
                           <1>
                                   add
                                        edi, DestinationFile_DirEntry
                                   ;mov ecx, 21
                           <1>
2855 0000AEE0 B115
                           <1>
                                 mov cl, 21
                                  rep
2856 0000AEE2 F3A4
                           <1>
                                        movsb
                           <1>
2857
                           <1> msftdf restore current dir 2:
2858
2859 0000AEE4 803D[620D0100]00
                           <1> cmp byte [Restore_CDIR], 0
```

```
2860 0000AEEB 760D
                                 <1>
                                          mov esi, [msftdf_drv_offset] call restore_current_directory
2861 0000AEED 8B35[70640100]
                                 <1>
2862 0000AEF3 E803C1FFFF
                                <1>
                                         jc short msftdf_rest_cdir_err_retn
2863 0000AEF8 72B2
                                <1>
2864
                                 <1>
2865
                                 <1> msftdf_df2_check_move_cmd_phase:
2866 0000AEFA 803D[6E640100]01 <1>
                                      cmp byte [move cmd phase], 1
                                                 short msftdf_phase_1_return
2867 0000AF01 74AA
                                <1>
                                           jе
2868
                                 <1>
                                <1> msftdf_df2_check_directory:
2869
                                          mov esi, DestinationFile_Directory
2870 0000AF03 BE[ED630100]
                                <1>
2871 0000AF08 803E20
                                 <1>
                                           cmp
                                                 byte [esi], 20h
                                           jna short msftdf_make_dfde_locate_ffe_on_directory
2872 0000AF0B 760F
                                <1>
2873
                                <1> msftdf_df2_change_directory:
2874 0000AF0D FE05[620D0100]
                                <1> inc byte [Restore_CDIR]
<1> xor ah, ah; CD_COMMAND sign -> 0
2875 0000AF13 30E4
                                       call change_current_directory
jc short msftdf_return
2876 0000AF15 E888EFFFFF
                                <1>
2877 0000AF1A 7290
                                 <1>
2878
                                 <1>
                                 <1> ;msftdf_df2_change_prompt_dir_string:
2879
2880
                                 <1>;
                                          call change_prompt_dir_string
2881
                                 <1>
                                 <1> msftdf_make_dfde_locate_ffe_on_directory:
2882
2883
                                 <1>
                                          ; Current directory fcluster <> Directory buffer cluster
                                           ; Current directory will be reloaded by
2884
                                 <1>
2885
                                 <1>
                                           ; 'locate_current_dir_file' procedure
2886
                                 <1>
2887
                                 <1>
                                          ;xor ax, ax
                                          xor eax, eax mov ecx, eax
2888 0000AF1C 31C0
                                 <1>
2889 0000AF1E 89C1
                                 <1>
2890 0000AF20 6649
                                 <1>
                                           dec cx; FFFFh
2891
                                 <1>
                                                ; CX = FFFFh -> find first deleted or free entry
2892
                                 <1>
                                                 ; ESI would be ASCIIZ filename address if the call
2893
                                 <1>
                                                 ; would not be for first free or deleted dir entry
2894 0000AF22 E8CFF1FFF
                                 <1>
                                          call locate_current_dir_file
2895 0000AF27 733F
                                 <1>
                                          jnc msftdf_make_dfde_set_ff_dir_entry
                                <1>
2897
                                <1>
                                          ;cmp eax, 2
                                            cmp al, 2
2898 0000AF29 3C02
                                <1>
2899 0000AF2B 7537
                                <1>
                                           jne short msftdf_error_retn
2900
                                <1>
                                 <1> msftdf add_new_dir_entry_check_fs:
2901
                                <1> mov esi, [msftdf_drv_offset]
2902 0000AF2D 8B35[70640100]
2903 0000AF33 A1[B1600100]
                                <1>
                                           mov eax, [DirBuff_Cluster]
2904 0000AF38 807E0300
                                <1>
                                          cmp
                                                byte [esi+LD_FATType], 0
                                                 short msftdf_add_new_subdir_cluster
2905 0000AF3C 7711
                                <1>
                                         jа
2906
                                <1>
2907
                                 <1> msftdf_add_new_fs_subdir_section:
                                       ;CL=\overline{0}, CH=\overline{E5h} --> \overline{deleted} entry, CH=0 --> free entry
2908
                                 <1>
2909
                                 <1>
                                           ;xorcx, cx
                                           xor ch, ch; cx = 0 --> add a new subdir section
2910 0000AF3E 30ED
                                <1>
                                          call add_new_fs_section
    jc short msftdf_dsfde_error_retn
2911 0000AF40 E8830C0000
                                 <1>
2912 0000AF45 721E
                                <1>
2913
                                <1>
                                          ;mov [createfile_LastDirCluster], eax
2914
                                 <1>
2915 0000AF47 E8A30E0000
                                <1>
                                          call load_FS_sub_directory
                                 <1>
                                          ;mov ebx, Directory Buffer
                                           jnc
2917 0000AF4C 7318
                                 <1>
                                                 short msftdf_add_new_fs_subdir_section_ok
2918 0000AF4E C3
                                <1>
                                           retn
2919
                                <1>
2920
                                <1> msftdf_add_new_subdir_cluster:
2921 0000AF4F E881150000
                                <1>
                                      call add_new_cluster
2922 0000AF54 720F
                                <1>
                                                 short msftdf_dsfde_error_retn
                                           jс
2923
                                 <1>
2924
                                 <1>
                                         ;mov [createfile_LastDirCluster], eax
2925
                                 <1>
2926 0000AF56 E8570E0000
                                 <1>
                                           call load FAT sub directory
                                           jnc short msftdf_add_new_subdir_cluster_ok
2927 0000AF5B 7309
                                 <1>
2928
                                 <1>
                                           ; EBX = Directory buffer address
2929
                                 <1>
2930
                                 <1> msftdf_ansdc_update_parent_dir_lmdt:
                                 <1> msftdf make_dfde_err_upd_pdir_lmdt:
2931
2932 0000AF5D 50
                                          push eax
                                 <1>
                                           call update parent dir lmdt
2933 0000AF5E E854FAFFFF
                                 <1>
2934 0000AF63 58
                                 <1>
                                           pop
                                                 eax
2935
                                 <1>
                                 <1> msftdf_error_retn:
2937 0000AF64 F9
                                         stc
                                 <1>
2938
                                 <1> msftdf_dsfde_restore_cdir_failed:
2939
                                 <1> msftdf_dsfde_error_retn:
2940 0000AF65 C3
                                 <1>
                                           retn
                                 <1>
2942
                                 <1> msftdf_add_new_fs_subdir_section_ok:
2943
                                 <1> msftdf_add_new_subdir_cluster_ok:
                                          mov edi, ebx; Directory buffer address
2944 0000AF66 89DF
                                 <1>
2945
                                 <1>
                                 <1> msftdf make dfde set ff dir entry:
                                           mov edx, [Current_Dir_FCluster]
2947 0000AF68 8B15[80590100]
                                 <1>
2948 0000AF6E 8915[D4640100]
                                 <1>
                                                 [createfile FFCluster], edx
                                           mov
                                          ; EDI = Directory entry offset
                                 <1>
2950 0000AF74 BE[3E640100]
                                 <1>
                                           mov esi, DestinationFile_DirEntry
2951 0000AF79 B908000000
                                                 ecx, 8
                                 <1>
                                          mov
2952 0000AF7E F3A5
                                 <1>
                                          rep
                                                 movsd
                                 <1>
2954 0000AF80 C605[AC600100]02
                                 <1>
                                                 byte [DirBuff ValidData], 2
                                          mov
2955 0000AF87 E890F9FFFF
                                           call save_directory_buffer
                                 <1>
                                                 short msftdf make dfde err upd pdir lmdt
2956 0000AF8C 72CF
                                 <1>
                                           jс
2957
                                 <1>
2958
                                 <1> msftdf_make_dfde_update_pdir_lmdt:
                                           call update_parent_dir_lmdt
2959 0000AF8E E824FAFFFF
                                 <1>
2960
                                 <1>
2961
                                 <1> msftdf dsfde restore current dir 1:
2962 0000AF93 803D[620D0100]00
                                 <1>
                                           cmp byte [Restore CDIR], 0
2963 0000AF9A 760D
                                 <1>
                                                 short msftdf_dsfde_check_directory
2964 0000AF9C 8B35[70640100]
                                 <1>
                                                 esi, [msftdf drv offset]
                                           mov
```

```
<1>
                                          call restore_current_directory
2966 0000AFA7 72BC
                                <1>
                                                short msftdf dsfde restore cdir failed
2967
                                <1>
2968
                                <1> msftdf_dsfde_check_directory:
2969 0000AFA9 BE[6D630100]
                                2970 0000AFAE 803E20
                                <1>
                                           cmp
                                                 byte [esi], 20h
2971 0000AFB1 760F
                                           jna short msftdf_dsfde_find_file
                                <1>
2972
                                <1>
2973
                                <1> msftdf_dsfde_change_directory:
2974 0000AFB3 FE05[620D0100] <1> inc byte [Restore_CDIR]
2975 0000AFB9 28E4
                                <1>
                                           sub ah, ah; CD_COMMAND sign -> 0
2976 0000AFBB E8E2EEFFFF
                                 <1>
                                           call change current directory
2977 0000AFC0 72A3
                                <1>
                                          jc short msftdf_dsfde_error_retn
2978
                                <1>
                                 <1> ;msftdf dsfde_sf_change_prompt_dir_string:
2979
2980
                                 <1>;
                                          call change_prompt_dir_string
2981
2982
                                 <1> msftdf_dsfde_find_file:
                                        mov esi, SourceFile_Name ; Offset 66
mov ax, [esi+14] ; 80 -> SourceFile_AttributesMask
2983 0000AFC2 BE[AE630100]
                                 <1>
2984 0000AFC7 668B460E
                                <1>
                                           call find_first_file
2985 0000AFCB E817D3FFFF
                                <1>
2986 0000AFD0 7293
                                         jc short msftdf dsfde error retn
                                 <1>
2987
                                 <1>
2988
                                <1> msftdf_dsfde_delete_direntry:
2989 0000AFD2 8B35[70640100]
                                 <1> mov esi, [msftdf_drv_offset]
2990
                                 <1>
2991 0000AFD8 807E0300
                                <1>
                                           cmp
                                                byte [esi+LD FATType], 0
2992 0000AFDC 770A
                                <1>
                                        jа
                                                 short msftdf_delete_FAT_direntry
2993
                                 <1>
2994 0000AFDE 30DB
                                <1>
                                        xor bl, bl
                                       ; BL = 0 -> File
; EDI -> Director
2995
                                <1>
                                          ; EDI -> Directory buffer entry offset/address
2996
                                 <1>
2997 0000AFE0 E8E40B0000
                                <1>
                                           call delete_fs_directory_entry
2998 0000AFE5 7315
                                <1>
                                                short msftdf_dsfde_restore_current_dir_2
2999 0000AFE7 C3
                                 <1>
                                         retn
3000
                                 <1>
                                 <1> msftdf delete FAT direntry:
3002 0000AFE8 8A1D[75620100]
                                          mov bl, [FindFile_LongNameEntryLength]
                                 <1>
3003 0000AFEE 668B0D[A0620100] <1>
                                                cx, [FindFile DirEntryNumber]
                                          mov
                                          ; ESI = Logical DOS drive description table address
                                 <1>
3005
                                 <1>
                                          ; EDI = Directory buffer entry offset/address
                                          call delete_directory entry
3006 0000AFF5 E89AFCFFFF
                                 <1>
3007 0000AFFA 721C
                                          jс
                                 <1>
                                                short msftdf_retn
3008
                                 <1>
                                 <1> msftdf_dsfde_restore_current_dir_2:
3009
3010 0000AFFC 803D[620D0100]00 <1> cmp byte [Restore_CDIR], 0
3011 0000B003 7607
                                           jna short msftdf new dir fcluster retn
                                <1>
                                          ;mov esi, [msftdf_drv_offset]
call restore_current_directory
3012
                                <1>
3013 0000B005 E8F1BFFFFF
                                <1>
                                          jc short msftdf_retn
3014 0000B00A 720C
                                <1>
3015
                                <1>
                                 <1> msftdf_new_dir_fcluster retn:
3016
3017 0000B00C 31C9
                                <1> xor ecx, ecx
3018 0000B00E A1[D4640100]
                                <1>
                                           mov eax, [createfile_FFCluster]
3019 0000B013 BB[EC630100]
                                <1>
                                          mov ebx, DestinationFile_Drv
3020
                                 <1>
3021
                                 <1> msftdf retn:
3022 0000B018 C3
                                 <1>
                                          retn
3023
                                 <1>
3024
                                 <1>
3025
                                 <1> copy_source_file_to_destination_file:
                                       ; 17/10/2016
3026
                                 <1>
3027
                                 <1>
                                          ; 16/10/2016
                                         ; 15/10/2016
3028
                                 <1>
                                         ; 30/03/2016, 31/03/2016
; 24/03/2016, 25/03/2016, 28/03/2016
3029
                                 <1>
3030
                                 <1>
                                          ; 21/03/2016, 22/03/2016, 23/03/2016
3031
                                 <1>
3032
                                 <1>
                                          ; 16/03/2016, 17/03/2016, 18/03/2016
3033
                                 <1>
                                           ; 15/03/2016 (TRDOS 386 = TRDOS v2.0)
                                           ; 02/09/2011 (FILE.ASM 'copy_source_file_to_destination_file')
3034
                                 <1>
3035
                                 <1>
                                          ; 01/08/2010 - 18/05/2011
3036
                                 <1>
3037
                                 <1>
                                               Command Interpreter phase 1 enter ->
                                                       AL = 1 -> Caller is command interpreter
3038
                                 <1>
3039
                                 <1>
                                                       AL = 2 -> The second call, re-enter/continue
                                              Phase 1 -> Check source file
3040
                                 <1>
                                                         'found' is required
3041
                                           ; Phase 2 -> Check destination file,
3042
                                 <1>
3043
                                 <1>
                                                          save 'found' or 'not found' status
3044
                                 <1>
                                                          'permission denied' error will be return
3045
                                 <1>
                                                          if attributes have not for ordinary file
                                                          without readonly attribute
3046
                                 <1>
3047
                                 <1>
                                               Command Interpreter phase 1 return ->
3048
                                 <1>
                                                          DH = Source file attributes
3049
                                 <1>
                                                          DL = Destination file found status
3050
                                 <1>
                                                          EAX = 0
3051
                                 <1>
                                               Command Interpreter phase 2 enter ->
3052
                                 <1>
                                                          AL = 2 \rightarrow Continue from the last position
                                                          AH =
3053
                                 <1>
                                               Phase 3 -> Load source file or use read/write cluster method
3054
                                 <1>
3055
                                 <1>
                                               Phase 4 -> Create destination file if it is not found
                                               Phase 5 -> Open destination file
3056
                                 <1>
                                               Phase 6 -> Read from source and write to destination
3057
                                 <1>
3058
                                 <1>
                                               Phase 7 -> Unload source file, if it is loaded at memory
3059
                                 <1>
                                                   cf = 1 causes to return before the phase 7
                                                          but loaded file will be unloaded
3060
                                 <1>
3061
                                 <1>
                                                         (allocated memory block will be deallocated)
3062
                                 <1>
                                           ; INPUT ->
3063
                                 <1>
3064
                                 <1>
                                                 ESI = Source File Pathname (Asciiz)
3065
                                 <1>
                                                    EDI = Destination File Pathname (Asciiz)
3066
                                 <1>
                                                    AL = 0 --> Interrupt (System call)
                                                    AL > 0 --> Command Interpreter (Question)
3067
                                 <1>
3068
                                 <1>
                                                    AL = 1 --> Question Phase
3069
                                 <1>
                                                    AL = 2 \longrightarrow Progress Phase
```

2965 0000AFA2 E854C0FFFF

```
3070
                                 <1>
3071
                                 <1>
                                           ; OUTPUT ->
                                                 cf = 0 \rightarrow OK
3072
                                 <1>
                                           ;
3073
                                 <1>
                                                 EAX = Destination file first cluster
3074
                                 <1>
3075
                                 <1>
                                                    CL > 0 if there is file reading error before EOF
3076
                                 <1>
                                                         (incomplete copy)
                                                    {
m CH} > 0 if file is (full) loaded at memory
3077
                                 <1>
3078
                                 <1>
3079
                                 <1>
                                                 cf = 1 -> Error code in AL (EAX)
                                           ;
3080
                                 <1>
                                           ; (EBX, ECX, ESI, EDI register contents will be changed)
3081
                                 <1>
3082
                                 <1>
3083
                                 <1>
3084 0000B019 3C02
                                 <1>
                                                 al, 2
                                           cmp
                                                  csftdf2_check_cdrv
3085 0000B01B 0F845A020000
                                 <1>
                                 <1>
3087
                                 <1> ; Phase 1
3088
                                 <1>
3089 0000B021 A2[94640100]
                                 <1>
                                                 byte [copy_cmd_phase], al
                                           mov
3090
                                 <1>
3091 0000B026 57
                                           push edi; *
                                 <1>
3092
                                 <1>
3093
                                 <1> csftdf_parse_sf_path:
                                           mov edi, SourceFile_Drv call parse_path_name
3094 0000B027 BF[6C630100]
                                 <1>
3095 0000B02C E887F4FFF
                                 <1>
3096 0000B031 721C
                                 <1>
                                                 short csftdf_parse_sf_path_failed
                                           jс
3097
                                 <1>
3098
                                 <1> csftdf_parse_df_path:
3099 0000B033 5E
                                 <1>
                                          pop esi; * (pushed edi)
3100
                                 <1>
3101
                                 <1> csftdf sf check filename exists:
3102 0000B034 803D[AE630100]21
                                         cmp byte [SourceFile_Name], 21h
                                 <1>
3103 0000B03B 7215
                                 <1>
                                                  short csftdf_sf_file_not_found_error
3104
                                 <1>
3105 0000B03D BF[EC630100]
                                 <1>
                                           mov
                                                  edi, DestinationFile_Drv
3106 0000B042 E871F4FFFF
                                           call parse path name
                                 <1>
3107 0000B047 7310
                                 <1>
                                           jnc
                                                 short csftdf check sf cdrv
3108
                                 <1>
3109 0000B049 3C01
                                 <1>
                                                  al, 1; File or directory name is not existing
                                           cmp
3110 0000B04B 760C
                                 <1>
                                                 short csftdf_check_sf_cdrv
                                 <1>
3112
                                 <1> csftdf_parse_df_path_failed:
3113 0000B04D F9
                                 <1>
                                           stc
                                 <1> csftdf_sf_error_retn:
3114
3115 0000B04E C3
                                 <1>
                                           retn
3116
                                 <1>
                                 <1> csftdf_parse_sf_path_failed:
3117
                                           pop edi; *
jmp short csftdf_sf_error_retn
3118 0000B04F 5F
                                 <1>
3119 0000B050 EBFC
                                 <1>
3120
                                 <1>
3121
                                 <1> csftdf sf file not found error:
                                           mov eax, 2; File not found
3122 0000B052 B802000000
                                 <1>
3123 0000B057 EBF5
                                 <1>
                                           jmp short csftdf_sf_error_retn
3124
                                 <1>
3125
                                 <1> csftdf_check_sf_cdrv:
3126 0000B059 8A3D[86590100]
                                 <1>
                                           mov bh, [Current Drv]
3127
                                 <1>
3128 0000B05F 883D[97640100]
                                 <1>
                                                 [csftdf_cdrv], bh ; 23/03/2016
3129
                                 <1>
3130 0000B065 8A15[6C630100]
                                 <1>
                                           mov
                                                  dl, [SourceFile_Drv]
3131 0000B06B 38FA
                                 <1>
                                           cmp
                                                 dl, bh ; byte [Current Drv]
3132 0000B06D 7407
                                 <1>
                                                  short csftdf_sf_check_directory
                                           jе
3133
                                 <1>
                                           call change_current_drive
3134 0000B06F E8D0BEFFFF
                                 <1>
3135 0000B074 72D8
                                 <1>
                                                 short csftdf_sf_error_retn
                                           jс
3136
                                 <1>
3137
                                 <1> csftdf_sf_check_directory:
3138 0000B076 BE[6D630100]
                                 <1>
                                           mov esi, SourceFile_Directory
3139 0000B07B 803E20
                                 <1>
                                                 byte [esi], 20h
                                           cmp
                                           jna short csftdf find sf
3140 0000B07E 760F
                                 <1>
3141
                                 <1>
                                 <1> csftdf sf change directory:
3142
3143 0000B080 FE05[620D0100]
                                 <1>
                                      inc byte [Restore_CDIR]
                                                 ah, ah ; CD COMMAND sign -> 0
3144 0000B086 30E4
                                 <1>
                                           xor
3145 0000B088 E815EEFFFF
                                           call change_current_directory
                                 <1>
3146 0000B08D 72BF
                                 <1>
                                           jc
                                                 short csftdf_sf_error_retn
3147
                                 <1>
3148
                                 <1> ;csftdf_sf_change_prompt_dir_string:
3149
                                 <1> ; call change_prompt_dir_string
3150
                                 <1>
                                  <1> csftdf_find_sf:
3151
                            <1>
3152 0000B08F BE[AE630100]
                                                 esi, SourceFile Name
                                          mov
3153 0000B094 66B80018
                                <1>
                                                ax, 1800h ; Except volume label and dirs
3154 0000B098 E84AD2FFFF
                                <1>
                                          call find_first_file
3155 0000B09D 72AF
                                <1>
                                           jс
                                                 short csftdf_sf_error_retn
                                <1>
                                           and dx, dx; Ambiguous filename chars used sign (DX>0) jz short csftdf sf found
                                <1> csftdf_sf_ambgfn_check:
3157
3158 0000B09F 6621D2
                                <1>
3159 0000B0A2 7407
                                <1>
                                                 short csftdf_sf_found
3160
                                <1>
3161
                                 <1> csftdf_ambiguous_file_name_error:
                                <1>
3162 0000B0A4 B802000000
                                           mov eax, 2 ; File not found error
3163 0000B0A9 F9
                                <1>
3164 0000B0AA C3
                                <1>
                                          retn
3165
                                <1>
                                <1> csftdf sf found:
                                <1>
3167 0000B0AB A3[98640100]
                                          mov [csftdf_filesize], eax
3168
                                <1>
                                           or
3169 0000B0B0 09C0
                                <1>
                                                 eax, eax
3170 0000B0B2 7507
                                         jnz short csftdf_set_source_file_direntry
                                <1>
3171
                                 <1>
                                <1> csftdf_sf_file_size_zero:
3172
3173 0000B0B4 B814000000
                                <1>
                                          mov eax, 20; TRDOS zero length (file size) error
3174 0000B0B9 F9
                                 <1>
                                           stc
```

```
3175 0000B0BA C3
                               <1>
                                          retn
3176
                                <1>
3177
                               <1> csftdf_set_source_file_direntry:
3178 0000B0BB BE[78620100]
3178 0000B0BB BE[78620100]
3179 0000B0C0 BF[BE630100]
                               <1>
                                     mov esi, FindFile_DirEntry
                               <1>
                                         mov
                                               edi, SourceFile_DirEntry
3180 0000B0C5 B908000000
                                <1>
                                         mov
                                               ecx, 8
3181 0000B0CA F3A5
                               <1>
                                        rep movsd
3182
                                <1>
3183
                                <1> csftdf_sf_restore_cdrv:
                               3184
                            <1>
3185 0000B0CC 8A15[97640100]
                                       mov dl, [csftdf_cdrv]
                                         cmp dl, [Current_Drv]
je short csftdf_sf_restore_cdir
3186 0000B0D2 3A15[86590100]
                                <1>
                                     cmp dl, [Current_Drv]
je short csftdf_sf_resto
call change_current_drive
3187 0000B0D8 7407
                               <1>
3188 0000B0DA E865BEFFFF
                               <1>
3189 0000B0DF 724F
                                <1>
                                       jc short csftdf_df_error_retn ; 30/03/2016
3190
                                <1>
                                <1> csftdf sf restore cdir:
3191
3192 0000B0E1 803D[620D0100]00 <1> cmp byte [Restore_CDIR], 0
3193 0000B0E8 7612
                                <1>
                                               short csftdf_df_check_filename_exists
                                         jna
3194 0000B0EA 29C0
3195 0000B0EC BE00010900 <1>
                                         sub eax, eax
                                      mov esi, Logical_DOSDisks
                                        mov
                                               ah, dl ; byte [csftdf_cdrv]
                               <1>
3197 0000B0F3 01C6
                                        add
                                               esi, eax
                                     call restore_current_directory
jc short csftdf_df_error_retn
                          <1>
3198 0000B0F5 E801BFFFFF
3199 0000B0FA 7234
                               <1>
3200
                               <1>
                               <1> csftdf df check filename exists:
3202 0000B0FC 803D[2E640100]20 <1>
                                      cmp byte [DestinationFile_Name], 20h
3203 0000B103 7716
                                <1>
                                                short csftdf_check_df_cdrv
3204
                                <1>
3205
                               <1> csftdf_copy_sf_name:
3206 0000B105 BF[2E640100]
                                <1>
                                     mov edi, DestinationFile_Name
3207 0000B10A BE[AE630100]
                                               esi, SourceFile Name
                               <1>
                                         mov
3208 0000B10F B10C
                               <1>
                                         mov cl, 12
3209
                                <1>
3210
                               <1> csftdf_df_copy_sf_name_loop:
                             <1> lodsb
3211 0000B111 AC
                               <1>
3212 0000B112 AA
                                          stosb
3213 0000B113 08C0
                               <1>
                                          or
                                               al, al
3214 0000B115 7404
                               <1>
                                               short csftdf_check_df_cdrv
                                         jΖ
3215 0000B117 FEC9
                               <1>
                                         dec cl
                                        jnz csftdf df copy sf name loop
3216 0000B119 75F6
                               <1>
3217
                               <1>
3218
                               <1> csftdf_check_df_cdrv:
3219 0000B11B 8A15[EC630100]
                               <1> mov dl, [DestinationFile_Drv]
3220 0000B121 3A15[86590100]
                                               dl, [Current Drv]
                               <1>
                                         cmp
3221 0000B127 7408
                                               short csftdf_df_check_directory
                               <1>
                                         jе
3222
                                <1>
3223 0000B129 E816BEFFFF
                                <1>
                                         call change_current_drive
3224 0000B12E 7301
                                <1>
                                         jnc short csftdf_df_check_directory
3225
                                <1>
3226
                                <1> csftdf_df_error_retn:
3227 0000B130 C3
                                <1>
                                        retn
3228
                                <1>
                                <1> csftdf_df_check_directory:
3230 0000B131 BE[ED630100]
                                <1> mov esi, DestinationFile_Directory
3231 0000B136 803E20
                               <1>
                                         cmp byte [esi], 20h
3232 0000B139 760F
                               <1>
                                       jna short csftdf_find_df
3233
                                <1>
3234
                               <1> csftdf_df_change_directory:
3235 0000B13B FE05[620D0100]
                               <1> inc byte [Restore_CDIR]
3236 0000B141 28E4
                                <1>
                                          sub
                                               ah, ah ; CD COMMAND sign -> 0
                                         call change_current_directory
3237 0000B143 E85AEDFFFF
                                <1>
3238 0000B148 72E6
                                <1>
                                               short csftdf_df_error_retn
3239
                                <1>
3240
                                <1> ;csftdf_df_change_prompt_dir_string:
3241
                                <1>;
                                       call change_prompt_dir_string
3242
                                <1>
3243
                                <1> csftdf_find_df:
                                <1> ; 23/03/2016
3244
3245 0000B14A 29DB
                                <1>
                                         sub ebx, ebx
3246 0000B14C 8A3D[EC630100]
                                <1>
                                         mov
                                               bh, [DestinationFile_Drv]
                                         add ebx, Logical DOSDisks
3247 0000B152 81C300010900
                                <1>
3248 0000B158 891D[C4640100]
                                <1>
                                       mov [csftdf_df_drv_dt], ebx
3249
                                <1>
3250 0000B15E BE[2E640100]
                                <1>
                                         mov
                                                esi, DestinationFile_Name
                                <1> xor
3251 0000B163 6631C0
                                               ax, ax
                                                ; DestinationFile_AttributesMask -> any/zero
3252
                                <1>
3253 0000B166 E87CD1FFFF
                                <1>
                                         call find_first_file
3254 0000B16B 7218
                                <1>
                                        jc short csftdf_df_check_error_code
3255
                                <1>
                                <1> csftdf_df_ambgfn_check:
3256
3257 0000B16D 6609D2
                                <1>
                                                dx, dx; Ambiguous filename chars used sign (DX>0)
                                          or
                                <1>
3258 0000B170 752A
                                          jnz
                                               short csftdf_df_error_inv_fname
3259
                                <1>
                                <1> csftdf_df_found:
3260
3261 0000B172 C605[96640100]01 <1>
                                     mov byte [DestinationFileFound], 1
                                         ; 17/10/2016 (cl -> bl)
3262
                               <1>
3263 0000B179 80E31F
                                <1>
                                          and bl, 1Fh; Attributes, D-V-S-H-R
                                         jz short csftdf_df_save_first_cluster
3264 0000B17C 745F
                               <1>
3265
                               <1>
                                <1> csftdf df permission denied retn:
3266
                                         mov eax, 05h; Access/Permission denied.
3267 0000B17E B805000000
                               <1>
                               <1> csftdf df error stc retn:
3268
3269 0000B183 F9
                                <1>
                                          stc
3270 0000B184 C3
                               <1>
                                          retn
3271
                                <1>
3272
                                <1> csftdf df check error code:
3273
                                <1>
                                         ;cmp eax, 2
3274 0000B185 3C02
                                <1>
                                          cmp
                                               al, 2
3275 0000B187 75FA
                                                short csftdf df error stc retn
                                <1>
                                         jne
                                <1>
3277 0000B189 C605[96640100]00
                               <1>
                                         mov
                                               byte [DestinationFileFound], 0
3278
                                <1>
3279
                                <1>
                                         ; 15/10/2016
```

```
<1> mov esi, FindFile_Name; *
<1> call check_filename
<1> jnc short csftdf_df_valid_fname
3280 0000B190 BE[68620100] <1>
3281 0000B195 E80BD5FFFF <1>
3282 0000B19A 7307 <1>
3282 0000B19A 7307
                                <1> csftdf_df_error_inv_fname: ; 'invalid file name !'
3283
3284 0000B19C B81A000000
                                 <1> mov eax, ERR INV FILE NAME ; 26
3285 0000B1A1 F9
                                 <1>
                                           stc
3286 0000B1A2 C3
                                 <1>
                                           retn
3287
                                 <1>
3288
                                 <1> csftdf_df_valid_fname:
                                         ; 21/03/2016
3289
                                 <1>
                                           ; (Capitalized file name)
3290
                                 <1>
                                           ;mov esi, FindFile_Name ; * ; 15/10/2016
mov edi, DestinationFile_Name
3291
                                  <1>
3292 0000B1A3 BF[2E640100]
                                 <1>
3293 0000B1A8 A5
                                 <1>
                                           movsd
3294 0000B1A9 A5
                                 <1>
                                           movsd
3295 0000B1AA A5
                                 <1>
                                           movsd
3296
                                 <1>
                                           ;movsb
3297
                                  <1>
3298
                                  <1> csftdf_check_disk_free_size_0:
3299 0000B1AB A1[DA630100]
                                 <1>
                                         mov eax, [SourceFile_DirEntry+DirEntry_FileSize]
3300
                                 <1>
                                  <1> csftdf check disk free size 1:
3301
                                         ;sub ebx, ebx
3302
                                  <1>
3303
                                  <1>
                                           ;mov esi, Logical_DOSDisks
3304
                                  <1>
                                           ;mov bh, [DestinationFile_Drv]
3305
                                  <1>
                                           ;add esi, ebx
                                  <1>
3307 0000B1B0 8B35[C4640100]
                                           mov esi, [csftdf_df_drv_dt]; 23/03/2016
                                 <1>
                                  <1>
                                           movzx ecx, word [esi+LD_BPB+BytesPerSec] ; 17, LD BPB + 0Bh
3309 0000B1B6 0FB74E11
                                 <1>
3310 0000B1BA 01C8
                                 <1>
                                           add eax, ecx
3311 0000B1BC 48
                                                 eax ; file size (additional bytes) + 511 (round up)
                                 <1>
                                           dec
                                 <1> csftdf_check_disk_free_size_3: ; 16/03/2016
3312
3313 0000B1BD 29D2
                                <1>
                                           sub edx, edx
3314 0000B1BF F7F1
                                 <1>
                                           div ecx; bytes per sector
3315
                                 <1>
                                 <1> csftdf check disk free size:
                                       cmp eax, [esi+LD_FreeSectors]
3317 0000B1C1 3B4674
                                 <1>
3318 0000B1C4 0F8294000000
                                 <1>
                                                  csftdf_check_disk_free_size_ok
                                            jb
                                                  short csftdf_df_insufficient_disk_space
3319 0000B1CA 770A
                                 <1>
3320
                                 <1>
                                           cmp byte [esi+LD_FATType], 0 ; FS needs FDT sector also.
3321 0000B1CC 807E0300
                                 <1>
                                                    csftdf_check_disk_free_size_ok
3322 0000B1D0 0F8788000000
                                 <1>
                                           jа
3323
                                 <1>
3324
                                 <1> csftdf_df_insufficient_disk_space:
                                           mov eax, 27h; insufficient disk space jmp short csftdf_df_error_stc_retn
3325 0000B1D6 B827000000
                                 <1>
3326 0000B1DB EBA6
                                 <1>
3327
                                 <1>
                                 <1> csftdf df save first cluster:
3328
3329
                                 <1>
                                          ; ESI = FindFile DirEntry (for the old destination file)
                                           ; EAX = Old destination file size
3330
                                  <1>
3331
                                  <1>
                                           ; 24/03/2016
                                           ; EDI = Directory entry address (within Dir Buffer boundaries)
3332
                                 <1>
3333 0000B1DD 81EF00000800
                                 <1>
                                         sub edi, Directory_Buffer ; (<65536)</pre>
                                           shr di, 5 ; Convert entry offset to entry index/number
3334 0000B1E3 66C1EF05
                                 <1>
3335 0000B1E7 66893D[66640100]
                                 <1>
                                                 [DestinationFile_DirEntryNumber], di ; (<2048)
                                           mov
                                 <1>
                                 <1> csftdf_df_check_sf_df_fcluster:
3337
                                       mov dx, [esi+DirEntry_FstClusHI] shl edx, 16
3338 0000B1EE 668B5614
                                 <1>
3339 0000B1F2 C1E210
                                 <1>
3340 0000B1F5 668B561A
                                 <1>
                                           mov dx, [esi+DirEntry_FstClusLO]
3341 0000B1F9 8915[A8640100]
                                                 [csftdf df cluster], edx
                                 <1>
                                           mov
                                 <1> csftdf_df_check_sf_df_fcluster 1:
3342
3343 0000B1FF 668B15[D2630100]
                                 <1> mov dx, [SourceFile_DirEntry+DirEntry_FstClusHI]
                                           shl
3344 0000B206 C1E210
                                  <1>
                                                 edx, 16
3345 0000B209 668B15[D8630100]
                                 <1>
                                           mov
                                                  dx, [SourceFile_DirEntry+DirEntry_FstClusLO]
3346 0000B210 3B15[A8640100]
                                        cmp
jne
                                 <1>
                                                 edx, [csftdf df cluster]
3347 0000B216 7512
                                 <1>
                                                 short csftdf_df_check_sf_df_fcluster_ok
3348
                                 <1> csftdf_df_check_sf_df_drv:
3349 0000B218 8A15[6C630100]
                                 <1> mov dl, [SourceFile_Drv]
3350 0000B21E 3A15[EC630100]
                                 <1>
                                           cmp
                                                  dl, [DestinationFile_Drv]
                                                 short csftdf_df_check_sf df fcluster ok
3351 0000B224 7504
                                  <1>
                                           jne
3352
                                 <1>
3353
                                  <1>
                                           ; source and destination files are same !
3354
                                  <1>
                                           ; (they have same first cluster value on same logical disk)
3355
                                  <1>
3356 0000B226 31C0
                                  <1>
                                                 eax, eax; mov eax, 0 -> Bad command or file name!
3357 0000B228 F9
                                  <1>
                                            stc
3358 0000B229 C3
                                  <1>
                                            retn
3359
                                  <1>
                                  <1> csftdf_df_check_sf_df_fcluster_ok:
3360
                                  <1> csftdf_df_move_findfile_struct:
3361
                                 <1>
                                           ; mov esi, FindFile_DirEntry
3363 0000B22A BF[3E640100]
                                 <1>
                                           mov edi, DestinationFile_DirEntry
3364 0000B22F B908000000
                                 <1>
                                           mov
                                                 ecx, 8
3365 0000B234 F3A5
                                 <1>
                                           rep
                                                 movsd
                                 <1>
3367
                                 <1> csftdf_check_disk_free_size_2:
3368 0000B236 89C2
                                 <1>
                                                 edx, eax ; Old destination file size
3369
                                 <1>
                                           ;mov eax, [SourceFile DirEntry+DirEntry FileSize]
3370
                                 <1>
                                                 eax, [csftdf_filesize]; 23/03/2016
3371 0000B238 A1[98640100]
                                 <1>
                                           mov
                                 <1>
3372
                                           ;;sub ecx, ecx; 0
3373
                                  <1>
                                           ;mov esi, Logical DOSDisks
3374
                                  <1>
                                                  ch, [DestinationFile Drv]
3375
                                  <1>
                                           ;mov
3376
                                  <1>
                                           ;add
                                                 esi, ecx
3377
                                  <1>
                                                 [csftdf df drv dt], esi
3378
                                  <1>
                                           ;mov
3379
                                 <1>
3380 0000B23D 8B35[C4640100]
                                                  esi, [csftdf_df_drv_dt] ; 23/03/2016
                                 <1>
                                           mov
3381
                                  <1>
3382 0000B243 668B4E11
                                 <1>
                                           mov
                                                  cx, [esi+LD_BPB+BytesPerSec] ; 17, LD_BPB + OBh
3383 0000B247 01CA
                                 <1>
                                            add
                                                  edx, ecx; + 512
3384 0000B249 01C8
                                 <1>
                                           add
                                                 eax, ecx; + 512
```

```
3385 0000B24B 4A
                                <1>
                                                 edx ; old file size + 511 (round up)
                                           dec
3386 0000B24C 48
                                 <1>
                                           dec
                                                 eax ; new file size + 511 (round up)
3387 0000B24D F7D9
                                <1>
                                                 ecx ; -512 ; OFFFFFE00h
                                           neg
3388 0000B24F 21CA
                                <1>
                                                 edx, ecx ; = old sector count * 512
3389 0000B251 21C8
                                 <1>
                                                 eax, ecx; = new sector count * 512
                                           and
3390
                                 <1>
3391 0000B253 29D0
                                <1>
                                           sub
                                                 eax, edx; new file size - old file size (on disk)
3392 0000B255 7607
                                 <1>
                                                 short csftdf_check_disk_free_size_ok
                                          jna
3393
                                 <1>
3394 0000B257 F7D9
                                 <1>
                                         neg ecx; 512 (bytes per sector); 200h
3395
                                 <1>
                                           ; check free space for additional sectors
3396
                                           ; eax = number of additional sectors * bytes per sector
                                 <1>
                                           ; esi = Logical DOS drive number (of destination disk)
3397
                                 <1>
3398 0000B259 E95FFFFFF
                                 <1>
                                                    csftdf_check_disk_free_size_3
                                 <1>
3399
3400
                                 <1> csftdf_check_disk_free_size_ok:
                                        ; 18/03/201<del>6</del>
3401
                                 <1>
                                 <1> csftdf_df_check_copy_cmd_phase:
3402
                                       mov al, [copy_cmd_phase]
cmp al, 1
3403 0000B25E A0[94640100]
                                 <1>
3404 0000B263 3C01
                                <1>
3405 0000B265 7514
                                <1>
                                                 short csftdf2_check_cdrv
                                 <1>
3407 0000B267 31C0
                                <1>
                                          xor
                                                  eax, eax
3408 0000B269 A2[94640100]
                                 <1>
                                         mov
                                                 [copy_cmd_phase], al ; 0
                                 <1>
3409
3410 0000B26E 8A15[96640100]
                                 <1>
                                           mov
                                                  dl, [DestinationFileFound]
3411 0000B274 8A35[C9630100]
                                 <1>
                                         mov
                                                 dh, [SourceFile_DirEntry+11] ; Attributes
3412
                                 <1>
3413
                                 <1> csftdf_return:
3414 0000B27A C3
                                 <1>
                                           retn
3415
                                 <1>
3416
                                 <1> ; Phase 2
3417
                                 <1>
3418
                                 <1> csftdf2 check cdrv:
                                      ; 18/03<del>/</del>2016
3419
                                 <1>
3420
                                 <1>
                                           ; Here, destination drive and directory are ready !
                                          ; (checking/restoring is not needed)
3421
                                 <1>
3422
                                 <1>
                                         ; (Since at the end of the phase 1)
3423
                                 <1>
                                         mov dl, [DestinationFile_Drv]
3424
                                 <1>;
3425
                                 <1>;
                                          cmp dl, [Current_Drv]
3426
                                 <1>;
                                           jе
                                                 short csftdf2_df_check_directory
                                 <1>;
3427
                                           call change current drive
3428
                                 <1>;
3429
                                 <1>;
                                          jc short csftdf2_read_error
3430
                                 <1>;
                                 <1> ;csftdf2_df_check_directory:
3431
                                           mov esi, DestinationFile_Directory
                                 <1> ;
3432
3433
                                 <1>;
                                           cmp
                                                 byte [esi], 20h
3434
                                 <1>;
                                           jna short csftdf2_df_check_found_or_not
3435
                                 <1>;
                                 <1> ; csftdf2_df_change_directory:
<1> ; inc byte [Restore_CDIR]
<1> ; xor ah, ah ; CD_COMMAND sign -> 0
3436
3437
3438
3439
                                 <1>;
                                           call change_current_directory
3440
                                 <1>;
                                                 short csftdf2_stc_return
                                           jc
3441
3442
                                 <1> ;;csftdf2_df_change_prompt_dir_string:
3443
                                 <1> ;; call change_prompt_dir_string
3444
                                 <1>
3445
                                 <1> csftdf2_df_check_found_or_not:
                                      ; 21/03/20<del>1</del>6
                                 <1>
3447 0000B27B 803D[96640100]00
                                           cmp byte [DestinationFileFound], 0
                                <1>
3448 0000B282 7739
                                 <1>
                                                 short csftdf2_set_sf_percentage
                                           jа
3449
                                 <1>
3450
                                 <1> csftdf2_create_file:
3451 0000B284 BE[2E640100]
                                <1> mov esi, DestinationFile_Name
3452 0000B289 A1[98640100]
                                <1>
                                                 eax, [csftdf_filesize]
                                           mov
3453 0000B28E 30C9
                                 <1>
                                                 cl, cl ; 0
                                           xor
3454
                                <1>
3455 0000B290 31DB
                                 <1>
                                        xor ebx, ebx; 0
3456 0000B292 4B
                                 <1>
                                          dec ebx; OFFFFFFFh
                                 <1>
3457
3458
                                 <1>
                                         ; INPUT ->
                                         ; EAX -> File Size
3459
                                 <1>
                                                 ESI = ASCIIZ File name
3460
                                 <1>
3461
                                 <1>
                                                 CL = File attributes
                                           ;
                                                 EBX = FFFFFFFFh -> empty file sign for FAT fs
3462
                                 <1>
3463
                                 <1>
                                                 EBX <> FFFFFFFFh -> use file size for FAT fs
3464
                                 <1>
                                           ;
                                           ; OUTPUT ->
3465
                                 <1>
3466
                                  <1>
                                                 EAX = New file's first cluster
                                                  ESI = Logical Dos Drv Descr. Table Addr.
3467
                                 <1>
3468
                                 <1>
                                                 EBX = CreateFile_Size address
3469
                                 <1>
                                                 ECX = Sectors per cluster (<256)
                                                 EDX = Directory Entry Index/Number (<65536)
3470
                                 <1>
3471
                                 <1>
3472
                                 <1>
                                                 cf = 1 -> error code in AL (EAX)
3473
                                 <1>
3474 0000B293 E8EC050000
                                           call create_file
                                 <1>
3475
                                 <1>
                                           ;pop esi
                                                  csftdf2_rw_error
3476 0000B298 0F82A3050000
                                 <1>
                                            jс
3477
                                 <1>
3478
                                 <1> csftdf2_create_file_OK:
3479 0000B29E A3[A8640100]
                                 <1>
                                           mov [csftdf_df_cluster], eax
3480
                                 <1>
                                           ; 24/03/2016
                                 <1>
3482 0000B2A3 668915[66640100]
                                           mov [DestinationFile_DirEntryNumber], dx
                                 <1>
3483
                                 <1>
                                           ; 21/03/2016
3484
                                 <1>
3485 0000B2AA BE00000800
                                 <1>
                                           mov esi, Directory_Buffer
                                           shl
3486 0000B2AF C1E205
                                 <1>
                                                 edx, 5 ; 32 * index number
3487 0000B2B2 01D6
                                 <1>
                                           add
                                                 esi, edx
3488 0000B2B4 BF[3E640100]
                                 <1>
                                                 edi, DestinationFile_DirEntry
3489 0000B2B9 B108
                                 <1>
                                           mov
                                                cl, 8 ; 32 bytes
```

```
3490 0000B2BB F3A5
                                 <1>
                                           rep
                                                movsd
3491
                                 <1>
3492
                                 <1> csftdf2_set_sf_percentage:
3493
                                <1>
                                          ; 17/03/2016
3494 0000B2BD 31C0
                                <1>
                                           xor
                                                eax, eax
3495 0000B2BF A2[BC640100]
                                 <1>
                                           mov
                                                 [csftdf percentage], al; 0, reset
                                 <1>
3497 0000B2C4 A3[B4640100]
                                 <1>
                                                 [csftdf_sf_rbytes], eax ; 0, reset
                                          mov
3498 0000B2C9 A3[B8640100]
                                 <1>
                                                 [csftdf_df_wbytes], eax ; 0, reset
                                          mov
                                 <1>
3500 0000B2CE 8A25[6C630100]
                                 <1>
                                           mov
                                                 ah, [SourceFile_Drv]
3501 0000B2D4 BE00010900
                                 <1>
                                           mov
                                                 esi, Logical DOSDisks
3502 0000B2D9 01C6
                                 <1>
                                                 esi, eax
                                           add
3503
                                 <1>
3504 0000B2DB 8935[C0640100]
                                 <1>
                                                 [csftdf_sf_drv_dt], esi; 23/03/2016
                                          mov
3505
                                 <1>
3506 0000B2E1 668B15[D2630100]
                                 <1>
                                                 dx, [SourceFile_DirEntry+DirEntry_FstClusHI]
3507 0000B2E8 C1E210
                                 <1>
                                           shl
                                                 edx, 16
                                                 dx, [SourceFile DirEntry+DirEntry FstClusLO]
3508 0000B2EB 668B15[D8630100]
                                 <1>
                                           mov
3509 0000B2F2 8915[A4640100]
                                 <1>
                                                [csftdf_sf_cluster], edx
                                          mov
3510
                                 <1>
3511
                                 <1>
                                          ; 16/03/2016
3512
                                 <1>
                                          ; Note: Singlix FS boot sector parameters (for cluster
3513
                                 <1>
                                                 related calculations) has same offset
3514
                                 <1>
                                                 values from LD BPB as in FAT file system.
3515
                                 <1>
                                                 [esi+LD_BPB+SecPerClust] is 1 for Singlix FS.
                                 <1>
                                          movzx ecx, byte [esi+LD_BPB+SecPerClust]
3517 0000B2F8 0FB64E13
                                 <1>
3518 0000B2FC 880D[EA630100]
                                 <1>
                                                [SourceFile_SecPerClust], cl
3519
                                 <1>
                                           ; 17/03/2016
3520
                                 <1>
3521 0000B302 386E03
                                                [esi+LD FATType], ch ; 0
                                 <1>
                                           cmp
3522 0000B305 7707
                                 <1>
                                           jа
                                                 short csftdf2_set_sf_percent_rsize1
3523
                                 <1>
3524 0000B307 B800000100
                                 <1>
                                                 eax, 65536; read/write buffer size for Singlix FS
                                          mov
3525 0000B30C EB06
                                <1>
                                           jmp
                                                 short csftdf2_set_sf_percent_rsize2
                                <1>
                                <1> csftdf2_set_sf_percent_rsize1:
3527
3528 0000B30E 668B4611
                                <1>
                                          mov ax, [esi+LD_BPB+BytesPerSec]
3529 0000B312 F7E1
                                <1>
                                           mul
                                                ecx
                                           ; sub edx, edx
3530
                                <1>
                                 <1> csftdf2_set_sf_percent_rsize2:
3531
3532 0000B314 A3[AC640100]
                                <1>
                                          mov [csftdf_r_size], eax
3533
                                 <1>
3534
                                 <1> csftdf2_set_df_percentage:
3535
                                 <1>
                                        ;sub eax, eax
3536
                                 <1>
                                          ;mov ah, [DestinationFile Drv]
                                          ;mov edi, Logical_DOSDisks
3537
                                 <1>
3538
                                 <1>
                                           ;add edi, eax
3539
                                 <1>
                                          ;mov [csftdf_df_drv_dt], edi ; 17/03/2016
3540
                                 <1>
3541 0000B319 8B3D[C4640100]
                                 <1>
                                                edi, [csftdf_df_drv_dt] ; 23/03/2016
                                          mov
3542
                                 <1>
3543
                                 <1>
                                          ; 16/03/2016
3544
                                 <1>
                                          ; Note: Singlix FS boot sector parameters (for cluster
3545
                                 <1>
                                                 related calculations) has same offset
3546
                                                 values from LD BPB as in FAT file system.
                                          ;
3547
                                 <1>
                                                 [edi+LD_BPB+SecPerClust] is 1 for Singlix FS.
                                          ;
3548
                                 <1>
                                          ;movzx ecx, byte [edi+LD BPB+SecPerClust]
                                 <1>
3549
3550 0000B31F 8A4F13
                                 <1>
                                           mov cl, [edi+LD_BPB+SecPerClust]
3551 0000B322 880D[6A640100]
                                 <1>
                                          mov
                                                [DestinationFile SecPerClust], cl
3552
                                 <1>
3553
                                 <1>
                                          ; 17/03/2016
3554 0000B328 386F03
                                 <1>
                                          cmp [edi+LD_FATType], ch ; 0
3555 0000B32B 7707
                                 <1>
                                                 short csftdf2_set_df_percent_wsize1
                                           jа
                                 <1>
                                                 eax, 65536 ; read/write buffer size for Singlix FS
3557 0000B32D B800000100
                                <1>
                                          mov
3558 0000B332 EB06
                                 <1>
                                                 short csftdf2_set_df_percent_wsize2
                                           jmp
3559
                                <1>
3560
                                <1> csftdf2_set_df_percent_wsize1:
3561 0000B334 0FB74711
                                <1>
                                          movzx eax, word [edi+LD_BPB+BytesPerSec]
3562 0000B338 F7E1
                                <1>
                                           mul ecx
                                           ; sub edx, edx
3563
                                 <1>
3564
                                 <1> csftdf2_set_df_percent_wsize2:
3565 0000B33A A3[B0640100]
                                 <1>
                                          mov [csftdf_w_size], eax
                                 <1>
3567 0000B33F A1[98640100]
                                 <1>
                                           mov
                                                 eax, [csftdf_filesize]
                                 <1>
3568
3569 0000B344 3D00000100
                                 <1>
                                                 eax, 65536 ; 64KB ; small file
                                           cmp
3570 0000B349 721F
                                 <1>
                                           jb
                                                 short csftdf2 load file ; do not display percentage
                                 <1>
                                 <1> csftdf2_reset_wf_percent_ptr_chk_64k:
3573 0000B34B B201
                                          mov dl, 1; 25/03/2016
                                 <1>
3574
                                 <1>
3575 0000B34D 3D00000400
                                           cmp eax, 65536*4; 256KB
                                 <1>
3576 0000B352 7310
                                         jnb short csftdf2_enable_percentage_display ; big file
                                <1>
3577
                                 <1>
                                        ; 64-128KB file size for floppy disks
cmp byte [SourceFile_Drv], dl ; 1 ; read from floppy disk ?
3578
                                 <1>
3579 0000B354 3815[6C630100]
                                <1>
3580 0000B35A 7608
                                 <1>
                                                 short csftdf2_enable_percentage_display
3581
                                 <1>
3582 0000B35C 3815[EC630100]
                                 <1>
                                                 byte [DestinationFile_Drv], dl ; 1 ; write to floppy disk ?
                                           cmp
3583 0000B362 7706
                                <1>
                                                 short csftdf2_load_file
                                           jа
3584
                                 <1>
                                 <1> csftdf2_enable_percentage_display:
3585
3586 0000B364 8815[BC640100]
                                <1>
                                          mov [csftdf_percentage], dl; 1
3587
                                 <1>
3588
                                 <1> csftdf2_load_file:
                                       ; 13/05/2016
3589
                                 <1>
3590
                                 <1>
                                          ; 19/03/2016
                                        ; 18/03/2016
3591
                                 <1>
3592
                                <1>
                                          ; 17/03/2016
                                        mov ah, OFh
3593 0000B36A B40F
                                <1>
3594 0000B36C E83261FFFF
                                <1> call _int10h
```

```
; 13/05/2016
3595
                                <1>
3596 0000B371 883D[BD640100]
                                <1>
                                         mov
                                               [csftdf videopage], bh ; active video page
3598 0000B379 E82561FFFF
3597 0000B377 B403
                                               ah, 03h
                                <1>
                                         mov
                               <1>
                                         call int10h
3599 0000B37E 668915[BE640100]
                               <1>
                                         mov
                                               [csftdf_cursorpos], dx
3600
                                <1>
3601 0000B385 29C0
                                <1>
                                          sub eax, eax
3602 0000B387 A2[95640100]
                                <1>
                                         mov [csftdf_rw_err], al; 0
3603
                                <1>
3604
                                <1>; ///
                                <1> csftdf_sf_amb: ; 15/03/2016
3605
3606 0000B38C 8B0D[98640100]
                                <1>
                                         mov ecx, [csftdf filesize]
                                                                         ; 23/03/2016
3607
                                <1>
3608
                                <1>
                                         ; TRDOS 386 (TRDOS v2.0)
3609
                                <1>
                                         ; Allocate contiguous memory block for loading the file
3610
                                <1>
                                         ;mov ecx, [SourceFile DirEntry+DirEntry FileSize]
3611
                                <1>
                                <1>
3612
3613
                                <1>
                                          ; sub eax, eax ; First free memory aperture
3614
                                <1>
                                          ; eax = 0 (Allocate memory from the beginning)
3615
                                <1>
3616
                                <1>
                                         ; ecx = File (Allocation) size in bytes
3617
                                <1>
3618 0000B392 E8EAA0FFFF
                                <1>
                                          call allocate_memory_block
3619 0000B397 7304
                                <1>
                                          jnc short loc_check_sf_save_loading_parms
3620
                                <1>
3621 0000B399 29C0
                               <1>
                                          sub eax, eax
3622 0000B39B 29C9
                                         sub ecx, ecx
                                <1>
3623
                                <1>
                                <1> loc_check_sf_save_loading_parms:
3624
3625 0000B39D A3[9C640100]
                               <1>
                                         mov [csftdf_sf_mem_addr], eax ; loading address
3626 0000B3A2 890D[A0640100]
                                <1>
                                          mov
                                               [csftdf_sf_mem_bsize], ecx ; block size
                               <1> ; ///
3627
3628
                                <1>
                                       ; 19/03/2016
3629 0000B3A8 8B35[C0640100]
                                         mov esi, [csftdf sf drv dt]; logical dos drv desc. tbl.
                                <1>
3630
                                <1>
                                         ; 17/03/2016
3631
                                <1>
3632 0000B3AE 09C0
                                <1>
                                          or eax, eax; contiguous free memory block address
3633 0000B3B0 0F845B010000
                                <1>
                                                csftdf2_read_sf_cluster
                                          jΖ
                                <1>
3634
                                         ; 18/03/2016
3635
                                <1>
3636 0000B3B6 8B1D[9C640100]
                                <1>
                                        mov ebx, [csftdf_sf_mem_addr] ; memory block address
3637
                                <1>
3638 0000B3BC 807E0300
                                <1>
                                          cmp byte [esi+LD_FATType], 0
3639 0000B3C0 0F8605020000
                                <1>
                                         jna csftdf2_load_fs_file
3640
                                <1>
                                <1> csftdf2 load fat file:
3641
3642 0000B3C6 53
                                         push ebx; *
                                <1>
3643
                                <1>
3644
                                <1> csftdf2_load_fat_file_next:
3645 0000B3C7 BE[B1130100]
                                          mov esi, msg_reading
                               <1>
3646 0000B3CC E8FCAFFFFF
                                <1>
                                          call print msg
3647
                                <1>
3648 0000B3D1 803D[BC640100]00
                               <1>
                                         cmp byte [csftdf_percentage], 0
3649 0000B3D8 7605
                                <1>
                                               short csftdf2 load fat file 1
                                         jna
3650
                                <1>
3651 0000B3DA E87C000000
                               <1>
                                         call csftdf2 print percentage; 19/03/2016
3652
                                <1>
3653
                                <1> csftdf2_load_fat_file_1:
3654 0000B3DF 8B35[C0640100]
                               <1> mov esi, [csftdf_sf_drv_dt]
3655 0000B3E5 5B
                               <1>
                                         pop ebx; *
3656
                                <1>
                                <1> csftdf2_load_fat_file_2:
3657
3658 0000B3E6 E8B8000000
                               <1>
                                         call csftdf2_read_fat_file_sectors ; 19/03/2016
3659 0000B3EB 0F8250040000
                                <1>
                                         jc csftdf2_rw_error; eocc! or disk error!
3660
                                <1>
3661 0000B3F1 09D2
                                <1>
                                         or
                                                edx, edx; edx > 0 -> EOF
3662 0000B3F3 7520
                                <1>
                                               short csftdf2_load_fat_file_ok
                                         jnz
3663
                                <1>
3664 0000B3F5 803D[BC640100]00
                               <1>
                                          cmp
                                                byte [csftdf percentage], 0
                                                short csftdf2 load fat file 2
3665 0000B3FC 76E8
                                <1>
3666
                                <1>
3667 0000B3FE 53
                                         push ebx; *
                                <1>
3668
                                <1>
3669
                                <1>
                                         ; Set cursor position
                                         ; AH= 02h, BH= Page Number, DH= Row, DL= Column
3670
                                <1>
3671 0000B3FF 8A3D[BD640100]
                                <1>
                                          mov bh, [csftdf videopage]
                                          mov dx, [csftdf_cursorpos]
3672 0000B405 668B15[BE640100]
                                <1>
                                               ah, 2
3673 0000B40C B402
                                <1>
                                          mov
                                          call int10h
3674 0000B40E E89060FFFF
                                <1>
3675 0000B413 EBB2
                                <1>
                                          jmp short csftdf2_load_fat_file_next
3676
                                <1>
                                <1> csftdf2 load fat file ok:
3678 0000B415 803D[BC640100]00
                                          cmp byte [csftdf_percentage], 0
                                <1>
3679 0000B41C 0F8651020000
                                          jna csftdf2 save file ; 25/03/2016
                                <1>
3680
                                <1>
                                         ; "Reading... 100%"
                                <1>
3682 0000B422 BF[C9130100]
                                <1>
                                         mov edi, percentagestr
3683 0000B427 B031
                                <1>
                                               al, '1'
3684 0000B429 AA
                                <1>
                                          stosb
                                               al, '0'
3685 0000B42A B030
                                <1>
                                          mov
3686 0000B42C AA
                                <1>
                                          stosb
3687 0000B42D AA
                                <1>
                                          stosb
                                <1>
3689 0000B42E 8A3D[BD640100]
                                <1>
                                                bh, [csftdf_videopage]
                                         mov
3690 0000B434 668B15[BE640100]
                                <1>
                                          mov
                                                dx, [csftdf_cursorpos]
                                                ah, 2
3691 0000B43B B402
                                <1>
                                          mov
3692 0000B43D E86160FFFF
                                <1>
                                          call
                                                _int10h
3693
                                <1>
3694 0000B442 BE[B1130100]
                                <1>
                                                esi, msg_reading
                                          mov
3695 0000B447 E881AFFFFF
                                <1>
                                          call print_msg
                                <1>
3697 0000B44C BE[C9130100]
                                <1>
                                          mov
                                                esi, percentagestr
                                          call print_msg
3698 0000B451 E877AFFFFF
                                <1>
                                <1>
```

```
csftdf2_save_file ; 25/03/2016
                                            jmp
3701
                                 <1>
3702
                                 <1> csftdf2_print_percentage:
3703
                                <1>
                                       ; 09/12/2017
                                          ; 19/03/2016
3704
                                <1>
3705
                                <1>
                                          ; 18/03/2016
                                        mov al, 20h
3706 0000B45B B020
                                <1>
3707 0000B45D BF[C9130100]
                                <1>
                                        mov edi, percentagestr
3708 0000B462 AA
                                <1>
                                          stosb
3709 0000B463 AA
                                <1>
                                          stosb
3710 0000B464 A1[B4640100]
3711 0000B469 BA64000000
                                <1>
                                          mov eax, [csftdf_sf_rbytes]
                                <1>
                                          mov
                                                edx, 100
                                          mul
3712 0000B46E F7E2
                                <1>
                                                edx
3713 0000B470 8B0D[98640100]
                               <1>
                                          mov
                                                ecx, [csftdf_filesize]
3714 0000B476 F7F1
                                <1>
                                          div
                                                ecx
3715 0000B478 B10A
                                <1>
                                          mov
                                                cl, 10
3716 0000B47A F6F1
                                <1>
                                                cl
3717 0000B47C 80C430
                                          add
                                                ah, '0'
                                <1>
3718 0000B47F 8827
                                <1>
                                          mov
                                                 [edi], ah
3719 0000B481 20C0
                               <1>
                                          and
                                                al, al
3720 0000B483 740A
                               <1>
                                          jz
                                                short csftdf2_print_percent_1
3721 0000B485 4F
                                <1>
                                          dec
3722
                                <1>
                                          ;cbw
3723 0000B486 28E4
                                <1>
                                          sub
                                                ah, ah ; 09/12/2017
3724 0000B488 F6F1
                                <1>
                                          div
                                                cl
                                                ah, '0'
3725 0000B48A 80C430
                                <1>
                                          add
3726 0000B48D 8827
                                <1>
                                          mov
                                                [edi], ah
3727
                                <1>
                                          ;and al, al
3728
                                <1>
                                                short csftdf2_print_percent_1
                                          ;jz
3729
                                 <1>
                                          ;dec edi
3730
                                 <1>
                                          ;mov [edi], '1'; 100%
3731
                                 <1>
                                 <1> csftdf2_print_percent_1:
3732
3733 0000B48F BE[C9130100]
                                <1>
                                       mov esi, percentagestr
                                          ;call print_msg
3734
                                 <1>
3735
                                <1>
                                          ;retn
3736 0000B494 E934AFFFFF
                                          jmp print msg
                                <1>
3737
                                <1>
3738
                                <1> csftdf2_read_file_sectors:
                                      ; 19/03/201<del>6</del>
3739
                                <1>
3740 0000B499 807E0300
                                          cmp byte [esi+LD FATType], 0
                                <1>
                                         jna
3741 0000B49D 0F8627070000
                                <1>
                                                   csftdf2_read_fs_file_sectors
3742
                                <1>
3743
                                 <1> csftdf2_read_fat_file_sectors:
                                       ; 19/03/2016
3744
                                 <1>
3745
                                 <1>
                                          ; 18/03/2016
3746
                                 <1>
                                          ; return:
                                        ; CF = 0 \& EDX > 0 -> END OF FILE
; CF = 0 \& EDX = 0 -> not EOF
3747
                                 <1>
3748
                                 <1>
                                          ; CF = 1 -> read error (error code in AL)
3749
                                 <1>
3750
                                 <1>
3751
                                 <1> csftdf2 read fat file secs 0:
                                <1> mov edx, [csftdf_filesize]
3752 0000B4A3 8B15[98640100]
3753 0000B4A9 2B15[B4640100]
                                <1>
                                          sub
                                                edx, [csftdf_sf_rbytes]
                                          cmp
                                                edx, [csftdf_r_size]
3754 0000B4AF 3B15[AC640100]
                                <1>
3755 0000B4B5 7306
                                <1>
                                          jnb
                                                short csftdf2_read_fat_file_secs_1
3756 0000B4B7 8915[AC640100]
                                          mov [csftdf r size], edx
                                <1>
3757
                                <1>
3758
                                <1> csftdf2_read_fat_file_secs_1:
3759 0000B4BD A1[AC640100]
                                <1> mov eax, [csftdf_r_size]
3760 0000B4C2 29D2
                                <1>
                                          sub edx, edx
3761 0000B4C4 0FB74E11
                                <1>
                                          movzx ecx, word [esi+LD BPB+BytesPerSec]
                                <1>
3762 0000B4C8 01C8
                                          add eax, ecx
3763 0000B4CA 48
                                <1>
                                          dec
3764 0000B4CB F7F1
                                <1>
                                          div
                                                ecx
3765 0000B4CD 89C1
                                <1>
                                          mov
                                                 ecx, eax; sector count
3766 0000B4CF A1[A4640100]
                                <1>
                                          mov eax, [csftdf_sf_cluster]
3767
                                <1>
3768
                                <1>
                                          ; EBX = memory block address (current)
3769
                                <1>
3770 0000B4D4 E821090000
                                          call read fat file sectors
                                <1>
3771 0000B4D9 7235
                                 <1>
                                          jс
                                                short csftdf2_read_fat_file_secs_3
3772
                                <1>
3773
                                <1>
                                          ; EBX = next memory address
3774
                                <1>
3775 0000B4DB A1[B4640100]
                                                 eax, [csftdf_sf_rbytes]
                                <1>
                                          mov
3776 0000B4E0 0305[AC640100]
                                                eax, [csftdf r size]
                                <1>
                                          add
3777 0000B4E6 8B15[98640100]
                                <1>
                                          mov
                                                edx, [csftdf_filesize]
3778 0000B4EC 39D0
                                 <1>
                                          cmp
                                                 eax, edx
                                                short csftdf2_read_fat_file_secs_3 ; edx > 0
3779 0000B4EE 7320
                                 <1>
                                          jnb
3780 0000B4F0 A3[B4640100]
                                 <1>
                                          mov
                                                [csftdf_sf_rbytes], eax
3781
                                 <1>
                                          push ebx ; *
3782 0000B4F5 53
                                <1>
3783
                                <1>
                                          ; get next cluster (csftdf_r_size! bytes)
3784 0000B4F6 A1[A4640100]
                                <1>
                                                eax, [csftdf_sf_cluster]
                                          mov
3785 0000B4FB E8CC060000
                                <1>
                                          call
                                               get_next_cluster
3786 0000B500 5B
                                          pop ebx; *
                                <1>
3787 0000B501 7306
                                <1>
                                                short csftdf2_read_fat_file_secs_2
                                          jnc
3788
                                <1>
3789
                                          : 15/10/2016
                                <1>
3790
                                <1>
                                          ;Disk read error instad of drv not ready err
3791 0000B503 B811000000
                                          mov eax, 17; Read error!
                                <1>
3792 0000B508 C3
                                <1>
                                          retn
3793
                                <1>
3794
                                <1> csftdf2_read_fat_file_secs_2:
3795 0000B509 29D2
                                <1>
                                          sub edx, edx; 0
3796 0000B50B A3[A4640100]
                                <1>
                                          mov [csftdf_sf_cluster], eax ; next cluster
                                <1>
3798
                                 <1> csftdf2_read_fat_file_secs_3:
3799 0000B510 C3
                                <1>
                                          retn
3800
                                <1>
3801
                                 <1> csftdf2 read sf cluster:
3802
                                          ; 19/03/2016
                                <1>
3803 0000B511 BB00000700
                                 <1>
                                          mov ebx, Cluster_Buffer ; buffer address (64KB)
                                 <1>
3804
```

3700 0000B456 E918020000

<1>

```
3805 0000B516 803D[BC640100]00
                                                 byte [csftdf percentage], 0
                                <1>
                                           cmp
3806 0000B51D 760D
                                 <1>
                                                 short csftdf2 read sf clust 2
                                           jna
3807
                                 <1>
3808 0000B51F 53
                                 <1>
                                          push ebx; *
3809
                                 <1>
3810
                                 <1> csftdf2_read_sf_clust_next:
3811 0000B520 E836FFFFFF
                                          call csftdf2_print_percentage
                                 <1>
3812
                                 <1>
3813
                                 <1> csftdf2_read_sf_clust_0:
3814 0000B525 8B35[C0640100]
                                         mov esi, [csftdf sf drv dt]
                                 <1>
3815
                                 <1> csftdf2_read_sf_clust_1:
3816 0000B52B 5B
                                 <1>
                                          pop ebx ; *
3817
                                 <1>
3818
                                 <1> csftdf2_read_sf_clust_2:
3819 0000B52C 89DA
                                 <1>
                                          mov edx, ebx
3820 0000B52E 0315[AC640100]
                                                 edx, [csftdf_r_size]
                                <1>
                                           add
3821 0000B534 81FA00000800
                                                 edx, Cluster Buffer + 65536
                                <1>
                                           cmp
3822 0000B53A 772F
                                 <1>
                                                 short csftdf2_write_df_cluster
                                          jа
3823
                                 <1>
3824 0000B53C E858FFFFF
                                <1>
                                           call csftdf2 read file sectors; 19/03/2016
                                                     csftdf2_save_fat_file_err2 ; eocc! or disk error!
3825 0000B541 0F8280020000
                                <1>
                                 <1>
3827 0000B547 09D2
                                                  edx, edx ; edx > 0 -> EOF
                                 <1>
                                           or
3828 0000B549 7520
                                 <1>
                                           jnz
                                                  short csftdf2_write_df_cluster
3829
                                 <1>
3830 0000B54B 803D[BC640100]00
                                 <1>
                                           cmp
                                                 byte [csftdf_percentage], 0
                                                 short csftdf2_read_sf_clust_2
3831 0000B552 76D8
                                 <1>
                                           jna
3832
                                 <1>
3833 0000B554 53
                                 <1>
                                          push ebx; *
3834
                                 <1>
3835
                                 <1>
                                          ; Set cursor position
3836
                                 <1>
                                          ; AH= 02h, BH= Page Number, DH= Row, DL= Column
3837 0000B555 8A3D[BD640100]
                                          mov bh, [csftdf_videopage]
                                 <1>
3838 0000B55B 668B15[BE640100]
                                <1>
                                          mov dx, [csftdf_cursorpos]
                                                ah, 2
_int10h
3839 0000B562 B402
                                 <1>
                                          mov
3840 0000B564 E83A5FFFFF
                                 <1>
                                           call
3841 0000B569 EBB5
                                 <1>
                                          jmp short csftdf2_read_sf_clust_next
3842
                                 <1>
3843
                                 <1> csftdf2_write_df_cluster:
                                      ; 19/03/2016
3844
                                <1>
3845 0000B56B 8B35[C4640100]
                                <1>
                                           mov esi, [csftdf_df_drv_dt]
3846 0000B571 BB00000700
                                 <1>
                                          mov
                                                 ebx, Cluster_Buffer; buffer address (64KB)
3847
                                 <1>
3848
                                 <1> csftdf2_write_df_clust_next:
                                 <1> call csftdf2 write file sectors ; 19/03/2016
3849 0000B576 E855000000
                                                     csftdf2_save_fat_file_err2 ; eocc! or disk error!
3850 0000B57B 0F8246020000
                                 <1>
                                 <1>
3852 0000B581 09D2
                                 <1>
                                          or
                                                  edx, edx; edx > 0 \rightarrow EOF
3853 0000B583 750A
                                 <1>
                                          jnz
                                                 short csftdf2_rw_f_clust_ok
3854
                                 <1>
3855 0000B585 81FB00000800
                                                  ebx, Cluster_Buffer + 65536
                                 <1>
                                           cmp
3856 0000B58B 72E9
                                 <1>
                                           jb
                                                 short csftdf2_write_df_clust_next
3857
                                 <1>
3858 0000B58D EB82
                                 <1>
                                                 short csftdf2_read_sf_cluster
3859
                                 <1>
3860
                                 <1> csftdf2_rw_f_clust_ok:
3861 0000B58F 803D[BC640100]00
                                      cmp byte [csftdf percentage], 0
                                <1>
3862 0000B596 0F86B2010000
                                 <1>
                                                    csftdf2_save_fat_file_4 ; 25/03/2016
3863
                                 <1>
                                           ; "100%"
3864
                                <1>
3865 0000B59C BF[C9130100]
                                <1>
                                           mov edi, percentagestr
3866 0000B5A1 B031
                                 <1>
                                          mov
                                                al, '1'
3867 0000B5A3 AA
                                <1>
                                           stosb
3868 0000B5A4 B030
                                <1>
                                                al, '0'
                                          mov
3869 0000B5A6 AA
                                 <1>
                                          stosb
3870 0000B5A7 AA
                                 <1>
                                           stosb
                                 <1>
3872 0000B5A8 8A3D[BD640100]
                                                 bh, [csftdf_videopage]
                                <1>
                                           mov
3873 0000B5AE 668B15[BE640100]
                                 <1>
                                                 dx, [csftdf_cursorpos]
                                           mov
3874 0000B5B5 B402
                                 <1>
                                                 ah, 2
                                           mov
3875 0000B5B7 E8E75EFFFF
                                 <1>
                                                _int10h
3876
                                 <1>
3877 0000B5BC BE[C9130100]
                                 <1>
                                                 esi, percentagestr
                                           mov
3878 0000B5C1 E807AEFFFF
                                 <1>
                                           call print_msg
3879
                                 <1>
3880 0000B5C6 E983010000
                                 <1>
                                            jmp
                                                     csftdf2_save_fat_file_4
3881
                                 <1>
3882
                                 <1> csftdf2 load fs file:
3883
                                 <1>
                                        ; temporary - 18/03/2016
3884 0000B5CB E96F020000
                                            jmp csftdf2_read_error
                                 <1>
3885
                                 <1>
3886
                                 <1> csftdf2_write_file_sectors:
                                           ; 19/03/2016
                                 <1>
3888 0000B5D0 807E0300
                                 <1>
                                           cmp byte [esi+LD_FATType], 0
3889 0000B5D4 0F86F1050000
                                 <1>
                                           jna csftdf2 write fs file sectors
3890
                                 <1>
                                 <1> csftdf2 write_fat_file_sectors:
3891
                                         ; 19/03/201<del>6</del>
3892
                                 <1>
3893
                                 <1>
                                           ; 18/03/2016
3894
                                 <1>
                                           : return:
                                           ; CF = 0 \& EDX > 0 \rightarrow END OF FILE
3895
                                 <1>
                                          ; CF = 0 & EDX = 0 -> not EOF
; CF = 1 -> write error (error code in AL)
3896
                                 <1>
3897
                                 <1>
3898
                                 <1>
3899
                                 <1> csftdf2 write fat file secs 0:
                                                edx, [csftdf_filesize]
3900 0000B5DA 8B15[98640100]
                                 <1>
                                           mov
                                                 edx, [csftdf df wbytes]
3901 0000B5E0 2B15[B8640100]
                                 <1>
                                           sub
3902 0000B5E6 3B15[B0640100]
                                                 edx, [csftdf w size]
                                 <1>
                                           cmp
3903 0000B5EC 7306
                                 <1>
                                           jnb
                                                 short csftdf2_write_fat_file_secs_1
3904 0000B5EE 8915[B0640100]
                                 <1>
                                           mov
                                                [csftdf_w_size], edx
3905
                                 <1>
3906
                                 <1> csftdf2_write_fat_file_secs_1:
3907 0000B5F4 A1[B0640100]
                                           mov eax, [csftdf w size]
                                <1>
3908 0000B5F9 29D2
                                 <1>
                                           sub edx, edx
                                           movzx ecx, word [esi+LD_BPB+BytesPerSec]
3909 0000B5FB 0FB74E11
                                 <1>
```

```
3910 0000B5FF 01C8
                                <1>
                                          add
                                                eax, ecx
3911 0000B601 48
                                <1>
                                          dec
                                                eax
3912 0000B602 F7F1
                                <1>
                                          div
                                                ecx
                                                 ecx, eax ; sector count
3913 0000B604 89C1
                                <1>
                                          mov
3914 0000B606 A1[A8640100]
                                <1>
                                                eax, [csftdf_df_cluster]
                                          mov
3915
                                <1>
3916
                                 <1>
                                          ; EBX = memory block address (current)
3917
                                 <1>
3918 0000B60B E8A20F0000
                                <1>
                                          call write_fat_file_sectors
                                          jс
3919 0000B610 7259
                                 <1>
                                                short csftdf2_write_fat_file_secs_4
3920
                                 <1>
3921
                                 <1>
                                          ; EBX = next memory address
3922
                                 <1>
                                                eax, [csftdf df wbytes]
3923 0000B612 A1[B8640100]
                                <1>
                                          mov
3924 0000B617 0305[B0640100]
                                <1>
                                                eax, [csftdf_w_size]
                                          add
3925 0000B61D 8B15[98640100]
                                <1>
                                          mov
                                                 edx, [csftdf_filesize]
3926 0000B623 39D0
                                 <1>
                                          cmp
                                                eax, edx
3927 0000B625 7344
                                                short csftdf2_write_fat_file_secs_4
                                 <1>
                                          jnb
                                                [csftdf df wbytes], eax
3928 0000B627 A3[B8640100]
                                 <1>
                                          mov
3929
                                 <1>
3930 0000B62C A3[5A640100]
                                 <1>
                                          mov
                                                 [DestinationFile_DirEntry+DirEntry_FileSize], eax
3931
                                 <1>
3932 0000B631 53
                                          push ebx; *
                                 <1>
3933
                                 <1>
3934 0000B632 803D[96640100]01
                                <1>
                                                byte [DestinationFileFound], 1
                                          cmp
3935 0000B639 7210
                                 <1>
                                          jb
                                                 short csftdf2_write_fat_file_secs_2
                                 <1>
3937
                                <1>
                                          ; get next cluster (csftdf_w_size! bytes)
3938 0000B63B A1[A8640100]
                                <1>
                                                eax, [csftdf_df_cluster]
                                          mov
3939 0000B640 E887050000
                                <1>
                                          call get next cluster
3940 0000B645 731C
                                <1>
                                                short csftdf2_write_fat_file_secs_3
3941
                                 <1>
3942 0000B647 21C0
                                                 eax, eax; end of cluster chain!?
                                <1>
                                          and
3943 0000B649 7521
                                <1>
                                                short csftdf2_write_fat_file_secs_5 ; disk error !
3944
                                <1>
                                <1> csftdf2 write fat file secs 2:
3945
3946 0000B64B A1[A8640100]
                                <1>
                                          mov eax, [csftdf df cluster]; last cluster
3947 0000B650 E8800E0000
                                <1>
                                          call add_new_cluster
3948 0000B655 7215
                                <1>
                                                 short csftdf2_write_fat_file_secs_5
3949
                                <1>
3950
                                <1>
                                          ; NOTE: Destination file size may be bigger than
                                          ; source file size when the last reading fails after here.
3951
                                 <1>
                                          ; (The last -empty- cluster of destination file must be
3952
                                 <1>
                                          ; truncated and LMDT must be current date&time for partial
3953
                                 <1>
3954
                                 <1>
                                          ; copy result!)
3955 0000B657 8B15[B0640100]
                                 <1>
                                          mov edx, [csftdf_w_size] ; bytes per cluster
3956 0000B65D 0115[5A640100]
                                 <1>
                                          add
                                               [DestinationFile_DirEntry+DirEntry_FileSize], edx
3957
                                 <1>
3958
                                <1> csftdf2_write_fat_file_secs_3:
                                          pop ebx; *
3959 0000B663 5B
                                <1>
3960 0000B664 29D2
                                <1>
                                          sub
                                                edx, edx ; 0
3961 0000B666 A3[A8640100]
                                <1>
                                          mov
                                                [csftdf_df_cluster], eax ; next cluster
3962
                                <1>
3963
                                 <1> csftdf2_write_fat_file_secs_4:
3964 0000B66B C3
                                 <1>
                                          retn
3965
                                 <1>
                                 <1> csftdf2_write_fat_file_secs_5:
3966
                                      pop ebx; *
3967 0000B66C 5B
                                 <1>
3968
                                 <1>
                                          ; 16/10/2016 (1Dh -> 18)
                                          mov eax, 18; Write error!
3969 0000B66D B812000000
                                <1>
3970 0000B672 C3
                                <1>
3971
                                 <1>
3972
                                 <1> csftdf2_save_file:
                                        ; 09/1<del>2</del>/2017
3973
                                 <1>
                                          ; 25/03/2016
3974
                                 <1>
3975
                                 <1>
                                          ; 19/03/2016
                                 <1>
                                          ; 18/03/2016
3977 0000B673 8B35[C4640100]
                                                esi, [csftdf_df_drv_dt]; logical dos drv desc. tbl.
                                 <1>
                                          mov
3978
                                 <1>
3979 0000B679 8B1D[9C640100]
                                 <1>
                                                 ebx, [csftdf_sf_mem_addr] ; memory block address
                                          mov
3980
                                 <1>
3981 0000B67F 807E0300
                                 <1>
                                          cmp byte [esi+LD FATType], 0
3982 0000B683 0F86F4010000
                                 <1>
                                           jna
                                                   csftdf2_save_fs_file
3983
                                 <1>
3984
                                 <1> csftdf2_save_fat_file:
3985 0000B689 53
                                          push ebx; *
                                 <1>
                                 <1>
3987 0000B68A 803D[BC640100]00
                                 <1>
                                          cmp
                                               byte [csftdf_percentage], 0
                                                short csftdf2 save fat file 0
3988 0000B691 7724
                                 <1>
                                          jа
3989
                                 <1>
3990
                                 <1>
                                          ; Set cursor position
3991
                                 <1>
                                          ; AH= 02h, BH= Page Number, DH= Row, DL= Column
3992 0000B693 8A3D[BD640100]
                                                 bh, [csftdf_videopage]
                                <1>
                                          mov
3993 0000B699 668B15[BE640100]
                                 <1>
                                                 dx, [csftdf_cursorpos]
3994 0000B6A0 B402
                                 <1>
                                                ah, 2
                                          mov
3995 0000B6A2 E8FC5DFFFF
                                 <1>
                                          call
                                                _int10h
                                <1>
3997 0000B6A7 BE[BD130100]
                                <1>
                                                esi, msg_writing
                                         mov
3998 0000B6AC E81CADFFFF
                                <1>
                                          call print_msg
                                 <1>
4000
                                <1> csftdf2_save_fat_file_next:
4001 0000B6B1 8B35[C4640100]
                                          mov esi, [csftdf df drv dt]; 25/03/2016
                                <1>
                                <1>
4002
4003
                                 <1> csftdf2_save_fat_file_0:
                                          pop ebx; *
4004 0000B6B7 5B
                                <1>
4005
                                <1>
                                <1> csftdf2 save fat_file_1:
                                          call csftdf2_write_file_sectors; 19/03/2016
4007 0000B6B8 E813FFFFFF
                                <1>
4008 0000B6BD 0F827E010000
                                <1>
                                           jс
                                                 csftdf2_rw_error ; eocc! or disk error!
                                <1>
4010 0000B6C3 09D2
                                          or edx, edx; edx > 0 -> EOF
                                <1>
                                         jnz short csftdf2_save_fat_file 3; 25/03/2016
4011 0000B6C5 756D
                                 <1>
4012
                                <1>
4013 0000B6C7 803D[BC640100]00
                                <1>
                                          cmp byte [csftdf_percentage], 0
4014 0000B6CE 76E8
                                 <1>
                                          jna short csftdf2 save fat file 1
```

```
4015
4016 0000B6D0 B020
                                 <1>
                                                 al, 20h
                                          mov
4017 0000B6D2 BF[C9130100]
                                                edi, percentagestr
                                <1>
                                          mov
4018 0000B6D7 AA
                                <1>
                                          stosb
4019 0000B6D8 AA
                                <1>
                                          stosb
4020 0000B6D9 A1[B8640100]
                                <1>
                                          mov
                                                eax, [csftdf_df_wbytes]
4021 0000B6DE BA64000000
                                <1>
                                                edx, 100
                                          mov
4022 0000B6E3 F7E2
                                          mul
                                <1>
                                                edx
4023 0000B6E5 8B0D[98640100]
                                <1>
                                                ecx, [csftdf_filesize]
                                          mov
4024 0000B6EB F7F1
                                <1>
                                          div
                                                ecx
4025 0000B6ED B10A
                                <1>
                                          mov
                                                cl, 10
4026 0000B6EF F6F1
                                <1>
                                          div
                                                cl
4027 0000B6F1 80C430
                                                ah, '0'
                                <1>
                                          add
4028 0000B6F4 8827
                                <1>
                                          mov
                                                [edi], ah
4029 0000B6F6 20C0
                                <1>
                                          and
                                                al, al
4030 0000B6F8 740A
                                <1>
                                          jг
                                                 short csftdf2_save_fat_file_2
4031 0000B6FA 4F
                                <1>
                                          dec
                                          ;cbw
4032
                                <1>
4033 0000B6FB 30E4
                                <1>
                                                ah, ah ; 09/12/2017
                                          xor
4034 0000B6FD F6F1
                                <1>
                                          div
                                                cl
                                                ah, '0'
4035 0000B6FF 80C430
                                <1>
                                          add
4036 0000B702 8827
                                <1>
                                          mov
                                                [edi], ah
                                          ;and al, al
4037
                                <1>
4038
                                 <1>
                                          ;jz
                                                short csftdf2_save_fat_file_2
4039
                                 <1>
                                          ;dec edi
                                                 [edi], '1' ; 100%
4040
                                 <1>
                                          ;mov
4041
                                 <1>
                                 <1> csftdf2_save_fat_file_2:
4042
4043 0000B704 53
                                 <1>
                                          push ebx; *
4044
                                 <1>
4045 0000B705 E802000000
                                 <1>
                                           call csftdf2_print_wr_percentage ; 25/03/2016
4046
                                 <1>
4047 0000B70A EBA5
                                 <1>
                                            jmp
                                                    csftdf2_save_fat_file_next
4048
                                 <1>
4049
                                 <1> csftdf2_print_wr_percentage:
4050
                                 <1>
                                          ; Set cursor position
                                          ; AH= 02h, BH= Page Number, DH= Row, DL= Column
                                 <1>
4052 0000B70C 8A3D[BD640100]
                                 <1>
                                          mov bh, [csftdf_videopage]
                                                dx, [csftdf cursorpos]
4053 0000B712 668B15[BE640100]
                                <1>
                                          mov
                                          mov ah, 2
4054 0000B719 B402
                                <1>
4055 0000B71B E8835DFFFF
                                 <1>
                                          call _int10h
4056
                                 <1>
4057 0000B720 BE[BD130100]
                                 <1>
                                                 esi, msg_writing
                                          mov
4058 0000B725 E8A3ACFFFF
                                 <1>
                                          call print_msg
4059
                                 <1>
4060 0000B72A BE[C9130100]
                                 <1>
                                          mov
                                                 esi, percentagestr
4061
                                 <1>
                                          ;call print_msg
4062
                                 <1>
                                          ;retn
4063 0000B72F E999ACFFFF
                                 <1>
                                          jmp print_msg
                                 <1>
4064
4065
                                 <1> csftdf2_save_fat_file_3:
4066 0000B734 803D[BC640100]00
                                <1>
                                       cmp byte [csftdf_percentage], 0
4067 0000B73B 7611
                                <1>
                                           jna
                                                   csftdf2_save_fat_file_4 ; 25/03/2016
4068
                                <1>
4069
                                <1>
                                          ; "100%"
4070 0000B73D BF[C9130100]
                                <1>
                                          mov edi, percentagestr
                                                al, '1'
4071 0000B742 B031
                                <1>
                                          mov
4072 0000B744 AA
                                <1>
                                          stosb
4073 0000B745 B030
                                <1>
                                                al, '0'
                                          mov
4074 0000B747 AA
                                <1>
                                          stosb
4075 0000B748 AA
                                <1>
                                           stosb
                                 <1>
4077 0000B749 E8BEFFFFFF
                                <1>
                                          call csftdf2_print_wr_percentage
4078
                                 <1>
                                 <1> csftdf2_save_fat_file_4:
4079
4080 0000B74E 803D[96640100]00
                                <1>
                                                byte [DestinationFileFound], 0
                                          cmp
4081 0000B755 7647
                                <1>
                                                short csftdf2_save_fat_file_6
                                          jna
4082
                                 <1>
4083 0000B757 8B35[C4640100]
                                <1>
                                                 esi, [csftdf_df_drv_dt] ; 31/03/2016
4084
                                 <1>
4085 0000B75D A1[A8640100]
                                 <1>
                                          mov
                                                 eax, [csftdf_df_cluster] ; last cluster
4086 0000B762 E865040000
                                 <1>
                                          call
                                                 get next cluster
4087 0000B767 7235
                                 <1>
                                                 short csftdf2_save_fat_file_6 ; eocc! or disk error!
                                          jс
4088
                                 <1>
4089 0000B769 A1[A8640100]
                                 <1>
                                                eax, [csftdf_df_cluster] ; last cluster
                                          mov
4090
                                 <1>
                                          ;xor ecx, ecx
                                          ;mov [FAT ClusterCounter], ecx; 0; reset
4091
                                 <1>
4092
                                 <1>
                                          ;dec ecx ; OFFFFFFFh
                                          ;shr ecx, 4; 28 bit; 0fffffffh
mov ecx, 0fffffffh
4093
                                 <1>
4094 0000B76E B9FFFFFF0F
                                 <1>
4095 0000B773 E87E070000
                                 <1>
                                          call update_cluster
4096 0000B778 7224
                                 <1>
                                           jc
                                                 short csftdf2_save_fat_file_6 ; really last cluster!?
                                 <1>
4098 0000B77A A3[A8640100]
                                                 [csftdf df cluster], eax; next cluster
                                 <1>
4099
                                 <1>
                                          ; byte [FAT_BuffValidData] = 2
4100
                                 <1>
4101 0000B77F E82F0A0000
                                 <1>
                                          call save fat buffer
4102 0000B784 730E
                                          jnc short csftdf2_save_fat_file_5
                                 <1>
                                 <1>
4103
4104 0000B786 8B15[98640100]
                                <1>
                                                 edx, [csftdf_filesize]
                                          mov
                                                 [DestinationFile DirEntry+DirEntry FileSize], edx
4105 0000B78C 8915[5A640100]
                                 <1>
                                          mov
4106 0000B792 EB58
                                 <1>
                                          jmp
                                                short csftdf2_save_fat_file_err3
                                 <1>
4107
4108
                                 <1> csftdf2_save_fat_file_5:
                                          mov eax, [csftdf_df_cluster]
4109 0000B794 A1[A8640100]
                                <1>
4110
                                 <1>
                                          ; EAX = First cluster to be truncated/unlinked
4111
                                 <1>
4112
                                 <1>
                                          ; ESI = Logical dos drive description table address
4113 0000B799 E8580C0000
                                 <1>
                                          call truncate_cluster_chain
4114
                                <1>
4115
                                <1> csftdf2_save_fat_file_6:
4116
                                <1>
                                          ; 28/03/2016
4117 0000B79E BE[C9630100]
                                <1>
                                          mov esi, SourceFile_DirEntry+DirEntry_Attr ; +11 to + 18
4118 0000B7A3 BF[49640100]
                                <1>
                                          mov edi, DestinationFile_DirEntry+DirEntry_Attr ; +11 to + 18
                                 <1>
                                          movsb : +11
4119 0000B7A8 A4
```

```
4120 0000B7A9 A5
                                          movsd ; +12 .. +15
                                <1>
4121 0000B7AA 66A5
                                <1>
                                          movsw ; +16 .. +17
                                          ; + 18
                                <1>
4123 0000B7AC 83C604
                                <1>
                                          add esi, 4
4124 0000B7AF 83C704
                                <1>
                                          add edi, 4
4125 0000B7B2 A5
                                <1>
                                          movsd ; DirEntry_WrtTime ; +22 .. +25
                                <1>
4127 0000B7B3 8B15[98640100]
                                                 edx, [csftdf_filesize]
                                <1>
                                          mov
4128 0000B7B9 8915[5A640100]
                                <1>
                                                [DestinationFile_DirEntry+DirEntry_FileSize], edx
                                          mov
                                <1>
4130 0000B7BF E8BAF0FFFF
                                <1>
                                          call convert_current_date_time
                                          ; DX = Date in dos dir entry format
4131
                                 <1>
4132
                                <1>
                                          ; AX = Time in dos dir entry format
4133 0000B7C4 EB4D
                                <1>
                                          jmp short csftdf2_save_fat_file_7
                                <1>
4134
4135
                                <1> csftdf2_save_fat_file_err1:
4136 0000B7C6 5B
                                <1>
                                         pop ebx; *
                                <1> csftdf2_save_fat_file_err2:
4137
4138 0000B7C7 A1[B8640100]
                                <1>
                                          mov eax, [csftdf_df_wbytes]
4139 0000B7CC 8B15[5A640100]
                                <1>
                                          mov edx, [DestinationFile_DirEntry+DirEntry_FileSize]
                                          cmp edx, eax
4140 0000B7D2 39C2
                                <1>
                                jna short csftdf2_save_fat_file_err3
mov eax, [csftdf_df_cluster] ; last (empty) cluster
4141 0000B7D4 7616
4142 0000B7D6 A1[A8640100]
                                          ; ESI = Logical dos drive description table address
4144 0000B7DB E8160C0000
                                          call truncate_cluster_chain
                                                short csftdf2 save fat file err3
4145 0000B7E0 720A
                                          jс
4146 0000B7E2 A1[B8640100]
                                          mov eax, [csftdf df wbytes]
4147 0000B7E7 A3[5A640100]
                                         mov [DestinationFile_DirEntry+DirEntry_FileSize], eax
                                <1>
                                 <1> csftdf2_save_fat_file_err3:
4148
4149 0000B7EC E88DF0FFFF
                                <1> call convert current date time
4150
                                 <1>
                                          ; DX = Date in dos dir entry format
4151
                                 <1>
                                          ; AX = Time in dos dir entry format
                                     mov
mov
mov
mov
4152 0000B7F1 C605[4B640100]00
                                          mov byte [DestinationFile_DirEntry+DirEntry_CrtTimeTenth], 0
                                <1>
4153 0000B7F8 66A3[4C640100]
                                <1>
                                      mov [DestinationFile_DirEntry+DirEntry_CrtTime], ax
                                               [DestinationFile_DirEntry+DirEntry_CrtDate], dx
4154 0000B7FE 668915[4E640100]
                                <1>
4155 0000B805 66A3[54640100]
                                                [DestinationFile DirEntry+DirEntry WrtTime], ax
                                <1>
4156 0000B80B 668915[56640100]
                                <1>
                                                [DestinationFile_DirEntry+DirEntry_WrtDate], dx
4157 0000B812 F9
                                 <1>
4158
                                 <1> csftdf2_save_fat_file_7:
4159 0000B813 9C
                                <1> pushf
                                | Push
| mov
| <1> mov
| <1> mov
| <1> mov
4160 0000B814 668915[50640100]
                                          mov [DestinationFile_DirEntry+DirEntry_LastAccDate], dx
4161 0000B81B BE[3E640100]
                                                esi, DestinationFile DirEntry
                                          mov edi, Directory Buffer
4162 0000B820 BF00000800
                                          movzx ecx, word [DestinationFile DirEntryNumber]; (<2048)
4163 0000B825 0FB70D[66640100] <1>
4164 0000B82C 66C1E105
                                <1>
                                          shl cx, 5; 32 * directory entry number
4165 0000B830 01CF
                                <1>
                                          add
                                                edi, ecx
                                          ;mov ecx, 8
4166
                                <1>
4167 0000B832 66B90800
                                <1>
                                          mov
                                                cx, 8
4168 0000B836 F3A5
                                <1>
                                          rep
                                                movsd
4169 0000B838 9D
                                <1>
                                          popf
4170 0000B839 730B
                                <1>
                                          jnc short csftdf2_write_file_OK
4171
                                <1>
4172
                                <1> csftdf2_write_error:
4173
                                <1> ; 18/03/2016
4174 0000B83B B01D
                                <1>
                                          mov al, 1Dh ; write error
4175 0000B83D EB02
                                <1>
                                               short csftdf2_rw_error
                                          jmp
4176
                                <1>
                                          ; 16/03/2016
4177
                                <1>
4178
                                 <1> csftdf2_read_error:
4179 0000B83F B011
                                <1> mov al, 17; Drive not ready or read error!
4180
                                 <1> csftdf2_rw_error:
4181 0000B841 A2[95640100]
                                <1>
                                          mov [csftdf_rw_err], al
4182
                                 <1>
4183
                                 <1> csftdf2_write_file_OK:
                                      ; 18/03/2016
                                 <1>
4184
                                          mov byte [DirBuff_ValidData], 2
4185 0000B846 C605[AC600100]02
                                <1>
4186 0000B84D E8CAF0FFFF
                                 <1>
                                         call save_directory_buffer
                                 <1>
4187
4188
                                 <1>
                                          ; Update last modification date&time of destination
4189
                                 <1>
                                          ; file's (parent) directory
4190 0000B852 E860F1FFFF
                                 <1>
                                          call update_parent_dir_lmdt
4191
                                 <1>
4192 0000B857 A1[9C640100]
                                                 eax, [csftdf_sf_mem_addr] ; start address
                                 <1>
                                          mov
4193
                                 <1>
4194 0000B85C 21C0
                                 <1>
                                          and
                                                eax, eax
4195 0000B85E 750E
                                                short csftdf2_dealloc_mblock
                                 <1>
                                          jnz
                                 <1>
4197 0000B860 88C5
                                 <1>
                                          mov
                                                ch, al ; 0 (Cluster r/w, not full loading)
4198
                                 <1> csftdf2_dealloc_retn:
4199 0000B862 8A0D[95640100]
                                     mov cl, [csftdf_rw_err]
                                <1>
4200 0000B868 A1[A8640100]
                                 <1>
                                          mov
                                                eax, [csftdf_df_cluster]
4201 0000B86D C3
                                 <1>
                                          retn
4202
                                 <1>
4203
                                 <1> csftdf2_dealloc_mblock:
4204 0000B86E 8B0D[A0640100]
                                          mov ecx, [csftdf sf mem bsize]; block size
                                 <1>
                                          call deallocate memory block
4205 0000B874 E8159EFFFF
                                 <1>
                                           mov ch, OFFh; (File was full loaded at memory)
4206 0000B879 B5FF
                                 <1>
4207 0000B87B EBE5
                                 <1>
                                          jmp short csftdf2_dealloc_retn
4208
                                 <1>
4209
                                 <1> csftdf2 save fs file:
4210
                                 <1>
                                          ; 16/10/2016 (1Dh -> 18)
4211
                                 <1>
                                          ; temporary - (21/03/2016)
4212 0000B87D B812000000
                                          mov eax, 18; write error
                                 <1>
4213 0000B882 F9
                                 <1>
4214 0000B883 C3
                                 <1>
                                          retn
4215
                                 <1>
4216
                                 <1> create file:
                                          ; 16/10/2016
4217
                                 <1>
4218
                                 <1>
                                          ; 24/03/2016, 31/03/2016
4219
                                          ; 20/03/2016, 21/03/2016, 23/03/2016
                                 <1>
4220
                                 <1>
                                          ; 19/03/2016 (TRDOS 396 = TRDOS v2.0)
4221
                                 <1>
                                          ; 03/09/2011 (FILE.ASM, 'proc_create_file')
4222
                                 <1>
                                          ; 09/08/2010
4223
                                 <1>
4224
                                 <1>
                                          ; INPUT ->
```

```
4225
                                 <1>
                                                 EAX = File Size
                                        ;
4226
                                 <1>
                                                 ESI = ASCIIZ File Name
                                                CL = File Attributes
4227
                                 <1>
                                          ;
4228
                                 <1>
                                              EBX = FFFFFFFFh -> create empty file
4229
                                 <1>
                                                             (only for FAT fs)
                                          ; OUTPUT ->
4230
                                 <1>
                                          ; CF = 0 \rightarrow
4231
                                 <1>
4232
                                 <1>
                                                 EAX = New file's first cluster
4233
                                 <1>
                                                 ESI = Logical Dos Drv Descr. Table Addr.
                                                EBX = offset CreateFile Size
4234
                                 <1>
4235
                                 <1>
                                               ECX = Sectors per cluster (<256)
                                                 EDX = Directory entry index/number (<65536)
4236
                                 <1>
                                              CF = 1 -> error code in AL
4237
                                 <1>
4238
                                 <1>
                                          test cl, 18h (directory or volume name)
4239
                                 <1>;
4240
                                 <1>;
                                           jnz
                                                short loc_createfile_access_denied
4241 0000B884 80E107
                                           and cl, 07h; S, H, R
                                <1>
4242 0000B887 880D[E4640100]
                                          mov [createfile_attrib], cl
                                <1>
4243
                                 <1>
4244 0000B88D 89D9
                                <1>
                                          mov ecx, ebx
4245 0000B88F 89F3
                                <1>
                                        mov ebx, esi ; ASCIIZ File Name address
                                        sub
4246 0000B891 29D2
                                 <1>
                                                edx, edx
                                          mov dh, [Current_Drv]
mov esi, Logical_DOSDisks
4247 0000B893 8A35[86590100]
                                <1>
4248 0000B899 BE00010900
                                <1>
                                         add esi, edx
4249 0000B89E 01D6
                                 <1>
4250
                                 <1>
4251 0000B8A0 8815[EF640100]
                                <1>
                                          mov [createfile UpdatePDir], dl; 0; 31/03/2016
4252
                                 <1>
4253
                                          ; LD_DiskType = 0 for write protection (read only)
                                 <1>
                                          cmp byte [esi+LD_DiskType], 1; 0 = Invalid
4254 0000B8A6 807E0101
                                <1>
4255 0000B8AA 730A
                                <1>
                                          jnb short loc_createfile_check_file_sytem
                                 <1>
                                          ; 16/10/2016 (TRDOS Error code: 30, disk write protected)
                                          mov eax, 30; 13h, MSDOS err: Disk write-protected
4257 0000B8AC B81E000000
                                <1>
4258 0000B8B1 66BA0000
                                <1>
                                          ; err retn: EDX = 0, EBX = File name offset
4259
                                <1>
4260
                                <1>
                                          ; ESI -> Dos drive description table address
4261 0000B8B5 C3
                                 <1>
                                          retn
                                 <1>
4262
                                 <1> ;loc createfile access denied:
4263
                                        mov eax, 05h; access denied (invalid attributes input)
4264
                                 <1>;
4265
                                 <1>;
                                           stc
4266
                                 <1>;
                                          retn
4267
                                 <1>
                                <1> loc_createfile_check_file_sytem:
4268
4269 0000B8B6 807E0301
                                <1>
                                      cmp byte [esi+LD_FATType], 1
4270 0000B8BA 730A
                                <1>
                                          jnb
                                                short loc_createfile_chk_empty_FAT_file_sign1
                                <1>
                                      mov [createfile_size], eax
; ESI = Logical Dos Drive Description Table address
; EBX = ASCIIZ File Name address
4272 0000B8BC A3[D0640100]
                                <1>
4273
                                 <1>
4274
                                <1>
                                        jmp create_fs_file
4275 0000B8C1 E9FE020000
                                 <1>
4276
                                 <1>
                                 <1> loc_createfile_chk_empty_FAT_file_sign1:
4277
                                      ; ECX = FFFFFFFF -> create empty file if drive has FAT fs
4278
                                <1>
4279 0000B8C6 41
                                 <1>
                                          inc
                                                ecx
4280 0000B8C7 7506
                                 <1>
                                           jnz
                                                 short loc_createfile_chk_empty_FAT_file_sign2
4281 0000B8C9 890D[D0640100]
                                <1>
                                                [createfile_size], ecx ; 0 ; empty file
                                          mov
4282
                                 <1>
4283
                                 <1> loc_createfile_chk_empty_FAT_file_sign2:
                                <1> ; 23/03/<del>2</del>016
4284
4285 0000B8CF 668B4E11
                                <1>
                                          mov cx, [esi+LD_BPB+BytesPerSec]
4286 0000B8D3 66890D[EC640100]
                                <1>
                                         mov
                                                [createfile BytesPerSec], cx
4287
                                 <1>
                                       ; EBX = ASCIIZ File Name address
movzx edx, byte [esi+LD_BPB+SecPerClust]
4288
                                <1>
4289 0000B8DA 0FB65613
                                <1>
4290 0000B8DE 8815[E5640100]
                                <1>
                                          mov [createfile_SecPerClust], dl
4291 0000B8E4 8B4E74
                                <1>
                                          mov ecx, [esi+LD_FreeSectors]
4292 0000B8E7 39D1
                                <1>
                                          cmp ecx, edx ; byte [createfile_SecPerClust]
4293 0000B8E9 7306
                                <1>
                                                short loc_create_fat_file
                                         jnb
4294
                                 <1>
4295
                                 <1> loc_createfile_insufficient_disk_space:
4296 0000B8EB B827000000
                                 <1> mov eax, 27h
                                 <1> loc_createfile_gffc_retn:
4297
4298 0000B8F0 C3
                                 <1>
4299
                                 <1>
4300
                                 <1> loc_create_fat_file:
4301 0000B8F1 891D[C8640100]
                                <1> mov [createfile_Name_Offset], ebx
4302 0000B8F7 890D[CC640100]
                                <1>
                                          mov [createfile_FreeSectors], ecx
4303
                                 <1>
4304
                                 <1> loc_createfile_gffc_1:
4305 0000B8FD E821050000
                                       call get_first_free_cluster
                                 <1>
4306 0000B902 72EC
                                 <1>
                                           jс
                                                 short loc_createfile_gffc_retn
4307
                                 <1>
4308 0000B904 A3[D4640100]
                                                 [createfile FFCluster], eax
                                 <1>
4309
                                 <1>
4310
                                 <1> loc_createfile_locate_ffe_on_directory:
                                          ; Current directory fcluster <> Directory buffer cluster
4311
                                 <1>
4312
                                           ; Current directory will be reloaded by
                                 <1>
4313
                                 <1>
                                           ; 'locate_current_dir_file' procedure
4314
                                 <1>
4315
                                 <1>
                                           ; ESI = Logical Dos Drv Desc. Table Adress
4316 0000B909 56
                                 <1>
                                           push esi;
4317 0000B90A 31C0
                                 <1>
                                           xor
                                                 eax, eax
                                 <1>
4318
4319 0000B90C A3[A2600100]
                                 <1>
                                                 dword [FAT_ClusterCounter], eax ; 0
                                          mov
4320
                                 <1>
                                           ; 21/03/2016
4321 0000B911 A2[EE640100]
                                 <1>
                                          mov byte [createfile_wfc], al; 0
4322
                                 <1>
4323 0000B916 89C1
                                 <1>
                                                ecx, eax
                                          dec cx; FFFFh
4324 0000B918 6649
                                 <1>
                                          ; CX = FFFFh -> find first deleted or free entry
4325
                                 <1>
4326
                                 <1>
                                          ; ESI would be ASCIIZ filename address if the call
                                          ; would not be for first free or deleted dir entry
4327
                                 <1>
4328 0000B91A E8D7E7FFFF
                                 <1>
                                           call locate_current_dir_file
4329 0000B91F 0F83EE000000
                                 <1>
                                           jnc loc_createfile_set_ff_dir_entry
```

```
4330 0000B925 5E
                                <1>
                                          pop esi; *
                                <1>
                                          ; ESI = Logical DOS Drv. Description Table Address
4332 0000B926 83F802
                               <1>
                                          cmp eax, 2
4333 0000B929 7402
                               <1>
                                                short loc_createfile_add_new_cluster
                                <1> loc_createfile_locate_file_stc_retn:
4334
4335 0000B92B F9
                                <1>
                                         stc
4336 0000B92C C3
                                <1>
4337
                                <1>
4338
                                <1> loc_createfile_add_new_cluster:
4339 0000B92D 803D[85590100]02
                                         cmp byte [Current_FATType], 2
                               <1>
4340
                                <1>
                                          ;cmp byte [esi+LD_FATType], 2
4341 0000B934 770C
                                 <1>
                                                short loc createfile add new cluster check fsc
                                          jа
4342 0000B936 803D[84590100]01
                                <1>
                                                byte [Current_Dir_Level], 1
                                          cmp
4343
                                <1>
                                         ;cmp byte [esi+LD_CDirLevel], 1
                                <1>
4344 0000B93D 7303
                                        jnb short loc_createfile_add_new_cluster_check_fsc
4345
                                <1>
4346
                                <1>
                                         ;mov eax, 12
4347 0000B93F B00C
                                <1>
                                         mov al, 12; No more files
4348
                                <1>
4349
                                <1> loc_createfile_anc_retn:
4350 0000B941 C3
                                <1>
                                <1>
4352
                                <1> loc_createfile_add_new_cluster_check_fsc:
4353 0000B942 8B0D[CC640100]
                                <1>
                                        mov ecx, [createfile_FreeSectors]
4354 0000B948 0FB605[E5640100] <1>
                                          movzx eax, byte [createfile_SecPerClust]
4355 0000B94F 66D1E0
                                <1>
                                          shl ax, 1; AX = 2 * AX
                                          cmp ecx, eax
4356 0000B952 39C1
                                <1>
4357 0000B954 7295
                                <1>
                                         jb short loc_createfile_insufficient_disk_space
4358
                                <1>
4359
                                <1> loc_createfile_add_new_subdir_cluster:
                            <1>
4360 0000B956 8B15[B1600100]
                                       mov edx, [DirBuff_Cluster]
4361 0000B95C 8915[D8640100]
                                <1>
                                         mov
                                                [createfile_LastDirCluster], edx
                                <1>
4362
4363 0000B962 A1[D4640100]
                                <1>
                                               eax, [createfile_FFCluster]
                                         mov
4364 0000B967 E846040000
                                <1>
                                         call load FAT sub directory
4365 0000B96C 72D3
                                <1>
                                          jс
                                                short loc_createfile_anc_retn
4366
                                <1>
4367
                                <1> pass_createfile_add_new_subdir_cluster:
4368
                                         ;movzx eax, word [esi+LD_BPB+BytesPerSec]
                                 <1>
                                          movzx eax, word [createfile BytesPerSec]; 23/03/2016
4369 0000B96E 0FB705[EC640100]
                               <1>
                                          mul ecx ; ecx = directory buffer sector count
4370 0000B975 F7E1
                                <1>
                                                ecx, eax
4371 0000B977 89C1
                                <1>
                                         mov
                                         shr ecx, 2 ; dword count
4372 0000B979 C1E902
                                <1>
4373 0000B97C 29C0
                                         sub eax, eax; 0
                                <1>
4374 0000B97E F3AB
                                <1>
                                         rep
                                                stosd
4375
                                <1>
                                        mov
4376 0000B980 C605[AC600100]02 <1>
                                                byte [DirBuff ValidData], 2
                                        call save_directory_buffer
4377 0000B987 E890EFFFFF
                                <1>
4378 0000B98C 72B3
                                <1>
                                                short loc_createfile_anc_retn
4379
                                <1>
4380
                                <1> loc_createfile_save_added_subdir_cluster:
4381 0000B98E A1[D8640100]
                                <1> mov eax, [createfile_LastDirCluster]
4382 0000B993 8B0D[D4640100]
                               <1>
                                         mov
                                                ecx, [createfile_FFCluster]
4383 0000B999 E858050000
                               <1>
                                         call update_cluster
                                       jnc short loc_createfile_save_fat_buffer_0
4384 0000B99E 7304
                                <1>
4385 0000B9A0 09C0
                                <1>
                                          or
                                                eax, eax ; EAX = 0 -> cluster value is 0 or eocc
                                        jnz short loc_createfile_save_fat_buffer_stc_retn
4386 0000B9A2 751A
                               <1>
4387
                                <1>
4388
                                <1> loc_createfile_save_fat_buffer_0:
                               <1> mov eax, [createfile_FFCluster]
4389 0000B9A4 A1[D4640100]
4390 0000B9A9 A3[D8640100]
                               <1>
                                          mov
                                               [createfile_LastDirCluster], eax
                                     mov
                                         mov ecx, OFFFFFFFh; 28 bit call update_cluster
4391 0000B9AE B9FFFFFF0F
                                <1>
4392 0000B9B3 E83E050000
                                <1>
                                     jnc short loc_createfile_save_fat_buffer_1
or eax, eax; Was it free cluster
4393 0000B9B8 7306
                                <1>
4394 0000B9BA 09C0
                                <1>
4395 0000B9BC 7402
                                <1>
                                                short loc_createfile_save_fat_buffer_1
                                          jΖ
4396
                                <1>
4397
                                <1> loc_createfile_save_fat_buffer_stc_retn:
4398 0000B9BE F9
                                <1>
4399
                                <1> loc_createfile_save_fat_buffer_retn:
4400
                                <1> loc_createfile_gffc_2_stc_retn:
4401 0000B9BF C3
                                <1>
                                         retn
4402
                                <1>
                                <1> loc createfile save fat buffer 1:
4403
                                        ; byte [FAT_BuffValidData] = 2
                                <1>
4404
4405 0000B9C0 E8EE070000
                                <1>
                                          call save_fat_buffer
4406 0000B9C5 72F8
                                <1>
                                                short loc_createfile_save_fat_buffer_retn
                                          jс
                                <1>
4408 0000B9C7 803D[A2600100]01
                                <1>
                                                byte [FAT_ClusterCounter], 1
4409 0000B9CE 7222
                                                short loc_createfile_save_fat_buffer_2
                                <1>
                                          jb
4410
                                 <1>
4411
                                 <1>
                                          ; {\tt ESI} = Logical DOS Drive Description Table address
4412 0000B9D0 A1[A2600100]
                                 <1>
                                                 eax, [FAT_ClusterCounter]
                                          mov
4413
                                 <1>
4414 0000B9D5 C605[A2600100]00
                                <1>
                                                byte [FAT_ClusterCounter], 0 ; 21/03/2016
                                          mov
4415
                                <1>
                                          mov
                                                bx, OFF01h; add free clusters
4416 0000B9DC 66BB01FF
                                <1>
4417 0000B9E0 E863080000
                                <1>
                                          call calculate fat freespace
4418
                                <1>
4419
                                <1>
                                          ;inc eax; OFFFFFFFF -> 0; recalculation is needed!
4420
                                <1>
                                          ;jnz short loc_createfile_save_fat_buffer_2
4421
                                 <1>
                                          ; ecx > 0 -> Recalculation is needed
4422
                                <1>
4423 0000B9E5 09C9
                                <1>
                                                 ecx, ecx
4424 0000B9E7 7409
                                <1>
                                          jΖ
                                                short loc_createfile_save_fat_buffer_2
4425
                                <1>
                                          mov bx, OFFOOh; ; recalculate free space
4426 0000B9E9 66BB00FF
                                <1>
4427 0000B9ED E856080000
                                <1>
                                          call calculate_fat_freespace
4428
                                <1>
4429
                                <1> loc_createfile_save_fat_buffer_2:
4430
                                <1>
                                         ;call update_parent_dir_lmdt
4431
                                <1>
                                <1> loc_createfile_gffc_2:
4432
4433 0000B9F2 E82C040000
                                <1>
                                        call get_first_free_cluster
4434 0000B9F7 72C6
                                <1>
                                          jc short loc_createfile_gffc_2_stc_retn
```

```
4435
                                  <1>
4436 0000B9F9 A3[D4640100]
                                  <1>
                                                  [createfile FFCluster], eax
                                           mov
                                  <1>
4437
4438 0000B9FE A1[D8640100]
                                 <1>
                                                  eax, [createfile_LastDirCluster]
                                 <1>
4439
4440 0000BA03 E8AA030000
                                 <1>
                                           call
                                                 load_FAT_sub_directory
4441 0000BA08 72B5
                                 <1>
                                                  short loc_createfile_gffc_2_stc_retn
                                           jс
4442
                                 <1>
4443 0000BA0A BF00000800
                                 <1>
                                                  edi, Directory_Buffer
4444
                                 <1>
4445 0000BA0F 6629DB
                                 <1>
                                           sub
                                                  bx, bx ; directory entry index/number = 0
4446
                                  <1>
                                           push esi; *; 23/03/2016
4447 0000BA12 56
                                 <1>
4448
                                 <1>
                                 <1> loc createfile_set_ff_dir_entry:
4449
4450 0000BA13 66891D[E6640100]
                                 <1>
                                           mov [createfile_DirIndex], bx
                                  <1>
                                            ; EDI = Directory entry address
4452
                                 <1>
4453 0000BA1A 8B35[C8640100]
                                 <1>
                                                 esi, [createfile_Name_Offset]
4454 0000BA20 A1[D4640100]
                                 <1>
                                                 eax, [createfile FFCluster]
                                           mov
                                           mov [createfile Cluster], eax; 24/03/2016
4455 0000BA25 A3[DC640100]
                                 <1>
4456 0000BA2A B5FF
                                  <1>
                                           mov
                                                 ch, OFFh
4457 0000BA2C 8A0D[E4640100]
                                 <1>
                                           mov cl, [createfile_attrib] ; file attributes
                                 <1>
                                           ; CH > 0 \rightarrow File size is in [EBX]
4459 0000BA32 BB[D0640100]
                                           mov ebx, createfile_size
                                 <1>
4460
                                  <1>
4461 0000BA37 E803EEFFFF
                                  <1>
                                           call make_directory_entry
4462
                                 <1>
4463 0000BA3C 5E
                                  <1>
                                           pop
                                                  esi ; * ; ESI = Logical Dos Drv Desc. Table address
                                  <1>
4464
4465 0000BA3D C605[AC600100]02
                                 <1>
                                                 byte [DirBuff_ValidData], 2
                                           mov
4466 0000BA44 E8D3EEFFFF
                                  <1>
                                           call
                                                 save directory buffer
4467 0000BA49 7221
                                  <1>
                                                  short loc_createfile_set_ff_dir_entry_retn
                                           jс
4468
                                 <1>
4469 0000BA4B C605[EF640100]01
                                                  byte [createfile UpdatePDir], 1; 31/03/2016
                                 <1>
                                           mov
4470
                                 <1>
4471
                                 <1> loc createfile get set write file cluster:
4472 0000BA52 A1[D0640100]
                                           mov eax, [createfile size]
                                 <1>
4473 0000BA57 09C0
                                 <1>
                                           or
                                                  eax, eax
4474 0000BA59 7570
                                           jnz short loc_createfile_get_set_wfc_cont
                                 <1>
4475 0000BA5B 40
                                 <1>
                                           inc eax
4476
                                  <1>
                                           ; 23/03/2016
4477 0000BA5C 0FB61D[E5640100]
                                           movzx ebx, byte [createfile SecPerClust]
                                 <1>
4478
                                  <1>
                                           ;movzx ecx, word [esi+LD_BPB+BytesPerSec] ; 512
4479 0000BA63 0FB70D[EC640100]
                                 <1>
                                            movzx ecx, word [createfile_BytesPerSec] ; 512
                                                loc_createfile_set_cluster_count
4480 0000BA6A EB7C
                                 <1>
                                 <1>
4482
                                 <1> loc_createfile_set_ff_dir_entry_retn:
4483 0000BA6C C3
                                 <1>
4484
                                 <1>
4485
                                 <1> loc_createfile_write_fcluster_to_disk:
                                          add eax, [esi+LD_DATABegin]; convert to physical address mov ebx, Cluster_Buffer
4486 0000BA6D 034668
                                 <1>
4487 0000BA70 BB00000700
                                 <1>
                                           ; ESI = Logical DOS Drv. Desc. Tbl. address
4488
                                 <1>
4489
                                 <1>
                                           ; EAX = Disk address
4490
                                 <1>
                                           ; EBX = Sector Buffer
                                           ; ECX = sectors per cluster
                                  <1>
4492 0000BA75 E8D33D0000
                                 <1>
                                          call disk_write
4493 0000BA7A 7211
                                  <1>
                                                  short loc_createfile_dsk_wr_err
                                           jс
4494
                                 <1>
4495
                                 <1> loc_createfile_update_fat_cluster:
                                 <1>
                                           ; 21/03/2016
4497 0000BA7C 803D[EE640100]00
                                                 byte [createfile_wfc], 0
                                 <1>
                                           cmp
4498 0000BA83 7712
                                  <1>
                                                  short loc_createfile_update_fat_cluster_n1
                                           jа
4499
                                 <1>
4500 0000BA85 FE05[EE640100]
                                 <1>
                                           inc
                                                  byte [createfile_wfc] ; 1
4501 0000BA8B EB24
                                 <1>
                                           jmp
                                                 short loc_createfile_update_fat_cluster_n2
4502
                                 <1>
4503
                                 <1> loc_createfile_dsk_wr_err:
4504
                                 <1>
                                          ; 16/10/2016 (1Dh -> 18)
4505
                                 <1>
                                           ; 23/03/2016
4506 0000BA8D B812000000
                                                 eax, 18; Drive not ready or write error!
                                 <1>
                                           mov
4507 0000BA92 E9BD000000
                                 <1>
                                           jmp
                                                 loc_createfile_stc_retn
4508
                                 <1>
                                 <1> loc_createfile_update_fat_cluster_n1:
4509
4510 0000BA97 A1[E0640100]
                                 <1>
                                           mov
                                                 eax, [createfile_PCluster]
4511 0000BA9C 8B0D[DC640100]
                                 <1>
                                                 ecx, [createfile Cluster]
                                           mov
                                           call update_cluster
4512 0000BAA2 E84F040000
                                 <1>
                                           jnc short loc_createfile_update_fat_cluster_n2
or eax, eax; EAX = 0 -> cluster value is 0 or
4513 0000BAA7 7308
                                 <1>
                                                  eax, eax \overline{;} EAX = 0 \overline{-} cluster value is \overline{0} or eocc
4514 0000BAA9 09C0
                                 <1>
4515 0000BAAB 0F85A3000000
                                 <1>
                                           jnz loc_createfile_stc_retn
                                  <1>
                                 <1> loc_createfile_update_fat_cluster_n2:
4518 0000BAB1 A1[DC640100]
                                 <1>
                                           mov eax, [createfile_Cluster]
4519 0000BAB6 B9FFFFFF0F
                                 <1>
                                           mov ecx, OFFFFFFh
4520 0000BABB E836040000
                                 <1>
                                           call update_cluster
                                           jnc short loc createfile save fat buffer 3
4521 0000BAC0 734E
                                <1>
4522 0000BAC2 09C0
                                <1>
                                         or
                                                 eax, eax; EAX = 0 -> cluster value is 0 or eocc
4523 0000BAC4 744A
                                                 short loc createfile save fat buffer 3
                                 <1>
4524
                                 <1>
4525
                                 <1> loc_createfile_upd_fat_fcluster_stc_retn:
4526 0000BAC6 E989000000
                                 <1>
                                          jmp loc_createfile_stc_retn
                                 <1>
4527
                                  <1> loc createfile get set wfc cont:
4528
                                           ;movzx ecx, word [esi+LD_BPB+BytesPerSec] ; 512
4529
                                 <1>
4530 0000BACB 0FB70D[EC640100]
                                 <1>
                                           movzx ecx, word [createfile_BytesPerSec] ; 512
4531 0000BAD2 01C8
                                  <1>
                                           add eax, ecx
                                          dec eax ; add eax, 511
4532 0000BAD4 48
                                 <1>
4533 0000BAD5 29D2
                                 <1>
                                           sub
                                                 edx, edx
4534 0000BAD7 F7F1
                                 <1>
                                          div ecx
4535 0000BAD9 0FB61D[E5640100] <1>
                                         movzx ebx, byte [createfile_SecPerClust]
4536 0000BAE0 01D8
                                 <1>
                                          add
                                                 eax, ebx
                                 <1>
4537 0000BAE2 48
                                           dec
                                                  eax ; add eax, SecPerClust - 1
4538 0000BAE3 6631D2
                                 <1>
                                                 dx, dx
4539 0000BAE6 F7F3
                                 <1>
                                           div
                                                 ebx
```

```
4540
4541
                                <1> loc createfile set cluster count:
                                         mov [createfile_CCount], eax
4542 0000BAE8 A3[E8640100]
                                <1>
4543
                                <1>
                                               edi, Cluster_Buffer
4544 0000BAED BF00000700
                                <1>
                                         mov
4545 0000BAF2 89C8
                               <1>
                                         mov
                                               eax, ecx ; Bytes per Sector
4546 0000BAF4 F7E3
                                         mul ebx; Sectors per Cluster
                               <1>
                                      ; EAX = Bytes per Cluster
4547
                               <1>
4548 0000BAF6 89C1
                               <1>
                                         mov ecx, eax
                                         shr ecx, 2; dword count
4549 0000BAF8 C1E902
                               <1>
4550 0000BAFB 31C0
                               <1>
                                         xor eax, eax
4551 0000BAFD F3AB
                                <1>
                                         rep
                                               stosd ; clear cluster buffer
4552
                                <1>
4553 0000BAFF A1[DC640100]
                               <1>
                                                eax, [createfile_Cluster] ; 24/03/2016
                                         mov
4554
                                <1>
4555 0000BB04 89D9
                                         mov
                                <1>
                                                ecx, ebx
                                <1>
                                <1> loc_createfile_get_set_wf_fclust_cont:
4557
                                        sub eax, 2
4558 0000BB06 83E802
                                <1>
                                         mul ecx
4559 0000BB09 F7E1
                               <1>
4560
                               <1>
                                         ; EAX = Logical DOS disk address (offset)
                                         jmp
4561 0000BB0B E95DFFFFFF
                               <1>
                                                   loc_createfile_write_fcluster_to_disk
4562
                                <1>
4563
                                <1> loc_createfile_save_fat_buffer_3:
                                      ; byte [FAT_BuffValidData] = 2
4564
                                <1>
4565 0000BB10 E89E060000
                                <1>
                                          call save_fat_buffer
4566 0000BB15 723D
                               <1>
                                         jc loc_createfile_stc_retn
4567
                                <1>
                                         ; 21/03/2016
4568
                                <1>
4569 0000BB17 803D[A2600100]01 <1>
                                         cmp byte [FAT ClusterCounter], 1
4570 0000BB1E 721B
                               <1>
                                                short loc_createfile_save_fat_buffer_4
4571
                                <1>
                                         ; ESI = Logical DOS Drive Description Table address
4572
                                <1>
4573 0000BB20 A1[A2600100]
                               <1>
                                         mov eax, [FAT ClusterCounter]
                                                bx, 0FF01h; add free clusters
4574 0000BB25 66BB01FF
                                <1>
                                         mov
                                          call calculate fat freespace
4575 0000BB29 E81A070000
                                <1>
4576
                                <1>
4577
                                <1>
                                         ;inc eax; OFFFFFFFF -> 0; recalculation is needed!
4578
                                <1>
                                         ;jnz short loc_createfile_save_fat_buffer_4
4579
                                <1>
4580
                                <1>
                                         ; ecx > 0 -> Recalculation is needed
4581 0000BB2E 09C9
                                <1>
                                         or ecx, ecx
4582 0000BB30 7409
                                <1>
                                                short loc_createfile_save_fat_buffer_4
                                          jz
4583
                                <1>
                                         mov bx, 0FF00h; ; recalculate free space
call calculate_fat_freespace
4584 0000BB32 66BB00FF
                                <1>
4585 0000BB36 E80D070000
                                <1>
                                <1>
                                <1> loc_createfile_save_fat_buffer_4:
4587
4588 0000BB3B FF0D[E8640100]
                                <1>
                                         dec   dword [createfile_CCount]
                                               short loc createfile upd dir modif date time
4589
                                <1>
                                         ;jz
                                               short loc_createfile_stc_retn_cc ; 31/03/2016
4590 0000BB41 743F
                                <1>
4591
                                <1>
                               <1> loc_createfile_get_set_write_next_cluster:
4592
                                     call get_first_free_cluster
4593 0000BB43 E8DB020000
                               <1>
4594 0000BB48 720A
                                <1>
                                          jc
                                               short loc_createfile_stc_retn
4595
                               <1>
                               <1> loc createfile_get_set_write_next_cluster_1:
4597 0000BB4A 83F8FF
                                       cmp eax, OFFFFFFFh
                               <1>
4598 0000BB4D 7213
                               <1>
                                                short loc_createfile_get_set_write_next_cluster_2
                                          jb
4599
                               <1>
4600
                                <1> loc_createfile_wnc_insufficient_disk_space:
4601 0000BB4F B827000000
                                <1>
                                        mov eax, 27h; Insufficient disk space
4602
                                <1>
4603
                                <1> loc_createfile_stc_retn:
4604 0000BB54 803D[EE640100]01
                               <1> cmp byte [createfile_wfc], 1
4605 0000BB5B 7324
                                <1>
                                          jnb
                                               short loc_createfile_err_retn
4606 0000BB5D C3
                                <1>
                                         retn
4607
                                <1>
4608
                                <1> loc_createfile_wnc_inv_format_retn:
                               <1> ; mov eax, 28
4609
4610 0000BB5E B01C
                               <1>
                                         mov al, 28 ; Invalid format
4611 0000BB60 EBF2
                                <1>
                                         jmp short loc_createfile_stc_retn
                               <1>
4612
4613
                               <1> loc_createfile_get_set_write_next_cluster_2:
                                       cmp eax, 2
4614 0000BB62 83F802
                               <1>
4615 0000BB65 72F7
                               <1>
                                          jb
                                                short loc_createfile_wnc_inv_format_retn
4616
                                <1>
                                <1> loc_createfile_get_set_write_next_cluster_3:
4617
                                     mov ecx, [createfile_Cluster]
4618 0000BB67 8B0D[DC640100]
                                <1>
4619 0000BB6D A3[DC640100]
                                         mov [createfile Cluster], eax
                               <1>
4620 0000BB72 890D[E0640100]
                                <1>
                                         mov [createfile_PCluster], ecx
4621 0000BB78 0FB60D[E5640100]
                                          movzx ecx, byte [createfile_SecPerClust]
                                <1>
4622 0000BB7F EB85
                                <1>
                                                short loc_createfile_get_set_wf_fclust_cont
4623
                                <1>
4624
                                <1> loc_createfile_err_retn:
4625 0000BB81 F9
                                <1>
                                          stc
4626
                                <1>
4627
                                <1> ;loc_createfile_upd_dir_modif_date_time:
                                <1> loc createfile stc retn cc: ; 31/03/2016
4628
4629 0000BB82 9C
                                          pushf; cpu is here for an error return or completion
                                <1>
4630 0000BB83 50
                                <1>
                                          push eax ; error code if cf = 1
4631
                                <1>
4632
                                <1>
                                          ;call update_parent_dir_lmdt
4633
                                <1>
                                <1> ;loc_createfile_stc_retn_cc:
4634
                                         mov eax, [FAT ClusterCounter]
4635 0000BB84 A1[A2600100]
                                <1>
4636 0000BB89 09C0
                                <1>
                                          or
                                                eax, eax
4637 0000BB8B 741A
                                <1>
                                          jz
                                                short loc_createfile_stc_retn_pop_eax
4638 0000BB8D 8A3D[86590100]
                                <1>
                                         mov
                                                bh, [Current_Drv]
                                        mov bl, 01h; BL = 1 -> add clusters
4639 0000BB93 B301
                                <1>
4640
                                <1>
                                        ; NOTE: EAX value will be added to Free Cluster Count
4641
                                <1>
                                         ; (If EAX value is negative, Free Cluster Count will be decreased)
                                          call calculate fat freespace
4642 0000BB95 E8AE060000
                                <1>
                                         ; ESI = Logical DOS Drive Description Table Address
4643
                                <1>
4644
                                <1>
                                           ;jc short loc createfile stc retn pop eax cf
```

```
and
                                             ecx, ecx; cx = 0 \rightarrow valid free sector count
4646 0000BB9C 7409
                              <1>
                                        jz
                                             short loc createfile stc retn pop eax
4647
                              <1>
4648
                              <1> loc_createfile_stc_retn_recalc_FAT_freespace:
                                     mov bx, 0FF00h; bh = 0FFh ->
4649 0000BB9E 66BB00FF
                              <1>
4650
                              <1>
                                       ; ESI = Logical DOS Drv DT Addr
                                       ; BL = 0 -> Recalculate
                              <1>
4652 0000BBA2 E8A1060000
                                      call calculate_fat_freespace
                              <1>
4653
                              <1>
                              <1> loc_createfile_stc_retn_pop_eax:
4654
4655 0000BBA7 58
                                    pop
                              <1>
4656 0000BBA8 9D
                              <1>
                                       popf
4657 0000BBA9 7218
                              <1>
                                              short loc_createfile_retn
                                       jс
4658
                              <1>
                              <1> loc_createfile_retn_fcluster:
4659
4660 0000BBAB A1[D4640100]
                              <1>     mov     eax, [createfile_FFCluster]
<1>     mov     ebx, createfile size
4661 0000BBB0 BB[D0640100]
                                       mov ebx, createfile size
                                    ;movzx ecx, byte [esi+LD_BPB+SecPerClust]
4662
                               <1>
4663 0000BBB5 0FB60D[E5640100]
                               <1>
                                       movzx ecx, byte [createfile_SecPerClust] ; 23/03/2016
                                      movzx edx, word [createfile_DirIndex]
4664 0000BBBC 0FB715[E6640100]
                              <1>
4665
                              <1>
                               <1> loc createfile retn:
4667 0000BBC3 C3
                               <1>
                                       retn
4668
                               <1>
4669
                               <1> create fs file:
4670
                               <1>
                                    ; temporary (21/03/2016)
4671 0000BBC4 C3
                               <1>
4672
                               <1>
4673
                               <1> delete_fs_file:
4674
                              <1> ; temporary (28/02/2016)
4675 0000BBC5 C3
                              <1>
4676
                               <1>
4677
                               <1> rename_fs_file_or_directory:
4678 0000BBC6 C3
                              <1>
4679
                               <1>
                               <1> make_fs_directory:
4680
                               <1> ; temporary (21/02/2016)
4681
4682 0000BBC7 C3
                               <1>
                                       retn
4683
                               <1>
4684
                               <1> add_new_fs_section:
4685
                               <1>
                                     ; temporary (11/03/2016)
4686 0000BBC8 C3
                               <1>
4687
                               <1>
4688
                               <1> delete_fs_directory_entry:
                               <1> ; temporary (11/03/2016)
4689
4690 0000BBC9 C3
                               <1>
                                        retn
4691
                               <1>
                               <1> csftdf2_read_fs_file_sectors:
4692
                                       ; temporary (19/03/2016)
4693
                              <1>
4694 0000BBCA C3
                               <1>
                                        retn
4695
                               <1>
4696
                               <1> csftdf2_write_fs_file_sectors:
                               <1> ; temporary (19/03/2016)
4697
4698 0000BBCB C3
                               <1>
2309
                                  %include 'trdosk5.s' ; 24/01/2016
                               1
  2
                               <1>; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.0) - File System Procedures : trdosk5s
  3
                               4
                               <1> ; Last Update: 23/10/2016
  5
                               6
                               <1>; Beginning: 24/01/2016
                               <1> ; ------
  8
                               <1> ; Assembler: NASM version 2.11 (trdos386.s)
  9
 10
                               <1>; Derived from TRDOS Operating System v1.0 (8086) source code by Erdogan Tan
 11
                               <1>; DRV_FAT.ASM (21/08/2011)
                               12
                               <1>; DRV_FAT.ASM (c) 2005-2011 Erdogan TAN [ 07/07/2009 ] Last Update: 21/08/2011
 13
 14
                               <1>
 15
                               <1> get_next_cluster:
                                     ; 15/10/2016
 16
                               <1>
 17
                               <1>
                                       ; 23/03/2016
                                       ; 01/02/2016 (TRDOS 386 = TRDOS v2.0)
                               <1>
 18
                                      ; 05/07/2011
 19
                               <1>
                                      ; 07/07/2009
; 2005
; INPUT ->
 20
                               <1>
 21
                               <1>
 22
                               <1>
 23
                               <1>
                                      ;
                                             EAX = Cluster Number (32 bit)
                                             ESI = Logical DOS Drive Parameters Table
 24
                               <1>
                                       ; OUTPUT ->
 25
                               <1>
 26
                               <1>
                                             cf = 0 -> No Error, EAX valid
                                              cf = 1 & EAX = 0 \rightarrow End Of Cluster Chain cf = 1 & EAX > 0 \rightarrow Error
 27
                               <1>
 28
                               <1>
 29
                               <1>
                                              ECX = Current/Previous cluster (if CF = 0)
 30
                               <1>
                                              EAX = Next Cluster Number (32 bit)
                                        ;
 31
                               <1>
 32
                               <1>
                                        ; (Modified registers: EAX, ECX, EBX, EDX)
 33
                               <1>
 34 0000BBCC A3[96600100]
                              <1>
                                              [FAT_CurrentCluster], eax
                              <1> check_next_cluster_fat_type:
 35
 36 0000BBD1 29D2
                                        sub edx, edx; 0
                              <1>
 37 0000BBD3 807E0302
                              <1>
                                         cmp byte [esi+LD FATType], 2
  38 0000BBD7 7250
                                        jb short get_FAT12_next_cluster
                              <1>
  39 0000BBD9 0F87AF000000
                                        ja get_FAT32_next_cluster
                              <1>
                              <1> get_FAT16_next_cluster:
 41 0000BBDF BB00030000
                                       mov ebx, 300h;768 div ebx
                              <1>
  42 0000BBE4 F7F3
                              <1>
                                       ; EAX = Count of 3 FAT sectors
 43
                              <1>
 44
                              <1>
                                       ; EDX = Cluster Offset (< 768)
 45 0000BBE6 66D1E2
                                       shl dx, 1; Multiply by 2
                              <1>
 46 0000BBE9 89D3
                              <1>
                                       mov ebx, edx; Byte Offset
                                       add ebx, F. mov dx, 3
  47 0000BBEB 81C3001C0900
                              <1>
                                             ebx, FAT Buffer
 48 0000BBF1 66BA0300
                              <1>
  49 0000BBF5 F7E2
                              <1>
                                       mul edx
 50
                               <1>
                                       ; EAX = FAT Sector (<= 256)
```

4645 0000BB9A 21C9

```
; EDX = 0
                                          mov cl, [esi+LD_Name]
cmp byte [FAT_BuffValidData], 0
  52 0000BBF7 8A0E
                                 <1>
  53 0000BBF9 803D[9A600100]00
                                <1>
  54 0000BC00 0F86CC000000
                                <1>
                                         jna load_FAT_sectors0
                                     cmp cl, [FAT_BuffDrvName]
jne load_FAT_sectors0
cmp eax, [FAT_BuffSector]
jne load_FAT_sectors1
;movzx eax, word [ebx]
mov ax, [ebx]
; 01/02/2016
. DRV FAT_ASM (21/08/2011) b
  55 0000BC06 3A0D[9B600100]
                                          cmp cl, [FAT_BuffDrvName]
                                <1>
  56 0000BC0C 0F85C0000000
                                <1>
  57 0000BC12 3B05[9E600100]
                                <1>
  58 0000BC18 0F85BA000000
                                <1>
  59
                                 <1>
  60 0000BC1E 668B03
                                 <1>
  61
                                 <1>
                                          ; DRV FAT.ASM (21/08/2011) had a FATal bug here!
  62
                                 <1>
  63
                                 <1>
                                          ; (cmp ah, 0Fh) ! (ax \geq FF7h)
                                          ; (how can i do a such mistake!?)
  64
                                 <1>
                                          ;cmp al, 0F7h
;jb short loc_pass_gnc_FAT16_eoc_check
  65
                                 <1>
  66
                                 <1>
                                          ;cmp ah, 0FFh
  67
                                 <1>
                                          ;jb short loc_pass_gnc_FAT16_eoc_check
  68
                                 <1>
                                                ax, OFFF7h
  69 0000BC21 6683F8F7
                                <1>
                                          cmp
                                          jb
                                                short loc_pass_gnc_FAT16_eoc check
  70 0000BC25 725A
                                <1>
                                <1>
  71
                                          ; ax >= FFF7h (cluster 0002h to FFF6h is valid, in use)
  72 0000BC27 EB56
                                <1>
                                          jmp short loc_pass_gnc_FAT16_eoc_check_xor_eax
  73
                                <1>
  74
                                <1> get_FAT12_next_cluster:
 75 0000BC29 BB00040000
                                <1>
                                      mov ebx, 400h;1024
                                          ; EAX = Count of 3 FAT sectors
                                        ; EDX = Cluster Offset (< 1024)
                                          mov bx, ax ; Byte Offset
105
                                 <1>
                                          ;jb
                                short loc_pass_gnc_FAT16_eoc_check
 106
                                          ;jb short loc_pass_gnc_FAT16_eoc_check
 108 0000BC79 663DF70F
 109 0000BC7D 7202
                                                short loc_pass_gnc_FAT16_eoc_check
                                          ; ax >= FF7h (cluster 0002h to FF6h is valid, in use)
 110
 111
                                 <1>
 112
                                 <1> loc_pass_gnc_FAT16_eoc_check_xor_eax:
                                <1> xor eax, eax; 0
 113 0000BC7F 31C0
                                 <1> loc_pass_gnc_FAT16_eoc_check:
 114
                                 <1> loc_pass_gnc_FAT32_eoc_check:
 115
 116 0000BC81 8B0D[96600100]
                                <1>
                                        mov ecx, [FAT_CurrentCluster]
 117 0000BC87 F5
                                <1>
 118 0000BC88 C3
                                <1>
                                          retn
                                <1>
 119
                                <1> get_FAT12_nc_even:
 120
 121 0000BC89 80E40F
                                <1> and ah, 0Fh
 122 0000BC8C EBEB
                                 <1>
                                          jmp short loc_gnc_fat12_eoc_check
                                <1>
 123
                                <1> get_FAT32_next_cluster:
                                <1> mov ebx, 180h ;384
 125 0000BC8E BB80010000
 126 0000BC93 F7F3
                                <1>
                                          div
                                                ebx
                                <1>
                                          ; EAX = Count of 3 FAT sectors
                                      ; EDX = Cluster Offset (< 384)
shl dx, 2; Multiply by 4
mov ebx, edx; Byte Offset
                                <1>
 128
 129 0000BC95 66C1E202
                                <1>
 130 0000BC99 89D3
                                <1>
 131 0000BC9B 81C3001C0900
                                <1>
                                        add ebx, FAT_Buffer
 132 0000BCA1 66BA0300
                                 <1>
                                          mov
                                                 dx, 3
 133 0000BCA5 F7E2
                                          mul edx
                                 <1>
                                 <1>
                                           ; EAX = FAT Sector (<= 2097152) ; (FFFFFF7h * 4) / 512
 135
                                 <1>
                                                for 32KB cluster size:
                                                EAX \le 1024 = (4GB / 32KB) * 4) / 512
 136
                                 <1>
                                          ; EDX = 0
                                 <1>
 138 0000BCA7 8A0E
                                                cl, [esi+LD_Name]
                                 <1>
                                          mov
                                                 byte [FAT_BuffValidData], 0
 139 0000BCA9 803D[9A600100]00
                                 <1>
                                          cmp
 140 0000BCB0 7620
                                                short load FAT sectors0
                                 <1>
                                          jna
                                                cl, [FAT BuffDrvName]
 141 0000BCB2 3A0D[9B600100]
                                 <1>
                                          cmp
                                                short load FAT sectors0
 142 0000BCB8 7518
                                 <1>
                                          jne
 143 0000BCBA 3B05[9E600100]
                                                 eax, [FAT_BuffSector] ; 0, 3, 6, 9 ...
                                 <1>
                                          cmp
                                                 short load FAT sectors1
 144 0000BCC0 7516
                                 <1>
                                          jne
 145 0000BCC2 8B03
                                 <1>
                                                 eax, [ebx]
                                          mov
 146 0000BCC4 25FFFFFF0F
                                 <1>
                                          and
                                                 eax, OFFFFFFFh ; 28 bit Cluster
 147 0000BCC9 3DF7FFFF0F
                                                eax, OFFFFFF7h
                                 <1>
                                          cmp
                                 <1>
                                                short loc_pass_gnc_FAT32_eoc_check
 148 0000BCCE 72B1
                                          jb
 149
                                 <1>
                                           ; eax >= FFFFFF7h (cluster 0002h to FFFFFF6h is valid)
 150 0000BCD0 EBAD
                                 <1>
                                          jmp short loc_pass_gnc_FAT16_eoc_check_xor_eax
 151
                                 <1>
 152
                                 <1> load_FAT_sectors0:
                                         mov [FAT_BuffDrvName], cl
 153 0000BCD2 880D[9B600100]
                                 <1>
                                 <1> load FAT sectors1:
 155 0000BCD8 A3[9E600100]
                                 <1>
                                       mov [FAT_BuffSector], eax
```

51

```
<1> mov ebx, eax
<1> add eax, [esi+LD_FATBegin]
<1> cmp byte [esi+LD_FATType], 2
<1> ja short load_FAT_sectors3
<1> movzx ecx, word [esi+LD_BPB+BPB_FATSz16]
<1> jmp short load_FAT_sectors4
156 0000BCDD 89C3
157 0000BCDF 034660
158 0000BCE2 807E0302
159 0000BCE6 7706
160 0000BCE8 0FB74E1C
161 0000BCEC EB03
                             <1> load_FAT_sectors3:
<1> mov ecx, [
<1> load_FAT_sectors4:
162
163 0000BCEE 8B4E2A
                                           mov ecx, [esi+LD_BPB+BPB_FATSz32]
164
                                <1>
                                           sub ecx, ebx ; [FAT_BuffSector]
165 0000BCF1 29D9
                                           cmp ecx, 3
jna short load_FAT_sectors5
166 0000BCF3 83F903
                                <1>
167 0000BCF6 7605
                                <1>
                                         mov ecx, 3
168 0000BCF8 B903000000
                                <1>
169
                                <1> load_FAT_sectors5:
                                170 0000BCFD BB001C0900
171 0000BD02 E8553B0000
                                      jnc short load_FAT_sectors_ok; 15/10/2016 (15h -> 17); 23/03/2016 (15h)

mov eax, 17; Drive not ready or read error
172 0000BD07 730D
                                <1>
173
                                 <1>
174
                                 <1>
175 0000BD09 B811000000
                                <1>
                                       mov byte [FAT_BuffValidData], 0
retn
176 0000BD0E C605[9A600100]00 <1>
177 0000BD15 C3
                                 <1>
                                 <1> load_FAT_sectors_ok:
178
179 0000BD16 C605[9A600100]01
                                <1>
                                      mov byte [FAT_BuffValidData], 1
180 0000BD1D A1[96600100]
                                           mov eax, [FAT_CurrentCluster]
                                 <1>
                                         jmp check_next_cluster_fat_type
181 0000BD22 E9AAFEFFFF
                                 <1>
                                 <1>
                                 <1> load_FAT_root_directory:
183
                                       ; 23/10/2016
184
                                 <1>
                                           ; 15/10/2016
185
                                 <1>
                                         ; 07/02/2016
186
                                 <1>
                                         ; 02/02/2016
; 01/02/2016 (TRDOS 386 = TRDOS v2.0)
187
                                 <1>
188
                                 <1>
189
                                 <1>
                                         ; 21/05/2011
                                          ; 22/08/2009
190
                                 <1>
191
                                 <1>
                                         ; INPUT ->
192
                                 <1>
193
                                 <1>
                                         ; ESI = Logical DOS Drive Description Table
194
                                 <1>
                                           ; OUTPUT ->
                                           ; cf = 1 -> Root directory could not be loaded
195
                                 <1>
196
                                 <1>
                                                    EAX > 0 -> Error number
197
                                 <1>
                                                  cf = 0 \rightarrow EAX = 0
                                           ;
                                                 ECX = Directory buffer size in sectors (CL)
                                 <1>
198
                                           ;
                                                  EBX = Directory buffer address
199
                                 <1>
200
                                 <1>
                                                  NOTE: DirBuffer_Size is in bytes ! (word)
                                           ;
201
                                 <1>
202
                                 <1>
                                           ; (Modified registers: EAX, ECX, EBX, EDX)
                                 <1>
203
204
                                 <1>
                                           ; NOTE: Only for FAT12 and FAT16 file systems !
                                           ; (FAT32 fs root dir must be loaded as sub directory)
205
                                 <1>
206
                                 <1>
207 0000BD27 8A1E
                                 <1>
                                                  bl, [esi+LD Name]
                                           mov
208 0000BD29 8A7E03
                                 <1>
                                           mov
                                                  bh, [esi+LD_FATType]
209
                                 <1>
210
                                 <1>
                                           ;mov [DirBuff_DRV], bl
211
                                 <1>
                                                  [DirBuff_FATType], bh
                                           ; mov
212 0000BD2C 66891D[AA600100]
                                 <1>
                                                 [DirBuff_DRV], bx
                                           mov
213
                                 <1>
214
                                 <1>
                                           ;cmp bh, 2
215
                                           ;ja short load_FAT32_root_dir0 ; FAT32 root dir
                                 <1>
216
                                 <1>
217
                                 <1> load FAT root dir0: ; 23/10/2016
218 0000BD33 0FB75617
                                           movzx edx, word [esi+LD_BPB+RootDirEnts]
                                 <1>
219
                                 <1>
220
                                                 dx, dx ; 0 for FAT32 file systems
                                 <1>
                                           ;or
221
                                 <1>
                                                 short load_FAT32_root_dir0 ; FAT32 root dir
                                 <1>
                                                 dx, 512; Number of Root Dir Entries
223 0000BD37 6681FA0002
                                 <1>
                                           cmp
224 0000BD3C 7414
                                 <1>
                                                  short lrd_mov_ecx_32
                                           jе
                                           mov
225 0000BD3E 89D0
                                <1>
                                                 eax, edx
                                       ; 23/10/2016
mov ecx, eax
226
                                <1>
227 0000BD40 89C1
                                 <1>
                                          add cx, 15; round up
228 0000BD42 6683C10F
                                <1>
                                         shr cx, 4 ; 16 entries per sector (512/32)
229 0000BD46 66C1E904
                                <1>
                                        ; ecx = Root directory size in sectors shl ax, 5; Root directory size in bytes
                                 <1>
230
231 0000BD4A 66C1E005
                                <1>
                                       dec dx ; Last entry number of root dir
232 0000BD4E 664A
                                <1>
233
                                 <1>
                                         ; cx = Dir Buffer sector count
234 0000BD50 EB0B
                                 <1>
                                          jmp short lrd_check_dir_buffer
235
                                 <1>
236
                                 <1> lrd_mov_ecx_32:
237 0000BD52 B920000000
                                 <1>
                                           mov ecx, 32
238 0000BD57 664A
                                 <1>
                                           dec
                                                  dx ; 511
239 0000BD59 66B80040
                                 <1>
                                                 ax, 32*512
                                 <1>
240
                                 <1> lrd_check_dir_buffer:
241
242 0000BD5D 29DB
                                 <1> sub ebx, ebx; 0
243 0000BD5F 881D[AC600100]
                                 <1>
                                                 [DirBuff_ValidData], bl ; 0
                                           mov
                                      mov
mov
244 0000BD65 668915[AF600100]
                                 <1>
                                                  [DirBuff_LastEntry], dx
245 0000BD6C 891D[B1600100]
                                                 [DirBuff Cluster], ebx; 0
                                 <1>
                                           mov
                                       mov [DirBuffer_Size], ax
246 0000BD72 66A3[B5600100]
                                 <1>
                                 <1>
248 0000BD78 8B4664
                                 <1>
                                                  eax, [esi+LD_ROOTBegin]
                                          mov
249
                                 <1> read_directory:
                                       mov
250 0000BD7B BB00000800
                                 <1>
                                                 ebx, Directory_Buffer
                                                  ecx ; Directory buffer sector count
251 0000BD80 51
                                 <1>
                                           push
252 0000BD81 53
                                           push ebx
                                 <1>
253 0000BD82 E8D53A0000
                                           call disk_read
                                <1>
254 0000BD87 5B
                                 <1>
                                           pop
                                                  ebx
                                                  short load_DirBuff_error
255 0000BD88 720B
                                           jс
                                 <1>
256
                                 <1>
257
                                 <1> validate DirBuff and return:
258 0000BD8A 59
                                 <1> pop ecx ; Number of loaded sectors
259 0000BD8B C605[AC600100]01
                                <1>
                                                 byte [DirBuff_ValidData], 1
                                           xor = eax, eax ; 0 = no error
260 0000BD92 31C0
                                 <1>
```

```
261 0000BD94 C3
                                <1>
                                           retn
                                 <1>
263
                                <1> load_DirBuff_error:
264 0000BD95 89C8
                                <1>
                                          mov eax, ecx; remaining sectors
                                          pop ecx; sector count sub ecx, eax; Number of loaded sectors
265 0000BD97 59
                                <1>
266 0000BD98 29C1
                                <1>
                                          ; 15/10/2016 (15h -> 17)
267
                                <1>
268 0000BD9A B811000000
                                         mov eax, 17; DRV NOT READY OR READ ERROR!
                                <1>
269 0000BD9F F9
                                 <1>
270 0000BDA0 C3
                                 <1>
                                           retn
271
                                 <1>
272
                                 <1> load FAT32 root directory:
273
                                 <1>
                                        ; 02/02/2016 (TRDOS 386 = TRDOS v2.0)
274
                                 <1>
                                         ; INPUT ->
275
                                 <1>
                                                ESI = Logical DOS Drive Description Table
276
                                 <1>
                                          ; OUTPUT ->
277
                                 <1>
278
                                         ; cf = 1 -> Root directory could not be loaded
                                 <1>
279
                                 <1>
                                                     EAX > 0 -> Error number
280
                                 <1>
                                                 cf = 0 \rightarrow EAX = 0
                                           ;
281
                                 <1>
                                                 ECX = Directory buffer size in sectors (CL)
282
                                 <1>
                                                 EBX = Directory buffer address
                                           ;
283
                                 <1>
                                                 NOTE: DirBuffer_Size is in bytes ! (word)
                                          ;
284
                                 <1>
285
                                 <1>
                                          ; (Modified registers: EAX, ECX, EBX, EDX)
286
                                 <1>
                                 <1>
288 0000BDA1 8A1E
                                 <1>
                                          mov
                                                 bl, [esi+LD_Name]
289 0000BDA3 8A7E03
                                 <1>
                                                 bh, [esi+LD_FATType]
                                          mov
290
                                 <1>
291
                                 <1>
                                          ;mov [DirBuff_DRV], bl
                                                [DirBuff_FATType], bh [DirBuff_DRV], bx
292
                                 <1>
                                           ;mov
293 0000BDA6 66891D[AA600100]
                                 <1>
                                           mov
294
                                 <1>
                                 <1> load_FAT32_root_dir0:
295
                                          mov eax, [esi+LD_BPB+FAT32_RootFClust] jmp short load_FAT_sub_dir0
296 0000BDAD 8B4632
                                 <1>
297 0000BDB0 EB0C
                                 <1>
298
                                 <1>
299
                                 <1> load_FAT_sub_directory:
                                      -; 01/02/2016 (TRDOS 386 = TRDOS v2.0)
300
                                 <1>
301
                                 <1>
                                          ; 05/07/2011
                                         ; 23/08/2009
302
                                 <1>
303
                                 <1>
304
                                 <1>
                                         ; INPUT ->
                                                 ESI = Logical DOS Drive Description Table
305
                                 <1>
                                        ;
306
                                 <1>
                                                 EAX = Cluster Number
                                          ; OUTPUT ->
307
                                 <1>
                                                 cf = 1 -> Sub directory could not be loaded
308
                                 <1>
309
                                 <1>
                                                   EAX > 0 -> Error number
                                                 cf = 0 \rightarrow EAX = 0
310
                                 <1>
                                           ;
311
                                 <1>
                                                 ECX = Directory buffer size in sectors (CL)
312
                                 <1>
                                                 EBX = Directory buffer address
313
                                 <1>
314
                                 <1>
                                                 NOTE: DirBuffer_Size is in bytes ! (word)
315
                                 <1>
316
                                 <1>
                                          ; (Modified registers: EAX, ECX, EBX, EDX)
                                 <1>
318 0000BDB2 8A1E
                                                 bl, [esi+LD_Name]
                                 <1>
                                           mov
319 0000BDB4 8A7E03
                                 <1>
                                                 bh, [esi+LD_FATType]
320
                                 <1>
321
                                 <1>
                                           ;mov [DirBuff_DRV], bl
322
                                 <1>
                                           ;mov
                                                 [DirBuff_FATType], bh
323 0000BDB7 66891D[AA600100]
                                                [DirBuff_DRV], bx
                                <1>
                                           mov
324
                                 <1>
                                 <1> load FAT sub dir0:
325
326 0000BDBE 0FB64E13
                                 <1>
                                          movzx ecx, byte [esi+LD_BPB+SecPerClust]
                                <1>
328 0000BDC2 882D[AC600100]
                                                  [DirBuff_ValidData], ch; 0
                                <1>
                                          mov
329 0000BDC8 A3[B1600100]
                                 <1>
                                                 [DirBuff_Cluster], eax
                                          mov
330
                                 <1>
331 0000BDCD 0FB74611
                                 <1>
                                          movzx eax, word [esi+LD_BPB+BytesPerSec]
332 0000BDD1 F7E1
                                 <1>
                                           mul
                                                 ecx
333 0000BDD3 C1E805
                                                 eax, 5; directory entry count (dir size / 32)
                                 <1>
                                           shr
334 0000BDD6 6648
                                 <1>
                                          dec ax ; last entry
335 0000BDD8 66A3[AF600100]
                                 <1>
                                                 [DirBuff_LastEntry], ax
                                          mov
336
                                 <1>
337 0000BDDE A1[B1600100]
                                <1>
                                                 eax, [DirBuff Cluster]
                                          mov
338 0000BDE3 83E802
                                 <1>
                                          sub
                                                 eax, 2
339 0000BDE6 F7E1
                                 <1>
                                           mul
                                                 ecx
340 0000BDE8 034668
                                           add eax, [esi+LD DATABegin]
                                 <1>
341
                                 <1>
                                           ; ecx = sector per cluster (dir buffer size = 32 sectors)
                                                 short read_directory
342 0000BDEB EB8E
                                           jmp
                                 <1>
343
                                 <1>
344
                                 <1> ; DRV_FS.ASM
345
                                 <1>
346
                                 <1> load_current_FS_directory:
347 0000BDED C3
                                 <1>
                                          retn
                                 <1> load_FS_root_directory:
348
349 0000BDEE C3
                                 <1>
                                          retn
                                 <1> load FS sub directory:
350
351 0000BDEF C3
                                 <1>
352
                                 <1>
353
                                 <1> read_cluster:
                                         ; 15/10/2016
354
                                 <1>
355
                                 <1>
                                          ; 18/03/2016
356
                                 <1>
                                          ; 16/03/2016
                                          ; 17/02/2016
357
                                 <1>
                                 <1>
                                          ; 15/02/2016 (TRDOS 386 = TRDOS v2.0)
358
359
                                 <1>
                                           ; INPUT ->
360
                                 <1>
361
                                 <1>
                                                 EAX = Cluster Number (Sector index for SINGLIX FS)
362
                                 <1>
                                                  ESI = Logical DOS Drive Description Table address
363
                                 <1>
                                                 EBX = Cluster (File R/W) Buffer address (max. 64KB)
                                 <1>
                                                  Only for SINGLIX FS:
364
365
                                 <1>
                                                 EDX = File Number (The 1st FDT address)
```

```
; OUTPUT ->
366
                                  <1>
                                           ; cf = 1 \rightarrow Cluster can not be loaded at the buffer
367
                                  <1>
368
                                                    EAX > 0 -> Error number
                                  <1>
369
                                  <1>
                                                  cf = 0 -> Cluster has been loaded at the buffer
370
                                  <1>
371
                                 <1>
                                           ; (Modified registers: EAX, ECX, EBX, EDX)
                                 <1>
373 0000BDF0 0FB64E13
                                           movzx ecx, byte [esi+LD_BPB+BPB SecPerClust]
                                 <1>
374
                                 <1>
                                            ; CL = 1 = [esi+LD_FS_Reserved2] ; SectPerClust for Singlix FS
375
                                 <1>
376
                                 <1> read_file_sectors: ; 16/03/2016
                                           cmp byte [esi+LD_FATType], 0
jna short read_fs_cluster
377 0000BDF4 807E0300
                                 <1>
378 0000BDF8 761C
                                 <1>
379
                                 <1>
                                 <1> read fat file_sectors: ; 18/03/2016
380
381 0000BDFA 83E802
                                 <1>
                                           sub eax, 2; Beginning cluster number is always 2
382 0000BDFD 0FB65613
                                <1>
                                           movzx edx, byte [esi+LD BPB+BPB SecPerClust]; 18/03/2016
383 0000BE01 F7E2
                                           mul
                                 <1>
                                                  edx
                                                  eax, [esi+LD DATABegin]; absolute address of the cluster
384 0000BE03 034668
                                 <1>
                                           add
385
                                 <1>
386
                                 <1>
                                           ; EAX = Disk sector address
                                           ; ECX = Sector count
387
                                  <1>
                                           ; EBX = Buffer address
388
                                 <1>
                                           ; (EDX = 0)
389
                                 <1>
                                           ; ESI = Logical DOS drive description table address
390
                                 <1>
391
                                 <1>
392 0000BE06 E8513A0000
                                 <1>
                                           call disk read
393 0000BE0B 7306
                                 <1>
                                          jnc short rclust_retn
                                 <1>
394
                                           ; 15/10/2016 (15h -> 17)
395
                                 <1>
396 0000BE0D B811000000
                                 <1>
                                                  eax, 17; Drive not ready or read error!
397 0000BE12 C3
                                 <1>
                                           retn
398
                                 <1>
399
                                 <1> rclust retn:
400 0000BE13 29C0
                                 <1>
                                            sub
                                                  eax, eax ; 0
401 0000BE15 C3
                                 <1>
                                            retn
402
                                 <1>
403
                                 <1> read_fs_cluster:
404
                                  <1>
                                           ; 15/02/2016 (TRDOS 386 = TRDOS v2.0)
405
                                  <1>
                                           ; Singlix FS
406
                                  <1>
407
                                  <1>
                                           ; EAX = Cluster number is sector index number of the file (eax)
                                  <1>
408
409
                                  <1>
                                           ; EDX = File number is the first File Descriptor Table address
410
                                  <1>
                                                  of the file. (Absolute address of the FDT).
411
                                  <1>
412
                                  <1>
                                           ; eax = sector index (0 for the first sector)
                                           ; edx = FDT0 address
                                 <1>
413
414
                                 <1>
                                                  ; 64 KB buffer = 128 sectors (limit)
415 0000BE16 B980000000
                                 <1>
                                                 ecx, 128; maximum count of sectors (before eof)
                                           mov
416 0000BE1B E801000000
                                 <1>
                                            call read_fs_sectors
417 0000BE20 C3
                                 <1>
                                           retn
418
                                 <1>
419
                                 <1> read_fs_sectors:
                                         \frac{15}{02}; \frac{15}{02}2016 (TRDOS 386 = TRDOS v2.0)
420
                                 <1>
421 0000BE21 F9
                                 <1>
422 0000BE22 C3
                                 <1>
                                           retn
                                 <1>
423
424
                                  <1> get_first_free_cluster:
                                       - ; 0\overline{2}/03/\overline{2}016
425
                                  <1>
426
                                  <1>
                                           ; 21/02/2016 (TRDOS 386 = TRDOS v2.0)
                                          ; 26/10/2010 (DRV_FAT.ASM, 'proc_get_first_free_cluster')
; 10/07/2010
427
                                  <1>
                                 <1>
428
429
                                  <1>
                                          ; INPUT ->
430
                                  <1>
                                                 ESI = Logical DOS Drive Description Table address
                                           ;
431
                                  <1>
                                           ; OUTPUT ->
432
                                  <1>
                                                cf = 1 \rightarrow Error code in AL (EAX)
                                           ;
                                 <1>
                                                  cf = 0 \rightarrow
433
                                                    EAX = Cluster number
434
                                  <1>
                                                    If EAX = FFFFFFFFh -> no free space
435
                                  <1>
                                            ;
                                                  If the drive has FAT32 fs:
436
                                 <1>
437
                                  <1>
                                                    EBX = FAT32 FSI sector buffer address (if > 0)
                                            ;
                                 <1>
438
439 0000BE23 8B4678
                                 <1>
                                            mov
                                                  eax, [esi+LD_Clusters]
440 0000BE26 40
                                 <1>
                                                  eax ; add eax, 1
                                           inc
441 0000BE27 A3[34630100]
                                 <1>
                                            mov
                                                   [gffc_last_free_cluster], eax
                                  <1>
443 0000BE2C 31DB
                                 <1>
                                            xor
                                                  ebx, ebx; 0; 02/03/2016
                                  <1>
444
445 0000BE2E 807E0302
                                                  byte [esi+LD FATType], 2
                                 <1>
                                            cmp
446 0000BE32 760E
                                  <1>
                                                  short loc_gffc_get_first_fat_free_cluster0
                                  <1>
448
                                  <1> loc_gffc_get_first_fat32_free_cluster:
                                           ; 02/03/2016
449
                                            call get_fat32_fsinfo_sector_parms
450 0000BE34 E844060000
                                  <1>
451 0000BE39 7207
                                  <1>
                                                  short loc_gffc_get_first_fat_free_cluster0
                                  <1>
                                  <1> loc_gffc_check_fsinfo_parms:
453
                                           ;; mov ebx, DOSBootSectorBuff
454
                                  <1>
455
                                  <1>
                                            ;cmp dword [ebx], 41615252h
456
                                  <1>
                                           ;jne short loc_gffc_fat32_fsinfo_err
457
                                  <1>
                                           ;cmp dword [ebx+484], 61417272h
458
                                  <1>
                                            ;jne short loc_gffc_fat32_fsinfo_err
459
                                  <1>
                                           ;mov eax, [ebx+492] ; FSI_Next_Free
460
                                  <1>
                                           ;EAX = First free cluster
461
                                  <1>
                                            ; (from FAT32 FSInfo sector)
462 0000BE3B 89D0
                                           mov eax, edx; FSI Next Free (First Free Cluster)
                                  <1>
                                                  eax, Offfffffff ; invalid (unknown) !
463 0000BE3D 83F8FF
                                 <1>
                                            cmp
464 0000BE40 7205
                                 <1>
                                                  short loc_gffc_get_first_fat_free_cluster1
465
                                 <1>
                                            ; Start from the 1st cluster of the FAT(32) file system
466
                                 <1>
467
                                  <1> loc_gffc_get_first_fat_free_cluster0:
468 0000BE42 B802000000
                                 <1>
                                           mov eax, 2
                                  <1>
                                            ;xor edx, edx
469
470
                                  <1>
```

```
471
                                 <1> loc_gffc_get_first_fat_free_cluster1:
472 0000BE47 53
                                 <1>
                                          push ebx ; 02/03/2016
473
                                 <1>
474
                                 <1> loc_gffc_get_first_fat_free_cluster2:
                                          mov [gffc_first_free_cluster], eax
475 0000BE48 A3[30630100]
                                 <1>
476 0000BE4D A3[2C630100]
                                 <1>
                                           mov
                                                  [gffc_next_free_cluster], eax
                                          ; EBX = FAT32 FSINFO sector buffer address
478
                                 <1>
479
                                 <1>
                                          ; (EBX = 0, if the drive has not got FAT32 fs or
                                 <1>
                                           ; FAT32 FSINFO sector buffer is invalid.)
480
481
                                 <1>
482
                                 <1> loc_gffc_get_first_fat_free_cluster3:
483 0000BE52 E875FDFFFF
                                          call get_next_cluster
                                 <1>
                                           jnc short loc gffc get first fat free cluster4
484 0000BE57 7307
                                <1>
485 0000BE59 09C0
                                           or
                                 <1>
                                                 eax, eax
486 0000BE5B 740B
                                <1>
                                           jΖ
                                                 short loc_gffc_first_free_fat_cluster_next
487 0000BE5D 5B
                                <1>
                                          pop
                                                 ebx ; 02/03/2016
488 0000BE5E F5
                                <1>
                                          cmc
                                                 ; stc
489 0000BE5F C3
                                <1>
                                           retn
490
                                <1>
                                <1> loc_gffc_get_first_fat_free_cluster4:
491
                                      and eax, eax; next cluster value jnz short loc_gffc_first_free_fat_cluster_next
492 0000BE60 21C0
                                <1>
493 0000BE62 7504
                                <1>
494 0000BE64 89C8
                                <1>
                                                 eax, ecx; current (previous cluster) value
495 0000BE66 EB22
                                <1>
                                         jmp short loc_gffc_check_for_set
496
                                 <1>
                                <1> loc_gffc_first_free_fat_cluster_next:
498 0000BE68 A1[2C630100]
                                <1>
                                          mov
                                                 eax, [gffc_next_free_cluster]
499 0000BE6D 3B05[34630100]
                                <1>
                                                  eax, [gffc_last_free_cluster]
                                           cmp
                                                 short retn_stc_from_get_first_free_cluster
500 0000BE73 7308
                                <1>
                                           inb
501
                                 <1> pass_gffc_last_cluster_eax_check:
502 0000BE75 40
                                 <1>
                                           inc eax; add eax, 1
503 0000BE76 A3[2C630100]
                                <1>
                                                 [gffc_next_free_cluster], eax
                                           mov
504 0000BE7B EBD5
                                <1>
                                           jmp short loc_gffc_get_first_fat_free_cluster3
505
                                 <1>
506
                                <1> retn_stc_from_get_first_free_cluster:
507 0000BE7D A1[30630100]
                                <1>
                                          mov eax, [gffc_first_free_cluster]
508 0000BE82 83F802
                                <1>
                                           cmp
                                                 eax, 2
509 0000BE85 7709
                                 <1>
                                                 short loc_gffc_check_previous_clusters
                                           jа
510 0000BE87 29C0
                                <1>
                                           sub
                                                 eax, eax
511 0000BE89 48
                                <1>
                                           dec eax ; FFFFFFFFh
                                 <1>
                                 <1> loc_gffc_check_for_set:
513
514
                                 <1>
                                         ; 02/03/2016
515 0000BE8A 5B
                                           pop ebx
                                 <1>
516
                                 <1>
517
                                 <1>
                                          ; EBX = FAT32 FSINFO sector buffer address
                                          ; (EBX = 0, if the drive has not got FAT32 fs or
518
                                 <1>
519
                                 <1>
                                           ; FAT32 FSINFO sector buffer is invalid.)
                                 <1>
521 0000BE8B 09DB
                                 <1>
                                                  ebx, ebx
522 0000BE8D 750E
                                 <1>
                                           jnz
                                                 short loc_gffc_set_ffree_fat32_cluster
523
                                 <1>
524
                                 <1>
                                           ;cmp byte [esi+LD_FATType], 3
525
                                 <1>
                                                short loc_gffc_set_ffree_fat32_cluster
                                           ;jnb
526
                                 <1>
527
                                 <1>
                                           ;xor ebx, ebx; 0
                                 <1>
528
529
                                 <1> loc_gffc_retn:
530 0000BE8F C3
                                 <1>
                                          retn
531
                                 <1>
                                 <1> loc gffc check previous clusters:
533 0000BE90 48
                                 <1>
                                          dec eax; sub eax, 1
534 0000BE91 A3[34630100]
                                 <1>
                                                 [gffc_last_free_cluster], eax
535 0000BE96 B802000000
                                 <1>
                                           mov
                                                 eax, 2
536
                                 <1>
                                           ;xor edx, edx
537 0000BE9B EBAB
                                 <1>
                                          jmp short loc_gffc_get_first_fat_free_cluster2
                                 <1>
538
539
                                 <1> loc_gffc_set_ffree_fat32_cluster:
                                          ;call set_first_free_cluster
540
                                 <1>
                                           ;retn
541
                                 <1>
542
                                 <1>
                                           ;jmp short set_first_free_cluster
                                 <1>
543
544
                                 <1> set_first_free_cluster:
                                       ; 15/10/2016
545
                                 <1>
                                           ; 23/03/2016
546
                                 <1>
547
                                 <1>
                                          ; 02/03/2016
                                         ; 29/02/2016
548
                                 <1>
549
                                 <1>
                                          ; 26/02/2016
                                          ; 21/02/2016 (TRDOS 386 = TRDOS v2.0)
550
                                 <1>
551
                                 <1>
                                           ; 21/08/2011 (DRV_FAT.ASM, 'proc_set_first_free_cluster')
                                           ; 11/07/2010
552
                                 <1>
                                           ; INPUT ->
553
                                 <1>
554
                                 <1>
                                                  ESI = Logical DOS Drive Description Table address
555
                                                  EAX = First free cluster
                                 <1>
556
                                 <1>
                                                 EBX = FSINFO sector buffer address
557
                                 <1>
                                                 ;; If EBX > 0, it is FSINFO sector buffer address
558
                                 <1>
                                                  ;;EBX = 0, if FSINFO sector is not loaded
559
                                 <1>
                                           ; OUTPUT->
                                                 ESI = Logical DOS Drive Description Table address
560
                                 <1>
                                                  If EBX > 0, it is FSINFO sector buffer address
561
                                 <1>
562
                                 <1>
                                                  EBX = 0, if FSINFO sector could not be loaded
                                                  CF = 1 \rightarrow Error code in AL (EAX)
563
                                 <1>
                                 <1>
                                                  CF = 0 \rightarrow first free cluster is successfully updated
564
565
                                 <1>
566
                                 <1>
                                           ;cmp
                                                 byte [esi+LD_FATType], 3
567
                                 <1>
                                           ;jb
                                                 short loc_sffc_invalid_drive
568
                                 <1>
                                           ; Save First Free Cluster value for 'update cluster'
569
                                 <1>
570 0000BE9D 89463E
                                 <1>
                                                  [esi+LD_BPB+BPB_Reserved+4], eax ; First free Cluster
571
                                 <1>
572
                                 <1>
                                                  ebx, ebx
                                           ;or
573
                                 <1>
                                           ;jnz
                                                  short loc sffc read fsinfo sector
                                 <1>
574
575 0000BEA0 813B52526141
                                 <1>
                                                   dword [ebx], 41615252h
                                           cmp
```

```
576 0000BEA6 7540
                                <1>
                                          jne
                                                short loc_sffc_read_fsinfo_sector
577 0000BEA8 81BBE4010000727241- <1>
                                          cmp
                                                dword [ebx+484], 61417272h
577 0000BEB1 61
                                <1>
578 0000BEB2 7534
                                <1>
                                                short loc_sffc_read_fsinfo_sector
579
                                <1>
                                                eax, [ebx+492] ; FSI_Next_Free
580 0000BEB4 3B83EC010000
                                <1>
                                          cmp
581 0000BEBA 741F
                                <1>
                                                short loc_sffc_retn
                                          jе
582
                                <1>
583
                                <1> loc_sffc_write_fsinfo_sector:
                                         ; EBX = FSINFO sector buffer
584
                                <1>
585
                                <1>
                                          ; [CFS_FAT32FSINFOSEC] is set in 'get_fat32_fsinfo_sector_parms'
586 0000BEBC 8983EC010000
                                         mov [ebx+492], eax
                                <1>
587 0000BEC2 A1[44630100]
                                <1>
                                                eax, [CFS_FAT32FSINFOSEC]
                                         mov
588 0000BEC7 B901000000
                               <1>
                                       mov ecx, 1
589 0000BECC 53
                                       push ebx
                                <1>
590 0000BECD E87B390000
                                          call disk write
                                <1>
591 0000BED2 7208
                                <1>
                                          jс
                                                short loc_sffc_read_fsinfo_sector_err1
592 0000BED4 5B
                                <1>
                                                ebx
                                        pop
593
                                <1>
594 0000BED5 8B83EC010000
                                <1>
                                               eax, [ebx+492] ; First (Next) Free Cluster
                                         mov
595
                                <1>
                                <1> loc sffc retn:
597 0000BEDB C3
                                <1>
                                         retn
                                <1>
599
                                <1> ;loc_sffc_invalid_drive:
600
                                <1>;
                                         mov eax, OFh ; MSDOS Error : Invalid drive
                                          push edx
601
                                <1>;
602
                                <1>
                                <1> loc_sffc_read_fsinfo_sector_err1:
603
604 0000BEDC BB00000000
                                       mov \overline{ebx}, 0
                                <1>
605
                                <1>
                                         ; 15/10/2016 (1Dh -> 18)
606
                                <1>
                                          ; 23/03/2016 (1Dh)
607 0000BEE1 B812000000
                                          mov eax, 18; Drive not ready or write error
                                <1>
608
                                <1>
609
                                <1> loc_sffc_read_fsinfo_sector_err2:
610 0000BEE6 5A
                                <1>
                                         pop
                                                edx
611 0000BEE7 C3
                                <1>
                                          retn
                                <1>
612
613
                                <1> loc_sffc_read_fsinfo_sector:
                                         push eax
614 0000BEE8 50
                                <1>
615
                                <1>
616 0000BEE9 E88F050000
                                <1>
                                          call get fat32 fsinfo sector parms
617 0000BEEE 72F6
                                                 short loc sffc read fsinfo sector err2
                                <1>
                                          jс
618
                                <1>
619 0000BEF0 58
                                <1>
                                         pop
                                               eax
                                          ; EDX = First (Next) Free Cluster value from FSINFO sector
620
                                <1>
621
                                <1>
                                          ; EAX = First Free Cluster value from 'get_next_cluster'
                                <1>
                                          ; (edx = old value)
622
                                               eax, edx ; First free Cluster (eax = new value)
623 0000BEF1 39D0
                                <1>
                                               short loc sffc write fsinfo sector
624 0000BEF3 75C7
                                <1>
                                          jne
625
                                <1>
626 0000BEF5 C3
                                <1>
                                          retn
627
                                <1>
628
                                <1> update_cluster:
                                      ; 23/10/2016
629
                                <1>
630
                                <1>
                                          ; 23/03/2016
631
                                <1>
                                         ; 02/03/2016
                                        ; 01/03/2016
                                <1>
632
                                        ; 29/02/2016
; 27/02/2016
633
                                <1>
                                <1>
634
                                        ; 26/02/2016
635
                                <1>
                                        ; 22/02/2016 (TRDOS 386 = TRDOS v2.0)
636
                                <1>
                                         ; 11/08/2011
                                <1>
637
638
                                <1>
                                        ; 09/02/2005
                                        ; INPUT ->
639
                                <1>
640
                                <1>
                                                EAX = Cluster Number
641
                                <1>
                                                ECX = New Cluster Value
                                <1>
                                                ESI = Logical Dos Drive Parameters Table
642
643
                                <1>
                                <1>
                                                /// dword [FAT_ClusterCounter] ///
644
645
                                <1>
                                          ; OUTPUT ->
646
                                <1>
                                              cf = 0 \rightarrow No Error, EAX is valid
647
                                <1>
648
                                <1>
                                                cf = 1 & EAX = 0 -> End Of Cluster Chain
                                                cf = 1 & EAX > 0 -> Error
649
                                <1>
                                                       (ECX -> any value)
650
                                <1>
651
                                <1>
                                                EAX = Next Cluster
652
                                <1>
                                                ECX = New Cluster Value
653
                                <1>
                                                 /// [FAT_ClusterCounter] is updated,
654
                                <1>
655
                                <1>
                                                 /// decreased when a free cluster is assigned,
656
                                 <1>
                                                 /// increased if an assigned cluster is freed.
                                <1>
658
                                <1>
659
                                <1>
                                          ; (Modified registers: EAX, EBX, -ECX-, EDX)
660
                                <1>
661 0000BEF6 A3[96600100]
                                <1>
                                                 [FAT CurrentCluster], eax
662 0000BEFB 890D[38630100]
                                <1>
                                        mov
                                                [ClusterValue], ecx
                                <1>
                                <1> loc update cluster check fat buffer:
664
665 0000BF01 8A1E
                                <1>
                                          mov bl, [esi+LD_Name]
666 0000BF03 381D[9B600100]
                                <1>
                                          cmp
                                                 [FAT BuffDrvName], bl
667 0000BF09 741A
                                <1>
                                          jе
                                                 short loc_update_cluster_check_fat_type
                                                byte [FAT BuffValidData], 2
668 0000BF0B 803D[9A600100]02
                                <1>
                                          cmp
669 0000BF12 0F84C2000000
                                          jе
                                <1>
                                                 loc_uc_save_fat_buffer
670
                                <1>
                                <1> loc uc reset fat buffer validation:
672 0000BF18 C605[9A600100]00
                                <1>
                                          mov byte [FAT_BuffValidData], 0
673
                                <1>
                                <1> loc_uc_check_fat_type_reset_drvname:
674
675 0000BF1F 881D[9B600100]
                                <1>
                                          mov [FAT_BuffDrvName], bl
676
                                <1>
                                <1> loc_update_cluster_check_fat_type:
677
678 0000BF25 29D2
                                <1>
                                        sub edx, edx; 26/02/2016
679 0000BF27 8A5E03
                                <1>
                                          mov bl, [esi+LD FATType]
```

```
680 0000BF2A 83F802
                               <1>
                                               eax, 2
                                         cmp
681 0000BF2D 0F82BE000000
                               <1>
                                         jb
                                                 update_cluster_inv_data
682 0000BF33 80FB02
                               <1>
                                         cmp
683 0000BF36 0F877A010000
                              <1>
                                              update_fat32_cluster
                               <1>
684
                                         ;cmp bl, 1
685
                               <1>
                                         ;jb
                                               short update cluster inv data
686 0000BF3C 8B4E78
                               <1>
                                               ecx, [esi+LD_Clusters]
                                         mov
687 0000BF3F 41
                               <1>
                                        inc
                                              ecx
688 0000BF40 890D[A6600100]
                               <1>
                                               [LastCluster], ecx
                                         mov
                                              eax, ecx; dword [LastCluster]
689 0000BF46 39C8
                               <1>
                                        cmp
690 0000BF48 0F87A6000000
                               <1>
                                        ja return_uc_fat_stc
691
                               <1>
                                         ; TRDOS v1 has a FATal bug here !
692
                               <1>
                                               ; or bl, bl ; cmp bl, 0
693
                               <1>
                                               ; jz short update_fat12_cluster
694
                               <1>
                                        ; !! It would destroy FAT12 floppy disk fs here !!
                                         ; ('A:' disks of TRDOS v1 operating system project
695
                               <1>
                                         ; had 'singlix fs', so, I could not differ this mistake
696
                               <1>
697
                               <1>
                                         ; on a drive 'A:')
                                         cmp bl, 1; correct comparison is this!
698 0000BF4E 80FB01
                               <1>
699 0000BF51 0F86A2000000
                               <1>
                                                 update_fat12_cluster
                                          jna
700
                               <1>
701
                               <1> update fat16 cluster:
702
                               <1> pass_uc_fat16_errc:
703
                               <1>
                                        ;sub edx, edx
704 0000BF57 BB00030000
                                        mov ebx, 300h ;768
                               <1>
705 0000BF5C F7F3
                               <1>
                                         div
                                              ebx
                               <1>
                                        ; EAX = Count of 3 FAT sectors
707
                              <1>
                                       ; DX = Cluster offset in FAT buffer
708 0000BF5E 6689D3
                               <1>
                                        mov bx, dx
709 0000BF61 66D1E3
                                        shl bx, 1; Multiply by 2
                              <1>
710 0000BF64 66BA0300
                              <1>
                                      mov dx, 3
                                       mul
711 0000BF68 F7E2
                               <1>
                                              edx
                                        ; EAX = FAT Sector
712
                               <1>
713
                               <1>
                                        ; EDX = 0
714
                               <1>
                                        ; EBX = Byte offset in FAT buffer
715 0000BF6A 8A0D[9A600100]
                               <1>
                                         mov cl, [FAT_BuffValidData]
716 0000BF70 80F902
                               <1>
                                         cmp cl, 2
717 0000BF73 750A
                               <1>
                                         jne short loc_uc_check_fat16_buff_sector_load
718
                               <1>
719
                               <1> loc_uc_check_fat16_buff_sector_save:
720 0000BF75 3B05[9E600100]
                               <1>
                                      cmp eax, [FAT_BuffSector]
721 0000BF7B 755D
                               <1>
                                         jne
                                               short loc_uc_save_fat_buffer
722 0000BF7D EB15
                               <1>
                                              short loc_update_fat16_cell
                                         jmp
723
                               <1>
724
                               <1> loc_uc_check_fat16_buff_sector_load:
725 0000BF7F 80F901
                                      cmp cl, 1 ; byte [FAT_BuffValidData]
                               <1>
                                         jne loc_uc_load_fat sectors
726 0000BF82 0F85FB010000
                               <1>
727 0000BF88 3B05[9E600100]
                               <1>
                                         cmp eax, [FAT_BuffSector]
                                        jne
                                                  loc uc load fat sectors
728 0000BF8E 0F85EF010000
                               <1>
729
                               <1>
                               <1> loc_update_fat16_cell:
730
                               <1> loc_update_fat16_buffer:
<1> add ebx, FAT_Buffer; 26/02/2016
731
732 0000BF94 81C3001C0900
733
                               <1>
                                         ;movzx eax, word [ebx]
734 0000BF9A 668B03
                               <1>
                                        mov ax, [ebx]
735
                               <1>
                                        ; 01/03/2016
736 0000BF9D 89C2
                               <1>
                                        mov edx, eax; old value of the cluster
737 0000BF9F A3[96600100]
                                               [FAT_CurrentCluster], eax
                               <1>
                                        mov
                                               ecx, [ClusterValue] ; 32 bits
738 0000BFA4 8B0D[38630100]
                               <1>
                                         mov
                                               [ebx], cx ; 16 bits !
739 0000BFAA 66890B
                               <1>
                                        mov
740
                               <1>
741 0000BFAD C605[9A600100]02
                               <1>
                                               byte [FAT_BuffValidData], 2
                                        mov
742
                               <1>
743 0000BFB4 6683F802
                               <1>
                                               ax, 2
                                               short return_uc_fat_stc
744 0000BFB8 723A
                               <1>
                                         jb
745 0000BFBA 3B05[A6600100]
                               <1>
                                               eax, [LastCluster]
                                         cmp
746 0000BFC0 7732
                               <1>
                                               short return_uc_fat_stc
                                         jа
747
                               <1>
748
                               <1> loc_fat_buffer_updated:
                                       ; 01/03/2016
749
                               <1>
750 0000BFC2 F8
                               <1>
                                         clc
                               <1> loc fat_buffer_stc_1:
751
                                      pushf
752 0000BFC3 9C
                               <1>
753 0000BFC4 21C9
                               <1>
754 0000BFC6 7506
                               <1>
                                               short loc_fat_buffer_updated_1
                                        jnz
755
                               <1>
756
                               <1>
                                        ; 01/03/2016
757
                               <1>
                                       ; new value of the cluster = 0 (free)
758
                                <1>
                                         ; increase free(d) cluster count
759 0000BFC8 FF05[A2600100]
                                        inc dword [FAT_ClusterCounter]
                               <1>
760
                               <1>
                                761
762 0000BFCE 09D2
                               <1>
763 0000BFD0 7506
                               <1>
                                         jnz short loc_fat_buffer_updated_2
                                         ; old value of the cluster = 0 (it was free cluster)
764
                               <1>
765
                               <1>
                                         ; decrease free(d) cluster count
766 0000BFD2 FF0D[A2600100]
                               <1>
                                              dword [FAT ClusterCounter] ; it may be negative number
767
                               <1>
768
                               <1> loc_fat_buffer_updated_2:
769 0000BFD8 9D
                               <1>
                                         popf
770 0000BFD9 C3
                               <1>
771
                               <1>
                               <1> loc_uc_save_fat_buffer:
772
                                       ; byte [FAT BuffValidData] = 2
773
                               <1>
                                         call save fat buffer
774 0000BFDA E8D4010000
                               <1>
                                                loc fat sectors rw error2
775 0000BFDF 0F8297010000
                               <1>
                                         ;mov byte [FAT BuffValidData], 1
                               <1>
                                              eax, [FAT_CurrentCluster]
777 0000BFE5 A1[96600100]
                               <1>
                                        mov
                                               ecx, [ClusterValue]
778
                               <1>
                                         ;mov
779
                               <1>
                                         ;jmp short loc update cluster check fat buffer
780 0000BFEA 8A1E
                               <1>
                                         mov bl, [esi+LD_Name]; 01/03/2016
                                         jmp
781 0000BFEC E927FFFFF
                               <1>
                                                loc_uc_reset_fat_buffer_validation
782
                               <1>
783
                               <1> update_cluster_inv_data:
784
                               <1>
                                         ;mov eax, ODh
```

```
785 0000BFF1 B00D
                                        mov al, ODh ; Invalid Data
786 0000BFF3 C3
                               <1>
                              <1>
788
                              <1> return_uc_fat_stc:
                                     ; 0<del>1</del>/03<del>/</del>2016
789
                               <1>
790 0000BFF4 31C0
                              <1>
                                        xor
                                              eax, eax
                                        stc
791 0000BFF6 F9
                              <1>
792 0000BFF7 EBCA
                              <1>
                                              short loc_fat_buffer_stc_1
                                        jmp
793
                               <1>
                               <1> update_fat12_cluster:
794
795
                               <1> pass_uc_fat12_errc:
796
                               <1>
                                        ;sub edx, edx
                                        mov ebx, 400h;1024
797 0000BFF9 BB00040000
                              <1>
798 0000BFFE F7F3
                              <1>
                                      div ebx
                                      ; EAX = Count of 3 FAT sectors
                              <1>
800
                              <1>
                                        ; DX = Cluster offset in FAT buffer
                                      mov cx, 3
                           <1>
801 0000C000 66B90300
                                     mov bx, ax
802 0000C004 6689C3
                              <1>
                                              ax, cx ; 3
803 0000C007 6689C8
                              <1>
                                        mov
                                      mul dx ; Multiply by 3
804 0000C00A 66F7E2
                              <1>
                                     shr ax, 1; Divide by 2 xchg bx, ax
805 0000C00D 66D1E8
                              <1>
                              <1>
806 0000C010 6693
                              <1>
807
                                        ; EAX = Count of 3 FAT sectors
                              <1>
                                     ; EBX = Byte Offset in FAT buffer
                                      mul cx ; 3 * AX
809 0000C012 66F7E1
                               <1>
810
                               <1>
                                        ; EAX = FAT Beginning Sector
                               <1>
                                        ; EDX = 0
812 0000C015 8A0D[9A600100]
                                      mov cl, [FAT_BuffValidData]
                               <1>
813
                               <1>
                                        ; TRDOS v1 has a FATal bug here !
                                        ; (it does not have 'cmp cl, 2' instruction here !
814
                               <1>
815
                               <1>
                                        ; while 'jne' is existing !)
                                        cmp cl, 2; 2 = dirty buffer (must be written to disk)
jne short loc_uc_check_fat12_buff_sector_load
816 0000C01B 80F902
                               <1>
817 0000C01E 750A
                               <1>
                               <1>
819
                               <1> loc_uc_check_fat12_buff_sector_save:
                                        cmp eax, [FAT BuffSector]
820 0000C020 3B05[9E600100]
                              <1>
821 0000C026 75B2
                               <1>
                                         jne short loc uc save fat buffer
822 0000C028 EB15
                               <1>
                                        jmp short loc_update_fat12_cell
823
                               <1>
824
                               <1> loc_uc_check_fat12_buff_sector_load:
825 0000C02A 80F901
                              <1>
                                     cmp cl, 1 ; byte ptr [FAT_BuffValidData]
                                         jne loc uc load fat sectors
826 0000C02D 0F8550010000
                               <1>
827 0000C033 3B05[9E600100]
                                        cmp eax, [FAT_BuffSector]
                              <1>
                                        jne loc_uc_load_fat_sectors
828 0000C039 0F8544010000
                              <1>
                               <1>
830
                               <1> loc_update_fat12_cell:
                                    add ebx, FAT_Buffer; 26/02/2016
831 0000C03F 81C3001C0900
                              <1>
832 0000C045 668B0D[96600100]
                                              cx, [FAT CurrentCluster]
                              <1>
                                        mov
833 0000C04C 66D1E9
                               <1>
                                        shr
                                              cx, 1
834 0000C04F 668B03
                              <1>
                                        mov
                                              ax, [ebx]
835 0000C052 6689C2
                              <1>
                                    mov
                                              dx, ax
                                      jnc
836 0000C055 7344
                               <1>
                                              short uc_fat12_nc_even
837
                              <1>
838 0000C057 6683E00F
                              <1>
                                        and
                                              ax, 0Fh
839 0000C05B 8B0D[38630100]
                              <1>
                                              ecx, [ClusterValue]; 32 bits
                                        mov
840 0000C061 66C1E104
                              <1>
                                        shl
                                              cx, 4
841 0000C065 6609C1
                                        or
                              <1>
                                              cx, ax
842 0000C068 6689D0
                              <1>
                                        mov
                                              ax, dx
843 0000C06B 66890B
                              <1>
                                               [ebx], cx ; 16 bits !
                                        mov
844 0000C06E 66C1E804
                              <1>
                                       shr ax, 4; al(bit4..7) +ah(bit0..7)
845
                              <1>
                               <1> update_fat12_buffer:
847 0000C072 A3[96600100]
                                    mov [FAT_CurrentCluster], eax
                              <1>
848 0000C077 89C2
                               <1>
                                               edx, eax; 01/03/2016
849 0000C079 C605[9A600100]02
                              <1>
                                              byte [FAT_BuffValidData], 2
                                        mov
850 0000C080 6683F802
                               <1>
                                        cmp
                                              ax, 2
851 0000C084 0F826AFFFFFF
                              <1>
                                        jb
                                              return uc fat stc
852 0000C08A 3B05[A6600100]
                              <1>
                                        cmp
                                              eax, [LastCluster]
                                      ja return_uc_fat_stc
jmp loc_fat_buffer_updated
853 0000C090 0F875EFFFFFF
                               <1>
854 0000C096 E927FFFFF
                               <1>
855
                               <1>
                               <1> uc_fat12_nc_even:
856
857 0000C09B 662500F0
                              <1> and ax, 0F000h
858 0000C09F 8B0D[38630100]
                              <1>
                                        mov ecx, [ClusterValue]; 32 bits
859 0000C0A5 80E50F
                              <1>
                                        and
                                              ch, OFh
860 0000C0A8 6609C1
                              <1>
                                        or
                                              cx, ax
                                    mov
861 0000C0AB 6689D0
                              <1>
                                              ax, dx
862 0000C0AE 66890B
                              <1>
                                     mov
                                              [ebx], cx ; 16 bits !
863 0000C0B1 80E40F
                               <1>
                                              ah, 0Fh; al(bit0..7)+ah(bit0..3)
                                        jmp short update_fat12_buffer
864 0000C0B4 EBBC
                               <1>
865
                               <1>
                               <1> update_fat32_cluster:
                          <1>
                                              ecx, [esi+LD_Clusters]
867 0000C0B6 8B4E78
                                        mov
868 0000C0B9 41
                              <1>
                                              ecx
869 0000C0BA 890D[A6600100]
                              <1>
                                        mov
                                              [LastCluster], ecx
870
                              <1>
871 0000C0C0 39C8
                              <1>
                                        cmp
                                              eax, ecx
872 0000C0C2 0F872CFFFFF
                                        ja
                                              return uc fat stc
                              <1>
873
                               <1>
874
                              <1> pass_uc_fat32_errc:
875
                              <1>
                                        ;sub edx, edx
                                              ebx, 180h ;384
876 0000C0C8 BB80010000
                              <1>
                                        mov
877 0000C0CD F7F3
                              <1>
                                        div
                                              ebx
                                        ; EAX = Count of 3 FAT sectors
                              <1>
                                        ; DX = Cluster offset in FAT buffer
879
                              <1>
880 0000C0CF 89D3
                              <1>
                                        mov ebx, edx
881 0000C0D1 C1E302
                              <1>
                                        shl
                                              ebx, 2 ; Multiply by 4
                                      mov edx, 3
882 0000C0D4 BA03000000
                              <1>
883 0000C0D9 F7E2
                              <1>
                                        mul
                                              edx
                                        ; EBX = Cluster Offset in FAT buffer
884
                              <1>
885
                              <1>
                                       ; EAX = FAT Sector
886
                               <1>
                                        ; EDX = 0
887 0000C0DB 8A0D[9A600100]
                              <1>
                                        mov cl, [FAT_BuffValidData]
                                        cmp cl, 2
888 0000C0E1 80F902
                              <1>
889 0000C0E4 750E
                               <1>
                                        jne short loc_uc_check_fat32_buff_sector_load
```

```
890
                                <1>
                                <1> loc uc check fat32 buff sector save:
892 0000C0E6 3B05[9E600100]
                                         cmp eax, [FAT BuffSector]
                                <1>
893 0000C0EC 0F85E8FEFFFF
                                <1>
                                          jne loc_uc_save_fat_buffer
894 0000C0F2 EB11
                                <1>
                                          jmp short loc_update_fat32_cell
895
                                <1>
                                <1> loc uc check fat32 buff sector load:
896
                                      cmp cl, 1 ; byte [FAT_BuffValidData]
897 0000C0F4 80F901
                                <1>
898 0000C0F7 0F8586000000
                                <1>
                                          jne loc_uc_load_fat_sectors
                                          cmp eax, [FAT_BuffSector]
899 0000C0FD 3B05[9E600100]
                                <1>
                                         jne loc_uc_load_fat_sectors
900 0000C103 757E
                                <1>
                                <1>
902
                                <1> loc_update_fat32_cell:
                                <1> loc update fat32 buffer:
903
904 0000C105 81C3001C0900
                                          add ebx, FAT_Buffer; 26/02/2016
                                <1>
905 0000C10B 8B03
                                <1>
                                          mov
                                                eax, [ebx]
906 0000C10D 25FFFFF0F
                                <1>
                                          and
                                               eax, OFFFFFFFh; 28 bit cluster value
907
                                <1>
                                                edx, [FAT CurrentCluster]; 01/03/2016
908 0000C112 8B15[96600100]
                                <1>
909
                                <1>
910 0000C118 A3[96600100]
                                <1>
                                         mov
                                                [FAT_CurrentCluster], eax
911 0000C11D 8B0D[38630100]
                                <1>
                                         mov
                                                ecx, [ClusterValue]
912 0000C123 890B
                                <1>
                                                [ebx], ecx; 29/02/2016
                                         mov
                                <1>
914 0000C125 C605[9A600100]02
                                <1>
                                                byte [FAT_BuffValidData], 2
                                         mov
915
                                <1>
                                          ; 01/03/2016
                                <1>
                                                eax, eax ; was it free cluster ?
917 0000C12C 21C0
                                <1>
                                          and
918 0000C12E 7514
                                <1>
                                          jnz
                                                short loc_upd_fat32_c0
919
                                <1>
920
                                <1>
                                          ;or
                                                ecx, ecx; it will be left free ?!
                                          ;jz
921
                                <1>
                                                short loc_upd_fat32_c3
922
                                <1>
923 0000C130 3B563E
                                <1>
                                                edx, [esi+LD_BPB+BPB_Reserved+4] ; First free cluster
                                          cmp
924 0000C133 7520
                                <1>
                                                short loc_upd_fat32_c3
                                          jne
925
                                <1>
926 0000C135 3B15[A6600100]
                               <1>
                                                edx, [LastCluster]
                                          cmp
927 0000C13B 7207
                                                short loc_upd_fat32_c0
                                <1>
                                          jb
928
                                <1>
929 0000C13D BA02000000
                               <1>
                                                edx, 2 ; rewind !
                                          mov
                                                short loc_upd_fat32_c2
930 0000C142 EB0E
                               <1>
                                         jmp
931
                                <1>
                               <1> loc_upd_fat32_c0:
932
933 0000C144 FF463E
                               <1>
                                         inc dword [esi+LD_BPB+BPB_Reserved+4]; set it to next cluster
934 0000C147 EB0C
                                <1>
                                          jmp
                                               short loc_upd_fat32_c3
935
                                <1>
                               <1> loc upd fat32 c1:
936
                                      or
937 0000C149 09C9
                               <1>
                                                ecx, ecx; will it be free cluster?
                                                short loc upd fat32 c3
938 0000C14B 7508
                                <1>
                                <1>
940 0000C14D 3B563E
                                <1>
                                                edx, [esi+LD_BPB+BPB_Reserved+4] ; First free cluster
941 0000C150 7303
                                <1>
                                          jnb
                                                short loc upd fat32 c3
942
                                <1>
943
                                <1> loc_upd_fat32_c2:
944 0000C152 89563E
                                <1>
                                         mov [esi+LD BPB+BPB Reserved+4], edx
945
                                <1>
946
                                <1> loc_upd_fat32_c3:
947 0000C155 89C2
                                <1>
                                        mov edx, eax
948
                                <1>
949
                                <1> loc_upd_fat32_c4:
950 0000C157 83F802
                                       cmp eax, 2
                                <1>
                                          jb
951 0000C15A 0F8294FEFFFF
                                <1>
                                                return uc fat stc
952
                                <1>
953
                                <1> pass_uc_fat32_c_zero_check_2:
                                     cmp eax, [LastCluster]
954 0000C160 3B05[A6600100]
                                <1>
955 0000C166 0F8788FEFFFF
                                <1>
                                                return_uc_fat_stc
                                <1>
957 0000C16C E951FEFFFF
                                <1>
                                                loc_fat_buffer_updated
                                         jmp
958
                                <1>
959
                                <1> loc_fat_sectors_rw_error1:
                                        ;mov byte [FAT BuffValidData], 0
960
                                <1>
961
                                <1>
                                         ; 23/10/2016 (15h -> 17)
962
                                <1>
                                          ; 23/03/2016
963 0000C171 B811000000
                                <1>
                                         mov eax, 17; Drive not ready or read error
964 0000C176 8825[9A600100]
                                <1>
                                               [FAT BuffValidData], ah; 0
                                        mov
965
                                <1>
966
                                <1> loc fat sectors rw error2:
967
                                <1>
                                       ;mov eax, error code
                                <1>
968
                                          ; mov
                                               edx, 0
969 0000C17C 8B0D[38630100]
                                <1>
                                         mov
                                               ecx, [ClusterValue]
970 0000C182 C3
                                <1>
                                         retn
971
                                <1>
                                <1> loc_uc_load_fat_sectors:
973 0000C183 A3[9E600100]
                                               [FAT_BuffSector], eax
974
                                <1>
975
                                <1> load_uc_fat_sectors_zero:
976 0000C188 034660
                                <1>
                                         add eax, [esi+LD FATBegin]
977 0000C18B BB001C0900
                                                ebx, FAT_Buffer
                                <1>
                                          mov
                                                ecx, 3
978 0000C190 B903000000
                                <1>
                                          mov
                                          call disk read
979 0000C195 E8C2360000
                                <1>
980 0000C19A 72D5
                                <1>
                                                short loc_fat_sectors_rw_error1
                                <1>
982 0000C19C C605[9A600100]01
                                                   byte [FAT_BuffValidData], 1
                                <1>
                                           mov
                                          mov eax, [FAT CurrentCluster]
983 0000C1A3 A1[96600100]
                                <1>
984 0000C1A8 8B0D[38630100]
                                <1>
                                         mov ecx, [ClusterValue]
                                                   loc_update_cluster_check_fat_type
985 0000C1AE E972FDFFFF
                                <1>
                                          jmp
                                <1>
987
                                <1> save fat buffer:
                                         ; <del>1</del>5/10/2016
988
                                <1>
                                          ; 01/03/2016
989
                                <1>
                                         ; 22/02/2016 (TRDOS 386 = TRDOS v2.0)
990
                                <1>
991
                                <1>
                                         ; 11/08/2011
992
                                <1>
                                         ; 09/02/2005
                                        ; INPUT ->
993
                                <1>
994
                                <1>
                                                None
                                        ;
```

```
; OUTPUT ->
 995
                                 <1>
 996
                                 <1>
                                          ;
                                                cf = 0 \rightarrow OK.
                                                 cf = 1 -> error code in AL (EAX)
 997
                                 <1>
                                          ;
 998
                                 <1>
999
                                 <1>
                                                 EBX = FAT Buffer address
                                          ;
1000
                                 <1>
                                          ; (EAX, EDX, ECX will be modified)
1001
                                 <1>
1002
                                 <1>
1003
                                 <1>
                                           ;cmp byte [FAT_BuffValidData], 2
                                           ;je short loc_save_fat_buff
                                 <1>
1004
1005
                                 <1>
1006
                                 <1> ;loc_save_fat_buffer_retn:
                                 <1>; xor eax, eax
1007
                                 <1>;
1008
1009
                                 <1>
                                <1> loc_save_fat_buff:
1010
1011 0000C1B3 31D2
                                <1> xor edx, edx
1011 0000C1B5 31D2
1012 0000C1B5 8A35[9B600100] <1>
                                          mov dh, [FAT_BuffDrvName]
1013 0000C1BB 80FE41
                                          cmp dh, 'A'
jb short loc_save_fat_buffer_inv_data_retn
                                <1>
                                      jb short lo
sub dh, 'A'
push esi; *
mov esi,
1014 0000C1BE 722E
                                <1>
1015 0000C1C0 80EE41
                                <1>
1016 0000C1C3 56
1017 0000C1C4 BE00010900
                                <1>
                                <1>
                                          mov esi, Logical_DOSDisks
1018 0000C1C9 01D6
                                <1>
                                          add esi, edx
1019
                                <1>
1020 0000C1CB 8A5603
                                <1>
                                          mov
                                                dl, [esi+LD_FATType]
                                      mov dl, [es
and dl, dl
1021 0000C1CE 20D2
                                <1>
1022 0000C1D0 741B
                                <1>
                                                 short loc_save_fat_buffer_inv_data_pop_retn
1023
                                <1>
1024 0000C1D2 A1[9E600100]
                                <1> mov eax, [
<1> cmp dl, 2
                                          mov eax, [FAT_BuffSector]
1025 0000C1D7 80FA02
1026 0000C1DA 770A
                                 <1>
                                         ja short loc_save_fat32_buff
                                <1>
1027
1028
                                <1> loc_save_fat_12_16_buff:
                                      ; 01/03/2016
1029
                                 <1>
                                          ; TRDOS v1 has a FATal bug here!
1030
                                 <1>
                                          ; Correct code: mov dx, word ptr [FAT BuffSector]+2
1031
                                 <1>
                                        ; (DX:AX in TRDOS v1 -> EAX in TRDOS v2)
1032
                                 <1>
1033
                                <1>
                           movzx ecx, word [esi+LD_BPB+FATSecs]
1034 0000C1DC 0FB74E1C
                                        sub ecx, eax
1035 0000C1E0 29C1
                                      ; TRDOS v1 has a bug here... ('pop esi' was forgotten!)
; jna short loc_save_fat_buffer_inv_data_retn; wrong addr!
jna short loc_save_fat_buffer_inv_data_pop_retn; correct addr.
jmp short loc_save_fat_buffer_check_rs3
1036
1037
1038 0000C1E2 7609
1039 0000C1E4 EB15
1040
                                <1>
                            1041
1042 0000C1E6 8B4E2A
1043 0000C1E9 29C1
                                <1>
                                           sub
                                                 ecx, eax
                             <1>
1044 0000C1EB 770E
                                          ja short loc_save_fat_buffer_check_rs3
1045
                                <1>
                                <1> loc_save_fat_buffer_inv_data_pop_retn:
<1> pop esi; *
1046
1047 0000C1ED 5E
1048
                                <1> loc_save_fat_buffer_inv_data_retn:
                                       mov eax, ODh ; Invalid DATA
1049 0000C1EE B80D000000
                                <1>
1050 0000C1F3 C3
                                <1>
                         <1> loc_save_fat_buff_remain_sectors_3:
<1> mov ecx, 3
<1> jmp short loc save fat buff
1052
                                      mov ecx, 3
jmp short loc_save_fat_buff_continue
1053 0000C1F4 B903000000
1054 0000C1F9 EB05
                           1055
1057 0000C1FB 83F903
1058 0000C1FE 77F4
1059
                                <1>
                     1060
1061 0000C200 BB001C0900
1062 0000C205 034660
                                                eax, [esi+LD_FATBegin]
1063 0000C208 51
1064 0000C209 E83F360000
1065 0000C20E 59
                                         jс
1066 0000C20F 722B
                                                 short loc_save_FAT_buff_write_err
1067
                                <1>
1068 0000C211 807E0302
                                <1>
                                        cmp byte [esi+LD_FATType], 2
1069 0000C215 7605
                                <1>
                                        jna short loc_calc_2nd_fat12_16_addr
1070
                                <1>
                                <1> loc_calc_2nd_fat32_addr:
                              <1> mov eax, [esi+LD_BPB+FAT32_FAT_Size]
1072 0000C217 8B462A
1073 0000C21A EB04
                                          jmp short loc calc 2nd fat addr
                                <1>
1074
                                <1>
                                <1> loc_calc_2nd_fat12_16_addr:
1075
1076 0000C21C 0FB7461C
                                 <1> movzx eax, word [esi+LD_BPB+FATSecs]
1077
                                 <1>
1078
                                 <1> loc_calc_2nd_fat_addr:
                                          add eax, [esi+LD_FATBegin] add eax, [FAT_BuffSector]
1079 0000C220 034660
                                 <1>
1080 0000C223 0305[9E600100]
                                 <1>
                                          mov ebx, FAT Buffer
1081 0000C229 BB001C0900
                                 <1>
1082
                                 <1>
                                          ; ecx = 1 to 3
1083 0000C22E E81A360000
                                 <1>
                                           call disk_write
                                           jc short loc save FAT buff write err
1084 0000C233 7207
                                 <1>
1085
                                           ; Valid buffer (1 = valid but do not save)
                                 <1>
                                          mov byte [FAT BuffValidData], 1
1086 0000C235 C605[9A600100]01
                                 <1>
1087
                                 <1>
                                 <1> loc save FAT buff write err:
1088
1089 0000C23C 5E
                                 <1>
                                          pop esi; *
1090 0000C23D BB001C0900
                                 <1>
                                           mov
                                                ebx, FAT Buffer
                                 <1>
                                           ; 15/10/2016 (1Dh -> 18)
1092
                                           ; 23/03/2016 (1Dh)
                                 <1>
1093 0000C242 B812000000
                                 <1>
                                           mov eax, 18; Drive not ready or write error
1094 0000C247 C3
                                 <1>
                                          retn
1095
                                 <1>
1096
                                 <1> calculate fat freespace:
                                          ; 23/03/2016
1097
                                 <1>
1098
                                 <1>
                                           ; 02/03/2016
1099
                                 <1>
                                           ; 01/03/2016
```

```
; 29/02/2016
1100
                                <1>
1101
                                 <1>
                                         ; 22/02/2016 (TRDOS 386 = TRDOS v2.0)
                                         ; 30/04/2011
1102
                                <1>
1103
                                <1>
                                        ; 03/04/2010
                                        ; 2005
1104
                                <1>
1105
                                <1>
                                         ; INPUT ->
                                                EAX = Cluster count to be added or subtracted
1106
                                <1>
                                                If BH = FFh, ESI = TR-DOS Logical Drive Description Table
1107
1108
                                <1>
                                                If BH < FFh, BH = TR-DOS Logical Drive Number
                                <1>
1109
1110
                                <1>
                                                0 = Calculate, 1 = Add, 2 = Subtract, 3 = Get (Not Set/Calc)
                                          ; OUTPUT ->
1111
                                 <1>
1112
                                <1>
                                               EAX = Free Space in sectors
1113
                                <1>
                                                ESI = Logical Dos Drive Description Table address
1114
                                <1>
                                                BH = Logical Dos Drive Number (same with input value of BH)
1115
                                <1>
                                                BL = Type of operation (same with input value of BL)
                                                ECX = 0 \rightarrow valid
1116
                                <1>
                                <1>
1117
                                                ECX > 0 -> error or invalid
1118
                                <1>
                                                If EAX = FFFFFFFFh, it is 're-calculation needed'
1119
                                <1>
                                                                       sign due to r/w error
1120
                                <1>
1121 0000C248 66891D[3E630100]
                                                [CFS OPType], bx
                                <1>
                                         mov
1122 0000C24F A3[40630100]
                                <1>
                                                [CFS_CC], eax
                                         mov
                                <1>
1124 0000C254 80FFFF
                                <1>
                                                bh, OFFh
                                          cmp
1125 0000C257 740B
                                <1>
                                          jе
                                                short pass_calculate_freespace_get_drive_dt_offset
1126
                                <1>
1127
                                <1> loc_calculate_freespace_get_drive_dt_offset:
                                     xor eax, eax
1128 0000C259 31C0
                                <1>
1129 0000C25B 88FC
                               <1>
                                          mov ah, bh
1130 0000C25D BE00010900
                               <1>
                                          mov esi, Logical_DOSDisks
                                        add esi, eax
1131 0000C262 01C6
                                <1>
1132
                                <1>
1133
                                <1> pass_calculate_freespace_get_drive_dt_offset:
                                     or bl, bl
1134 0000C264 08DB
                                <1>
1135 0000C266 7435
                                <1>
                                          jΖ
                                                short loc_reset_fcc
                                <1>
1137
                                <1> loc_get_free_sectors:
1138 0000C268 8B4674
                                <1>
                                        mov eax, [esi+LD_FreeSectors]
1139
                                <1>
1140
                                <1>
                                         ;xor ecx, ecx
1141
                                <1>
                                         ;dec ecx ; OFFFFFFFh
1142
                                <1>
                                         ;cmp eax, ecx; 29/02/2016
1143
                                <1>
                                         ;je short loc_get_free_sectors_retn ; recalculation is needed!
1144
                                <1>
                                          ; 23/03/2016
1145
                                <1>
1146 0000C26B 8B4E70
                                <1>
                                         mov ecx, [esi+LD TotalSectors]
1147 0000C26E 39C1
                                <1>
                                                ecx, eax ; Total sectors must be greater than Free sectors !
                                          cmp
1148 0000C270 7707
                                <1>
                                                short loc_get_free_sectors_check_optype
                                <1>
1150 0000C272 31C0
                                                eax, eax
                               <1>
                                          xor
1151 0000C274 48
                                <1>
                                          dec
                                                eax ; OFFFFFFFF ; recalculation is needed!
1152 0000C275 894674
                                <1>
                                          mov
                                                [esi+LD_FreeSectors], eax ; reset (for recalculation)
1153
                                <1>
1154
                                <1> loc_get_free_sectors_retn:
1155 0000C278 C3
                                <1>
                                          retn
                                <1>
1157
                                <1> loc_get_free_sectors_check_optype:
1158 0000C279 80FB03
                                <1>
                                               bl, 3
                                          cmp
1159 0000C27C 7203
                               <1>
                                                short loc_set_fcc
                                          jb
1160
                                <1>
1161 0000C27E 29C9
                                <1>
                                          sub
                                                ecx, ecx; 0
1162
                                <1>
1163 0000C280 C3
                                <1>
1164
                                <1>
1165
                                <1> loc_set_fcc:
1166 0000C281 807E0302
                                <1> cmp byte [esi+LD_FATType], 2
1167 0000C285 0F87DF000000
                                <1>
                                                   loc_update_FAT32_fs_info_fcc
1168
                                <1>
1169
                                <1>
                                         ;mov eax, [esi+LD FreeSectors]
1170 0000C28B 0FB64E13
                                <1>
                                         movzx ecx, byte [esi+LD_BPB+SecPerClust]
1171 0000C28F 29D2
                                <1>
                                          sub edx, edx
1172 0000C291 F7F1
                                <1>
                                          div
                                                ecx
1173
                                <1>
                                         ; ; DX -> Remain sectors < SecPerClust
; ; DX > 0 -> invalid free sector count
1174
                                <1>
1175
                                <1>
1176
                                <1>
                                         ;jnz short loc_reset_fcc
1177
                                <1>
                                <1> ;pass set fcc div32:
1178
1179 0000C293 A3[B7600100]
                                      mov [FreeClusterCount], eax
                                <1>
1180 0000C298 E988000000
                                                 loc_set_free_sectors_FAT12_FAT16
                                <1>
                                          jmp
                                 <1>
                                <1> loc_reset_fcc:
1183 0000C29D 31C0
                                <1>
1184 0000C29F A3[B7600100]
                               <1>
                                         mov
                                                [FreeClusterCount], eax ; 0
1185 0000C2A4 8B5678
                               <1>
                                         mov
                                                edx, [esi+LD_Clusters]
1186 0000C2A7 42
                               <1>
                                        inc edx
1187 0000C2A8 8915[A6600100]
                                                [LastCluster], edx
                               <1>
                                       mov
1188
                                <1>
                                                byte [esi+LD_FATType], 2
1189 0000C2AE 807E0302
                               <1>
                                         cmp
1190 0000C2B2 7647
                                <1>
                                                short loc_count_free_fat_clusters_0
1191
                                <1>
1192 0000C2B4 48
                                <1>
                                          dec
                                                eax ; FFFFFFFFh
                                                [CFS FAT32FC], eax
1193 0000C2B5 A3[48630100]
                              <1>
1194
                                <1>
                                          ; 29/02/2016
1195
                                <1>
1196 0000C2BA 89463A
                               <1>
                                               [esi+LD BPB+BPB Reserved], eax ; reset
                                         mov
1197 0000C2BD 89463E
                                <1>
                                                [esi+LD_BPB+BPB_Reserved+4], eax ; reset
                                          mov
1198
                                <1>
1199 0000C2C0 B802000000
                                <1>
                                          mov
                                                eax, 2
1200
                                <1>
1201
                                <1> loc count fc next cluster 0:
                               <1>
1202 0000C2C5 50
                                         push eax
1203 0000C2C6 E801F9FFFF
                               <1>
                                          call get next cluster
1204 0000C2CB 7310
                                <1>
                                          jnc short loc_check_fat32_ff_cluster
```

```
<1>
<1>
<1>
1205 0000C2CD 09C0
                                              eax, eax
                                        or
1206 0000C2CF 741E
                                          jг
                                              short pass_inc_cfs_fcc_0
1207
1208
                               <1> loc_put_fcc_unknown_sign:
1209 0000C2D1 58
                                       pop eax
; "Free count is Unknown" sign
                                <1>
1210
                                <1>
1211
                                <1>
                                         ;mov dword [FreeClusterCount], OFFFFFFFh
1212
                                <1>
1213
                                <1>
                                         ; 29/02/2016
1214
                                         ; Save Free Cluster Count value in FAT32 'BPB Reserved' area
                                <1>
1215
                                <1>
                                        ;mov [esi+LD_BPB+BPB_Reserved], OFFFFFFFFh; unknown!
1216 0000C2D2 8B15[48630100]
                                         mov edx, [CFS FAT32FC] ; First Free Cluster
                                <1>
1217
                                <1>
                                         ; Save First Free Cluster value in FAT32 'BPB Reserved+4' area
                                        mov [esi+LD BPB+BPB Reserved+4], edx
1218 0000C2D8 89563E
                               <1>
                                <1>
1219
1220 0000C2DB EB7D
                                          jmp loc_put_fcc_invalid_sign
                                <1>
                                <1>
mov [CFS_FAT32FC], eax
;mov dword [FreeClusterCount], 1
1229 0000C2ED EB27
                                <1>
                                       jmp short pass_inc_cfs_fcc_1
1230
                                <1>
1231
                                <1> pass_inc_cfs_fcc_0:
1232 0000C2EF 58
                                <1>
                                        pop eax
1233
                                <1>
1234
                                <1> pass_inc_cfs_fcc_0c:
1235 0000C2F0 40
1236 0000C2F1 3B05[A6600100] <1>
                               <1> inc eax; add eax, 1
                                          cmp eax, [LastCluster]
jna short loc_count_fc_next_cluster_0
1238 0000C2F9 EB6F
                               <1>
                                        jmp short loc_update_FAT32_fs_info_fcc
1239
                                <1>
                               <1> loc_count_free_fat_clusters_0:
1240
                              <1> ; mov eax, 2
                               <1>
1242 0000C2FB B002
                                         mov al, 2
1243
                                <1>
1244
                               <1> loc_count_fc_next_cluster:
                         <1> push eax
<1> call get_
<1> jc shor
1245 0000C2FD 50
1246 0000C2FE E8C9F8FFFF
                                         call get_next_cluster
1247 0000C303 720C
                                         jс
                                                short loc_count_fcc_stc
1248
                               <1>
                               <1> loc_count_free_clusters_1:
<1> and eax, eax
<1> jnz short pass_inc_cfs_fcc
1249
                       <1>
<1>
<1>
1250 0000C305 21C0
1251 0000C307 750C
                               <1>
1253 0000C309 FF05[B7600100]
                               <1>
                                         inc dword [FreeClusterCount]
                                        jmp short pass_inc_cfs_fcc
1254 0000C30F EB04
                               <1>
1255
                                <1>
1256
                                <1> loc_count_fcc_stc:
1257 0000C311 09C0
                               <1> or eax, eax
1258 0000C313 75BC
                               <1>
                                          jnz short loc_put_fcc_unknown_sign; 29/02/2016
1259
                                <1>
1260
                                <1> pass_inc_cfs_fcc:
                                        pop eax
1261 0000C315 58
                                <1>
1262
                                <1>
1263
                                <1> pass_inc_cfs_fcc_1:
1264 0000C316 40
                               <1> inc eax; add eax, 1
1265 0000C317 3B05[A6600100] <1>
                                         cmp eax, [LastCluster]
                                        jna short loc_count_fc_next_cluster
1266 0000C31D 76DE
                                <1>
1267
                                <1>
1268
                               <1> loc_set_free_sectors:
                               <1> cmp byte [esi+LD_FATType], 2
<1> ja short loc_update_FAT32_fs
1269 0000C31F 807E0302
1270 0000C323 7745
                                               short loc_update_FAT32_fs_info_fcc
1271
                                <1>
1272
                                <1> loc_set_free_sectors_FAT12_FAT16:
1275 0000032E A1[40630100]
                               <1> mov eax, [CFS_CC]
<1> cmp byte [CFS_OPType], 1
<1> ja short pass_FAT_add_fcc
<1> add [FreeClusterCount], eax
1276 0000C333 803D[3E630100]01 <1>
1277 0000C33A 7708
1278 0000C33C 0105[B7600100]
1279 0000C342 EB06
                                <1>
                                        jmp short pass_FAT_add_sub_fcc
1280
                                <1>
                                <1> pass_FAT_add_fcc:
1282 0000C344 2905[B7600100]
                                <1>
                                         sub [FreeClusterCount], eax
1283
                                <1>
                                <1> pass_FAT_add_sub_fcc:
1284
1285 0000C34A 0FB64613
                                <1> movzx eax, byte [esi+LD_BPB+SecPerClust]
1286 0000C34E 8B15[B7600100]
                                                edx, [FreeClusterCount]
                                 <1>
                                          mov
1287 0000C354 F7E2
                                <1>
                                          mul
                                                edx
                                <1>
1289 0000C356 31C9
                                <1>
                                          xor
                                                ecx, ecx
1290 0000C358 EB05
                                <1>
                                          jmp
                                               short loc_cfs_retn_params
                                <1>
1292
                                <1> loc_put_fcc_invalid_sign:
1293 0000C35A 29C0
                                <1>
                                                sub eax, eax; 0
                                          dec eax ; FFFFFFFFh
1294 0000C35C 48
                                <1>
                                <1> loc_fat32_ffc_recalc_needed:
1295
1296 0000C35D 89C1
                                <1>
                                         mov ecx, eax
1297
                                <1>
                                <1> loc cfs retn params:
1298
1299 0000C35F 894674
                                <1>
                                         mov [esi+LD FreeSectors], eax
1300 0000C362 0FB71D[3E630100]
                                <1>
                                          movzx ebx, word [CFS_OPType]
1301 0000C369 C3
                                <1>
1302
                                <1>
1303
                                <1> loc_update_FAT32_fs_info_fcc:
                                <1> loc_check_fcc_FSINFO_op:
1304
1305
                                <1>
                                         ; 29/02/2016
1306
                                <1>
                                         ; EAX = Free cluster count (before this update) ; value from disk
1307
                                <1>
                                          ; EDX = First Free Cluster (before this update) ; value from disk
1308 0000C36A 803D[3E630100]01
                                          cmp byte [CFS OPType], 1
                                <1>
1309 0000C371 7221
                                <1>
                                               short loc cfs FAT32 get rcalc parms ; 0 = recalculated
                                          jb
```

```
1310 0000C373 7406
                                                   short loc_check_fcc_FSINFO_op1 ; 1 = add
                                  <1>
                                           jе
                                  <1> loc check fcc FSINFO op2: ; subtract
                                  <1> neg dword [CFS_CC]; prepare to subtract; 2 = sub (add negative)
1312 0000C375 F71D[40630100]
1313
                                  <1> loc_check_fcc_FSINFO_op1:
                                         ; 01/03/2016
1314
                                  <1>
1315 0000C37B 31D2
                                  <1>
                                             xor edx, edx; 0

1316 0000C37D 4A
1317 0000C37E 8B463A
1318 0000C381 39D0
1319 0000C383 73D5
                                            add eax, [CFS_CC]; free cluster count on disk + current count
1320 0000C385 0305[40630100] <1>
1321 0000C38B 72CD
                                             jc short loc_put_fcc_invalid_sign
1322
                                  <1>
1323 0000C38D A3[B7600100]
                                             mov [FreeClusterCount], eax
                                 <1>
1324 0000C392 EB0E
                                  <1>
                                            jmp short loc_cfs_write_FSINFO_sector
1325
                                  <1>
                                  <1> loc cfs FAT32_get_rcalc_parms:
1327 0000C394 8B15[48630100] <1>
                                        mov edx, [CFS_FAT32FC]
1328 0000C39A A1[B7600100]
                                 <1>
                                             mov
                                                   eax, [FreeClusterCount]
                                            mov [esi+LD BPB+BPB Reserved+4], edx; First Free Cluster
1329 0000C39F 89563E
                                 <1>
                                 <1> loc_cfs_write_FSINFO_sector:
1330
                                 1331 0000C3A2 89463A
                                            ; 01/03/2016
1332
                                  <1>
1333 0000C3A5 E8AA000000
                                 <1>
                                          call set_fat32_fsinfo_sector_parms
                                         jc short loc_put_fcc_invalid_sign
1334 0000C3AA 72AE
                                  <1>
1335
                                  <1>
1336
                                  <1> loc set FAT32 free sectors:
                                         ; 29/02/2016
1337
                                  <1>
1338
                                  <1>
                                            ;mov eax, [FreeClusterCount]
1339
                                            ;mov ecx, eax
                                   <1>
1340
                                   <1>
                                            ;cmp eax, OFFFFFFFF ; Invalid !
1341
                                   <1>
                                            ;je short loc_cfs_retn_params
                                  <1>
1342
1343 0000C3AC 8B0D[B7600100]
                                  <1>
                                          mov ecx, [FreeClusterCount]
                                mul ecx
<1> ; 29/02/2016
<1> xor ecx, ecx; 0
<1> or edx, edx ^
<1> jn7
<1>
1344 0000C3B2 0FB64613
                                            movzx eax, byte [esi+LD_BPB+SecPerClust]
1345 0000C3B6 F7E1
1347 0000C3B8 31C9
1348 0000C3BA 09D2
                                                   edx, edx ; 0 ?
1349 0000C3BC 759C
                                            jnz loc_put_fcc_invalid_sign
1350 0000C3BE 394670
                                 <1>
                                             cmp [esi+LD_TotalSectors], eax ; Volume size in sectors
                                       jna
;
1351 0000C3C1 7697
                                  <1>
                                                    short loc_put_fcc_invalid_sign
1352
                                 <1>
1353
                                 <1> loc_set_FAT32_free_sectors_ok:
1354 0000C3C3 31D2
                                  <1>
                                            xor edx, edx; 0
1355 0000C3C5 EB98
                                  <1>
                                             jmp
                                                   short loc_cfs_retn_params
1356
                                  <1>
1357
                                  <1>
                                  <1> get last cluster:
1358
                                          - ; <del>2</del>2/10/2016
1359
                                  <1>
1360
                                  <1>
                                            ; 27/02/2016 (TRDOS 386 = TRDOS v2.0)
                                          ; 12/06/2010 (DRV_FAT.ASM, 'proc_get_last_custer') ; 06/06/2010
1361
                                   <1>
1362
                                  <1>
                                           ; INPUT ->
1363
                                  <1>
1364
                                  <1>
                                                   EAX = First Cluster Number
                                          ;
1365
                                  <1>
                                                   ESI = Logical Dos Drive Parameters Table
                                            ; OUTPUT ->
1366
                                   <1>
                                   <1>
                                          ;
                                                   cf = 0 -> No Error, EAX is valid
1367
1368
                                  <1>
                                                    cf = 1 \rightarrow EAX > 0 \rightarrow Error
1369
                                  <1>
                                                   EAX = Last Cluster Number
                                             ;
1370
                                  <1>
                                                   ECX = Previous Cluster -just before the last cluster-
1371
                                   <1>
                                                     ; 22/10/2016
                                             ;
1372
                                   <1>
                                                    [glc_index] = cluster index number of the last cluster
                                            ;
1373
                                   <1>
1374
                                   <1>
                                            ; (Modified registers: EAX, ECX, EBX, EDX)
1375
                                   <1>
1376 0000C3C7 89C1
                                   <1>
                                             mov
                                                    ecx, eax
1377
                                  <1>
1378 0000C3C9 C705[50630100]FFFF- <1>
                                                   dword [glc_index], OFFFFFFFF ; 22/10/2016
1378 0000C3D1 FFFF
                                  <1>
1379
                                   <1>
1380
                                  <1> loc_glc_get_next_cluster_1:
                                        mov [glc_prevcluster], ecx
1381 0000C3D3 890D[4C630100]
                                  <1>
                                            ; 22/10/2016
                                  <1>
1383 0000C3D9 FF05[50630100]
                                            inc dword [glc_index]
                                  <1>
1384
                                  <1>
                                  <1> loc_glc_get_next_cluster_2:
1386 0000C3DF E8E8F7FFFF
                                  <1>
                                         call get_next_cluster
                                            ; ecx = current/previous cluster
1387
                                  <1>
1388
                                            ; eax = next/last cluster
                                  <1>
1389 0000C3E4 73ED
                                           jnc short loc_glc_get_next_cluster_1
                                   <1>
                                   <1>
1391 0000C3E6 09C0
                                  <1>
                                                   eax, eax
                                             or
                                   <1>
1392 0000C3E8 7509
                                                 short loc_glc_stc_retn
                                             jnz
1393
                                   <1>
                                           ; ecx = previous cluster
1394
                                  <1>
1395 0000C3EA 89C8
                                  <1>
                                             mov eax, ecx
1396
                                  <1>
1397
                                  <1>
                                            ; previous cluster becomes last cluster (ecx -> eax)
1398
                                   <1>
                                            ; previous of previous cluster becomes previous cluster (ecx)
1399
                                   <1>
                                  <1> loc_glc_prev_cluster_retn:
1400
1401 0000C3EC 8B0D[4C630100]
                                            mov ecx, [glc_prevcluster]
                                  <1>
1402 0000C3F2 C3
                                  <1>
1403
                                  <1>
                                  <1> loc_glc_stc_retn:
1404
1405 0000C3F3 F5
                                  <1>
                                           cmc ;stc
1406 0000C3F4 EBF6
                                  <1>
                                             jmp short loc_glc_prev_cluster_retn
1407
                                  <1>
                                  <1> truncate_cluster_chain:
1408
                                          ; 01/03/2016
1409
                                  <1>
1410
                                   <1>
                                            ; 28/02/2016 \text{ (TRDOS } 386 = \text{TRDOS } v2.0)
1411
                                  <1>
                                            ; 22/01/2011 (DRV_FAT.ASM, 'proc_truncate_cluster_chain')
1412
                                   <1>
                                           ; 11/09/2010
                                   <1>
                                          ; INPUT ->
1413
```

```
1414
                                  <1>
                                                  ESI = Logical dos drive description table address
1415
                                  <1>
                                                  EAX = First cluster to be truncated/unlinked
                                           ;
                                           ; OUTPUT ->
1416
                                  <1>
1417
                                  <1>
                                                  ESI = Logical dos drive description table address
1418
                                  <1>
                                                  ECX = Count of truncated/removed clusters
1419
                                  <1>
                                                  CF = 0 \rightarrow EAX = Free sectors
                                                  CF = 1 -> Error code in EAX (AL)
1420
                                  <1>
1421
                                  <1>
1422
                                  <1>
                                            ; NOTE: This procedure does not update 1m date&time !
1423
                                  <1>
1424
                                  <1> loc_truncate_cc:
1425 0000C3F6 31C9
                                  <1>
                                           xor ecx, ecx; mov ecx, 0
                                  <1>
                                           ;mov byte [FAT_BuffValidData], 0
1426
1427 0000C3F8 890D[A2600100]
                                  <1>
                                            mov [FAT_ClusterCounter], ecx; 0; reset
                                  <1>
1428
1429
                                  <1> loc_tcc_unlink_clusters:
                                           call update cluster
1430 0000C3FE E8F3FAFFFF
                                  <1>
                                           ; EAX = Next Cluster
1431
                                  <1>
                                           ; ECX = Cluster Value
1432
                                  <1>
1433
                                  <1>
                                           ; Note:
                                          ; Returns count of unlinked clusters in
1434
                                  <1>
1435
                                  <1>
                                           ; dword ptr FAT ClusterCounter
1436 0000C403 73F9
                                  <1>
                                           jnc short loc_tcc_unlink_clusters
1437
                                  <1>
1438
                                  <1> pass_tcc_unlink_clusters:
1439 0000C405 A2[57630100]
                                  <1>
                                           mov byte [TCC_FATErr], al
1440 0000C40A 803D[9A600100]02
                                 <1>
                                           cmp byte [FAT BuffValidData], 2
1441 0000C411 750E
                                  <1>
                                           jne short loc_tcc_calculate_FAT_freespace
1442 0000C413 E89BFDFFFF
                                  <1>
                                            call save_fat_buffer
1443 0000C418 7307
                                           jnc short loc tcc calculate FAT freespace
                                  <1>
1444 0000C41A A2[57630100]
                                  <1>
                                           mov
                                                  byte [TCC_FATErr], al ; Error
1445
                                  <1>
                                           ;mov byte [FAT_BuffValidData], 0
1446
                                  <1>
1447
                                  <1>
                                           ; 01/03/2016
                                           jmp short loc_tcc_recalculate_FAT_freespace
1448 0000C41F EB12
                                  <1>
1449
                                  <1>
                                  <1> loc tcc calculate FAT freespace:
                                           mov eax, [FAT_ClusterCounter] ; signed (+-) number
1451 0000C421 A1[A2600100]
                                 <1>
1452 0000C426 66BB01FF
                                 <1>
                                                  bx, OFF01h; BH = FFh -> ESI = Dos drv desc. table
                                            mov
                                                           ; BL = 1 \rightarrow add cluster
1453
                                 <1>
                                            call calculate_fat_freespace
1454 0000C42A E819FEFFFF
                                 <1>
                                                  ecx, ecx; cx = 0 \rightarrow valid free sector count
1455 0000C42F 21C9
                                  <1>
                                            and
1456 0000C431 7409
                                 <1>
                                                  short pass_truncate_cc_recalc_FAT_freespace
                                            jz
1457
                                  <1>
1458
                                  <1> loc_tcc_recalculate_FAT_freespace:
1459 0000C433 66BB00FF
                                  <1>
                                           mov bx, 0FF00h; recalculate!
                                            call calculate_fat_freespace
1460 0000C437 E80CFEFFFF
                                  <1>
1461
                                  <1>
                                  <1> loc tcc calculate FAT freespace err:
1462
                                  <1> pass_truncate_cc_recalc_FAT_freespace:
1463
1464 0000C43C 8B0D[A2600100]
                                  <1>
                                           mov ecx, [FAT_ClusterCounter]
1465
                                  <1>
1466 0000C442 803D[57630100]00
                                                  byte [TCC_FATErr], 0
                                  <1>
                                            cmp
1467 0000C449 7608
                                  <1>
                                                  short loc_tcc_unlink_clusters_retn
1468
                                  <1>
1469
                                  <1> loc_tcc_unlink_clusters_error:
1470 0000C44B 0FB605[57630100]
                                  <1>
                                           movzx eax, byte [TCC FATErr]
1471 0000C452 F9
                                  <1>
                                           stc
1472
                                  <1> loc_tcc_unlink_clusters_retn:
1473 0000C453 C3
                                  <1>
                                           retn
1474
                                  <1>
1475
                                  <1> set fat32 fsinfo sector parms:
                                           ; 15/10/2016
1476
                                  <1>
1477
                                  <1>
                                           ; 23/03/2016
1478
                                  <1>
                                           ; 29/02/2016 (TRDOS 386 = TRDOS v2.0)
                                           ; INPUT ->
1479
                                  <1>
1480
                                  <1>
                                                  ESI = Logical dos drive description table address
                                                  [esi+LD_BPB+BPB_Reserved] = Free Cluster Count
1481
                                  <1>
1482
                                  <1>
                                                   [esi+LD_BPB+BPB_Reserved+4] = First Free Cluster
1483
                                  <1>
                                           ; OUTPUT ->
1484
                                  <1>
                                                  ESI = Logical dos drive description table address
1485
                                  <1>
                                                  CF = 0 \rightarrow OK..
                                                  CF = 1 -> Error code in EAX (AL)
                                  <1>
1486
1487
                                  <1>
1488
                                  <1>
                                            ; (Modified registers: EAX, EBX, ECX, EDX)
1489
                                  <1>
1490 0000C454 E824000000
                                  <1>
                                            call get fat32 fsinfo sector parms
1491 0000C459 7221
                                  <1>
                                                  short update_fat32_fsinfo_sector_retn
1492
                                  <1>
1493 0000C45B 8B463A
                                  <1>
                                                  eax, [esi+LD BPB+BPB Reserved] ; Free Cluster Count
                                            mov
1494 0000C45E 8B563E
                                  <1>
                                            mov
                                                  edx, [esi+LD_BPB+BPB_Reserved+4] ; First free Cluster
                                  <1>
                                  <1>
                                             ;mov ebx, DOSBootSectorBuff
1497 0000C461 8983E8010000
                                  <1>
                                            mov [ebx+488], eax
1498 0000C467 8993EC010000
                                  <1>
                                                  [ebx+492], edx
                                           mov
1499
                                  <1>
1500 0000C46D A1[44630100]
                                  <1>
                                                  eax, [CFS_FAT32FSINFOSEC]
                                           mov
1501 0000C472 B901000000
                                  <1>
                                           mov
                                                  ecx, 1
1502 0000C477 E8D1330000
                                  <1>
                                            call
                                                 disk_write
1503
                                  <1>
                                                   short update_fat32_fsinfo_sector_retn
                                           ;jnc
1504
                                  <1>
                                           ; 15/10/2016 (1Dh -> 18)
1505
                                  <1>
                                  <1>
1506
                                            ; 23/03/2016 (1Dh)
1507
                                  <1>
                                            ;mov eax, 18; Drive not ready or write error
1508
                                  <1>
1509
                                  <1> update_fat32_fsinfo_sector_retn:
1510 0000C47C C3
                                  <1>
                                            retn
1511
                                  <1>
1512
                                  <1> get_fat32_fsinfo_sector_parms:
                                          ; 15/10/2016
1513
                                  <1>
1514
                                  <1>
                                            ; 23/03/2016
1515
                                  <1>
                                           ; 01/03/2016
1516
                                  <1>
                                           ; 29/02/2016 (TRDOS 386 = TRDOS v2.0)
                                           ; INPUT ->
                                  <1>
1517
                                  <1>
1518
                                                  ESI = Logical dos drive description table address
```

```
; OUTPUT ->
1519
                                  <1>
1520
                                  <1>
                                                  ESI = Logical dos drive description table address
                                           ;
                                                  EBX = FSINFO sector buffer address (DOSBootSectorBuff)
1521
                                  <1>
1522
                                  <1>
                                                  CF = 0 \rightarrow OK..
                                                   EAX = FsInfo sector address
1523
                                  <1>
1524
                                  <1>
                                                     ECX = Free cluster count
                                                    EDX = First free cluster
1525
                                  <1>
                                                  CF = 1 -> Error code in AL (EAX)
                                  <1>
1526
1527
                                  <1>
                                                   EBX = 0
1528
                                  <1>
                                            ;
1529
                                  <1>
                                                   [CFS_FAT32FSINFOSEC] = FAT32 FSINFO sector address
1530
                                  <1>
1531
                                  <1>
                                            ; (Modified registers: EAX, EBX, ECX, EDX)
1532
                                  <1>
                                           movzx eax, word [esi+LD_BPB+FAT32 FSInfoSec]
1533 0000C47D 0FB74636
                                  <1>
1534 0000C481 03466C
                                  <1>
                                            add eax, [esi+LD_StartSector]
                                                 [CFS FAT32FSINFOSEC], eax
1535 0000C484 A3[44630100]
                                  <1>
1536
                                  <1>
1537 0000C489 BB[965E0100]
                                                     ebx, DOSBootSectorBuff
                                  <1>
1538 0000C48E B901000000
                                 <1>
                                           mov ecx, 1
                                            call disk read
1539 0000C493 E8C4330000
                                 <1>
                                            jс
1540 0000C498 7232
                                  <1>
                                                  short loc_read_FAT32_fsinfo_sec_err
1541
                                  <1>
1542 0000C49A BB[965E0100]
                                  <1>
                                                   ebx, DOSBootSectorBuff
1543
                                  <1>
1544 0000C49F 813B52526141
                                  <1>
                                            cmp
                                                   dword [ebx], 41615252h
1545 0000C4A5 751E
                                  <1>
                                                  short loc_read_FAT32_fsinfo_sec_stc
                                            jne
1546
                                  <1>
1547 0000C4A7 81BBE4010000727241- <1>
                                                   dword [ebx+484], 61417272h
                                            cmp
1547 0000C4B0 61
                                 <1>
1548 0000C4B1 7512
                                  <1>
                                            jne
                                                   short loc_read_FAT32_fsinfo_sec_stc
1549
                                  <1>
1550 0000C4B3 A1[44630100]
                                                   eax, [CFS_FAT32FSINFOSEC]
                                 <1>
                                            mov
1551 0000C4B8 8B8BE8010000
                                 <1>
                                                  ecx, [ebx+488] ; free cluster count
1552 0000C4BE 8B93EC010000
                                  <1>
                                                  edx, [ebx+492] ; first (next) free cluster
                                            mov
1553
                                  <1>
1554 0000C4C4 C3
                                  <1>
                                            retn
1555
                                  <1>
1556
                                  <1> loc_read_FAT32_fsinfo_sec_stc:
                                         ; \overline{15/10/2016} (0Bh -> 28)
1557
                                 <1>
1558 0000C4C5 B81C000000
                                 <1>
                                            mov
                                                  eax, 28 ; Invalid format!
1559 0000C4CA EB05
                                  <1>
                                            jmp
                                                  short loc_read_FAT32_fsinfo_sec_stc_retn
1560
                                  <1>
1561
                                  <1> loc_read_FAT32_fsinfo_sec_err:
                                         ; \overline{15}/10/\overline{2}016 (1\overline{5}h - \overline{>} 17)
1562
                                  <1>
1563
                                  <1>
                                            ; 23/03/2016 (15h)
1564 0000C4CC B811000000
                                  <1>
                                           mov eax, 17; Drive not ready or read error
1565
                                 <1>
1566
                                 <1> loc_read_FAT32_fsinfo_sec_stc_retn:
1567 0000C4D1 29DB
                                          sub ebx, ebx; 0
                                 <1>
1568 0000C4D3 F9
                                  <1>
                                            stc
1569 0000C4D4 C3
                                  <1>
                                            retn
1570
                                  <1>
1571
                                  <1> add_new_cluster:
                                        ; 15/10/2016
1572
                                  <1>
1573
                                  <1>
                                           ; 16/05/2016
                                           ; 18/03/2016, 24/03/2016
1574
                                  <1>
1575
                                  <1>
                                          ; 11/03/2016 (TRDOS 386 = TRDOS v2.0)
                                          ; 30/07/2011 (DRV_FAT.ASM)
; 11/09/2010
1576
                                  <1>
1577
                                  <1>
                                          ; INPUT ->
1578
                                  <1>
1579
                                  <1>
                                                  ESI = Logical dos drv desc. table address
                                          ;
1580
                                  <1>
                                                  EAX = Last cluster
                                           ;
1581
                                  <1>
                                           ; OUTPUT ->
1582
                                  <1>
                                          ;
                                                  ESI = Logical dos drv desc. table address
1583
                                  <1>
                                                  EAX = New Last cluster (next cluster)
                                                  cf = 1 -> error code in EAX (AL)
1584
                                  <1>
                                                  cf = 1 \rightarrow DX = sectors per cluster
1585
                                  <1>
1586
                                  <1>
                                                  ECX = Free sectors
1587
                                  <1>
                                           ; NOTE:
1588
                                  <1>
                                            ; This procedure does not update lm date&time !
1589
                                  <1>
1590
                                  <1>
                                            ; (Modified registers: EAX, EBX, ECX, EDX, EDI)
1591
                                  <1>
1592
                                  <1>
1593 0000C4D5 A3[74640100]
                                  <1>
                                            mov
                                                 [FAT_anc_LCluster], eax
                                  <1>
1595 0000C4DA E844F9FFFF
                                  <1>
                                            call get_first_free_cluster
1596 0000C4DF 720B
                                  <1>
                                                  short loc_add_new_cluster_retn
                                            jс
                                            ; EAX >= 2 and EAX < FFFFFFFF is valid
1597
                                  <1>
1598
                                  <1>
1599 0000C4E1 89C2
                                                  edx, eax
                                  <1>
                                            mov
1600
                                  <1>
1601 0000C4E3 42
                                  <1>
                                  <1>
1602
                                            ;jnz short loc_add_new_cluster_check_ffc_eax
1603 0000C4E4 7516
                                            jnz short loc_add_new_cluster_save_fcc
                                  <1>
1604
                                  <1>
1605
                                  <1> loc_add_new_cluster_no_disk_space_retn:
1606 0000C4E6 B827000000
                                  <1>
                                            mov eax, 27h; MSDOS err => insufficient disk space
1607
                                  <1> loc add new cluster stc retn:
1608 0000C4EB F9
                                  <1>
                                           stc
1609
                                  <1> loc add new cluster retn:
1610 0000C4EC 0FB65E13
                                           movzx ebx, byte [esi+LD_BPB+SecPerClust]
                                  <1>
1611 0000C4F0 8B4E74
                                 <1>
                                            mov ecx, [esi+LD_FreeSectors]
                                           ;xor edx, edx
1612
                                  <1>
1613
                                  <1>
                                            ;stc
1614 0000C4F3 C3
                                  <1>
                                           retn
1615
                                  <1>
1616
                                  <1> loc_anc_invalid_format_stc_retn:
1617 0000C4F4 F9
                                  <1>
                                           stc
                                  <1> loc add new cluster invalid format retn:
1618
1619
                                  <1>
                                           ; 15/10/2016 (0Bh -> 28)
1620 0000C4F5 B81C000000
                                  <1>
                                            mov eax, 28; Invalid format
1621 0000C4FA EBF0
                                  <1>
                                            jmp short loc_add_new_cluster_retn
                                  <1>
1622
```

```
1623
                                 <1> ;loc_add_new_cluster_check_ffc_eax:
1624
                                 <1>;
                                          cmp eax, 2
1625
                                 <1>;
                                                short loc_add_new_cluster_invalid_format_retn
                                          jb
1626
                                 <1>
                                 <1> loc_add_new_cluster_save_fcc:
1627
1628 0000C4FC A3[78640100]
                                <1>
                                         mov [FAT_anc_FFCluster], eax
                                <1>
1630 0000C501 83E802
                                         sub eax, 2
                                <1>
1631 0000C504 0FB65E13
                                <1>
                                          movzx ebx, byte [esi+LD_BPB+SecPerClust]
1632 0000C508 F7E3
                                          mul ebx
                                <1>
1633 0000C50A 09D2
                                <1>
                                          or edx, edx
1634 0000C50C 75E6
                                <1>
                                          jnz short loc and invalid format std retn
1635
                                <1>
1636
                                <1> loc_add_new_cluster_allocate_cluster:
                                      ; 18/03/2016
1637
                                <1>
1638 0000C50E 92
                                <1>
                                          xchg = edx, eax ; eax = 0
                                          ; 16/05/2016
1639
                                 <1>
                                        ;cmp [ClusterBuffer_Valid], al ; 0
1640
                                 <1>
1641
                                 <1>
                                          ;jna short loc_anc_clear_cluster_buffer
                                          ;; 'copy' command,
1642
                                 <1>
1643
                                 <1>
                                          ;; writing destination file clust after reading source file clust
                                          ;mov [ClusterBuffer_Valid], al ; 0 ; reset
;jmp short loc_add_new_cluster_write_nc_to_disk
1644
                                 <1>
                                 <1>
1645
                                 <1>
                                 <1> loc_anc_clear_cluster_buffer:
1647
                                        ; 11/03/2016
1648
                                <1>
                                <1>
                                          ; Clear buffer
1650 0000C50F BF00000700
                                          mov edi, Cluster_Buffer; 70000h (for current TRDOS 386 version)
                                <1>
1651 0000C514 89D9
                                <1>
                                                ecx, ebx ; sector count
1652 0000C516 C1E107
                                          shl ecx, 7; 1 sector = 512 bytes -> 128 double words
                                <1>
1653
                                <1>
                                          ;xor eax, eax; 0
1654 0000C519 F3AB
                                <1>
                                          rep
                                                stosd
1655
                                <1>
1656
                                <1> loc_add_new_cluster_write_nc_to_disk:
                                          ; 11/03/2016
1657
                                <1>
1658
                                <1>
                                          ; xchg eax, edx; edx = 0, eax = sector offset
1659 0000C51B 89D0
                                <1>
                                          mov eax, edx
                                          add eax, [esi+LD_DATABegin]
1660 0000C51D 034668
                                <1>
1661 0000C520 72D3
                                <1>
                                        jc short loc_add_new_cluster_invalid_format_retn
1662
                                <1>
1663 0000C522 89D9
                                <1>
                                          mov ecx, ebx; ECX = sectors per cluster (<256)
1664 0000C524 BB00000700
                                <1>
                                          mov
                                                ebx, Cluster Buffer
                                          call disk write
1665 0000C529 E81F330000
                                <1>
1666 0000C52E 7307
                                         jnc short loc add new cluster update fat nlc
                                <1>
1667
                                <1>
                                          ; 15/10/2016 (1Dh -> 18)
1668
                                <1>
1669 0000C530 B812000000
                                <1>
                                          mov eax, 18; Write Error
1670 0000C535 EBB4
                                <1>
                                          jmp short loc_add_new_cluster_stc_retn
1671
                                <1>
                                <1> loc_add_new_cluster_update_fat_nlc:
1673 0000C537 A1[78640100]
                                <1>
                                          mov eax, [FAT_anc_FFCluster]
1674 0000C53C 31C9
                                <1>
                                          xor
                                                ecx, ecx
1675 0000C53E 890D[A2600100]
                                          mov [FAT_ClusterCounter], ecx; 0; reset
                                <1>
1676 0000C544 49
                                <1>
                                          dec ecx; OFFFFFFFh
1677 0000C545 E8ACF9FFFF
                                <1>
                                      call
jnc
or
                                          call update_cluster
1678 0000C54A 7304
                                <1>
                                                short loc_add_new_cluster_update_fat_plc
1679 0000C54C 09C0
                                                eax, eax ; EAX = 0 -> cluster value is 0 or eocc
                                <1>
1680 0000C54E 759B
                                <1>
                                        jnz short loc_add_new_cluster_stc_retn
1681
                                <1>
1682
                                <1> loc_add_new_cluster_update_fat_plc:
1683 0000C550 A1[74640100]
                                <1> mov eax, [FAT_anc_LCluster]
1684 0000C555 8B0D[78640100]
                                <1>
                                          mov
                                                ecx, [FAT anc FFCluster]
1685 0000C55B E896F9FFFF
                                <1>
                                          call update cluster
1686 0000C560 7314
                                <1>
                                          jnc short loc_add_new_cluster_save_fat_buffer
                                         or
                                                eax, eax \overline{;} EAX = \overline{0} -> cluster value is 0 or eocc
1687 0000C562 09C0
                                <1>
1688 0000C564 7410
                                <1>
                                          jг
                                                short loc_add_new_cluster_save_fat_buffer
1689
                                <1>
                                 <1> loc_anc_save_fat_buffer_err_retn:
1690
1691
                                 <1>
                                          ;cmp byte [FAT_ClusterCounter], 1
1692
                                 <1>
                                          ;jb short loc_add_new_cluster_retn
1693
                                 <1>
1694 0000C566 66BB00FF
                                 <1>
                                                bx, 0FF00h ; recalculate free space (BL = 0)
                                                          ; (BH = FFh -> Use ESI as Drv Param. Tbl.)
1695
                                <1>
1696 0000C56A 50
                                <1>
                                          push eax
                                          call calculate fat freespace
1697 0000C56B E8D8FCFFFF
                                <1>
1698 0000C570 58
                                 <1>
                                          pop eax
                                          jmp loc_add_new_cluster_stc_retn
1699 0000C571 E975FFFFF
                                <1>
1700
                                 <1>
1701
                                 <1> loc_add_new_cluster_save_fat_buffer:
1702
                                          ;cmp byte [FAT BuffValidData], 2
                                 <1>
1703
                                 <1>
                                          ;jne short loc_add_new_cluster_calc_FAT_freespace
1704
                                 <1>
                                          ;Byte [FAT_BuffValidData] = 2
1705 0000C576 E838FCFFFF
                                          call save_fat_buffer
                                <1>
1706 0000C57B 72E9
                                <1>
                                                short loc anc save fat buffer err retn
                                <1>
1708
                                <1> loc_add_new_cluster_calc_FAT_freespace:
                                <1>
                                         ;mov eax, 1 ; Only one Cluster
                               <1>
1710 0000C57D A1[A2600100]
                                                eax, [FAT_ClusterCounter]
                                          mov
1711 0000C582 66BB01FF
                                <1>
                                                bx, OFFO1h; BH = FFh -> ESI -> Dos drv desc. table
                                          mov
1712
                                <1>
                                                ; BL = 1 \rightarrow add cluster
1713 0000C586 B301
                                <1>
                                        mov bl, 01h; BL = 1 \rightarrow add clusters
1714
                                <1>
                                          ; NOTE: EAX value will be added to Free Cluster Count
1715
                                          ; (Free Cluster Count is decreased when EAX value is negative)
                                <1>
1716 0000C588 E8BBFCFFFF
                              <1>
                                           call calculate_fat_freespace
1717
                                <1>
                                          ; ECX = 0 \rightarrow no error, ECX > 0 \rightarrow error or invalid return
                                          and ecx, ecx; ECX = 0 -> valid free sector count
1718 0000C58D 21C9
                                <1>
1719 0000C58F 7409
                                <1>
                                          jz short loc_add_new_cluster_return_cluster_number
                                <1>
1720
1721
                                <1> loc_add_new_cluster_recalc_FAT_freespace:
1722 0000C591 66BB00FF
                                         mov bx, 0FF00h ; recalculate free space
                                <1>
                                           call calculate_fat_freespace
1723 0000C595 E8AEFCFFFF
                                <1>
1724
                                <1>
                                          ; cf = 0
1725
                                <1> loc_add_new_cluster_return_cluster_number:
1726 0000C59A 89C1
                                        mov ecx, eax ; Free sector count
                                <1>
1727 0000C59C A1[78640100]
                                <1>
                                          mov eax, [FAT_anc_FFCluster]
```

```
movzx ebx, byte [esi+LD BPB+SecPerClust]
1728 0000C5A1 0FB65E13
                                 <1>
                                           ;mov edi, Cluster_Buffer
xor edx, edx
                                 <1>
1730 0000C5A5 31D2
                                 <1>
1731 0000C5A7 C3
                                 <1>
1732
                                 <1>
1733
                                 <1> write_cluster:
                                       ; 15/10/2016
1734
                                 <1>
                                           ; 21/03/2016 (TRDOS 386 = TRDOS v2.0)
1735
                                 <1>
1736
                                 <1>
1737
                                  <1>
                                           ; INPUT ->
1738
                                  <1>
                                                 EAX = Cluster Number (Sector index for SINGLIX FS)
1739
                                                  ESI = Logical DOS Drive Description Table address
                                  <1>
1740
                                 <1>
                                                 EBX = Cluster (File R/W) Buffer address (max. 64KB)
                                           ;
1741
                                 <1>
                                                Only for SINGLIX FS:
                                         ;
1742
                                  <1>
                                                 EDX = File Number (The 1st FDT address)
                                           ; OUTPUT ->
1743
                                  <1>
                                           ; cf = 1 -> Cluster can not be written onto disk
1744
                                  <1>
                                                    EAX > 0 -> Error number
1745
                                  <1>
1746
                                 <1>
                                                  cf = 0 -> Cluster has been written successfully
                                           ;
1747
                                 <1>
1748
                                 <1>
                                           ; (Modified registers: EAX, ECX, EBX, EDX)
1749
                                  <1>
                                           movzx ecx, byte [esi+LD BPB+BPB SecPerClust]
1750 0000C5A8 0FB64E13
                                 <1>
1751
                                 <1>
                                           ; CL = 1 = [esi+LD_FS_Reserved2] ; SectPerClust for Singlix FS
1752
                                 <1>
                                 <1> write_file_sectors: ; 16/03/2016
1753
1754 0000C5AC 807E0300
                                 <1>
                                       cmp byte [esi+LD_FATType], 0
1755 0000C5B0 761C
                                 <1>
                                           jna short write_fs_cluster
1756
                                 <1>
1757
                                <1> write_fat_file_sectors:
1758 0000C5B2 83E802
                                <1>
                                           sub eax, 2; Beginning cluster number is always 2
1759 0000C5B5 0FB65613
                                           movzx edx, byte [esi+LD BPB+BPB SecPerClust]; 18/03/2016
                                 <1>
1760 0000C5B9 F7E2
                                           mul edx
                                <1>
1761 0000C5BB 034668
                                 <1>
                                         add eax, [esi+LD_DATABegin] ; absolute address of the cluster
1762
                                 <1>
1763
                                           ; EAX = Disk sector address
                                 <1>
                                           ; ECX = Sector count
1764
                                 <1>
                                           ; EBX = Buffer address
1765
                                 <1>
                                           ; (EDX = 0)
1766
                                 <1>
1767
                                 <1>
                                           ; ESI = Logical DOS drive description table address
1768
                                 <1>
                                           call disk_write
jnc short wclust_retn
1769 0000C5BE E88A320000
                                 <1>
1770 0000C5C3 7306
                                 <1>
1771
                                 <1>
1772
                                 <1>
                                           ; 15/10/2016 (1Dh -> 18)
1773 0000C5C5 B812000000
                                           mov eax, 18; Drive not ready or write error!
                                 <1>
1774 0000C5CA C3
                                 <1>
1775
                                 <1>
1776
                                 <1> wclust_retn:
1777 0000C5CB 29C0
                                 <1>
                                         sub eax, eax; 0
1778 0000C5CD C3
                                 <1>
1779
                                 <1>
                                 <1> write_fs_cluster:
1780
                                       ; \overline{2}1/03/2016 (TRDOS 386 = TRDOS v2.0)
1781
                                 <1>
1782
                                  <1>
                                           ; Singlix FS
1783
                                  <1>
1784
                                  <1>
                                           ; EAX = Cluster number is sector index number of the file (eax)
1785
                                  <1>
1786
                                  <1>
                                           ; EDX = File number is the first File Descriptor Table address
                                           ; of the file. (Absolute address of the FDT).
1787
                                  <1>
1788
                                  <1>
1789
                                  <1>
                                           ; eax = sector index (0 for the first sector)
                                           ; edx = FDT0 address
1790
                                 <1>
1791
                                  <1>
                                                 ; 64 KB buffer = 128 sectors (limit)
                                           mov ecx, 128 ; maximum count of sectors (before eof)
call write_fs_sectors
1792 0000C5CE B980000000
                                 <1>
1793 0000C5D3 E801000000
                                 <1>
1794 0000C5D8 C3
                                 <1>
                                           retn
1795
                                 <1>
1796
                                 <1> write_fs_sectors:
1797
                                 <1>
                                        \frac{1}{21}, \frac{21}{03}, \frac{2016}{2016} (TRDOS 386 = TRDOS v2.0)
1798 0000C5D9 F9
                                 <1>
                                           stc
1799 0000C5DA C3
                                 <1>
                                           retn
1800
                                 <1>
1801
                                  <1> get_cluster_by_index:
                                         = ; 29/\overline{04/2016} (TRDOS 386 = TRDOS v2.0)
1802
                                 <1>
1803
                                  <1>
                                           ; INPUT ->
1804
                                  <1>
                                                 EAX = Beginning cluster
                                                 EDX = Sector index in disk/file section
1805
                                  <1>
1806
                                  <1>
                                                   (Only for SINGLIX file system!)
1807
                                  <1>
                                                  ECX = Cluster sequence number after the beginning cluster
                                           ;
1808
                                  <1>
                                                  ESI = Logical DOS Drive Description Table address
                                           ; OUTPUT ->
1809
                                  <1>
                                                  EAX = Cluster number
1810
                                  <1>
1811
                                  <1>
                                                  cf = 1 -> Error code in AL (EAX)
1812
                                  <1>
1813
                                  <1>
                                           ; (Modified registers: EAX, ECX, EBX, EDX)
                                 <1>
1815 0000C5DB 807E0301
                                 <1>
                                                  byte [esi+LD_FATType], 1
                                           cmp
                                           jb
1816 0000C5DF 721E
                                 <1>
                                                   short get_fs_section_by_index
                                 <1>
1818 0000C5E1 3B4E78
                                 <1>
                                           cmp
                                                  ecx, [esi+LD_Clusters]
1819 0000C5E4 7207
                                 <1>
                                           jb
                                                  short gcbi_1
                                 <1> gcbi_0:
1820
1821 0000C5E6 F9
                                 <1>
                                           stc
1822 0000C5E7 B823000000
                                 <1>
                                                  eax, 23h; Cluster not available!
                                           mov
1823
                                 <1>
                                                         ; MSDOS error code: FCB unavailable
1824 0000C5EC C3
                                <1>
                                           retn
1825
                                 <1> gcbi 1:
1826 0000C5ED 51
                                 <1>
                                           push
1827 0000C5EE E8D9F5FFFF
                                           call get_next_cluster
                                <1>
1828 0000C5F3 59
                                 <1>
                                                 ecx
1829 0000C5F4 7203
                                 <1>
                                                 short gcbi 3
                                           jс
1830 0000C5F6 E2F5
                                 <1>
                                           loop
                                                gcbi_1
                                 <1> qcbi 2:
1831
1832 0000C5F8 C3
                                 <1>
                                           retn
```

```
1833
                                                              <1> gcbi_3:
                                                              <1>
1834 0000C5F9 09C0
                                                                               or
                                                                                            eax, eax
1835 0000C5FB 74E9
                                                                                            short gcbi_0
                                                              <1>
                                                                                jz
1836 0000C5FD F5
                                                              <1>
                                                                                CMC
                                                                                           ; stc
1837 0000C5FE C3
                                                              <1>
                                                                               retn
1838
                                                              <1>
                                                              <1> get fs section by index:
1839
                                                                           \frac{1}{3} 29/04/\frac{1}{2}016 (TRDOS 386 = TRDOS v2.0)
1840
                                                              <1>
1841
                                                              <1>
                                                                                ; INPUT ->
1842
                                                              <1>
                                                                                            EAX = Beginning FDT number/address
                                                                               ;
1843
                                                              <1>
                                                                                            EDX = Sector index in disk/file section
1844
                                                              <1>
                                                                                            ECX = Sector sequence number after the beginning FDT
                                                                                ;
                                                                                           ESI = Logical DOS Drive Description Table address
1845
                                                              <1>
                                                                                ;
1846
                                                              <1>
                                                                             ; OUTPUT ->
                                                                                           EAX = FDT number/address
1847
                                                              <1>
                                                                             ;
1848
                                                              <1>
                                                                                            EDX = Sector index of the section (0,1,2,3,4...)
                                                                                           cf = 1 -> Error code in AL (EAX)
1849
                                                              <1>
                                                                                ;
1850
                                                              <1>
1851
                                                              <1>
                                                                                ; (Modified registers: EAX, ECX, EBX, EDX)
1852
                                                              <1>
1853 0000C5FF B8FFFFFFF
                                                              <1>
                                                                                mov
                                                                                            eax, OFFFFFFFh
1854 0000C604 C3
                                                              <1>
                                                                                retn
1855
                                                              <1>
                                                              <1> get_last_section:
1856
                                                                            -; \overline{22}/10/2016 (TRDOS 386 = TRDOS v2.0)
1857
                                                              <1>
1858
                                                              <1>
                                                                                ; INPUT ->
1859
                                                              <1>
                                                                                           EAX = (The 1st) FDT number/address
                                                                               ;
                                                                             ;
1860
                                                              <1>
                                                                                           ESI = Logical DOS Drive Description Table address
1861
                                                              <1>
                                                                                ; OUTPUT ->
1862
                                                              <1>
                                                                                           EAX = FDT number/address of the last section
                                                                                ;
1863
                                                              <1>
                                                                                            EDX = Last sector of the section (0,1,2,3,4...)
1864
                                                               <1>
                                                                                           [glc index] = sector index number of the last sector
1865
                                                              <1>
                                                                                                                    (for file, not for the last section)
1866
                                                              <1>
1867
                                                              <1>
                                                                                            cf = 1 -> Error code in AL (EAX)
                                                                               ;
1868
                                                              <1>
1869
                                                               <1>
                                                                               ; (Modified registers: EAX, ECX, EBX, EDX)
1870
                                                              <1>
1871 0000C605 B800000000
                                                              <1>
                                                                                           eax, 0
                                                                                mov
                                                                                mov edx, 0
1872 0000C60A BA00000000
                                                              <1>
1873 0000C60F C3
                                                              <1>
                                                                                retn
2310
                                                                     %include 'trdosk6.s'; 24/01/2016
                                                              1
                                                               <1> ; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.2) - MAIN PROGRAM : trdosk6.s
     2
                                                               <1> ; ------
     3
     4
                                                               <1> ; Last Update: 03/08/2020
                                                               <1> ; ------
                                                               <1> ; Beginning: 24/01/2016
     6
                                                               <1> ; ------
     7
     8
                                                               <1> : Assembler: NASM version 2.11 (trdos386.s)
                                                               9
   10
                                                               <1>; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
                                                               <1>; u1.s (27/17/2015), u2.s (03/01/2016)
   11
                                                               12
   13
                                                               <1>; Derived from TRDOS Operating System v1.0 (8086) source code by Erdogan Tan
   14
                                                               <1>; TRDOS2.ASM (09/11/2011)
                                                                                                                     ______
   15
                                                              <1>; INT_21H.ASM (c) 2009-2011 Erdogan TAN [14/11/2009] Last Update: 08/11/2011
   16
   17
                                                               <1>
                                                              <1> sysent: ; < enter to system call >
   18
                                                                        ; 17/03/2017
   19
                                                              <1>
   20
                                                               <1>
                                                                               ; 03/03/2017
                                                                               ; 19/02/2017
   21
                                                              <1>
   22
                                                               <1>
                                                                             ; 13/01/2017
                                                                             ; 06/06/2016
   23
                                                              <1>
   24
                                                              <1>
                                                                               ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
                                                                               ; 16/04/2015 - 19/10/2015 (Retro UNIX 386 v1)
   25
                                                               <1>
   26
                                                              <1>
                                                                              ; 10/04/2013 - 18/01/2014 (Retro UNIX 8086 v1)
   27
                                                              <1>
                                                                                ; 'unkni' or 'sysent' is sytem entry from various traps.
   28
                                                              <1>
   29
                                                              <1>
                                                                                ; The trap type is determined and an indirect jump is made to
   30
                                                               <1>
                                                                                ; the appropriate system call handler. If there is a trap inside
                                                                                ; the system a jump to panic is made. All user registers are saved % \left( 1\right) =\left( 1\right) +\left( 1\right) 
   31
                                                              <1>
   32
                                                              <1>
                                                                                ; and u.sp points to the end of the users stack. The sys (trap)
                                                                                ; instructor is decoded to get the the system code part (see
   33
                                                               <1>
                                                                                ; trap instruction in the PDP-11 handbook) and from this
   34
                                                              <1>
   35
                                                               <1>
                                                                                ; the indirect jump address is calculated. If a bad system call is
   36
                                                              <1>
                                                                                ; made, i.e., the limits of the jump table are exceeded, 'badsys'
   37
                                                               <1>
                                                                                ; is called. If the call is legitimate control passes to the
   38
                                                                                ; appropriate system routine.
                                                               <1>
   39
                                                               <1>
                                                                                 ; Calling sequence:
                                                               <1>
                                                                                             Through a trap caused by any sys call outside the system.
    41
                                                               <1>
                                                               <1>
    42
   43
                                                               <1>
                                                                                            Arguments of particular system call.
   44
                                                               <1>
    45
                                                               <1>
   46
                                                                                ; Retro UNIX 8086 v1 modification:
                                                              <1>
   47
                                                              <1>
                                                                                               System call number is in EAX register.
   48
                                                              <1>
                                                                                               Other parameters are in EDX, EBX, ECX, ESI, EDI, EBP
   49
                                                              <1>
                                                                                             registers depending of function details.
   50
                                                               <1>
   51
                                                              <1>
                                                                                ; 16/04/2015
                                                              <1>
   52
   53 0000C610 368925[5C030300]
                                                              <1>
                                                                                 mov [ss:u.sp], esp; Kernel stack points to return address
   54
                                                              <1>
   55
                                                              <1>
                                                                                ; save user registers
                                                                                push ds
   56 0000C617 1E
                                                              <1>
                                                                                push
   57 0000C618 06
                                                              <1>
    58 0000C619 0FA0
                                                                                push fs
                                                              <1>
   59 0000C61B 0FA8
                                                              <1>
                                                                                push gs
    60 0000C61D 60
                                                              <1>
                                                                                pushad ; eax, ecx, edx, ebx, esp -before pushad-, ebp, esi, edi
   61
                                                              <1>
    62
                                                              <1>
                                                                                 ; ESPACE = [ss:u.sp] - esp ; 4*12 = 48 ; 17/09/2015 ; 06/06/2016
   63
                                                              <1>
                                                                                             (ESPACE is size of space in kernel stack
```

```
64
                                <1>
                                                for saving/restoring user registers.)
                                <1>
                                          ;
                                          push eax ; 01/07/2015
 66 0000C61E 50
                                <1>
 67 0000C61F 66B81000
                                <1>
                                          mov ax, KDATA
                                                  ds, ax
 68 0000C623 8ED8
                                <1>
                                         mov
 69 0000C625 8EC0
                                <1>
                                           mov
                                                    es, ax
 70 0000C627 8EE0
                                <1>
                                          mov
                                                 fs, ax
 71 0000C629 8EE8
                                <1>
                                          mov
                                                 gs, ax
 72 0000C62B A1[C0580100]
                                <1>
                                         mov eax, [k_page_dir]
                                         mov cr3, eax
 73 0000C630 0F22D8
                                <1>
                                         pop eax ; 01/07/2015
 74 0000C633 58
                                <1>
                                <1>
                                          ; 19/10/2015
 76 0000C634 FC
                                <1>
                                         cld
 77
                                <1>
 78 0000C635 FE05[5B030300]
                                <1>
                                          inc
                                               byte [sysflg]
                                                ; incb sysflg / indicate a system routine is in progress
 79
                                <1>
 80 0000C63B FB
                                            sti ; 18/01/2014
                                <1>
 81 0000C63C 0F85DF9DFFFF
                                          jnz panic ; 24/05/2013
                                <1>
                                <1>
                                                ; beq 1f
                                                ; jmp panic ; / called if trap inside system
 83
                                <1>
 84
                                <1> ;1:
                                <1>
                                          ; 17/03/2017
 86 0000C642 80642438FE
                                          and byte [esp+ESPACE+8], ~1; clear carry flag
                                <1>
                                <1>
                                          ; 16/04/2015
 88
                                <1>
 89 0000C647 A3[64030300]
                                <1>
                                               [u.r0], eax
                                          mov
 90 0000C64C 8925[60030300]
                                <1>
                                                [u.usp], esp; kernel stack points to user's registers
 91
                                <1>
                                          ; 13/01/2017 (TRDOS 386 Feaure only !)
                                <1>
 93 0000C652 803D[D4030300]00
                                <1>
                                          cmp byte [u.t_lock], 0 ; timer interrupt lock ?
 94 0000C659 0F879D010000
                                <1>
                                                                      ; yes, sys release only !!!
 95
                                <1>
 96
                                                ; mov $s.syst+2,clockp
                                <1>
 97
                                <1>
                                                ; mov r0,-(sp) / save user registers
 98
                                <1>
                                                ; mov sp,u.r0 / pointer to bottom of users stack
                                                         ; / in u.r0
 99
                                <1>
                                                ; mov r1,-(sp)
100
                                <1>
101
                                <1>
                                                ; mov r2,-(sp)
102
                                <1>
                                                ; mov r3,-(sp)
103
                                <1>
                                                ; mov r4,-(sp)
                                                ; mov r5, -(sp)
104
                                <1>
                                                ; mov ac, -(sp) / "accumulator" register for extended
105
                                <1>
                                                           ; / arithmetic unit
                                <1>
106
                                                ; mov mq,-(sp) / "multiplier quotient" register for the
107
                                <1>
108
                                <1>
                                                           ; / extended arithmetic unit
                                                ; mov sc,-(sp) / "step count" register for the extended
109
                                <1>
                                                           ; / arithmetic unit
110
                                <1>
                                <1>
                                                ; mov sp,u.sp / u.sp points to top of users stack
111
112
                                <1>
                                                ; mov 18.(sp),r0 / store pc in r0
                                                                                    10400xxx
113
                                <1>
                                                ; mov -(r0), r0 / sys inst in r0
114
                                <1>
                                                ; sub $sys,r0 / get xxx code
115 0000C65F C1E002
                                <1>
                                               eax, 2
                                                ; asl r0 \!\!\!/ multiply by 2 to jump indirect in bytes
116
                                <1>
117 0000C662 3DB8000000
                                <1>
                                                eax, end_of_syscalls - syscalls
118
                                <1>
                                                ; cmp r0,$2f-1f / limit of table (35) exceeded
119
                                <1>
                                               short badsys
                                                ; bhis badsys / yes, bad system call
                                <1>
121 0000C667 F5
                                <1>
                                          CMC
122 0000C668 9C
                                <1>
                                          pushf
123 0000C669 50
                                          push eax
                                <1>
124 0000C66A 8B2D[5C030300]
                                <1>
                                                ebp, [u.sp] ; Kernel stack at the beginning of sys call
125 0000C670 B0FE
                                                al, OFEh ; 111111110b
                                <1>
                                          mov
                                                al, 0 ; al = al + cf
                                          adc
126 0000C672 1400
                                <1>
127 0000C674 204508
                                <1>
                                          and [ebp+8], al ; flags (reset carry flag)
128
                                <1>
                                               ; bic $341,20.(sp) / set users processor priority to 0
129
                                <1>
                                                             ; / and clear carry bit
130 0000C677 5D
                                <1>
                                         pop
                                                ebp ; eax
131 0000C678 9D
                                <1>
                                         popf
132 0000C679 0F8208020000
                                <1>
                                          jс
                                                   badsys
133 0000C67F A1[64030300]
                                <1>
                                          mov eax, [u.r0]
134
                                <1>
                                          ; system call registers: EAX, EDX, ECX, EBX, ESI, EDI
135 0000C684 FFA5[8AC60000]
                                          jmp dword [ebp+syscalls]
                                <1>
                                <1>
                                                ; jmp *1f(r0) / jump indirect thru table of addresses
136
137
                                <1>
                                                            ; / to proper system routine.
138
                                <1> syscalls: ; 1:
                                         ; 31/12/2017
139
                                <1>
140
                                <1>
                                         ; 28/02/2017
                                        ; 20/02/2017
141
                                <1>
                                <1>
                                         ; 19/02/2017
142
                                         ; 15/10/2016
143
                                <1>
144
                                <1>
                                        ; 20/05/2016
145
                                <1>
                                        ; 19/05/2016
                                          ; 16/05/2016
146
                                <1>
147
                                <1>
                                          ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
                                          ; 21/09/2015
                                <1>
148
149
                                <1>
                                          ; 01/07/2015
150
                                <1>
                                          ; 16/04/2015 (32 bit address modification)
                                          151 0000C68A [41E70000]
                                <1>
152 0000C68E [E9C80000]
                                <1>
153 0000C692 [BECA0000]
                                                     ; 2
                                <1>
                                          dd sysfork
154 0000C696 [F1CE0000]
                                          dd sysread ; 3
                                <1>
155 0000C69A [10CF0000]
                                          dd syswrite ; 4
                                <1>
156 0000C69E [A7CC0000]
                                          dd sysopen ; 5
                                <1>
157 0000C6A2 [C8CE0000]
                                <1>
                                          dd sysclose ; 6
                                          dd syswait ; 7
dd syscreat ; 8
158 0000C6A6 [40CA0000]
                                <1>
159 0000C6AA [D6CB0000]
                                <1>
160 0000C6AE [95F50000]
                                <1>
                                          dd sysrename ; 9 ; TRDOS 386, Rename File (31/12/2017)
161 0000C6B2 [10F10000]
                                          dd sysdelete ; 10 ; TRDOS 386, Delete File (29/12/2017)
                                <1>
162 0000C6B6 [24DB0000]
                                          dd sysexec ; 11
                                <1>
                                          dd syschdir ; 12
163 0000C6BA [3AF20000]
                                <1>
164 0000C6BE [FFF30000]
                                          dd systime \,; 13 ; TRDOS 386, Get Sys Date&Time (30/12/2017)
                                <1>
165 0000C6C2 [8ACE0000]
                                <1>
                                          dd sysmkdir ; 14
166 0000C6C6 [6EF20000]
                                          dd syschmod ; 15 ; TRDOS 386, Change Attributes (30/12/2017)
                                <1>
167 0000C6CA [77F10000]
                                <1>
                                          dd sysrmdir ; 16 ; TRDOS 386, Remove Directory (29/12/2017)
168 0000C6CE [FFDD0000]
                                <1>
                                          dd sysbreak ; 17
```

```
169 0000C6D2 [54F30000]
                                  <1>
                                            dd sysdrive ; 18 ; TRDOS 386, Get/Set Current Drv (30/12/2017)
                                            dd sysseek ; 19
dd systell ; 20
170 0000C6D6 [40DE0000]
                                  <1>
171 0000C6DA [52DE0000]
                                  <1>
172 0000C6DE [B6F60000]
                                  <1>
                                            dd sysmem
                                                       ; 21 ; TRDOS 386, Get Total&Free Mem (31/12/2017)
                                            dd sysprompt ; 22 ; TRDOS 386, Change Cmd Prompt (31/12/2017)
173 0000C6E2 [ECF60000]
                                  <1>
174 0000C6E6 [2EF70000]
                                  <1>
                                            dd syspath ; 23 ; TRDOS 386, Get/Set Run Path (31/12/2017)
175 0000C6EA [9BF70000]
                                                         ; 24 ; TRDOS 386, Get/Set Env Vars (31/12/2017)
                                  <1>
                                            dd sysenv
                                            dd sysstime ; 25 ; TRDOS 386, Set Sys Date&Time (30/12/2017)
176 0000C6EE [80F40000]
                                  <1>
177 0000C6F2 [B8DE0000]
                                  <1>
                                            dd sysquit ; 26
                                            dd sysintr ; 27
178 0000C6F6 [ACDE0000]
                                  <1>
179 0000C6FA [A3F30000]
                                  <1>
                                            dd sysdir
                                                       ; 28 ; TRDOS 386, Get Curr Drive&Dir (30/12/2017)
180 0000C6FE [A7CF0000]
                                  <1>
                                            dd sysemt
                                                         ; 29
181 0000C702 [DEF30000]
                                            dd sysldrvt ; 30 ; TRDOS 386, Get Logical DOS DDT (30/12/2017)
                                  <1>
182 0000C706 [58D10000]
                                  <1>
                                            dd sysvideo ; 31 ; TRDOS 386 Video Functions (16/05/2016)
183 0000C70A [6B010100]
                                            dd sysaudio ; 32 ; TRDOS 386 Audio Functions (16/05/2016)
                                  <1>
184 0000C70E [C0CF0000]
                                  <1>
                                            dd systimer ; 33 ; TRDOS 386 Timer Functions (18/05/2016)
185 0000C712 [F9DE0000]
                                  <1>
                                            dd syssleep ; 34 ; Retro UNIX 8086 v1 feature only !
                                                              ; 11/06/2014
186
                                  <1>
187 0000C716 [28DF0000]
                                                        ; 35 ; Retro UNIX 386 v1 feature only !
                                  <1>
                                            dd sysmsg
                                                              ; 01/07/2015
                                  <1>
188
189 0000C71A [FFDF0000]
                                  <1>
                                            dd sysgeterr ; 36 ; Retro UNIX 386 v1 feature only !
190
                                  <1>
                                                              ; 21/09/2015 - get last error number
191 0000C71E [E7F00000]
                                            dd sysfpstat ; 37 ; TRDOS 386 FPU state option (28/02/2017)
                                  <1>
192 0000C722 [50E70000]
                                  <1>
                                            dd syspri ; 38 ; change priority - TRDOS 386 (20/05/2016)
                                            dd sysrele ; 39 ; TRDOS 386 (19/05/2016) (0 -> 39)
193 0000C726 [FCC70000]
                                  <1>
194 0000C72A [83E80000]
                                  <1>
                                            dd sysfff
                                                         ; 40 ; Find First File - TRDOS 386 (15/10/2016)
195 0000C72E [62E90000]
                                  <1>
                                            dd sysfnf
                                                         ; 41 ; Find Next File - TRDOS 386 (15/10/2016)
196 0000C732 [D2EF0000]
                                            dd sysalloc ; 42 ; Allocate contiguous memory block/pages
                                  <1>
                                  <1>
                                                              ; TRDOS 386 (19/02/2017) DMA buff fuctions
197
                                            dd sysdalloc ; 43 ; Deallocate contiguous memory block/pages
198 0000C736 [90F00000]
                                  <1>
199
                                  <1>
                                                              ; TRDOS 386 (19/02/2017) DMA buff fuctions
200 0000C73A [CBF00000]
                                  <1>
                                            dd syscalbac ; 44 ; IRQ Callback and Signal Response Byte
                                                              ; service setup - TRDOS 386 (20/02/2017)
201
                                  <1>
202
                                  <1>
                                                               ; 28/08/2017 (20/08/2017)
                                                        ; 45 ; TRDOS 386 - (ISA) DMA service
203 0000C73E [FC090100]
                                  <1>
                                            dd sysdma
204
                                  <1>
205
                                  <1> end_of_syscalls:
206
                                  <1>
207
                                  <1> error:
                                          ; 18/05/2016
208
                                  <1>
209
                                  <1>
                                           ; 13/05/2016
                                           ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
; 16/04/2015 - 17/09/2015 (Retro UNIX 386 v1)
210
                                  <1>
211
                                  <1>
                                            ; 10/04/2013 - 07/08/2013 (Retro UNIX 8086 v1)
212
                                  <1>
213
                                  <1>
214
                                  <1>
                                            ; 'error' merely sets the error bit off the processor status (c-bit)
                                            ; then falls right into the 'sysret', 'sysrele' return sequence.
215
                                  <1>
                                  <1>
216
217
                                  <1>
                                           ; INPUTS -> none
218
                                            ; OUTPUTS ->
                                  <1>
                                                  processor status - carry (c) bit is set (means error)
219
                                  <1>
220
                                  <1>
221
                                  <1>
                                            ; 26/05/2013 (Stack pointer must be reset here!
                                  <1>
                                                         Because, jumps to error procedure
222
223
                                  <1>
                                                         disrupts push-pop nesting balance)
                                            ;
224
                                  <1>
225 0000C742 8B2D[5C030300]
                                  <1>
                                                   ebp, [u.sp]; interrupt (system call) return (iretd) address
                                            mov
226 0000C748 804D0801
                                  <1>
                                                  byte [ebp+8], 1 ; set carry bit of flags register
227
                                  <1>
                                                                 ; (system call will return with cf = 1)
                                                   ; bis \$1,20.(r1) / set c bit in processor status word below
228
                                  <1>
229
                                  <1>
                                                                  ; / users stack
230
                                  <1>
                                            ; 17/09/2015
231 0000C74C 83ED30
                                                  ebp, ESPACE ; 48 ; total size of stack frame ('sysdefs.inc')
                                  <1>
                                            sub
232
                                  <1>
                                                                 ; for saving/restoring user registers
233
                                  <1>
                                                 ebp, [u.usp]
                                            ; cmp
234
                                  <1>
                                            ;je
                                                   short err0
235 0000C74F 892D[60030300]
                                                   [u.usp], ebp
                                  <1>
                                            mov
236
                                  <1> ;err0:
237
                                  <1>
                                            ; 01/09/2015
238 0000C755 8B25[60030300]
                                  <1>
                                                                    ; Retro Unix 8086 v1 modification!
                                            mov esp, [u.usp]
239
                                  <1>
                                                                    ; 10/04/2013
                                                                    ; (If an I/O error occurs during disk I/O,
240
                                  <1>
                                                                    ; related procedures will jump to 'error'
                                  <1>
241
242
                                  <1>
                                                                    ; procedure directly without returning to
                                                                    ; the caller procedure. So, stack pointer
243
                                  <1>
244
                                  <1>
                                                                           ; must be restored here.)
245
                                  <1>
                                           ; 13/05/2016
                                            ; NOTE: (The last) error code is in 'u.error', it can be retrieved by
246
                                  <1>
247
                                                   'get last error' system call later.
                                  <1>
248
                                  <1>
                                            ; 03/09/2015 - 09/06/2015 - 07/08/2013
249
                                  <1>
250 0000C75B C605[C6030300]00
                                  <1>
                                            mov byte [u.kcall], 0 ; namei_r, mkdir_w reset
251
                                  <1>
252
                                  <1> sysret: ; < return from system call>
253
                                          ; 01/03/2017
                                  <1>
254
                                  <1>
                                            ; 28/02/2017
                                           ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
255
                                  <1>
                                            ; 16/04/2015 - 10/09/2015 (Retro UNIX 386 v1)
256
                                  <1>
                                            ; 10/04/2013 - 23/02/2014 (Retro UNIX 8086 v1)
257
                                  <1>
258
                                  <1>
259
                                  <1>
                                            ; 'sysret' first checks to see if process is about to be
                                            ; terminated (u.bsys). If it is, 'sysexit' is called.
260
                                  <1>
                                            ; If not, following happens:
261
                                  <1>
                                                   1) The user's stack pointer is restored.
262
                                  <1>
263
                                  <1>
                                                   2) r1=0 and 'iget' is called to see if last mentioned
264
                                  <1>
                                                      i-node has been modified. If it has, it is written out
                                                      via 'ppoke'.
265
                                  <1>
                                  <1>
                                                   3) If the super block has been modified, it is written out
266
267
                                  <1>
                                                      via 'ppoke'.
                                                   4) If the dismountable file system's super block has been
268
                                  <1>
269
                                  <1>
                                                     modified, it is written out to the specified device
270
                                  <1>
                                                     via 'ppoke'.
                                                   5) A check is made if user's time quantum (uquant) ran out
271
                                  <1>
272
                                  <1>
                                                      during his execution. If so, 'tswap' is called to give
273
                                  <1>
                                                      another user a chance to run.
```

```
274
                                               6) 'sysret' now goes into 'sysrele'.
                               <1>
275
                                <1>
                                                   (See 'sysrele' for conclusion.)
                                        ;
276
                               <1>
277
                               <1>
                                         ; Calling sequence:
278
                               <1>
                                         ; jump table or 'br sysret'
279
                               <1>
                                         ; Arguments:
280
                               <1>
281
                               <1>
                                        i ......
282
                               <1>
                                         ; ((AX=r1 for 'iget' input))
283
                               <1>
284
                               <1>
285 0000C762 31C0
                               <1>
                                               eax, eax ; 28/02/2017
                                         xor
                               <1> sysret0: ; 29/07/2015 (eax = 0, jump from sysexec)
286
287 0000C764 FEC0
                               <1>
                                        inc al ; 04/05/2013
288 0000C766 3805[B2030300]
                                         cmp [u.bsys], al; 1
                               <1>
                                               ; tstb u.bsys / is a process about to be terminated because
289
                               <1>
290 0000C76C 0F8377010000
                                         jnb sysexit ; 04/05/2013
                               <1>
291
                                              ; bne sysexit / of an error? yes, go to sysexit
                               <1>
292
                               <1>
                                         ;mov esp, [u.usp]; 24/05/2013 (that is not needed here)
293
                               <1>
                                               ; mov u.sp,sp / no point stack to users stack
294 0000C772 FEC8
                               <1>
                                              al ; mov ax, 0
                                               ; clr r1 / zero r1 to check last mentioned i-node
295
                               <1>
                                         call iget
296 0000C774 E8CD300000
                               <1>
297
                               <1>
                                               ; jsr r0, iget / if last mentioned i-node has been modified
                                                          ; / it is written out
298
                               <1>
                                        ; 10/01/2017
299
                               <1>
300
                               <1>
                                        ; 09/01/2017
301
                               <1> ;sysrele: ; < release >
302
                                       ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
                               <1>
                                         ; 16/04/2015 - 14/10/2015 (Retro UNIX 386 v1)
303
                               <1>
304
                               <1>
                                         ; 10/04/2013 - 07/03/2014 (Retro UNIX 8086 v1)
305
                                <1>
                                        ; 'sysrele' first calls 'tswap' if the time quantum for a user is
306
                               <1>
307
                               <1>
                                        ; zero (see 'sysret'). It then restores the user's registers and
                                        ; turns off the system flag. It then checked to see if there is
308
                               <1>
309
                               <1>
                                         ; an interrupt from the user by calling 'isintr'. If there is,
310
                                        ; the output gets flashed (see isintr) and interrupt action is
311
                               <1>
                                       ; taken by a branch to 'intract'. If there is no interrupt from
312
                               <1>
                                        ; the user, a rti is made.
313
                               <1>
314
                               <1>
                                       ; Calling sequence:
                                        ; Fall through a 'bne' in 'sysret' & ?
315
                                <1>
316
                               <1>
                                         ; Arguments:
317
                               <1>
318
                               <1>
                                        319
                               <1>
                                        ; 23/02/2014 (swapret)
320
                               <1>
321
                               <1>
                                        ; 22/09/2013
322
                               <1> sysrel0: ;1:
323 0000C779 803D[A8030300]00
                                        cmp byte [u.quant], 0; 16/05/2013
                               <1>
324
                               <1>
                                              ; tstb uquant / is the time quantum 0?
325 0000C780 7705
                               <1>
                                                  short swapret
                                          jа
                                              ; bne 1f / no, don't swap it out
326
                               <1>
327
                               <1> sysrelease: ; 07/12/2013 (jump from 'clock')
328 0000C782 E8821E0000
                               <1>
                                         call tswap
329
                               <1>
                                               ; jsr r0, tswap / yes, swap it out
330
                               <1>
                               <1>; Retro Unix 8086 v1 feature: return from 'swap' to 'swapret' address.
331
332
                                <1> swapret: ;1:
                                     ; 10/09/2015
333
                               <1>
334
                               <1>
                                        ; 01/09/2015
                                       ; 14/05/2015
335
                                <1>
336
                               <1>
                                        ; 16/04/2015 (Retro UNIX 386 v1 - 32 bit, pm modifications)
337
                               <1>
                                        ; 26/05/2013 (Retro UNIX 8086 v1)
338
                               <1>
                                        ; cli
339
                               <1>
                                         ; 24/07/2015
340
                               <1>
                                        ;; 'esp' must be already equal to '[u.usp]' here !
                               <1>
341
342
                               <1>
                                         ;; mov esp, [u.usp]
343
                               <1>
344
                               <1>
                                         ; 22/09/2013
345 0000C787 E8BB300000
                               <1>
                                        call isintr
                                         ; 20/10/2013
346
                               <1>
347 0000C78C 7405
                               <1>
                                        jz
                                              short sysrel1
348 0000C78E E83F010000
                               <1>
                                        call intract
                                               ; jsr r0, isintr / is there an interrupt from the user
349
                               <1>
350
                               <1>
                                               ; br intract / yes, output gets flushed, take interrupt
                                                             ; / action
351
                               <1>
                                <1> sysrel1:
352
353 0000C793 FA
                                              ; 14/10/2015
                               <1> cli
354
                                <1> sysrel2:
                                         ; 28/02/2017
355
                                <1>
                               <1>
                                         ; Check if there is a (delayed) callback for current user/process
357 0000C794 A0[D7030300]
                               <1>
                                         mov al, [u.irqwait]
358 0000C799 240F
                               <1>
                                         and
                                              al, OFh; is there a waiting IRQ callback service?
359 0000C79B 7444
                               <1>
                                         jz
                                               short sysrel8 ; no
360
                               <1>
                               <1>
                                        ; Set return to IRQ callback service and return from the service
361
362 0000C79D 0FB6D8
                               <1>
                                        movzx ebx, al
363 0000C7A0 883D[D7030300]
                               <1>
                                         mov [u.irqwait], bh; 0; reset
364 0000C7A6 8A9B[96160100]
                               <1>
                                         mov
                                               bl, [ebx+IRQenum]; (available) IRQ index +1 (1 to 9)
365
                               <1>
                                         ; 01/03/2017
366 0000C7AC FECB
                               <1>
                                         dec bl ; IRQ index number, 0 to 8
367 0000C7AE 7831
                               <1>
                                               short sysrel8 ; 0 -> FFh (not in use!?)
                                         js
368
                               <1>
                                         ;
369 0000C7B0 A0[B3030300]
                               <1>
                                         mov
                                               al, [u.uno] ; current process (user) number
370 0000C7B5 3883[F26A0100]
                               <1>
                                               [ebx+IRQ.owner], al
                                         cmp
371 0000C7BB 7524
                                               short sysrel8; it is not the current user/process!?
                               <1>
                                         jne
372 0000C7BD F683[046B0100]01
                               <1>
                                         test
                                               byte [ebx+IRQ.method], 1 ; callback ?
373 0000C7C4 741B
                                               short sysrel8; not a callback method !?
                               <1>
                                         jz
374
                               <1>
375 0000C7C6 8B93[166B0100]
                               <1>
                                               edx, [ebx+IRQ.addr] ; IRQ callback service address (virtual)
                                         mov
376 0000C7CC C605[D8030300]01
                               <1>
                                         mov
                                               byte [u.r lock], 1 ; IRQ callback service in progress flag
                               <1>
377
378 0000C7D3 E8D91E0000
                               <1>
                                         call wswap; save user's registers & status
```

```
379
                                                              (for return from IRQ callback service)
                                 <1>
                                 <1>
381 0000C7D8 8B2D[5C030300]
                                 <1>
                                          mov
                                                 ebp, [u.sp]; kernel's stack, points to EIP (user)
382 0000C7DE 895500
                                 <1>
                                                 [ebp], edx ; IRQ call back service address
                                 <1> sysrel8:
383
384 0000C7E1 FE0D[5B030300]
                                 <1>
                                                 byte [sysflg]
                                          dec
                                                 ; decb sysflg / turn system flag off
                                 <1>
386
                                 <1>
387 0000C7E7 A1[B8030300]
                                <1>
                                                 eax, [u.pgdir]
388 0000C7EC 0F22D8
                                                cr3, eax ; 1st PDE points to Kernel Page Table 0 (1st 4 MB)
                                <1>
                                          mov
389
                                <1>
                                                         ; (others are different than kernel page tables)
                                          ; 10/09/2015
390
                                 <1>
391 0000C7EF 61
                                <1>
                                          popad ; edi, esi, ebp, temp (icrement esp by 4), ebx, edx, ecx, eax
392
                                 <1>
                                                ; mov (sp)+,sc / restore user registers
393
                                 <1>
                                                ; mov (sp) + mq
                                                ; mov (sp)+,ac
394
                                 <1>
395
                                 <1>
                                                ; mov (sp) + , r5
396
                                 <1>
                                                ; mov (sp)+,r4
397
                                 <1>
                                                 ; mov (sp)+,r3
398
                                 <1>
                                                 ; mov (sp)+,r2
399
                                 <1>
400 0000C7F0 A1[64030300]
                                 <1>
                                                 eax, [u.r0] ; ((return value in EAX))
                                          mov
401 0000C7F5 0FA9
                                 <1>
                                          pop
                                                 qs
402 0000C7F7 0FA1
                                 <1>
                                          pop
403 0000C7F9 07
                                <1>
                                          pop
                                                 es
404 0000C7FA 1F
                                 <1>
                                                 ds
                                          pop
                                 <1>
                                                word [esp+8], 200h; 22/01/2017; force enabling interrupts
                                          ;or
406 0000C7FB CF
                                          iretd
                                <1>
407
                                 <1>
                                                 ; rti / no, return from interrupt
408
                                 <1>
409
                                 <1> sysrele:
                                       ; 24/03/2017
410
                                 <1>
                                          ; 28/02/2017
411
                                 <1>
412
                                 <1>
                                          ; 27/02/2017
                                          ; 29/01/2017
413
                                 <1>
                                          ; 14/01/2017
414
                                 <1>
415
                                 <1>
                                          ; 13/01/2017
                                          ; 09/01/2017, 10/01/2017, 12/01/2017
                                 <1>
416
                                          ; Major modification for TRDOS 386 (CallBack return)
417
                                 <1>
                                 <1>
418
419
                                 <1>
                                          ; 'sysrele' system call restores previously saved
420
                                 <1>
                                          ; registers and addresses of the process
                                          ; (Main purpose -in TRDOS 386- is to return from
                                 <1>
421
                                 <1>
                                          ; timer callback service routine in ring 3 -user mode-.)
422
423
                                 <1>
424
                                 <1>
                                          ; check if the process is in timer callback phase
425 0000C7FC 803D[D4030300]00
                                 <1>
                                          cmp byte [u.t lock], 0; TIMER INT LOCK
                                                short sysrel0 ; classic (Retro UNIX 386 type) sysrele
426
                                 <1>
                                          ;je
427 0000C803 7734
                                                short sysrel3
                                 <1>
                                          jа
                                          ; 27/02/2017
428
                                 <1>
429 0000C805 803D[D8030300]00
                                 <1>
                                          cmp byte [u.r_lock], 0 ; IRQ callback lock
430 0000C80C 0F8667FFFFF
                                 <1>
                                                sysrel0 ; classic sysrele ; 24/03/2017
                                          jna
431 0000C812 E859000000
                                 <1>
                                          call sysrel7
432 0000C817 803D[D8030300]00
                                 <1>
                                                byte [u.r_lock], 0 ; IRQ callback service lock
                                          cmp
433 0000C81E 7628
                                 <1>
                                          jna
                                                short sysrel4
434 0000C820 C605[D8030300]00
                                 <1>
                                                 byte [u.r_lock], 0 ; reset
                                          mov
                                 <1>
                                          ;mov byte [u.irqwait], 0 ; reset ; 28/02/2017
436 0000C827 A0[D9030300]
                                <1>
                                          mov
                                                al, [u.r_mode]
437 0000C82C 08C0
                                 <1>
                                          or
                                                 al, al
438 0000C82E 7518
                                <1>
                                          jnz
                                                short sysrel4
439 0000C830 FEC8
                                <1>
                                          dec
440 0000C832 A2[D9030300]
                                <1>
                                          mov
                                                 [u.r_mode], al ; OFFh ; not necessary !?
441 0000C837 EB32
                                <1>
                                          jmp
                                                 short sysrel6
                                 <1> sysrel3:
442
                                          ; 27/02/2017
443
                                <1>
444 0000C839 E832000000
                                <1>
                                          call sysrel7
                                <1>
                                          ; 14/01/2017
446 0000C83E 28C0
                                <1>
                                          sub
                                                al, al
447 0000C840 3805[D4030300]
                                <1>
                                                 [u.t_lock], al ; 0 ; TIMER INT LOCK
                                          cmp
448 0000C846 770E
                                <1>
                                                 short sysrel5 ; yes
                                          iа
449
                                <1> sysrel4:
450
                                 <1>
                                          ; 29/01/2017
                                                eax, [esp+28] ; eax
451 0000C848 8B44241C
                                <1>
                                          mov
                                                [u.r0], eax
452 0000C84C A3[64030300]
                                <1>
453 0000C851 E93EFFFFFF
                                <1>
                                          qmr
                                                sysrel2
                                 <1> sysrel5:
455 0000C856 A2[D4030300]
                                <1>
                                                [u.t lock], al ; 0 ; reset
                                          mov
456 0000C85B A0[D5030300]
                                <1>
                                          mov
                                                 al, [u.t_mode]
457 0000C860 20C0
                                 <1>
                                          and
                                                 al, al
                                                short sysrel2 ; OFFh ; user mode
458
                                <1>
                                          ;jnz
459 0000C862 75E4
                                <1>
                                          jnz
                                                 short sysrel4 ; 29/01/2017
460 0000C864 FEC8
                                 <1>
                                          dec
                                                 al
                                                [u.t_mode], al ; OFFh ; not necessary !?
461 0000C866 A2[D5030300]
                                <1>
                                          mov
                                 <1> sysrel6:
463
                                <1>
                                          ; cpu will continue from the interrupted sytem call addr
464 0000C86B 61
                                <1>
                                          popad ; edi, esi, ebp, esp, ebx, edx, ecx, eax
                                          add esp, 16; pass segment segisters: ds, es, fs, gs
465 0000C86C 83C410
                                <1>
466 0000C86F CF
                                <1>
                                          iretd
                                                  ; eip, cs, eflags
                                <1>
467
                                <1> sysrel7:
469 0000C870 0FB61D[B3030300]
                                <1>
                                          movzx ebx, byte [u.uno] ; current process number
470 0000C877 66C1E302
                                <1>
                                          shl bx, 2
                                          ;cmp [ebx+p.tcb-4], eax ; 0 ; is there callback address ?
471
                                <1>
472
                                <1>
                                          ;jna short sysrel0
                                          ; yes, reset callback address then restore process registers
473
                                <1>
474
                                <1>
                                          ;mov [ebx+p.tcb-4], eax; 0; reset
475 0000C87B 8B83[BC000300]
                                <1>
                                                eax, [ebx+p.upage-4]; UPAGE address
                                          mov
                                          cli ; disable interrupts till 'iretd'
476 0000C881 FA
                                <1>
477 0000C882 E9621E0000
                                <1>
                                          jmp rswap ; restore process 'u' structure
                                <1>
                                <1> badsys:
479
                                       ; 25/12/2016
480
                                <1>
                                          ; 18/04/2016 (TRDOS 386 = TRDOS v2.0)
481
                                <1>
                                 <1>
                                          ; 17/04/2011 (TRDOS v1.0, 'IFC.ASM')
482
                                 <1>
483
                                          ; 03/02/2011 ('trdos_ifc_routine')
```

```
484
                                  <1>
485
                                  <1>
                                           ; 16/04/2015 (Retro UNIX 386 v1, 'badsys')
                                           ; (EIP, EAX values will be shown on screen with error message)
486
                                  <1>
487
                                  <1>
                                           ; (EIP = 'CD 40h' instruction address -INT 40h-)
                                  <1>
                                           ; (EAX = Function number)
488
489
                                  <1>
490 0000C887 FE05[B2030300]
                                                  byte [u.bsys]
                                  <1>
                                           inc
                                 <1>
491
492 0000C88D 8B1D[5C030300]
                                 <1>
                                                   ebx, [u.sp]; esp at the beginning of 'sysent'
493 0000C893 8B03
                                                  eax, [ebx] ; EIP (return address, not 'INT 30h' address)
                                 <1>
                                           mov
494 0000C895 83E802
                                 <1>
                                            sub
                                                   eax, 2 ; CDh, ##h
495 0000C898 E8756AFFFF
                                  <1>
                                           call
                                                 dwordtohex
496 0000C89D 8915[6F140100]
                                 <1>
                                                   [eip_str], edx
                                           mov
497 0000C8A3 A3[73140100]
                                 <1>
                                                  [eip_str+4], eax
                                                  eax, [u.r0]
498 0000C8A8 A1[64030300]
                                 <1>
                                           mov
499 0000C8AD E8606AFFFF
                                 <1>
                                           call
                                                  dwordtohex
500 0000C8B2 8915[5E140100]
                                  <1>
                                           mov
                                                  [eax str], edx
501 0000C8B8 A3[62140100]
                                 <1>
                                           mov
                                                  [eax_str+4], eax
502
                                  <1>
503 0000C8BD 66C705[53140100]34- <1>
                                                  word [int num str], SYSCALL INT NUM; 25/12/2016
                                           mov
503 0000C8C5 30
                                 <1>
                                  <1>
505 0000C8C6 BE[25140100]
                                                   esi, ifc_msg; "invalid funtion call!" msg (trdosk9.s)
                                 <1>
                                            mov
506 0000C8CB E8FD9AFFFF
                                 <1>
                                            call print_msg
507
                                 <1>
508 0000C8D0 EB17
                                  <1>
                                                  sysexit
                                            jmp
509
                                 <1>
                                 <1> intract: ; / interrupt action
510
511
                                           ; 14/10/2015
                                  <1>
                                           ; 16/04/2015 (Retro UNIX 386 v1 - Beginning)
512
                                  <1>
513
                                  <1>
                                            ; 09/05/2013 - 07/12/2013 (Retro UNIX 8086 v1)
514
                                  <1>
                                           ; Retro UNIX 8086 v1 modification !
515
                                  <1>
516
                                  <1>
                                           ; (Process/task switching and quit routine by using
517
                                  <1>
                                           ; Retro UNIX 8086 v1 keyboard interrupt output.))
518
                                  <1>
                                           ; input -> 'u.quit' (also value of 'u.intr' > 0)
519
                                  <1>
                                  <1>
                                           ; output -> If value of 'u.quit' = FFFFh ('ctrl+brk' sign)
520
                                                         'intract' will jump to 'sysexit'.
521
                                  <1>
                                                       Intract will return to the caller
522
                                  <1>
523
                                  <1>
                                                         if value of 'u.quit' <> FFFFh.
                                           ; 14/10/2015
524
                                  <1>
525 0000C8D2 FB
                                  <1>
                                           sti
                                  <1>
                                            ; 07/12/2013
526
527 0000C8D3 66FF05[AC030300]
                                  <1>
                                           inc word [u.quit]
528 0000C8DA 7408
                                  <1>
                                            jΖ
                                                  short intrct0 ; FFFFh -> 0
                                            dec word [u.quit]
529 0000C8DC 66FF0D[AC030300]
                                  <1>
                                           ; 16/04/2015
                                 <1>
530
531 0000C8E3 C3
                                  <1>
                                            retn
532
                                 <1> intrct0:
533 0000C8E4 58
                                 <1>
                                                  eax ; call intract -> retn
                                            pop
534
                                 <1>
535 0000C8E5 31C0
                                 <1>
                                            xor
                                                  eax, eax
536 0000C8E7 FEC0
                                 <1>
                                            inc al ; mov ax, 1
537
                                  <1> ;;;
538
                                  <1>
                                            ; UNIX v1 original 'intract' routine...
539
                                  <1>
                                            ; / interrupt action
                                                  ;cmp *(sp),$rti / are you in a clock interrupt?
540
                                  <1>
541
                                  <1>
                                                  ; bne 1f / no, 1f
542
                                  <1>
                                                   ; cmp (sp)+,(sp)+ / pop clock pointer
                                            ; 1: / now in user area
543
                                  <1>
544
                                  <1>
                                                  ; mov r1,-(sp) / save r1
545
                                  <1>
                                                  ; mov u.ttyp,r1
                                  <1>
                                                       ; / pointer to tty buffer in control-to r1
546
547
                                  <1>
                                                  ; cmpb 6(r1),$177
548
                                  <1>
                                                        ; / is the interrupt char equal to "del"
                                                  ; beq 1f / yes, 1f
549
                                  <1>
                                                  ; clrb 6(r1)
550
                                  <1>
551
                                  <1>
                                                          ; / no, clear the byte
552
                                  <1>
                                                         ; / (must be a quit character)
                                                  ; mov (sp)+,r1 / restore r1
553
                                  <1>
                                                  ; clr u.quit / clear quit flag
554
                                  <1>
                                                  ; bis $20,2(sp)
555
                                  <1>
556
                                  <1>
                                                         ; / set trace for quit (sets t bit of
                                                  ; / ps-trace trap)
; rti ; / return from interrupt
557
                                  <1>
558
                                  <1>
559
                                  <1>
                                            ; 1: / interrupt char = del
560
                                  <1>
                                                  ; clrb 6(r1) / clear the interrupt byte
561
                                                            ; / in the buffer
                                  <1>
                                                  ; mov (sp)+,r1 / restore r1
562
                                  <1>
563
                                  <1>
                                                   ; cmp u.intr, $core / should control be
564
                                  <1>
                                                                ; / transferred to loc core?
                                                   ; blo 1f
565
                                  <1>
                                  <1>
                                                   ; jmp *u.intr / user to do rti yes,
567
                                  <1>
                                                                ; / transfer to loc core
568
                                  <1>
                                            ; 1:
                                                  ; sys 1 / exit
569
                                  <1>
570
                                  <1>
571
                                  <1> sysexit: ; <terminate process>
572
                                           ; 14/11/2017
                                  <1>
573
                                  <1>
                                            ; 27/05/2017
574
                                  <1>
                                            ; 10/04/2017
575
                                            ; 26/02/2017, 28/02/2017
                                  <1>
576
                                  <1>
                                            ; 02/01/2017, 23/01/2017
                                           ; 06/06/2016, 10/06/2016
577
                                  <1>
578
                                  <1>
                                            ; 19/05/2016, 23/05/2016
                                            ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
579
                                  <1>
                                            ; 16/04/2015 - 01/09/2015 (Retro UNIX 386 v1)
                                  <1>
580
                                            ; 19/04/2013 - 14/02/2014 (Retro UNIX 8086 v1)
581
                                  <1>
                                  <1>
582
                                            ; 'sysexit' terminates a process. First each file that
583
                                  <1>
584
                                  <1>
                                            ; the process has opened is closed by 'flose'. The process
585
                                  <1>
                                            ; status is then set to unused. The 'p.pid' table is then
586
                                  <1>
                                            ; searched to find children of the dying process. If any of
587
                                  <1>
                                            ; children are zombies (died by not waited for), they are
```

```
; set free. The 'p.pid' table is then searched to find the
588
                                <1>
589
                                          ; dying process's parent. When the parent is found, it is
590
                                          ; checked to see if it is free or it is a zombie. If it is
                                <1>
591
                                <1>
                                          ; one of these, the dying process just dies. If it is waiting
592
                                <1>
                                          ; for a child process to die, it notified that it doesn't
593
                                <1>
                                          ; have to wait anymore by setting it's status from 2 to 1
594
                                          ; (waiting to active). It is awakened and put on rung by
595
                                          ; 'putlu'. The dying process enters a zombie state in which
                                <1>
596
                                <1>
                                          ; it will never be run again but stays around until a 'wait'
597
                                <1>
                                          ; is completed by it's parent process. If the parent is not
598
                                <1>
                                          ; found, process just dies. This means 'swap' is called with
599
                                <1>
                                          ; 'u.uno=0'. What this does is the 'wswap' is not called
                                          ; to write out the process and 'rswap' reads the new process
600
                                <1>
601
                                <1>
                                         ; over the one that dies..i.e., the dying process is
                                <1>
602
                                         ; overwritten and destroyed.
603
                                <1>
604
                                <1>
                                         ; Calling sequence:
605
                                <1>
                                        ; sysexit or conditional branch.
606
                                <1>
                                          ; Arguments:
607
                                <1>
608
                                <1>
609
                                <1>
                                <1>
                                          ; Retro UNIX 8086 v1 modification:
610
                                <1>
                                                  System call number (=1) is in EAX register.
                                <1>
612
613
                                <1>
                                                  Other parameters are in EDX, EBX, ECX, ESI, EDI, EBP
614
                                <1>
                                                  registers depending of function details.
615
                                <1>
                                          ; ('swap' procedure is mostly different than original UNIX v1.)
616
                                <1>
617
                                <1>
618
                                <1>; / terminate process
                                       ; AX = 1
619
                                <1>
620 0000C8E9 6648
                                          dec ax; 0
                                <1>
621 0000C8EB 66A3[AA030300]
                                <1>
                                          mov [u.intr], ax; 0
                                                ; clr u.intr / clear interrupt control word
                                <1>
622
623
                                <1>
                                                ; clr r1 / clear r1
624
                                <1> sysexit_0:
                                     ; 23/01/2017
625
                                <1>
                                         ; 02/01/2017
626
                                <1>
                                         ; 10/06/2016
627
                                <1>
628
                                <1>
                                        ; 06/06/2016
                                        ; 23/05/2016
629
                                <1>
                                         ; 19/05/2016 - TRDOS 386 (TRDOS v2.0)
                                <1>
630
                                         ; Check and stop/clear timer event(s) of this (dying) process
631
                                <1>
632
                                <1>
                                         ; if there is.
633
                                <1>
634
                                <1>
                                         ; 02/01/2017
635 0000C8F1 FA
                                <1>
                                         cli ; disable interrupts
                                          ; 23/01/2017 - reset timer frequency (to 18.2Hz)
636
                                <1>
637 0000C8F2 B036
                                          mov al, 00110110b; 36h
                                <1>
638 0000C8F4 E643
                                <1>
                                          out 43h, al
639 0000C8F6 28C0
                                <1>
                                          sub
                                                al, al ; 0
640 0000C8F8 E640
                                <1>
                                          out
                                                40h, al ; LB
641 0000C8FA E640
                                <1>
                                               40h, al ; HB
                                <1>
642
                                         ;
643 0000C8FC 0FB61D[B3030300]
                                <1>
                                         movzx ebx, byte [u.uno]
                                          ;mov bl, [u.uno]; process number of dying process
                                <1>
645 0000C903 3883[FF000300]
                                <1>
                                          cmp
                                                byte [ebx+p.timer-1], al ; 0
646 0000C909 763A
                                <1>
                                                short sysexit_12; no timer events for this process
                                          jna
                                          mov byte [ebx+p.timer-1], al; 0; reset
647 0000C90B 8883[FF000300]
                                <1>
648
                                <1>
                                         ;mov al, [timer_events]
649
                                <1>
                                                al, al
                                         ;or
                                <1>
650
                                          ;jz
                                                short sysexit_12 ; no timer events
                                          ;mov cl, al
651
                                <1>
652 0000C911 8A0D[53650100]
                                <1>
                                                cl, [timer_events] ; 14/11/2017
                                         mov
653
                                <1>
                                          ;cli
                                                ; disable interrupts
654 0000C917 B410
                                <1>
                                          mov
                                                ah, 16; number of available timer events
655 0000C919 BE[60040300]
                                <1>
                                          mov
                                                esi, timer_set ; beginning address of timer events
                                <1> sysexit_7:
657 0000C91E 8A06
                                <1>
                                                 al, [esi] ; process number (of timer event)
                                          mov
658 0000C920 38D8
                                <1>
                                                 al, bl ; process number comparison
                                                short sysexit_10
659 0000C922 7411
                                <1>
                                          jе
660 0000C924 20C0
                                <1>
                                          and
                                                al, al
661 0000C926 7404
                                <1>
                                          jΖ
                                                 short sysexit_9
                                <1> sysexit_8:
662
663 0000C928 FEC9
                                <1>
                                          dec
664 0000C92A 7416
                                <1>
                                                 short sysexit 11
                                          jΖ
                                <1> sysexit_9:
665
666 0000C92C FECC
                                <1>
                                          dec
667 0000C92E 7415
                                                short sysexit_12
                                <1>
                                          jz
668 0000C930 83C610
                                <1>
                                          add
                                                 esi, 16
669 0000C933 EBE9
                                <1>
                                          jmp
                                                 short sysexit_7
670
                                <1>
671
                                <1> sysexit 10:
                                         ;mov byte [esi], 0
672
                                <1>
673 0000C935 66C7060000
                                <1>
                                          mov
                                                 word [esi], 0
                                               dword [esi+12], 0
674
                                <1>
                                          ;mov
675
                                <1>
                                                byte [timer events] ; 02/01/2017
676 0000C93A FE0D[53650100]
                                <1>
                                          dec
677
                                <1>
678 0000C940 EBE6
                                                 short sysexit 8
                                <1>
                                          jmp
679
                                <1>
                                <1> sysexit_11:
680
                                                ax, ax; 0; 26/02/2017
681 0000C942 6629C0
                                <1>
                                          sub
                                <1> sysexit 12:
682
                                          ; 26/02/2017 (Unlink IRQ callbacks belong to the user)
683
                                <1>
684 0000C945 803D[D6030300]00
                                <1>
                                          cmp byte [u.irqc], 0 ; Count of IRQ callbacks
685 0000C94C 7E2E
                                <1>
                                                 short sysexit_16 ; zero or invalid
686
                                <1>
                                          ; 28/02/2017
                                          ; clear IRQ callback flags (for 'sysrele' and 'sysret')
687
                                <1>
688 0000C94E A2[D7030300]
                                <1>
                                          mov
                                               [u.irqwait], al ; 0 ; force to clear waiting flag
689 0000C953 A2[D8030300]
                                <1>
                                          mov
                                                 [u.r_lock], al ; 0 ; force to clear busy flag
690 0000C958 BE[F26A0100]
                                <1>
                                          mov
                                                esi, IRQ.owner
                                <1> sysexit 13:
692 0000C95D AC
                                <1>
                                          lodsb
```

```
693 0000C95E 3A05[B3030300]
                                <1>
                                          cmp
                                                al, [u.uno] ; owner = current user ?
694 0000C964 750C
                                <1>
                                          jne
                                                 short sysexit 14
695 0000C966 C646FF00
                                                byte [esi-1], 0; owner = 0: Free
                                <1>
                                          mov
696 0000C96A FE0D[D6030300]
                                <1>
                                          dec
                                                byte [u.irqc]
697 0000C970 7408
                                <1>
                                          İΖ
                                                short sysexit 15
698
                                <1> sysexit_14:
699 0000C972 81FE[FA6A0100]
                                <1>
                                                esi, IRQ.owner + 8; the last IRQ index number?
                                          cmp
700 0000C978 76E3
                                                short sysexit_13 ; no
                                <1>
                                          jna
701
                                <1> sysexit_15:
702 0000C97A 30C0
                                <1> xor al, al; 0
703
                                <1> sysexit_16: ; 2:
704 0000C97C FB
                                <1>
                                          sti ; enable interrupts
705
                                <1>
706
                                <1>
                                          ; AX = 0
                                <1> sysexit_1: ; 1:
707
                                          ; AX = File descriptor
708
                                <1>
709
                                <1>
                                                 ; / rl has file descriptor (index to u.fp list)
710
                                <1>
                                                 ; / Search the whole list
711 0000C97D E89A130000
                                <1>
                                          call fclose
712
                                <1>
                                                 ; jsr r0,fclose / close all files the process opened
713
                                <1>
                                          ;; ignore error return
                                                ; br .+2 / ignore error return
714
                                <1>
715
                                <1>
                                          ;inc ax
716 0000C982 FEC0
                                <1>
                                          inc
717
                                <1>
                                                ; inc r1 / increment file descriptor
718
                                <1>
                                          ;cmp ax, 10
719 0000C984 3C0A
                                <1>
                                          cmp al, 10
                                                ; cmp r1,$10. / end of u.fp list?
720
                                <1>
721 0000C986 72F5
                                <1>
                                                short sysexit_1
722
                                <1>
                                                ; blt 1b / no, go back
723
                                <1>
                                          ;movzx ebx, byte [u.uno]
724 0000C988 8A1D[B3030300]
                                <1>
                                          mov bl, [u.uno]; 02/01/2017
                                                 ; movb u.uno,r1 / yes, move dying process's number to r1
725
                                <1>
726 0000C98E 88A3[AF000300]
                                <1>
                                               [ebx+p.stat-1], ah; 0, SFREE
                                                ; clrb p.stat-1(r1) / free the process
727
                                <1>
728
                                <1>
                                          ; 10/04/2017
729 0000C994 381D[696B0100]
                                <1>
                                          cmp [audio user], bl
730 0000C99A 7518
                                                short sysexit_17
                                <1>
                                          jne
731
                                 <1>
                                          ; reset audio device (current) owner and 'initializated' flag
                                          mov [audio_user], bh; 0
732 0000C99C 883D[696B0100]
                                <1>
                                          ; 27/05/2017
733
                                <1>
734 0000C9A2 8B0D[546B0100]
                                <1>
                                          mov ecx, [audio_buffer]
735 0000C9A8 09C9
                                                ecx, ecx
                                <1>
                                          or
                                                 short sysexit_17
736 0000C9AA 7408
                                <1>
                                          jΖ
737
                                <1>
                                          ; 'deallocate_user_pages' is not necessary in sysexit !!!
738
                                <1>
                                          ;push ebx
739
                                <1>
                                          ;mov ebx, ecx
740
                                <1>
                                          ;mov ecx, [audio_buff_size]
741
                                <1>
                                          ;call deallocate_user_pages
                                          ;; (Modified Registers -> EAX, EDX, ESI, EDI, EBX, ECX, EBP)
742
                                <1>
743 0000C9AC 29C9
                                <1>
                                          sub ecx, ecx
744 0000C9AE 890D[546B0100]
                                <1>
                                          mov
                                                 [audio buffer], ecx; 0
745
                                <1>
                                          ;pop ebx
746
                                <1> sysexit 17:
747
                                <1>
                                          ;shl bx, 1
748 0000C9B4 D0E3
                                <1>
                                                bl, 1
749
                                <1>
                                                ; asl r1 / use r1 for index into the below tables
750 0000C9B6 668B8B[1E000300]
                                <1>
                                          mov cx, [ebx+p.pid-2]
                                 <1>
                                                 ; mov p.pid-2(r1), r3 / move dying process's name to r3
751
752 0000C9BD 668B93[3E000300]
                                                dx, [ebx+p.ppid-2]
                                <1>
                                          mov
753
                                <1>
                                                 ; mov p.ppid-2(r1),r4 / move its parents name to r4
754
                                <1>
                                          ; xor bx, bx ; 0
755 0000C9C4 30DB
                                <1>
                                          xor bl, bl ; 0
756
                                <1>
                                                ; clr r2
757 0000C9C6 31F6
                                <1>
                                               esi, esi ; 0
                                          xor
758
                                <1>
                                                 ; clr r5 / initialize reg
759
                                <1> sysexit_2: ; 1:
760
                                                 ; / find children of this dying process,
                                <1>
761
                                <1>
                                                 ; / if they are zombies, free them
762
                                <1>
                                          ;add bx, 2
763 0000C9C8 80C302
                                <1>
                                          add bl, 2
764
                                <1>
                                                ; add $2,r2 / search parent process table
                                                          ; / for dying process's name
765
                                <1>
766 0000C9CB 66398B[3E000300]
                                <1>
                                                [ebx+p.ppid-2], cx
767
                                <1>
                                                 ; cmp p.ppid-2(r2),r3 / found it?
768 0000C9D2 7513
                                                 short sysexit 4
                                <1>
                                          jne
769
                                 <1>
                                                 ; bne 3f / no
770
                                <1>
                                          ; shr bx, 1
771 0000C9D4 D0EB
                                 <1>
                                                bl, 1
772
                                                 ; asr r2 / yes, it is a parent
                                <1>
773 0000C9D6 80BB[AF000300]03
                                <1>
                                                byte [ebx+p.stat-1], 3; SZOMB
                                 <1>
                                                 ; cmpb p.stat-1(r2),$3 / is the child of this
                                <1>
                                                                   ; / dying process a zombie
776 0000C9DD 7506
                                <1>
                                                 short sysexit_3
                                <1>
                                                 ; bne 2f / no
778 0000C9DF 88A3[AF000300]
                                                 [ebx+p.stat-1], ah; 0, SFREE
                                <1>
                                                ; clrb p.stat-1(r2) / yes, free the child process
779
                                <1>
780
                                <1> sysexit_3: ; 2:
781
                                <1>
                                          ;shr bx, 1
782 0000C9E5 D0E3
                                          shl bl, 1
                                <1>
783
                                <1>
                                                 ; asl r2
                                <1> sysexit 4: ; 3:
784
785
                                                ; / search the process name table
                                <1>
786
                                <1>
                                                 ; / for the dying process's parent
787 0000C9E7 663993[1E000300]
                                <1>
                                                [ebx+p.pid-2], dx
                                          cmp
                                                 ; cmp p.pid-2(r2),r4 / found it?
788
                                <1>
789 0000C9EE 7502
                                <1>
                                                short sysexit 5
                                          jne
790
                                <1>
                                                 ; bne 3f / no
791 0000C9F0 89DE
                                <1>
                                                 esi, ebx
792
                                <1>
                                                 ; mov r2,r5 / yes, put index to p.pid table (parents
                                                           ; / process # x2) in r5
793
                                <1>
794
                                <1> sysexit_5: ; 3:
795
                                <1>
                                          ;cmp bx, nproc + nproc
                                          cmp bl, nproc + nproc
796 0000C9F2 80FB20
                                <1>
797
                                <1>
                                                 ; cmp r2, $nproc+nproc / has whole table been searched?
```

```
798 0000C9F5 72D1
                                 <1>
                                                  short sysexit_2
799
                                 <1>
                                                  ; blt 1b / no, go back
800
                                                  ; mov r5,r1 / yes, r1 now has parents process \# x2
                                 <1>
801 0000C9F7 21F6
                                 <1>
                                                esi, esi ; r5=r1
802 0000C9F9 7436
                                 <1>
                                                 short sysexit 6
                                           jz
803
                                 <1>
                                                  ; beq 2f / no parent has been found.
804
                                 <1>
                                                       ; / The process just dies
805 0000C9FB 66D1EE
                                                 si, 1
                                 <1>
806
                                 <1>
                                                  ; asr r1 / set up index to p.stat
807 0000C9FE 8A86[AF000300]
                                                 al, [esi+p.stat-1]
                                 <1>
                                           mov
808
                                 <1>
                                                 ; movb p.stat-1(r1),r2 / move status of parent to r2
809 0000CA04 20C0
                                 <1>
                                           and
                                                 al, al
810 0000CA06 7429
                                 <1>
                                                 short sysexit 6
                                           jz
811
                                 <1>
                                                  ; beq 2f / if its been freed, 2f
812 0000CA08 3C03
                                 <1>
                                           cmp
                                                 al, 3
                                                  ; cmp r2,$3 / is parent a zombie?
813
                                 <1>
                                                  short sysexit 6
814 0000CA0A 7425
                                 <1>
                                                 ; beq 2f / yes, 2f
815
                                 <1>
816
                                 <1>
                                           ; BH = 0
817 0000CA0C 8A1D[B3030300]
                                 <1>
                                           mov bl, [u.uno]
818
                                 <1>
                                                  ; movb u.uno,r3 / move dying process's number to r3
819 0000CA12 C683[AF000300]03
                                 <1>
                                                 byte [ebx+p.stat-1], 3 ; SZOMB
                                           mov
                                                 ; movb $3,p.stat-1(r3) / make the process a zombie
820
                                 <1>
821 0000CA19 3C01
                                 <1>
                                                 al, 1 ; SRUN
822 0000CA1B 7414
                                 <1>
                                           jе
                                                 short sysexit_6
823
                                 <1>
                                           ;cmp
                                                al, 2
                                                 ; cmp r2,$2 / is the parent waiting for
824
                                 <1>
825
                                 <1>
                                                        ; / this child to die
826
                                 <1>
                                           ;jne short sysexit_6
                                                 ; bne 2f / yes, notify parent not to wait any more
827
                                 <1>
828
                                 <1>
                                           ; p.stat = 2 \longrightarrow waiting
                                           ; p.stat = 4 --> sleeping
829
                                 <1>
830 0000CA1D C686[AF000300]01
                                           mov byte [esi+p.stat-1], 1; SRUN
                                 <1>
831
                                 <1>
                                           ;dec byte [esi+p.stat-1]
832
                                 <1>
                                                 ; decb p.stat-1(r1) / awaken it by putting it (parent)
833 0000CA24 6689F0
                                 <1>
                                                 ax, si; r1 (process number in AL)
                                           mov
834
                                 <1>
835
                                 <1>
                                           ; mov
                                                 ebx, runq + 4
836
                                 <1>
                                                  ; mov $runq+4,r2 / on the runq
                                                  ebx, runq+2; normal run queue; 02/01/2017
837 0000CA27 BB[54030300]
                                 <1>
                                           mov
838 0000CA2C E8F01C0000
                                 <1>
                                           call putlu
                                                  ; jsr r0, putlu
839
                                 <1>
                                 <1> sysexit_6:
840
841
                                 <1>
                                                  ; / the process dies
842 0000CA31 C605[B3030300]00
                                 <1>
                                                 byte [u.uno], 0
                                           mov
843
                                 <1>
                                                  ; clrb u.uno / put zero as the process number,
                                                    ; / so "swap" will
844
                                 <1>
845 0000CA38 E8E61B0000
                                           call swap
                                 <1>
                                                  ; jsr r0, swap / overwrite process with another process
846
                                 <1>
847
                                 <1>
848
                                 <1> hlt_sys:
849
                                 <1>
                                          ;sti
850
                                 <1> hlts0:
851 0000CA3D F4
                                 <1>
                                          hlt
852 0000CA3E EBFD
                                 <1>
                                                 short hlts0
                                           jmp
853
                                 <1>
                                                 ; 0 / and thereby kill it; halt?
854
                                 <1>
855
                                 <1> syswait: ; < wait for a processs to die >
856
                                 <1>
                                        ; 17/09/2015
857
                                           ; 02/09/2015
                                 <1>
858
                                 <1>
                                          ; 01/09/2015
                                          ; 16/04/2015 (Retro UNIX 386 v1 - Beginning)
859
                                 <1>
                                 <1>
                                           ; 24/05/2013 - 05/02/2014 (Retro UNIX 8086 v1)
860
861
                                 <1>
                                 <1>
                                           ; 'syswait' waits for a process die.
862
863
                                 <1>
                                           ; It works in following way:
864
                                 <1>
                                           ; 1) From the parent process number, the parent's
                                 <1>
865
                                                 process name is found. The p.ppid table of parent
866
                                 <1>
                                                  names is then searched for this process name.
867
                                 <1>
                                                 If a match occurs, r2 contains child's process
868
                                 <1>
                                                 number. The child status is checked to see if it is
869
                                 <1>
                                                  a zombie, i.e; dead but not waited for (p.stat=3)
870
                                                 If it is, the child process is freed and it's name
                                 <1>
                                                 is put in (u.r0). A return is then made via 'sysret'.
871
                                 <1>
872
                                 <1>
                                                 If the child is not a zombie, nothing happens and
873
                                 <1>
                                                  the search goes on through the p.ppid table until
874
                                 <1>
                                                 all processes are checked or a zombie is found.
875
                                 <1>
                                                2) If no zombies are found, a check is made to see if
876
                                 <1>
                                                 there are any children at all. If there are none,
877
                                                  an error return is made. If there are, the parent's
                                 <1>
878
                                 <1>
                                                  status is set to 2 (waiting for child to die),
879
                                 <1>
                                                  the parent is swapped out, and a branch to 'syswait'
880
                                 <1>
                                                  is made to wait on the next process.
881
                                 <1>
882
                                 <1>
                                           ; Calling sequence:
883
                                 <1>
884
                                 <1>
                                           ; Arguments:
885
                                 <1>
886
                                 <1>
                                           ; Inputs: -
887
                                 <1>
                                           ; Outputs: if zombie found, it's name put in u.r0.
888
                                 <1>
889
                                 <1>
890
                                 <1>
891
                                 <1> ; / wait for a process to die
892
                                 <1>
893
                                 <1> syswait_0:
894 0000CA40 0FB61D[B3030300]
                                 <1>
                                          movzx ebx, byte [u.uno]; 01/09/2015
                                                 ; movb u.uno,r1 / put parents process number in r1
895
                                 <1>
896 0000CA47 D0E3
                                 <1>
                                                 bl. 1
                                           ;shl bx, 1
897
                                 <1>
898
                                 <1>
                                                 ; asl r1 / x2 to get index into p.pid table
899 0000CA49 668B83[1E000300]
                                 <1>
                                                 ax, [ebx+p.pid-2]
                                           mov
900
                                 <1>
                                                  ; mov p.pid-2(r1),r1 / get the name of this process
901 0000CA50 31F6
                                 <1>
                                                 esi, esi
902
                                 <1>
                                                 ; clr r2
```

```
903 0000CA52 31C9
                                 <1>
                                            xor ecx, ecx; 30/10/2013
904
                                  <1>
                                            ;xor cl, cl
905
                                                  ; clr r3 / initialize reg 3
                                  <1>
906
                                  <1> syswait_1: ; 1:
907 0000CA54 6683C602
                                           add si, 2
                                  <1>
908
                                  <1>
                                                  ; add $2,r2 / use r2 for index into p.ppid table
909
                                  <1>
                                                          ; / search table of parent processes
910
                                  <1>
                                                           ; / for this process name
911 0000CA58 663B86[3E000300]
                                  <1>
                                                  ax, [esi+p.ppid-2]
                                                  ; cmp p.ppid-2(r2),r1 / r2 will contain the childs
912
                                  <1>
913
                                  <1>
                                                                     ; / process number
914 0000CA5F 7535
                                  <1>
                                                  short syswait 3
                                            jne
915
                                  <1>
                                                   ;bne 3f / branch if no match of parent process name
916
                                  <1>
                                            ;inc
917 0000CA61 FEC1
                                  <1>
                                            inc
                                                  cl
918
                                  <1>
                                                  ;inc r3 / yes, a match, r3 indicates number of children
919 0000CA63 66D1EE
                                  <1>
                                            shr si, 1
                                                  ; asr r2 / r2/2 to get index to p.stat table
920
                                  <1>
921
                                  <1>
                                            ; The possible states ('p.stat' values) of a process are:
                                            ; 0 = free or unused
922
                                  <1>
923
                                  <1>
                                                  1 = active
                                                  2 = waiting for a child process to die
924
                                  <1>
                                            ;
925
                                                  3 = terminated, but not yet waited for (zombie).
                                  <1>
926 0000CA66 80BE[AF000300]03
                                  <1>
                                                 byte [esi+p.stat-1], 3; SZOMB, 05/02/2014
927
                                  <1>
                                                  ; cmpb p.stat-1(r2),$3 / is the child process a zombie?
928 0000CA6D 7524
                                  <1>
                                                  short syswait_2
                                  <1>
                                                   ; bne 2f / no, skip it
930 0000CA6F 88BE[AF000300]
                                  <1>
                                            mov
                                                  [esi+p.stat-1], bh; 0
931
                                  <1>
                                                  ; clrb p.stat-1(r2) / yes, free it
932 0000CA75 66D1E6
                                  <1>
                                            shl
                                                  si, 1
933
                                  <1>
                                                  ; asl r2 / r2x2 to get index into p.pid table
934 0000CA78 0FB786[1E000300]
                                  <1>
                                            movzx eax, word [esi+p.pid-2]
935 0000CA7F A3[64030300]
                                  <1>
                                            mov [u.r0], eax
                                  <1>
                                                   ; mov p.pid-2(r2),*u.r0
936
                                                               ; / put childs process name in (u.r0)
937
                                  <1>
938
                                  <1>
939
                                  <1>
                                            ; Retro UNIX 386 v1 modification ! (17/09/2015)
940
                                  <1>
                                            ; Parent process ID -p.ppid- field (of the child process)
941
                                  <1>
942
                                            ; must be cleared in order to prevent infinitive 'syswait'
                                  <1>
943
                                  <1>
                                            ; system call loop from the application/program if it calls
944
                                  <1>
                                            ; 'syswait' again (mistakenly) while there is not a zombie
                                            ; or running child process to wait. ('forktest.s', 17/09/2015)
945
                                  <1>
                                  <1>
946
947
                                  <1>
                                            ; Note: syswait will return with error if there is not a
948
                                  <1>
                                            ;
                                                    zombie or running process to wait.
949
                                  <1>
                                            ;
950 0000CA84 6629C0
                                  <1>
                                            sub
                                                  ax, ax
951 0000CA87 668986[3E000300]
                                  <1>
                                                   [esi+p.ppid-2], ax ; 0 ; 17/09/2015
952 0000CA8E E9D1FCFFFF
                                  <1>
                                            jmp
                                                  sysret0 ; ax = 0
953
                                  <1>
954
                                  <1>
                                            ;jmp
                                                 sysret
955
                                  <1>
                                                  ; br sysret1 / return cause child is dead
956
                                  <1> syswait_2: ; 2:
957 0000CA93 66D1E6
                                  <1>
                                            shl
                                                 si, 1
958
                                  <1>
                                                  ; asl r2 / r2x2 to get index into p.ppid table
959
                                  <1> syswait 3: ; 3:
960 0000CA96 6683FE20
                                  <1>
                                            cmp si, nproc+nproc
                                  <1>
                                                  ; cmp r2, $nproc+nproc / have all processes been checked?
961
962 0000CA9A 72B8
                                  <1>
                                            jb
                                                  short syswait 1
963
                                  <1>
                                                  ; blt 1b / no, continue search
964
                                  <1>
                                           ; and cx, cx
965 0000CA9C 20C9
                                  <1>
                                            and cl, cl
                                  <1>
                                                  ; tst r3 / one gets here if there are no children
966
967
                                  <1>
                                                         ; / or children that are still active
                                            ; 30/10/2013
968
                                  <1>
969 0000CA9E 750B
                                  <1>
                                            jnz short syswait_4
970
                                  <1>
                                                  error
                                                   ; beq error1 / there are no children, error
971
                                  <1>
972 0000CAA0 890D[64030300]
                                  <1>
                                                  [u.r0], ecx; 0
                                            mov
973 0000CAA6 E997FCFFFF
                                  <1>
                                            jmp
                                  <1> syswait 4:
975 0000CAAB 8A1D[B3030300]
                                            mov
                                  <1>
                                                  bl, [u.uno]
976
                                  <1>
                                                  ; movb u.uno,r1 / there are children so put
977
                                  <1>
                                                               ; / parent process number in r1
978 0000CAB1 FE83[AF000300]
                                  <1>
                                                  byte [ebx+p.stat-1]; 2, SWAIT, 05/02/2014
                                                  ; incb p.stat-1(r1) / it is waiting for
979
                                  <1>
980
                                  <1>
                                                                 ; / other children to die
981
                                  <1>
                                            ; 04/11/2013
982 0000CAB7 E8671B0000
                                            call swap
                                  <1>
983
                                  <1>
                                                   ; jsr r0, swap / swap it out, because it's waiting
                                                   syswait 0
 984 0000CABC EB82
                                  <1>
                                            qmj
                                                   ; br syswait / wait on next process
 985
                                  <1>
 986
                                  <1>
 987
                                  <1> sysfork: ; < create a new process >
                                            ; 02/01/2017 (TRDOS 386 modification)
988
                                  <1>
 989
                                  <1>
                                            ; 04/09/2015, 18/05/2015
990
                                  <1>
                                            ; 28/08/2015, 01/09/2015, 02/09/2015
                                            ; 09/05/2015, 10/05/2015, 14/05/2015
991
                                  <1>
992
                                            ; 06/05/2015 (Retro UNIX 386 v1 - Beginning)
                                  <1>
993
                                  <1>
                                            ; 24/05/2013 - 14/02/2014 (Retro UNIX 8086 v1)
 994
                                  <1>
995
                                  <1>
                                            ; 'sysfork' creates a new process. This process is referred
996
                                  <1>
                                            ; to as the child process. This new process core image is
997
                                  <1>
                                            ; a copy of that of the caller of 'sysfork'. The only
998
                                  <1>
                                            ; distinction is the return location and the fact that (u.r0)
                                            ; in the old process (parent) contains the process id (p.pid)
999
                                  <1>
1000
                                  <1>
                                            ; of the new process (child). This id is used by 'syswait'.
1001
                                  <1>
                                            ; 'sysfork' works in the following manner:
1002
                                                1) The process status table (p.stat) is searched to find
                                  <1>
1003
                                  <1>
                                                   a process number that is unused. If none are found
1004
                                  <1>
                                                  an error occurs.
1005
                                  <1>
                                                 2) when one is found, it becomes the child process number
1006
                                  <1>
                                                  and it's status (p.stat) is set to active.
1007
                                  <1>
                                                 3) If the parent had a control tty, the interrupt
```

```
1008
                                  <1>
                                                   character in that tty buffer is cleared.
1009
                                  <1>
                                                 4) The child process is put on the lowest priority run
1010
                                  <1>
                                                  queue via 'putlu'.
                                            ;
1011
                                  <1>
                                                 5) A new process name is gotten from 'mpid' (actually
1012
                                  <1>
                                                  it is a unique number) and is put in the child's unique
1013
                                  <1>
                                                  identifier; process id (p.pid).
                                                 6) The process name of the parent is then obtained and
1014
                                  <1>
1015
                                  <1>
                                                 placed in the unique identifier of the parent process
1016
                                  <1>
                                                  name is then put in 'u.r0'.
1017
                                                 7) The child process is then written out on disk by
                                  <1>
1018
                                  <1>
                                                  'wswap',i.e., the parent process is copied onto disk
1019
                                                  and the child is born. (The child process is written out on disk/drum with 'u.uno' being the child process
                                  <1>
1020
                                  <1>
                                                  number.)
1021
                                  <1>
                                                 8) The parent process number is then restored to 'u.uno'.
1022
                                  <1>
1023
                                  <1>
                                                 9) The child process name is put in 'u.r0'.
                                               10) The pc on the stack sp + 18 is incremented by 2 to
1024
1025
                                  <1>
                                                  create the return address for the parent process.
1026
                                  <1>
                                                11) The 'u.fp' list as then searched to see what files
                                                  the parent has opened. For each file the parent has
1027
                                  <1>
1028
                                  <1>
                                                   opened, the corresponding 'fsp' entry must be updated
                                                   to indicate that the child process also has opened
1029
                                  <1>
                                                   the file. A branch to 'sysret' is then made.
1030
                                  <1>
1031
                                  <1>
1032
                                  <1>
                                            ; Calling sequence:
1033
                                  <1>
                                               from shell ?
1034
                                  <1>
                                            ; Arguments:
1035
                                  <1>
1036
                                  <1>
                                            ; Inputs: -
1037
                                  <1>
                                            ; Outputs: *u.r0 - child process name
1038
                                  <1>
                                            ; ......
1039
                                  <1>
1040
                                  <1>
                                            ; Retro UNIX 8086 v1 modification:
1041
                                  <1>
                                                  AX = r0 = PID (>0) (at the return of 'sysfork')
                                                  = process id of child a parent process returns
1042
                                  <1>
1043
                                  <1>
                                                   = process id of parent when a child process returns
1044
                                  <1>
1045
                                  <1>
                                                    In original UNIX v1, sysfork is called and returns as
1046
                                  <1>
                                                  in following manner: (with an example: c library, fork)
1047
                                  <1>
1048
                                  <1>
1049
                                                               fork
                                  <1>
                                                         sys
1050
                                  <1>
                                                                br 1f / child process returns here
1051
                                  <1>
                                                         bes
                                                               2f / parent process returns here
1052
                                  <1>
                                                         / pid of new process in r0
1053
                                  <1>
                                                         rts pc
1054
                                  <1>
                                                   2: / parent process condionally branches here
                                                               $-1,r0 / pid = -1 means error return
1055
                                  <1>
                                                         mov
1056
                                  <1>
                                                              рс
                                                         rts
1057
                                  <1>
1058
                                  <1>
                                                  1: / child process brances here
1059
                                  <1>
                                                        clr 	 r0 	 / pid = 0 in child process
1060
                                  <1>
1061
                                  <1>
1062
                                  <1>
                                                  In UNIX v7x86 (386) by Robert Nordier (1999)
1063
                                  <1>
                                                         // pid = fork();
                                                         //
1064
                                  <1>
1065
                                  <1>
                                                         // pid == 0 in child process;
1066
                                  <1>
                                                         // pid == -1 means error return
1067
                                  <1>
                                                         // in child,
                                                               parents id is in par_uid if needed
1068
                                  <1>
                                                         //
1069
                                  <1>
1070
                                  <1>
                                                         _fork:
1071
                                  <1>
                                                                      $.fork,eax
                                                                mov
1072
                                  <1>
                                                                int
                                                                      $0x30
1073
                                  <1>
                                                                qmj
                                                                      1f
1074
                                  <1>
                                                                      2f
                                                                jnc
1075
                                  <1>
                                                                jmp
                                                                      cerror
1076
                                  <1>
                                                         1:
1077
                                  <1>
                                                                mov
                                                                      eax,_par_uid
1078
                                  <1>
                                                                xor
                                                                      eax,eax
1079
                                  <1>
                                                         2:
1080
                                  <1>
1081
                                  <1>
                                                  In Retro UNIX 8086 v1,
1082
                                  <1>
1083
                                  <1>
                                                   'sysfork' returns in following manner:
1084
                                  <1>
                                                                ax, sys_fork
1085
                                  <1>
1086
                                                                bx, offset @f; routine for child
                                  <1>
                                                         mov
1087
                                  <1>
                                                         int
                                                                20h
                                                         jс
1088
                                  <1>
                                                                error
1089
                                  <1>
1090
                                  <1>
                                                   ; Routine for parent process here (just after 'jc')
1091
                                  <1>
                                                         mov word ptr [pid of child], ax
                                                               next_routine_for_parent
1092
                                  <1>
                                                         jmp
1093
                                  <1>
1094
                                  <1>
                                                  @@: ; routine for child process here
1095
                                  <1>
1096
                                  <1>
                                                  NOTE: 'sysfork' returns to specified offset
                                                         for child process by using BX input.
1097
                                  <1>
                                                         (at first, parent process will return then
1098
                                  <1>
1099
                                  <1>
                                                         child process will return -after swapped in-
                                                         'syswait' is needed in parent process
1100
                                  <1>
1101
                                  <1>
                                                        if return from child process will be waited for.)
1102
                                  <1>
1103
                                  <1>
1104
                                  <1> ; / create a new process
1105
                                  <1>
                                         ; EBX = return address for child process
                                             ; (Retro UNIX 8086 v1 modification !)
                                  <1>
1106
1107 0000CABE 31F6
                                  <1>
                                                esi, esi
1108
                                  <1>
                                                  ; clr r1
1109
                                  <1> sysfork_1: ; 1: / search p.stat table for unused process number
1110 0000CAC0 46
                                  <1>
                                           inc esi
1111
                                  <1>
                                                  ; inc r1
```

```
1112 0000CAC1 80BE[AF000300]00
                                                 byte [esi+p.stat-1], 0 ; SFREE, 05/02/2014
                                <1>
1113
                                 <1>
                                                 ; tstb p.stat-1(r1) / is process active, unused, dead
                                                 short sysfork 2
1114 0000CAC8 760B
                                 <1>
1115
                                 <1>
                                                  ; beq 1f / it's unused so branch
1116 0000CACA 6683FE10
                                 <1>
                                                 si, nproc
                                           cmp
1117
                                 <1>
                                                  ; cmp r1, $nproc / all processes checked
1118 0000CACE 72F0
                                 <1>
                                                  short sysfork 1
                                                 ; blt 1b / no, branch back
                                 <1>
1119
1120
                                 <1>
1121
                                           ; Retro UNIX 8086 v1. modification:
                                 <1>
1122
                                 <1>
                                                  Parent process returns from 'sysfork' to address
                                                  which is just after 'sysfork' system call in parent
1123
                                 <1>
1124
                                 <1>
                                                  process. Child process returns to address which is put
                                           ;
1125
                                 <1>
                                                  in BX register by parent process for 'sysfork'.
1126
                                 <1>
                                                  ;add $2,18.(sp) / add 2 to pc when trap occured, points
1127
                                 <1>
                                                              ; / to old process return
1128
                                 <1>
                                                 ; br error1 / no room for a new process
1129
                                 <1>
1130 0000CAD0 E96DFCFFFF
                                 <1>
                                           jmp
                                                 error
1131
                                 <1> sysfork 2: ; 1:
                                           call allocate page
1132 0000CAD5 E8FD80FFFF
                                 <1>
1133 0000CADA 0F8262FCFFFF
                                 <1>
                                           jс
                                                 error
1134 0000CAE0 50
                                           push eax ; UPAGE (user structure page) address
                                 <1>
1135
                                 <1>
                                           ; Retro UNIX 386 v1 modification!
1136 0000CAE1 E80083FFFF
                                 <1>
                                           call duplicate_page_dir
1137
                                 <1>
                                                  ; EAX = New page directory
1138 0000CAE6 730B
                                 <1>
                                                short sysfork 3
1139 0000CAE8 58
                                 <1>
                                           pop eax ; UPAGE (user structure page) address
1140 0000CAE9 E8C782FFFF
                                 <1>
                                           call deallocate_page
1141 0000CAEE E94FFCFFFF
                                 <1>
                                           qmp
                                                 error
1142
                                 <1> sysfork_3:
1143
                                 <1>
                                           ; Retro UNIX 386 v1 modification !
1144 0000CAF3 56
                                 <1>
                                           push esi
1145 0000CAF4 E8B81B0000
                                 <1>
                                                wswap ; save current user (u) structure, user registers
                                                       ; and interrupt return components (for IRET)
1146
                                 <1>
1147 0000CAF9 8705[B8030300]
                                                 eax, [u.pgdir]; page directory of the child process
                                 <1>
1148 0000CAFF A3[BC030300]
                                                 [u.ppgdir], eax; page directory of the parent process
                                 <1>
                                           mov
1149 0000CB04 5E
                                 <1>
                                           pop
                                                 esi
1150 0000CB05 58
                                                       ; UPAGE (user structure page) address
                                 <1>
                                           pop
1151
                                 <1>
                                                 ; [u.usp] = esp
1152 0000CB06 89F7
                                 <1>
                                           mov
                                                 edi, esi
1153 0000CB08 66C1E702
                                 <1>
                                           shl
                                                 di, 2
                                                 [edi+p.upage-4], eax ; memory page for 'user' struct
1154 0000CB0C 8987[BC000300]
                                 <1>
                                           mov
1155 0000CB12 A3[B4030300]
                                 <1>
                                          mov [u.upage], eax; memory page for 'user' struct (child)
1156
                                 <1>
                                           ; 28/08/2015
1157 0000CB17 0FB605[B3030300]
                                 <1>
                                           movzx eax, byte [u.uno] ; parent process number
1158
                                 <1>
                                           ; movb u.uno,-(sp) / save parent process number
1159 0000CB1E 89C7
                                                edi, eax
                                 <1>
                                           mov
1160 0000CB20 50
                                 <1>
                                           pusheax ; **
1161 0000CB21 8A87[7F000300]
                                 <1>
                                           mov al, [edi+p.ttyc-1]; console tty (parent)
1162
                                 <1>
                                           ; 18/09/2015
1163
                                 <1>
                                           ;mov [esi+p.ttyc-1], al ; set child's console tty
                                                    [esi+p.waitc-1], ah ; 0 ; reset child's wait channel
1164
                                 <1>
                                           ; mov
1165 0000CB27 668986[7F000300]
                                 <1>
                                                 [esi+p.ttyc-1], ax ; al - set child's console tty
                                 <1>
                                                                 ; ah - reset child's wait channel
1166
1167 0000CB2E 89F0
                                 <1>
                                                 eax, esi
                                           mov
1168 0000CB30 A2[B3030300]
                                 <1>
                                           mov [u.uno], al; child process number
                                 <1>
                                                 ;movb r1,u.uno / set child process number to r1
1169
1170 0000CB35 FE86[AF000300]
                                 <1>
                                             inc byte [esi+p.stat-1]; 1, SRUN, 05/02/2014
                                               ; incb p.stat-1(r1) / set p.stat entry for child
1171
                                 <1>
1172
                                 <1>
                                                            ; / process to active status
                                                  ; mov u.ttyp,r2 / put pointer to parent process'
1173
                                 <1>
                                                              ; / control tty buffer in r2
1174
                                 <1>
                                                     ; beq 2f / branch, if no such tty assigned
1175
                                 <1>
                                                  ; clrb 6(r2) / clear interrupt character in tty buffer
1176
                                 <1>
1177
                                 <1>
                                           ; 2:
1178 0000CB3B 53
                                 <1>
                                           push
                                                ebx ; * return address for the child process
                                                  ; * Retro UNIX 8086 v1 feature only !
                                 <1>
1179
1180
                                 <1>
                                           ; (Retro UNIX 8086 v1 modification!)
                                                 ; mov $runq+4,r2
1181
                                 <1>
                                                  ebx, runq+2; normal run queue; 02/01/2017
1182 0000CB3C BB[54030300]
                                 <1>
1183 0000CB41 E8DB1B0000
                                 <1>
                                                 putlu
                                                 ; jsr r0, putlu / put child process on lowest priority
                                 <1>
1184
1185
                                 <1>
                                                          ; / run queue
1186 0000CB46 66D1E6
                                 <1>
                                           shl si, 1
                                                 ; asl r1 / multiply r1 by 2 to get index
1187
                                 <1>
                                 <1>
                                                        ; / into p.pid table
1189 0000CB49 66FF05[4E030300]
                                 <1>
                                                 word [mpid]
                                 <1>
                                                  ; inc mpid / increment m.pid; get a new process name
1190
1191 0000CB50 66A1[4E030300]
                                 <1>
                                                 ax, [mpid]
                                           mov
1192 0000CB56 668986[1E000300]
                                 <1>
                                           mov
                                                 [esi+p.pid-2], ax
                                  <1>
                                                  ;mov mpid,p.pid-2(r1) / put new process name
                                                                  ; / in child process' name slot
                                 <1>
1195 0000CB5D 5A
                                                  edx ; \star return address for the child process
                                 <1>
                                                   ; * Retro UNIX 8086 v1 feature only !
1196
                                 <1>
1197 0000CB5E 5B
                                 <1>
1198
                                 <1>
                                                 ebx, [esp] ; ** parent process number
                                           ; mov
1199
                                                  ; movb (sp),r2 / put parent process number in r2
                                 <1>
1200 0000CB5F 66D1E3
                                 <1>
1201
                                 <1>
                                                 ;asl r2 / multiply by 2 to get index into below tables
                                           ;movzx eax, word [ebx+p.pid-2]
1202
                                 <1>
1203 0000CB62 668B83[1E000300]
                                 <1>
                                           mov
                                                 ax, [ebx+p.pid-2]
                                                  ; mov p.pid-2(r2),r2 / get process name of parent
1204
                                 <1>
1205
                                 <1>
                                                                 ; / process
1206 0000CB69 668986[3E000300]
                                 <1>
                                                 [esi+p.ppid-2], ax
                                           mov
1207
                                 <1>
                                                  ; mov r2,p.ppid-2(r1) / put parent process name
                                 <1>
                                                         ; / in parent process slot for child
1209 0000CB70 A3[64030300]
                                                 [u.r0], eax
                                 <1>
                                           mov
1210
                                 <1>
                                                  ; mov r2, *u.r0 / put parent process name on stack
1211
                                                            ; / at location where r0 was saved
                                 <1>
1212 0000CB75 8B2D[5C030300]
                                 <1>
                                           mov
                                                  ebp, [u.sp]; points to return address (EIP for IRET)
                                                 [ebp], edx ; *, CS:EIP -> EIP
1213 0000CB7B 895500
                                 <1>
                                           mov
                                                          ; * return address for the child process
1214
                                 <1>
1215
                                                  ; mov $sysret1,-(sp) /
                                 <1>
1216
                                 <1>
                                                  ; mov \operatorname{sp},\operatorname{u.usp} / contents of \operatorname{sp} at the time when
```

```
1217
                                  <1>
                                                               ; / user is swapped out
1218
                                  <1>
                                                  ; mov $sstack,sp / point sp to swapping stack space
1219
                                            ; 04/09/2015 - 01/09/2015
                                  <1>
1220
                                  <1>
                                            ; [u.usp] = esp
1221 0000CB7E 68[62C70000]
                                  <1>
                                            push sysret; ***
1222 0000CB83 8925[60030300]
                                  <1>
                                            mov
                                                  [u.usp], esp; points to 'sysret' address (***)
                                                             ; (for child process)
                                  <1>
1224 0000CB89 31C0
                                  <1>
                                           xor
                                                  eax, eax
1225 0000CB8B 66A3[94030300]
                                  <1>
                                                  [u.ttyp], ax ; 0
                                           mov
1226
                                  <1>
1227 0000CB91 E81B1B0000
                                  <1>
                                            call wswap; Retro UNIX 8086 v1 modification!
1228
                                  <1>
                                                  ;jsr r0, wswap / put child process out on drum
                                                  ;jsr r0,unpack / unpack user stack
1229
                                  <1>
1230
                                  <1>
                                                  ;mov u.usp,sp / restore user stack pointer
1231
                                  <1>
                                                  ; tst (sp) + / bump stack pointer
1232
                                  <1>
                                            ; Retro UNIX 386 v1 modification !
1233 0000CB96 58
                                            pop eax ; ***
                                  <1>
1234 0000CB97 66D1E3
                                  <1>
                                            shl bx, 1
                                                   eax, [ebx+p.upage-4]; UPAGE address; 14/05/2015
1235 0000CB9A 8B83[BC000300]
                                  <1>
                                            call rswap ; restore parent process 'u' structure,
1236 0000CBA0 E8441B0000
                                  <1>
                                                       ; registers and return address (for IRET)
1237
                                  <1>
1238
                                  <1>
                                                  ;movb (sp)+,u.uno / put parent process number in u.uno
1239 0000CBA5 0FB705[4E030300]
                                            movzx eax, word [mpid]
                                  <1>
1240 0000CBAC A3[64030300]
                                  <1>
                                            mov [u.r0], eax
                                                  ; mov mpid, *u.r0 / put child process name on stack
                                  <1>
1241
1242
                                  <1>
                                                                ; / where r0 was saved
1243
                                  <1>
                                                  ; add $2,18.(sp) / add 2 to pc on stack; gives parent
1244
                                  <1>
                                                                  ; / process return
1245
                                  <1>
                                            ;xor ebx, ebx
1246 0000CBB1 31F6
                                  <1>
                                            xor
                                                  esi, esi
1247
                                  <1>
                                                  clr r1;
1248
                                  <1> sysfork 4: ; 1: / search u.fp list to find the files
                                                 ; / opened by the parent process
1249
                                  <1>
1250
                                  <1>
                                            ; 01/09/2015
1251
                                  <1>
                                           ;xor bh, bh
1252
                                  <1>
                                            ;mov bl, [esi+u.fp]
1253 0000CBB3 8A86[6A030300]
                                  <1>
                                            mov al, [esi+u.fp]
1254
                                                 ; movb u.fp(r1),r2 / get an open file for this process
                                  <1>
1255
                                  <1>
                                                     bl, bl
1256 0000CBB9 08C0
                                 <1>
                                            or al, al
1257 0000CBBB 740D
                                 <1>
                                                  short sysfork_5
                                                  ; beq 2f / file has not been opened by parent,
1258
                                  <1>
                                                         ; / so branch
1259
                                 <1>
1260 0000CBBD B40A
                                 <1>
                                                  ah, 10 ; Retro UNIX 386 v1 fsp structure size = 10 bytes
1261 0000CBBF F6E4
                                  <1>
                                           mul
                                                 ah
1262
                                  <1>
                                            ;movzx ebx, ax
1263 0000CBC1 6689C3
                                  <1>
                                            mov bx, ax
                                            ;shl
1264
                                  <1>
                                                  bx, 3
1265
                                  <1>
                                                  ; asl r2 / multiply by 8
                                                        ; asl r2 / to get index into fsp table
1266
                                  <1>
1267
                                  <1>
                                                        ; asl r2
1268 0000CBC4 FE83[4E010300]
                                  <1>
                                                   byte [ebx+fsp-2]
                                                  ; incb fsp-2(r2) / increment number of processes
1269
                                  <1>
1270
                                  <1>
                                                            ; / using file, because child will now be
1271
                                  <1>
                                                             ; / using this file
1272
                                  <1> sysfork_5: ; 2:
1273 0000CBCA 46
                                  <1>
                                              inc esi
1274
                                               ; inc r1 / get next open file
                                  <1>
1275 0000CBCB 6683FE0A
                                  <1>
                                                     si, 10
                                               ; cmp r1,$10. / 10. files is the maximum number which
1276
                                  <1>
1277
                                  <1>
                                                          ; / can be opened
1278 0000CBCF 72E2
                                  <1>
                                                  short sysfork 4
1279
                                                  ; blt 1b / check next entry
                                  <1>
1280 0000CBD1 E98CFBFFFF
                                  <1>
                                                sysret
                                                  ; br sysret1
1281
                                  <1>
1282
                                  <1>
1283
                                  <1> syscreat: ; < create file >
                                         ; 13/11/2017
1284
                                  <1>
1285
                                  <1>
                                           ; 27/10/2016
1286
                                  <1>
                                           ; 25/10/2016, 26/10/2016
1287
                                  <1>
                                           ; 15/10/2016, 16/10/2016, 17/10/2016
1288
                                  <1>
                                           ; 10/10/2016 (TRDOS 386 = TRDOS v2.0)
                                                       -derived from INT_21H.ASM-
1289
                                  <1>
1290
                                  <1>
                                                         ("loc_INT21h_create_file")
1291
                                  <1>
                                                 10/07/2011 (12/03/2011)
                                             ;
                                                  INT 21h Function AH = 3Ch
1292
                                  <1>
1293
                                  <1>
                                                  Create File
                                             ;
1294
                                  <1>
                                                  INPUT
1295
                                  <1>
                                                     CX = Attributes
1296
                                                        DS:DX= Address of zero terminaned path name
                                  <1>
1297
                                  <1>
                                            ; 27/12/2015 (Retro UNIX 386 v1.1)
1298
                                  <1>
                                            ; 14/05/2015 (Retro UNIX 386 v1 - Beginning)
1299
                                  <1>
1300
                                  <1>
                                            ; 27/05/2013 (Retro UNIX 8086 v1)
1301
                                  <1>
1302
                                  <1>
                                            ; 'syscreat' called with two arguments; name and mode.
1303
                                  <1>
                                            ; u.namep points to name of the file and mode is put
                                            ; on the stack. 'namei' is called to get i-number of the file.
1304
                                  <1>
1305
                                  <1>
                                            ; If the file aready exists, it's mode and owner remain
                                            ; unchanged, but it is truncated to zero length. If the file
1306
                                  <1>
1307
                                  <1>
                                            ; did not exist, an i-node is created with the new mode via
                                            ; 'maknod' whether or not the file already existed, it is
1308
                                  <1>
1309
                                  <1>
                                            ; open for writing. The fsp table is then searched for a free
                                            ; entry. When a free entry is found, proper data is placed
1310
                                  <1>
                                            ; in it and the number of this entry is put in the u.fp list.
1311
                                  <1>
1312
                                  <1>
                                            ; The index to the u.fp (also know as the file descriptor)
                                            ; is put in the user's r0.
1313
                                  <1>
1314
                                  <1>
                                            ; Calling sequence:
1315
                                  <1>
1316
                                  <1>
                                                syscreate; name; mode
1317
                                  <1>
                                            ; Arguments:
1318
                                  <1>
                                                  name - name of the file to be created
1319
                                  <1>
                                                  mode - mode of the file to be created
                                            ; Inputs: (arguments)
1320
                                  <1>
1321
                                  <1>
                                            ; Outputs: *u.r0 - index to u.fp list
```

```
1322
                                 <1>
                                                          (the file descriptor of new file)
1323
                                 <1>
                                          ; ......
1324
                                 <1>
1325
                                 <1>
                                           ; Retro UNIX 8086 v1 modification:
1326
                                 <1>
                                                 'syscreate' system call has two arguments; so,
                                                 \star 1st argument, name is pointed to by BX register
1327
                                 <1>
1328
                                 <1>
                                                * 2nd argument, mode is in CX register
1329
                                 <1>
1330
                                 <1>
                                                 AX register (will be restored via 'u.r0') will return
                                                 to the user with the file descriptor/number
1331
                                 <1>
                                           ;
1332
                                 <1>
                                                 (index to u.fp list).
1333
                                 <1>
1334
                                 <1>
                                          ;call arg2
                                          ; * name - 'u.namep' points to address of file/path name
1335
                                 <1>
                                                 in the user's program segment ('u.segmnt')
1336
                                 <1>
1337
                                 <1>
                                                     with offset in BX register (as sysopen argument 1).
                                          ; * mode - sysopen argument 2 is in CX register
1338
1339
                                 <1>
                                         ; which is on top of stack.
1340
                                 <1>
1341
                                 <1>
                                          ; TRDOS 386 (10/10/2016)
1342
                                 <1>
                                          ; INPUT ->
1343
                                 <1>
                                            ; CL = File Attributes
1344
                                 <1>
1345
                                 <1>
                                                     bit 0 (1) - Read only file (R)
                                                      bit 1 (1) - Hidden file (H)
1346
                                 <1>
1347
                                 <1>
                                                         bit 2 (1) - System file (R)
1348
                                 <1>
                                                      bit 3 (1) - Volume label/name (V)
1349
                                 <1>
                                           ;
                                                        bit 4 (1) - Subdirectory (D)
                                                    bit 5 (1) - File has been archived (A) EBX = Pointer to filename (ASCIIZ) -path-
1350
                                 <1>
1351
                                 <1>
                                           ;
1352
                                 <1>
                                          ; OUTPUT ->
1353
                                 <1>
                                                     eax = File/Device Handle/Number (index) (AL)
1354
                                 <1>
1355
                                 <1>
                                                     cf = 1 -> Error code in AL
1356
                                 <1>
                                          ; Modified Registers: EAX (at the return of system call)
1357
                                 <1>
1358
                                 <1>
1359
                                 <1>
                                          ; Note: If the file is existing and it has not any one
1360
                                 <1>
                                                of S,H,R,V,D attributes, it will be truncated
                                                 to zero length; otherwise, access error will be
1361
                                 <1>
1362
                                 <1>
                                                 returned.
1363
                                 <1>
1364
                                 <1> sysmkdir_0:
1365 0000CBD6 F6C108
                                <1>
                                     test cl, 08h; Volume name
1366 0000CBD9 740A
                                <1>
                                          jz short syscreat_0
1367
                                <1>
                                          ; Volume name or long name creation
1368
                                <1>
1369
                                <1>
                                          ; is not permitted (in TRDOS 386)!
1370 0000CBDB B80B000000
                                 <1>
                                          mov eax, ERR_FILE_ACCESS ; 11 ; 'permission denied !'
1371 0000CBE0 E926020000
                                <1>
                                           jmp sysopen_dev_err
1372
                                <1>
1373
                                 <1> syscreat_0:
1374
                                <1>
                                       ;mov [u.namep], ebx
1375 0000CBE5 51
                                <1>
                                          push ecx
                                        mov esi, ebx
1376 0000CBE6 89DE
                                <1>
1377
                                <1>
                                          ; file name is forced, change directory as temporary
1378
                                 <1>
                                          ;mov ax, 1
1379
                                          ;mov [FFF_Valid], ah ; 0 ; reset ; 17/10/2016
                                 <1>
1380
                                 <1>
                                          ;call set_working_path
1381 0000CBE8 E892300000
                                <1>
                                          call set_working_path_x ; 17/10/2016
1382 0000CBED 0F82D7000000
                                <1>
                                          jc syscreat_err
                                 <1>
                                          ; 16/10/2016
1384
                                 <1>
1385 0000CBF3 803D[77650100]00
                                <1>
                                         cmp byte [SWP_inv_fname], 0
1386 0000CBFA 776C
                                 <1>
                                          ja short syscreat_inv_fname; invalid file name!
1387
                                 <1>
1388
                                 <1>
                                          ; Here, we have a valid path and also a valid file name
1389
                                 <1>
                                          ; (Working dir has been changed if the path
1390
                                 <1>
                                          ; -file name string- had contained a dir name.)
1391
                                 <1>
1392 0000CBFC 6631C0
                                 <1>
                                          xor ax, ax
                                          ;mov esi, FindFile_Name
call find_first_file
1393
                                 <1>
1394 0000CBFF E8E3B6FFFF
                                 <1>
1395 0000CC04 59
                                 <1>
1396
                                 <1>
                                                ; ESI = Directory Entry (FindFile_DirEntry) Location
                                                 ; EDI = Directory Buffer Directory Entry Location
1397
                                 <1>
1398
                                 <1>
                                                ; EAX = File Size
1399
                                 <1>
                                                ; BL = Attributes of The File/Directory
                                                ; BH = Long Name Yes/No Status (>0 is YES)
1400
                                 <1>
                                                ; DX > 0 : Ambiguous filename chars are used
1401
                                 <1>
1402 0000CC05 7269
                                 <1>
                                           jc short syscreat_1 ; file not found (the good!)
                                 <1>
                                                               ; or another error (the bad')
1404
                                 <1>
1405
                                 <1>
                                           ; (& the uggly!) truncate file to zero length before open
1406
                                 <1>
                                           ;'*' and '?' already checked at 'set_working_path' stage
1407
                                 <1>
1408
                                 <1>
                                           ; and dx, dx
1409
                                 <1>
                                           ;jnz short sysmkdir_err ; permission denied
                                                                ; invalid filename chars
1410
                                 <1>
1411
                                 <1>
                                           ;test cl, 10h; subdirectory?
1412
                                 <1>
1413
                                 <1>
                                           ;jnz short sysmkdir_err
                                 <1>
1414
1415
                                 <1>
                                           ; BL = File Attributes:
                                                     bit 0 (1) - Read only file (R)
1416
                                 <1>
                                           ;
                                                       bit 1 (1) - Hidden file (H)
1417
                                 <1>
1418
                                 <1>
                                                         bit 2 (1) - System file (R)
                                 <1>
1419
                                                       bit 3 (1) - Volume label/name (V)
1420
                                 <1>
                                                         bit 4 (1) - Subdirectory (D)
1421
                                 <1>
                                                       bit 5 (1) - File has been archived
1422
                                 <1>
1423
                                 <1>
                                           ; * existing directory must not be truncated
                                           ; (we don't know it is empty or not, at this stage)
1424
                                 <1>
                                           ; * existing volume name (or a long name) can not be
1425
                                 <1>
1426
                                 <1>
                                           ; re-created or truncated by 'syscreat'
```

```
1427
                                 <1>
                                           ; * A file with S, H, R attributes must not be truncated
1428
                                 <1>
                                           ; (change attributes to normal, if you need truncate it)
1429
                                 <1>
1430 0000CC07 F6C31F
                                 <1>
                                           test bl, 00011111b ; check attributes of existing file
1431 0000CC0A 754E
                                 <1>
                                           jnz short sysmkdir err
1432
                                 <1>
                                          ;; normal file, OK to continue...
1433
                                <1>
1434
                                <1>
1435
                                <1>
                                          ; ESI = FindFile_DirEntry
                                          mov ax, [esi+DirEntry_FstClusHI]; 20
1436 0000CC0C 668B4614
                                <1>
1437 0000CC10 C1E010
                                <1>
                                          shl eax, 16; 13/11/2017
1438 0000CC13 668B461A
                                <1>
                                          mov
                                                ax, [esi+DirEntry FstClusLO] ; 26
1439
                                <1>
                                          ; EAX = First cluster to be truncated/unlinked
1440 0000CC17 57
                                          push edi
                                <1>
1441 0000CC18 51
                                <1>
                                          push ecx
1442 0000CC19 BE00010900
                                <1>
                                          mov
                                                esi, Logical_DOSDisks
                                          sub ecx, ecx
1443 0000CC1E 29C9
                                <1>
1444 0000CC20 8A2D[86590100]
                                <1>
                                          mov ch, [Current_Drv]
1445 0000CC26 01CE
                                <1>
                                          add
                                                esi, ecx
1446
                                <1>
                                          ; ESI = Logical dos drive description table address
1447 0000CC28 E8C9F7FFFF
                                <1>
                                          call truncate_cluster_chain
1448 0000CC2D 59
                                 <1>
                                          pop
                                                 ecx
1449 0000CC2E 5F
                                <1>
                                          pop
                                                edi
1450 0000CC2F 7230
                                <1>
                                          jс
                                                 short syscreate_truncate_err
                                 <1>
1451
1452
                                          ; 26/10/2016
                                 <1>
                                 <1>
                                          ; EDI = Directory entry address in directory buffer
                                          ; Update directory entry
1454
                                 <1>
1455 0000CC31 E848DCFFFF
                                <1>
                                          call convert_current_date_time
                                          ; OUTPUT -> DX = Date in dos dir entry format
1456
                                <1>
1457
                                <1>
                                                  AX = Time in dos dir entry format
                                                 [edi+DirEntry_WrtTime], ax
[edi+DirEntry_WrtDate], dx
1458 0000CC36 66894716
                                <1>
                                          mov
1459 0000CC3A 66895718
                                <1>
                                          mov
1460 0000CC3E 66895712
                                <1>
                                                [edi+DirEntry_LastAccDate], dx
                                      mov
                                      xor
1461 0000CC42 31C0
                                <1>
                                                eax, eax; file size = 0
1462 0000CC44 89471C
                                                [edi+DirEntry_FileSize], eax ; 0
                                <1>
                                          mov
1463 0000CC47 C605[AC600100]02 <1>
                                                byte [DirBuff ValidData], 2; data changed sign
                                          mov
1464 0000CC4E BE[78620100]
                                <1>
                                          mov
                                                esi, FindFile_DirEntry
                                                 dl, 1; open file for writing
1465 0000CC53 B201
                                <1>
                                          mov
1466 0000CC55 E9AA000000
                                <1>
                                          jmp sysopen_2
1467
                                <1>
                                 <1> sysmkdir err:
1468
                                      ; \overline{1} = write, 2 = read & write, >2 = invalid
1469
                                <1>
                                           mov eax, ERR FILE ACCESS ; 11 ; 'permission denied !'
1470 0000CC5A B80B000000
                                <1>
1471 0000CC5F EB73
                                <1>
                                           jmp short sysopen err
1472
                                <1>
                                <1> syscreate truncate err:
1473
1474 0000CC61 B812000000
                                      mov eax, ERR_DRV_WRITE ; 18 ; 'disk write error !'
                                <1>
1475 0000CC66 EB6C
                                           jmp short sysopen err
                                <1>
1476
                                <1>
1477
                                <1> syscreat_inv_fname: ; invalid file name chars
                                      ; \overline{1}6/1\overline{0}/2016
1478
                                 <1>
1479 0000CC68 B81A000000
                                           mov eax, ERR_INV_FILE_NAME ; 26 ; invalid file name chars
                                <1>
1480 0000CC6D 59
                                <1>
                                          pop
                                               ecx
1481 0000CC6E EB64
                                <1>
                                          jmp sysopen_err
1482
                                <1>
                                <1> syscreat 1:
1484
                                <1>
                                      ; Error code in EAX
                                          cmp al, 02h ; 'File not found' error
jne sysopen_err
1485 0000CC70 3C02
                                <1>
1486 0000CC72 7560
                                <1>
1487
                                <1>
1488 0000CC74 F6C110
                                          test cl, 10h; Directory
jnz sysmkdir_2
                                 <1>
1489 0000CC77 0F852C020000
                                <1>
1490
                                 <1>
                                 <1> syscreat_2:
1491
                                      mov esi, FindFile_Name
1492 0000CC7D BE[68620100]
                                <1>
                                          ;xoredx, edx
                                <1>
1494 0000CC82 31C0
                                           xor eax, eax ; File Size = 0
                                <1>
                                        xor ebx, ebx
dec ebx; FFFFFFFF -> create empty file
1495 0000CC84 31DB
                                <1>
1496 0000CC86 4B
                                <1>
1497
                                <1>
                                                                     (only for FAT fs)
                                          ; CL = File Attributes
1498
                                 <1>
                                          call create_file
1499 0000CC87 E8F8EBFFFF
                                <1>
                                        jc sysopen_err
1500 0000CC8C 7246
                                 <1>
1501
                                 <1>
                                                ; EAX = New file's first cluster
1502
                                 <1>
                                                 ; ESI = Logical Dos Drv Descr. Table Addr.
1503
                                 <1>
                                                ; EBX = offset CreateFile Size
1504
                                 <1>
                                                ; ECX = Sectors per cluster (<256)
1505
                                 <1>
                                                 ; EDX = Directory entry index/number (<65536)
1506
                                 <1>
                                           ; 26/10/2016
                                           ;mov esi, Directory_Buffer
1507
                                 <1>
                                           ;shl dx, 5; *32;add esi, edx
1508
                                 <1>
1509
                                 <1>
1510
                                 <1>
                                           ;; esi = directory entry address in directory buffer
1511
                                 <1>
1512
                                 <1>
                                           ; Here, directory entry has been created but last
1513
                                 <1>
                                           ; modification date & time of the parent dir has not
1514
                                 <1>
                                           ; been updated, yet!
1515
                                 <1>
                                           ; (Note: Directory and FAT buffers have been updated...)
1516
                                 <1>
1517 0000CC8E E824DDFFFF
                                           call update parent dir lmdt; now, it is OK too!
                                 <1>
1518
                                 <1>
1519
                                           ; 25/10/2016
                                 <1>
                                           mov ax, 1800h
1520 0000CC93 66B80018
                                 <1>
                                                 esi, FindFile Name
1521 0000CC97 BE[68620100]
                                 <1>
                                           mov
                                           call find_first_file
1522 0000CC9C E846B6FFFF
                                 <1>
1523 0000CCA1 7231
                                 <1>
                                                short sysopen_err
                                           jс
1524
                                 <1>
                                           ; Only possible error after here is
1525
                                 <1>
                                           ; "too many open files !" error.
1526
                                 <1>
1527
                                 <1>
1528
                                 <1>
                                          ; If "syscreat" will return with that error,
                                           ; (the file has been created but it could not be opened)
1529
                                 <1>
1530
                                 <1>
                                           ; the user must retry to open this file again
1531
                                 <1>
                                           ; or must close another file before using
```

```
; "sysopen" system call.
1532
                                <1>
1533
                                <1>
1534 0000CCA3 B201
                                <1>
                                         mov dl, 1; open file for writing
1535
                                <1>
                                          ; ESI = Directory Entry (FindFile_DirEntry) Location
1536
                                <1>
                                          ; EAX = File Size (= 0)
1537 0000CCA5 EB5D
                                <1>
                                          jmp short sysopen 2
1538
                                <1>
                                <1> sysopen: ;<open file>
1539
                                       ; 26/10/2016
1540
                                <1>
1541
                                <1>
                                         ; 24/10/2016
                                        ; 17/10/2016
1542
                                <1>
                                        ; 15/10/2016
                                <1>
1543
                                         ; 06/10/2016, 07/10/2016, 08/10/2016
1544
                                <1>
1545
                                <1>
                                        ; 05/10/2016 (TRDOS 386 = TRDOS v2.0)
                                                     -derived from INT 21H.ASM-
1546
                                <1>
                                        ;
1547
                                <1>
                                                      ("loc_INT21h_open_file")
                                          ; 26/02/2011
1548
                                 <1>
                                          ; INT 21h Function AH = 3Dh
1549
                                <1>
                                          ;
;
1550
                                <1>
                                                Open File
1551
                                <1>
                                               INPUT
                                                AL= File Access Value
1552
                                <1>
                                                   0- Open for reading
1553
                                 <1>
                                          ;
1554
                                <1>
                                          ;
                                                     1- Open for writing
1555
                                <1>
                                                       2- Open for reading and writing
1556
                                <1>
                                                      DS:DX= Pointer to filename (ASCIIZ)
                                           ;
1557
                                <1>
1558
                                <1>
                                          ; 14/05/2015 (Retro UNIX 386 v1 - Beginning)
                                         ; 22/05/2013 - 27/05/2013 (Retro UNIX 8086 v1)
1559
                                <1>
1560
                                <1>
1561
                                         ; 'sysopen' opens a file in following manner:
                                <1>
1562
                                <1>
                                              1) The second argument in a sysopen says whether to
                                                open the file ro read (0) or write (>0).
1563
                                 <1>
1564
                                <1>
                                               2) I-node of the particular file is obtained via 'namei'.
1565
                                <1>
                                         ; 3) The file is opened by 'iopen'.
1566
                                <1>
                                              4) Next housekeeping is performed on the fsp table
1567
                                <1>
                                               and the user's open file list - u.fp.
                                               a) u.fp and fsp are scanned for the next available slot.
1568
                                 <1>
1569
                                <1>
                                               b) An entry for the file is created in the fsp table.
1570
                                <1>
                                                c) The number of this entry is put on u.fp list.
                                                d) The file descriptor index to u.fp list is pointed
1571
                                <1>
1572
                                <1>
                                                   to by u.r0.
1573
                                 <1>
                                         ; Calling sequence:
1574
                                <1>
                                              sysopen; name; mode
1575
                                <1>
                                        ; Arguments:
1576
                                <1>
1577
                                <1>
                                          ; name - file name or path name
                                                mode - 0 to open for reading
1578
                                <1>
1579
                                                1 to open for writing
                                <1>
                                        : Outputs: (arguments)
1580
                                <1>
1581
                                <1>
                                         ; Outputs: *u.r0 - index to u.fp list (the file descriptor)
                                                      is put into r0's location on the stack.
1582
                                <1>
1583
                                 <1>
                                         ; ......
1584
                                <1>
1585
                                <1>
                                         ; Retro UNIX 8086 v1 modification:
1586
                                <1>
                                                'sysopen' system call has two arguments; so,
                                                \star 1st argument, name is pointed to by BX register
1587
                                <1>
1588
                                 <1>
                                                * 2nd argument, mode is in CX register
1589
                                <1>
1590
                                <1>
                                                AX register (will be restored via 'u.r0') will return
1591
                                <1>
                                                to the user with the file descriptor/number
1592
                                <1>
                                                (index to u.fp list).
1593
                                 <1>
1594
                                <1>
                                          ;call arg2
                                         ; * name - 'u.namep' points to address of file/path name
1595
                                <1>
                                        ; in the user's program segment ('u.segmnt')
1596
                                <1>
1597
                                <1>
                                                    with offset in BX register (as sysopen argument 1).
                                         ; * mode - sysopen argument 2 is in CX register
1598
                                <1>
1599
                                <1>
                                                    which is on top of stack.
1600
                                <1>
1601
                                <1>
                                          ; jsr r0,arg2 / get sys args into u.namep and on stack
1602
                                <1>
1603
                                 <1>
                                                ; system call registers: ebx, ecx (through 'sysenter')
1604
                                <1>
                                         ; TRDOS 386 (05/10/2016)
1605
                                <1>
1606
                                 <1>
                                         ;
                                           ; INPUT ->
1607
                                <1>
1608
                                 <1>
                                              CL = File Access Value (Open Mode)
                                                    0 - Open file for reading
1609
                                <1>
                                                     1 - Open file for writing
1610
                                 <1>
1611
                                 <1>
                                                        2 - Open device for reading
                                           ;
                                                    3 - Open device for writing
1612
                                 <1>
                                                      EBX = Pointer to filename/devicename (ASCIIZ)
1613
                                 <1>
                                          ; OUTPUT ->
1614
                                 <1>
1615
                                 <1>
                                                     eax = File/Device Handle/Number (index) (AL)
                                 <1>
1616
                                                     cf = 1 -> Error code in AL
1617
                                <1>
1618
                                <1>
                                          ; Modified Registers: EAX (at the return of system call)
1619
                                <1>
1620
                                <1>
1621 0000CCA7 80F901
                                                cl, 1; read file (0), write file (1)
                                <1>
                                          cmp
1622 0000CCAA 7614
                                <1>
                                                short sysopen_0
1623
                                <1>
1624 0000CCAC 80F903
                                                cl, 3
                                <1>
                                          cmp
                                                sysopen_device
1625 0000CCAF 0F8640010000
                                <1>
                                          jna
1626
                                <1>
1627
                                <1>
                                          ; Invalid access code
1628 0000CCB5 B817000000
                                          mov eax, ERR INV PARAMETER
                                <1>
1629 0000CCBA 0F874B010000
                                                sysopen dev err
                                <1>
1630
                                <1>
1631
                                <1> sysopen 0:
1632
                                <1>
                                          ;mov [u.namep], ebx
1633 0000CCC0 51
                                <1>
                                          push ecx
1634 0000CCC1 89DE
                                <1>
                                          mov
                                               esi, ebx
1635
                                <1>
                                          ; file name is forced, change directory as temporary
1636
                                <1>
                                          ;mov ax, 1
```

```
1637
                                         ;mov [FFF_Valid], ah ; 0 ; reset ; 17/10/2016
                                <1>
1638
                                <1>
                                         ;call set_working_path
                                         call set_working_path_x ; 17/10/2016
1639 0000CCC3 E8B72F0000
                               <1>
1640 0000CCC8 731E
                               <1>
                                         jnc short sysopen_1
                                <1>
1641
1642
                                <1> syscreat_err: ; ecx = file attributes (for 'syscreat')
1643 0000CCCA 59
                               <1>
                                         pop ecx ; open mode
1644 0000CCCB 21C0
                                          and
                                               eax, eax ; 0 -> Bad Path!
                               <1>
1645 0000CCCD 7505
                                <1>
                                               short sysopen_err
                                          jnz
                                          ; eax = 0
1646
                               <1>
1647 0000CCCF B80C000000
                                <1>
                                        mov eax, ERR_DIR_NOT_FOUND ; Directory not found !
1648
                                <1> sysopen err:
1649 0000CCD4 A3[64030300]
                                <1> mov [u.r0], eax
1650 0000CCD9 A3[C8030300]
                               <1>
                                               [u.error], eax
                                         call reset_working_path
1651 0000CCDE E871300000
                                <1>
1652 0000CCE3 E95AFAFFFF
                                <1>
                                         jmp
                                               error
1653
                                <1>
1654
                                <1> sysopen_1:
1655
                                <1>
                                       ;mov esi, FindFile_Name
1656 0000CCE8 66B80018
                               <1>
                                          mov ax, 1800h; Only files
1657 0000CCEC E8F6B5FFFF
                               <1>
                                          call find_first_file
1658 0000CCF1 5A
                                <1>
                                                edx
                                         pop
1659 0000CCF2 72E0
                                <1>
                                                short sysopen_err ; eax = 2 (File not found !)
                                         jс
                                <1>
1661
                                <1>
                                         ; check_open_file_attr_access_code
1662
                                <1>
1663 0000CCF4 F6C307
                                <1>
                                          test bl, 7 ; system, hidden, readonly
1664 0000CCF7 740B
                                <1>
                                          jz short sysopen_2
                                <1>
1666 0000CCF9 20D2
                                <1>
                                          and dl, dl; 0 = read mode
1667 0000CCFB 7407
                                <1>
                                                short sysopen_2
1668
                                <1>
                                          ; 1 = write, 2 = read & write, >2 = invalid
1669
                                <1>
1670 0000CCFD B80B000000
                                <1>
                                          mov eax, ERR_FILE_ACCESS ; 11 = 'permission denied !'
1671 0000CD02 EBD0
                                <1>
                                            jmp short sysopen_err
1672
                                <1>
1673
                                <1> sysopen 2:
1674
                                         ; esi = Directory Entry (FindFile_DirEntry) Location
                                <1>
1675 0000CD04 89F3
                                <1>
                                          mov ebx, esi
1676 0000CD06 31F6
                               <1>
                                                esi, esi ; 0
                                          xor
1677 0000CD08 31FF
                               <1>
                                           xor
                                                   edi, edi ; 0
1678
                                <1> sysopen_3: ; scan the list of entries in fsp table
1679 0000CD0A 80BE[6A030300]00 <1> cmp
                                                   byte [esi+u.fp], 0
                                                    short sysopen_4 ; empty slot
1680 0000CD11 760F
                                <1>
                                           jna
1681 0000CD13 6646
                                <1>
                                           inc
                                                   si
1682 0000CD15 6683FE0A
                                <1>
                                            cmp
                                                   si, 10
1683 0000CD19 72EF
                               <1>
                                        jb short sysopen_3
1684
                                <1> toomanyf:
                                                eax, ERR TOO MANY FILES ; too many open files !
1685 0000CD1B B80D000000
                                <1>
1686 0000CD20 EBB2
                                <1>
                                                short sysopen err
                                          jmp
1687
                                <1>
1688
                                <1> sysopen_4:
1689 0000CD22 80BF[E6680100]00
                               <1>
                                       cmp
                                                   byte [edi+OF_MODE], 0 ; Scan open files table
1690 0000CD29 760A
                                <1>
                                                  short sysopen_5
1691 0000CD2B 6647
                                <1>
                                          inc
                                                di
1692 0000CD2D 6683FF0A
                               <1>
                                          cmp
                                                di, OPENFILES ; max. number of open files (=10)
1693 0000CD31 72EF
                               <1>
                                          jb
                                                short sysopen 4
1694 0000CD33 EBE6
                                <1>
                                         jmp
                                                short toomanyf
1695
                                <1>
1696
                                <1> sysopen_5:
                                      inc
1697 0000CD35 FEC2
                                <1>
1698 0000CD37 8897[E6680100]
                                <1>
                                          mov
                                                   [edi+OF MODE], dl
1699 0000CD3D 8A15[26620100]
                               <1>
                                          mov dl, [FindFile_Drv]
1700 0000CD43 8897[DC680100]
                                <1>
                                                  [edi+OF_DRIVE], dl ; Logical DOS drive number
1701 0000CD49 66C1E702
                                <1>
                                         shl di, 2; *4 (dword offset)
1702
                                <1>
1703 0000CD4D 8987[2C690100]
                                <1>
                                         mov
                                                [edi+OF_SIZE], eax ; File size in bytes
1704
                                <1>
1705 0000CD53 668B4314
                                <1>
                                          mov ax, [ebx+DirEntry_FstClusHI]
1706 0000CD57 C1E010
                                <1>
                                          shl eax, 16
                                                ax, [ebx+DirEntry FstClusLO]
1707 0000CD5A 668B431A
                                <1>
                                          mov
1708 0000CD5E 8987[B4680100]
                                <1>
                                                 [edi+OF_FCLUSTER], eax ; First cluster
                                          mov
                                                  [edi+OF_CCLUSTER], eax ; Current cluster
1709 0000CD64 8987[CC690100]
                                <1>
                                          mov
1710
                                <1>
1711 0000CD6A 31DB
                                <1>
                                           xor ebx, ebx
                                                    [edi+OF_POINTER], ebx ; offset pointer (0)
1712 0000CD6C 899F[04690100]
                                <1>
                                           mov
1713 0000CD72 899F[F4690100]
                                <1>
                                                    [edi+OF CCINDEX], ebx; cluster index (0)
                                           mov
1714
                                <1>
1715 0000CD78 A1[98620100]
                                <1>
                                                eax, [FindFile_DirFirstCluster]
1716 0000CD7D 8987[54690100]
                                                [edi+OF_DIRFCLUSTER], eax
                                <1>
                                          mov
1717
                                <1>
1718 0000CD83 A1[9C620100]
                                 <1>
                                          mov
                                                eax, [FindFile_DirCluster]
1719 0000CD88 8987[7C690100]
                                <1>
                                          mov
                                               [edi+OF_DIRCLUSTER], eax
                                 <1>
1721
                                          ; Get (& Save) Volume ID
                                <1>
1722
                                <1>
                                          ; Important for files of removable drives
1723
                                <1>
                                         ; (In order to check the drive has same volume/disk)
1724 0000CD8E 88D7
                                <1>
                                        mov bh, dl
                                        add ebx, Logical_DOSDisks
mov al, [ebx+LD_FATType]
1725 0000CD90 81C300010900
                               <1>
1726 0000CD96 8A4303
                               <1>
                               <1>
1727 0000CD99 3C01
                                          cmp al, 1
                                            jb short sysopen_6_fs
1728 0000CD9B 7209
                               <1>
1729 0000CD9D 3C02
                               <1>
                                            cmp al, 2
1730 0000CD9F 770A
                               <1>
                                            ja short sysopen 6 fat32
                               <1> sysopen_6_fat:
                               <1>
1732 0000CDA1 8B432D
                                           mov eax, [ebx+LD_BPB+VolumeID]
1733 0000CDA4 EB08
                               <1>
                                            jmp short sysopen 7
                               <1> sysopen_6_fs:
1734
1735 0000CDA6 8B4328
                                <1>
                                           mov eax, [ebx+LD_FS_VolumeSerial]
1736 0000CDA9 EB03
                               <1>
                                            jmp short sysopen_7
1737
                                <1> sysopen_6_fat32:
1738 0000CDAB 8B4349
                                <1>
                                          mov eax, [ebx+LD_BPB+FAT32_VolID]
1739
                                <1> sysopen_7:
                                           mov [Current VolSerial], eax
1740 0000CDAE A3[7C590100]
                                <1>
                                <1>
1741
```

```
1742 0000CDB3 8987[A4690100]
                                         mov [edi+OF_VOLUMEID], eax
1743
                                         ; 24/10/2016
1744
                                <1>
1745 0000CDB9 66D1EF
                                <1>
                                         shr di, 1; 4/2, word offset
1746 0000CDBC 668B1D[A0620100]
                                <1>
                                               bx, [FindFile DirEntryNumber]
                                         mov
1747 0000CDC3 66899F[1C6A0100]
                               <1>
                                         mov
                                               [edi+OF_DIRENTRY], bx
                                <1>
1749 0000CDCA 31D2
                                <1>
                                         xor edx, edx
1750
                                <1>
                                         ;shr di, 2 ; /4 (byte offset)
                                         shr di, 1 ; 2/2, byte offset
1751 0000CDCC 66D1EF
                               <1>
1752 0000CDCF 8897[FA680100] <1>
                                               byte [edi+OF_OPENCOUNT], dl ; 0
                                         mov
1753 0000CDD5 8897[F0680100]
                                <1>
                                         mov
                                               byte [edi+OF_STATUS], dl ; 0
1754
                                <1>
1755 0000CDDB 89FB
                                <1>
                                         mov
                                               ebx, edi
1756 0000CDDD FEC3
                                <1>
                                         inc bl
1757
                                <1>
1758 0000CDDF 889E[6A030300]
                               <1>
                                                   [esi+u.fp], bl ; Open File Entry Number
1759 0000CDE5 8935[64030300]
                               <1>
                                                 [u.r0], esi; move index to u.fp list
                                         mov
1760
                                <1>
                                                         ; into eax on stack
1761
                                <1>
1762 0000CDEB E8642F0000
                               <1>
                                         call
                                                     reset_working_path
1763
                                <1>
1764 0000CDF0 E96DF9FFFF
                                         jmp sysret
                                <1>
1765
                                <1>
                                         ; (Retro UNIX 386 v1.0)
1766
                                <1>
                                         ; 'fsp' table (10 bytes/entry)
1767
                                <1>
                                         ; bit 15
1768
                                        ; ---|-----
1769
                                <1>
                                       ; r/w| i-number of open file
1770
                                <1>
                                         ; ---|-----
1771
                                <1>
1772
                                <1>
                                        ; device number
                                        ; ------
1773
                                <1>
1774
                                         ; offset pointer, r/w pointer to file (bit 0-15)
                                <1>
                                        ; -----
1775
                                <1>
                                        ; offset pointer, r/w pointer to file (bit 16-31)
1776
                                <1>
1777
                                <1>
                                         ; -----
                                         ; flag that says file | number of processes ; has been deleted | that have file open
1778
1779
                                <1>
                                         ; -----
1780
                                <1>
1781
                                <1>
1782
                                <1> sysopen_device:
                                      ; 15/10/2016
1783
                                <1>
                                         ; 08/10/2016
1784
                                <1>
                                         ; 07/10/2016 (TRDOS 386 = TRDOS v2.0)
1785
                               <1>
                              1786 0000CDF5 51
1787 0000CDF6 89E5
1788 0000CDF8 B91000000
1789 0000CDFD 29CC
1790 0000CDFF 89E7
1791 0000CE01 89DE
1792 0000CE03 E83F1A0000
1793 0000CE08 7310
1794
1795 0000CE0A 59
1796
                               <1> sysopen_dev_err:
1790
1797 0000CE0B A3[64030300]
1798 0000CE10 A3[C8030300]
                               <1>
                                        mov [u.r0], eax
                                         mov [u.error], eax
                               <1>
                                      jmp error
1799 0000CE15 E928F9FFFF
                               <1>
1800
                               <1> sysopen_dev_0:
                               <1> mov esi, edi ; Device name addr (max. 16 bytes, ASCIIZ)
1801 0000CE1A 89FE
                                                ; for example: "tty, TTY, /dev/tty"
1802
                          ; for example.

<1> call get_device_number

<1> mov esp, ebp

<1> pop ecx

<1> jnc short sysopen_dev_1

<1> mov eax, ERR_INV_DEV_NAME

<1> jmp short sysopen_dev_err
                               <1>
1803 0000CE1C E8D9310000
1804 0000CE21 89EC
1805 0000CE23 59
1806 0000CE24 7307
                                               eax, ERR_INV_DEV_NAME ; 24 ; 'invalid device name !'
1807 0000CE26 B818000000
1808 0000CE2B EBDE
1809
                                <1> sysopen_dev_1:
                                     ; eax = Device Number (AL)
1810
                                <1>
                                         ; cl = Open mode (2 = device read, 3 = device write)
1811
                                <1>
                                     xor ebx, ebx; 0
1812 0000CE2D 31DB
                                <1>
1813
                                <1> sysopen_dev_2: ; scan the list of entries
1814 0000CE2F 389B[6A030300]
                               <1> cmp [ebx+u.fp], bl; 0
1815 0000CE35 760E
                               <1>
                                          jna
                                                   short sysopen_dev_3 ; empty slot
                                       inc
cmp
1816 0000CE37 FEC3
                                <1>
                                                   bl
                                                bl, 10
1817 0000CE39 80FB0A
                               <1>
                                     jb short sysopen_dev_2;
mov eax, ERR_TOO_MANY_F;
jmp short sysopen_dev_eax
1818 0000CE3C 72F1
                               <1>
1819
                                <1>
                                         mov eax, ERR_TOO_MANY_FILES ; too many open files !
jmp short sysopen_dev_err
1820 0000CE3E B80D000000
                                <1>
1821 0000CE43 EBC6
                                <1>
                                <1> sysopen_dev_3:
1823 0000CE45 891D[64030300]
                                <1>
                                         mov [u.r0], ebx; File/Device index/handle/descriptor
                                         ; eax = device number (entry offset)
                                <1>
1825 0000CE4B 8AA8[78660100]
                                <1>
                                         mov ch, [eax+DEV_ACCESS] ; bit 0 = accessable by users
                                                                 ; bit 1 = read access perm
                                <1>
1826
                                                                 ; bit 2 = write access perm
1827
                                <1>
1828
                                <1>
                                                                 ; bit 3 = IOCTL permit to users
1829
                                <1>
                                                                 ; bit 4 = block device if set
1830
                                <1>
                                                                 ; bit 5 = 16 bit or 1024 byte
1831
                                                                 ; bit 6 = 32 bit or 2048 byte
                                <1>
1832
                                <1>
                                                                 ; bit 7 = installable device drv
1833 0000CE51 F6C501
                                <1>
                                         test ch, 1; accessable by normal users (except root)
1834 0000CE54 7510
                                                short sysopen_dev_4 ; yes, permission has been given
                                <1>
                                         jnz
1835 0000CE56 803D[B0030300]00
                                <1>
                                                byte [u.uid], 0 ; root?
1836 0000CE5D 7607
                                <1>
                                               short sysopen dev 4 ; superuser can open all devices
                                         jna
1837
                                <1> sysopen_dev_perm_err:
                                         mov
1838 0000CE5F B80B000000
                                <1>
                                               eax, ERR_DEV_ACCESS ; 11 = 'permission denied !'
1839 0000CE64 EBA5
                                <1>
                                         jmp
                                               short sysopen_dev_err
                                <1> sysopen_dev_4:
1840
                                         \overline{\text{shr}} \overline{\text{ch}}, 1; result: 1 = read, 2 = write, 3 = r & w
1841 0000CE66 D0ED
                               <1>
1842 0000CE68 FEC9
                               <1>
                                         dec
                                                cl ; result: 1 = read, 2 = write
1843 0000CE6A 84E9
                                <1>
                                         test cl, ch
1844 0000CE6C 74F1
                                <1>
                                         jz
                                                short sysopen_dev_perm_err
1845
                                <1>
1846 0000CE6E D0E5
                                <1>
                                               ch, 1 ; bit 0 = 0
                                         shl
```

```
1847
                                          ; eax = device number (entry offset)
                                 <1>
                                         call device_open
jc short sysopen_dev_perm_err
1848 0000CE70 E8A1320000
                                <1>
1849 0000CE75 72E8
                                <1>
1850
                                <1>
1851
                                <1>
                                          ; eax = device number (entry offset)
1852 0000CE77 0C80
                                <1>
                                          or al, 80h; set device bit (set bit 7 to 1)
1853 0000CE79 8B1D[64030300]
                                <1>
                                          mov
                                                ebx, [u.r0]
1854 0000CE7F 8883[6A030300]
                                                [ebx+u.fp], al
                                <1>
                                                                ; bit 7 (=1) points to device
                                          mov
1855
                                <1>
1856 0000CE85 E9D8F8FFFF
                                <1>
                                          jmp sysret
1857
                                 <1>
1858
                                 <1> sysmkdir: ; < make directory >
                                       ; 15/10/2016
1859
                                 <1>
1860
                                 <1>
                                          ; 10/10/2016 (TRDOS 386 = TRDOS v2.0)
                                                     -derived from INT 21H.ASM-
1861
                                 <1>
                                        ;
                                                     ("loc INT21h create file")
1862
                                 <1>
                                          ; 10/07/2011 (12/03/2011)
1863
                                 <1>
                                          ; INT 21h Function AH = 3Ch
1864
                                 <1>
1865
                                 <1>
                                                Create File
1866
                                 <1>
                                                INPUT
                                            ;
                                                 CX = Attributes
1867
                                 <1>
1868
                                 <1>
                                                       DS:DX= Address of zero terminaned path name
                                            ;
1869
                                 <1>
1870
                                 <1>
1871
                                 <1>
                                          ; 14/05/2015 (Retro UNIX 386 v1 - Beginning)
1872
                                 <1>
                                          ; 27/05/2013 - 02/08/2013 (Retro UNIX 8086 v1)
1873
                                         ; 'sysmkdir' creates an empty directory whose name is
1874
                                 <1>
1875
                                 <1>
                                          ; pointed to by arg 1. The mode of the directory is arg 2.
1876
                                          ; The special entries '.' and '..' are not present.
                                 <1>
1877
                                 <1>
                                         ; Errors are indicated if the directory already exists or
1878
                                 <1>
                                          ; user is not the super user.
1879
                                 <1>
1880
                                 <1>
                                         ; Calling sequence:
                                        ; sysmkdir; name; mode
1881
                                 <1>
1882
                                 <1>
                                          ; Arguments:
                                          ; name - points to the name of the directory
1883
                                 <1>
1884
                                 <1>
                                               mode - mode of the directory
1885
                                 <1>
                                          ; Inputs: (arguments)
                                          ; Outputs: -
1886
                                 <1>
                                          ; (sets 'directory' flag to 1;
1887
                                 <1>
                                               'set user id on execution' and 'executable' flags to 0)
1888
                                 <1>
                                          ;
1889
                                 <1>
                                          i ......
1890
                                 <1>
1891
                                 <1>
                                          ; Retro UNIX 8086 v1 modification:
                                                 'sysmkdir' system call has two arguments; so,
1892
                                 <1>
                                                 * 1st argument, name is pointed to by BX register
1893
                                 <1>
1894
                                                * 2nd argument, mode is in CX register
                                 <1>
1895
                                 <1>
1896
                                 <1>
                                          ; TRDOS 386 (10/10/2016)
1897
                                 <1>
                                          ; INPUT ->
1898
                                 <1>
                                            ; CL = Directory Attributes
1899
                                 <1>
                                                    bit 0 (1) - Read only file/dir (R)
1900
                                 <1>
                                                     bit 1 (1) - Hidden file/dir (H)
1901
                                 <1>
                                        ;
1902
                                 <1>
                                                       bit 2 (1) - System file/dir (R)
                                          ;
                                                     bit 3 (1) - Volume label/name (V)
1903
                                 <1>
1904
                                 <1>
                                                         bit 4 (1) - Subdirectory (D)
                                          ;
                                                    bit 5 (1) - File/Dir has been archived (A)
1905
                                 <1>
1906
                                 <1>
                                                   CX = 0 -> create normal directory
                                          ;
1907
                                 <1>
                                          ;
                                                      EBX = Pointer to directory name (ASCIIZ) -path-
1908
                                 <1>
                                          ;
                                          ; OUTPUT ->
1909
                                 <1>
1910
                                 <1>
                                                     eax = First cluster of the new directory
1911
                                 <1>
                                                     cf = 1 -> Error code in AL
                                          ;
1912
                                 <1>
1913
                                 <1>
                                          ; Modified Registers: EAX (at the return of system call)
1914
                                 <1>
1915
                                 <1>
                                          ; Note: If the file or directory is existing
1916
                                 <1>
                                                an access error will be returned.
                                          ;
1917
                                 <1>
1918 0000CE8A 6621C9
                                          and cx, cx ; if cx = 0 -> create a normal subdir
jz short sysmkdir_1
                                <1>
1919 0000CE8D 7413
                                <1>
1920
                                <1>
                                          test cl, 10h; if dir flags set, also use other flags
1921 0000CE8F F6C110
                                <1>
                                                sysmkdir_0 ; jump to head of 'syscreat'
1922 0000CE92 0F853EFDFFFF
                                <1>
                                          jnz
                                <1>
1924
                                          ; CX has wrong flags
                                <1>
1925 0000CE98 B817000000
                                 <1>
                                          mov eax, ERR_INV_FLAGS
1926 0000CE9D E969FFFFFF
                                          jmp sysopen_dev_err
                                 <1>
1927
                                 <1>
1928
                                 <1> sysmkdir 1:
                                                cl, 10h; set subdir flag and reset other flags
1929 0000CEA2 B110
                                 <1>
                                          mov
                                                 sysmkdir_0 ; jump to head of 'syscreat'
1930 0000CEA4 E92DFDFFFF
                                 <1>
                                          jmp
1931
                                 <1> sysmkdir 2:
                                          ; jump from 'syscreat' ; from 'syscreat_1'
1932
                                 <1>
                                 <1>
                                          ; CL = Directory attributes/flags
                                          mov
1934 0000CEA9 BE[68620100]
                                 <1>
                                                esi, FindFile_Name
                                                make sub directory
1935 0000CEAE E804D7FFFF
                                 <1>
                                          call
1936 0000CEB3 0F821BFEFFFF
                                                                  ; NOTE: Old type (TRDOS 8086)
                                 <1>
                                          jс
                                                 sysopen_err
1937
                                 <1>
                                                               ; error codes must be modified
                                                               ; for next TRDOS 386 versions
1938
                                 <1>
                                                               ; (10/10/2016)
1939
                                 <1>
1940
                                                               ; Old (MSDOS type)
                                 <1>
                                                               ; error codes (2011):
1941
                                 <1>
                                                                ; 2 = file not found
1942
                                 <1>
                                                               ; 3 = directory not found
1943
                                 <1>
1944
                                 <1>
                                                                ; 5 = access denied
                                                               ; 12 = no more files
1945
                                 <1>
                                                                ; 19 = disk write protected
1946
                                 <1>
1947
                                 <1>
                                                                ; 39 = insufficient disk space
1948
                                 <1>
                                                                ; 'sysdefs.s' ; 10/10/2016
1949
                                 <1>
1950 0000CEB9 A3[64030300]
                                 <1>
                                                 [u.r0], eax; New sub dir's first cluster
                                 <1>
1951
```

```
1952 0000CEBE E8912E0000
                                 <1>
                                             call
                                                        reset_working_path
                                 <1>
1954 0000CEC3 E99AF8FFFF
                                 <1>
                                           jmp sysret
1955
                                 <1>
1956
                                 <1> sysclose: ;<close file>
                                           ; 06/10/2016 (TRDOS 386 = TRDOS v2.0)
1957
                                 <1>
1958
                                         ; 14/05/2015 (Retro UNIX 386 v1 - Beginning)
1959
                                 <1>
1960
                                 <1>
                                          ; 22/05/2013 - 26/05/2013 (Retro UNIX 8086 v1)
1961
                                 <1>
                                          ; 'sysclose', given a file descriptor in 'u.r0', closes the
1962
                                 <1>
                                           ; associated file. The file descriptor (index to 'u.fp' list)
1963
                                 <1>
                                           ; is put in r1 and 'fclose' is called.
1964
                                 <1>
1965
                                 <1>
                                         ; Calling sequence:
1966
                                 <1>
1967
                                 <1>
                                                sysclose
1968
                                 <1>
                                           ; Arguments:
1969
                                 <1>
1970
                                 <1>
                                          ; Inputs: *u.r0 - file descriptor
1971
                                 <1>
                                           ; Outputs: -
1972
                                 <1>
                                          i ......
1973
                                 <1>
1974
                                          ; Retro UNIX 8086 v1 modification:
                                 <1>
1975
                                 <1>
                                                  The user/application program puts file descriptor
1976
                                 <1>
                                                   in BX register as 'sysclose' system call argument.
1977
                                 <1>
                                                   (argument transfer method 1)
1978
                                 <1>
                                           ; TRDOS 386 (06/10/2016)
1979
                                 <1>
1980
                                 <1>
                                           ; INPUT ->
1981
                                 <1>
1982
                                 <1>
                                                    EBX = File Handle/Number (file index) (AL)
1983
                                 <1>
                                           ; OUTPUT ->
                                                     cf = 0 \rightarrow EAX = 0
1984
                                 <1>
1985
                                 <1>
                                                     cf = 1 -> Error code in EAX (ERR_FILE_NOT_OPEN)
1986
                                 <1>
1987
                                 <1>
                                           ; Modified Registers: EAX (at the return of system call)
1988
                                 <1>
1989
                                 <1>
1990 0000CEC8 89D8
                                 <1>
                                                 eax, ebx
1991 0000CECA 31DB
                                          xor ebx, ebx
                                 <1>
1992 0000CECC 891D[64030300]
                                 <1>
                                          mov [u.r0], ebx; 0; return value of EAX
1993 0000CED2 E8450E0000
                                 <1>
                                          call fclose
1994 0000CED7 0F8385F8FFFF
                                 <1>
                                           jnc
                                                 sysret
1995 0000CEDD B80A000000
                                 <1>
                                                 eax, ERR_FILE_NOT_OPEN ; file not open !
1996 0000CEE2 A3[C8030300]
                                 <1>
                                          mov
                                                 [u.error], eax ;
1997 0000CEE7 A3[64030300]
                                                 [u.r0], eax ; ! invalid handle !
                                 <1>
                                           mov
1998 0000CEEC E951F8FFFF
                                           jmp error
                                 <1>
1999
                                 <1>
2000
                                 <1> sysread: ; < read from file >
2001
                                 <1>
                                       ; 11/10/2016 (TRDOS 386 = TRDOS v2.0)
                                                 -derived from INT_21H.ASM-
2002
                                 <1>
2003
                                 <1>
                                                       ("loc INT21h read file")
                                                 13/03/2011 (05/03/2011)
2004
                                 <1>
2005
                                 <1>
                                                INT 21h Function AH = 3Fh
2006
                                 <1>
                                           ;
                                                 Read from a File
2007
                                 <1>
                                                 INPUT
2008
                                 <1>
                                                  BX = File Handle
                                           ;
2009
                                 <1>
                                                  CX = Number of bytes to read
2010
                                 <1>
                                                       DS:DX= Buffer address
2011
                                 <1>
2012
                                 <1>
                                           ; Note: TRDOS 386 'sysread' has been derived from
                                                 Retro UNIX 386 v1 'sysread', except a few
2013
                                 <1>
                                          ;
2014
                                 <1>
                                           ;
                                                 code modifications.
2015
                                 <1>
                                          ; 13/05/2015 (Retro UNIX 386 v1)
2016
                                 <1>
                                           ; 11/05/2015 (Retro UNIX 386 v1 - Beginning)
2017
                                 <1>
                                          ; 23/05/2013 (Retro UNIX 8086 v1)
2018
                                 <1>
2019
                                 <1>
2020
                                 <1>
                                          ; 'sysread' is given a buffer to read into and the number of
2021
                                 <1>
                                          ; characters to be read. If finds the file from the file
2022
                                 <1>
                                           ; descriptor located in *u.r0 (r0). This file descriptor
2023
                                 <1>
                                           ; is returned from a successful open call (sysopen).
2024
                                 <1>
                                           ; The i-number of file is obtained via 'rw1' and the data
                                          ; is read into core via 'readi'.
2025
                                 <1>
2026
                                 <1>
2027
                                 <1>
                                          ; Calling sequence:
2028
                                 <1>
                                          ; sysread; buffer; nchars
2029
                                 <1>
                                          ; Arguments:
2030
                                 <1>
                                           ; buffer - location of contiguous bytes where
2031
                                                  input will be placed.
                                 <1>
2032
                                 <1>
                                                 nchars - number of bytes or characters to be read.
                                           ; Inputs: *u.r0 - file descriptor (& arguments)
2033
                                  <1>
                                           ; Outputs: *u.r0 - number of bytes read.
2034
                                 <1>
2035
                                 <1>
2036
                                 <1>
2037
                                 <1>
                                           ; Retro UNIX 8086 v1 modification:
2038
                                 <1>
                                                   'sysread' system call has three arguments; so,
                                                  * 1st argument, file descriptor is in BX register
2039
                                 <1>
2040
                                 <1>
                                                  * 2nd argument, buffer address/offset in CX register
                                                  ^{\star} 3rd argument, number of bytes is in DX register
2041
                                 <1>
2042
                                 <1>
2043
                                 <1>
                                                  AX register (will be restored via 'u.r0') will return
2044
                                 <1>
                                                  to the user with number of bytes read.
2045
                                 <1>
2046
                                 <1>
                                           ; TRDOS 386 (05/10/2016)
2047
                                 <1>
                                             ; INPUT ->
2048
                                 <1>
2049
                                                    EBX = File handle (descriptor/index)
                                 <1>
2050
                                 <1>
                                                    ECX = Buffer address
2051
                                 <1>
                                                      EDX = Number of bytes
                                           ; OUTPUT ->
2052
                                 <1>
2053
                                 <1>
                                                      EAX = Number of bytes have been read
2054
                                 <1>
                                                      cf = 1 -> Error code in AL
2055
                                 <1>
2056
                                 <1>
                                           ; Modified Registers: EAX (at the return of system call)
```

```
2057
                                 <1>
2058
                                 <1>
                                          ; EBX = File descriptor
2059
                                 <1>
2060 0000CEF1 E8740E0000
                                 <1>
                                          call getf1
                                          jc short device_read ; read data from device
2061 0000CEF6 7277
                                 <1>
2062
                                 <1>
                                          ; EAX = First cluster of the file
2063
                                 <1>
2064 0000CEF8 E83F000000
                                 <1>
                                          call rw1
2065 0000CEFD 730A
                                 <1>
                                                 short sysread_0
2066
                                 <1>
2067 0000CEFF A3[64030300]
                                 <1>
                                           mov
                                                 [u.r0], eax ; error code
2068 0000CF04 E939F8FFFF
                                 <1>
                                           jmp
                                                 error
2069
                                 <1>
2070
                                 <1> sysread_0:
2071 0000CF09 E825140000
                                 <1>
                                           call readi
2072 0000CF0E EB1D
                                 <1>
                                           jmp
                                                 short rw0
                                 <1>
                                 <1> syswrite: ; < write to file >
2074
                                          ; 23/10/2016
2075
                                 <1>
                                           ; 11/10/2016 (TRDOS 386 = TRDOS v2.0)
2076
                                 <1>
2077
                                 <1>
                                                      -derived from INT_21H.ASM-
2078
                                 <1>
                                                        ("loc INT21h write file")
2079
                                               13/03/2011 (05/03/2011)
                                 <1>
2080
                                 <1>
                                            ; INT 21h Function AH = 40h
2081
                                 <1>
                                           ;
                                                 Write to a File
2082
                                 <1>
                                                 INPUT
2083
                                 <1>
                                           ;
                                                  BX = File Handle
2084
                                           ;
                                                    CX = Number of bytes to write
                                 <1>
2085
                                 <1>
                                                       DS:DX= Buffer address
                                            ;
2086
                                 <1>
                                           ; Note: TRDOS 386 'sysrwrite' has been derived from
2087
                                 <1>
2088
                                                 Retro UNIX 386 v1 'syswrite', except a few
                                 <1>
                                           ;
2089
                                 <1>
                                                 code modifications.
                                           ;
2090
                                 <1>
2091
                                 <1>
                                           ; 13/05/2015 (Retro UNIX 386 v1)
2092
                                 <1>
                                           ; 11/05/2015 (Retro UNIX 386 v1 - Beginning)
2093
2094
                                 <1>
                                           ; 23/05/2013 (Retro UNIX 8086 v1)
2095
                                 <1>
2096
                                 <1>
                                          ; 'syswrite' is given a buffer to write onto an output file
2097
                                 <1>
                                           ; and the number of characters to write. If finds the file
2098
                                 <1>
                                           ; from the file descriptor located in *u.r0 (r0). This file
2099
                                 <1>
                                           ; descriptor is returned from a successful open or create call
                                           ; (sysopen or syscreat). The i-number of file is obtained via
2100
                                 <1>
2101
                                 <1>
                                           ; 'rwl' and buffer is written on the output file via 'write'.
2102
                                 <1>
2103
                                 <1>
                                          ; Calling sequence:
2104
                                 <1>
                                         ; syswrite; buffer; nchars
2105
                                 <1>
                                           ; Arguments:
2106
                                 <1>
                                           ; buffer - location of contiguous bytes to be writtten.
2107
                                 <1>
                                                nchars - number of characters to be written.
2108
                                 <1>
                                          ; Inputs: *u.r0 - file descriptor (& arguments)
                                           ; Outputs: *u.r0 - number of bytes written.
2109
                                 <1>
2110
                                 <1>
                                           ; ......
2111
                                 <1>
2112
                                 <1>
                                           ; Retro UNIX 8086 v1 modification:
                                                  'syswrite' system call has three arguments; so,
2113
                                 <1>
                                                 ^{\star} 1st argument, file descriptor is in BX register
                                 <1>
2114
2115
                                 <1>
                                                 * 2nd argument, buffer address/offset in CX register
2116
                                 <1>
                                                 \star 3rd argument, number of bytes is in DX register
2117
                                 <1>
2118
                                 <1>
                                                 AX register (will be restored via 'u.r0') will return
2119
                                 <1>
                                                 to the user with number of bytes written.
2120
                                 <1>
2121
                                 <1>
                                           ; INPUT ->
2122
                                 <1>
                                                    EBX = File handle (descriptor/index)
                                                    ECX = Buffer address
2123
                                 <1>
2124
                                 <1>
                                                       EDX = Number of bytes
2125
                                 <1>
                                           ; OUTPUT ->
2126
                                 <1>
                                                     EAX = Number of bytes have been written
                                           ;
2127
                                 <1>
                                                     cf = 1 -> Error code in AL
2128
                                 <1>
                                           ; Modified Registers: EAX (at the return of system call)
2129
                                 <1>
2130
                                 <1>
2131
                                 <1>
2132
                                 <1>
                                           ; EBX = File descriptor
2133 0000CF10 E8550E0000
                                 <1>
                                           call getf1
                                                short device_write ; write data to device
2134 0000CF15 7274
                                 <1>
                                           jс
2135
                                 <1>
                                           ; EAX = First cluster of the file
2136
                                           ; EBX = File number (Open file number); 23/10/2016
                                 <1>
2137
                                 <1>
2138 0000CF17 E820000000
                                 <1>
                                           call rw1
2139 0000CF1C 730A
                                 <1>
                                           jnc
                                                 short syswrite_0
2140 0000CF1E A3[64030300]
                                 <1>
                                                 [u.r0], eax; error code
2141 0000CF23 E91AF8FFFF
                                 <1>
                                           jmp
                                                 error
2142
                                 <1>
                                 <1> syswrite 0:
2144 0000CF28 E8321B0000
                                 <1>
                                          call writei
                                 <1> rw0: ; 1:
2145
2146 0000CF2D A1[8C030300]
                                 <1>
                                           mov eax, [u.nread]
2147 0000CF32 A3[64030300]
                                 <1>
                                           mov [u.r0], eax
                                 <1>
                                                sysret
2148 0000CF37 E926F8FFFF
                                           jmp
2149
                                 <1>
2150
                                 <1> rw1:
2151
                                 <1>
                                          ; 11/10/2016 (TRDOS 386 = TRDOS v2.0)
2152
                                 <1>
                                           ; 14/05/2015 (Retro UNIX 386 v1)
                                           ; 11/05/2015 (Retro UNIX 386 v1 - Beginning)
2153
                                 <1>
                                           ; 23/05/2013 - 24/05/2013 (Retro UNIX 8086 v1)
2154
                                 <1>
2155
                                 <1>
                                           ; System call registers: ebx, ecx, edx (through 'sysenter')
2156
                                 <1>
2157
                                 <1>
                                           ; EBX = File descriptor
2158
                                 <1>
                                           ;call getf1 ; calling point in 'getf' from 'rw1'
                                           ;jc short device rw ; read/write data from/to device
2159
                                 <1>
2160
                                 <1>
                                           ; EAX = First cluster of the file
2161
                                 <1>
```

```
2162 0000CF3C 83F802
                                  <1>
                                            cmp
                                                   eax, 2
2163 0000CF3F 7217
                                  <1>
                                            jb
                                                   short rw2
2164
                                  <1>
                                            ;
2165 0000CF41 890D[84030300]
                                  <1>
                                                                     ; buffer address/offset
2166
                                  <1>
                                                              ; (in the user's virtual memory space)
2167 0000CF47 8915[88030300]
                                  <1>
                                                   [u.count], edx
                                            mov
2168
                                  <1>
2169 0000CF4D C705[C8030300]0000- <1>
                                                       dword [u.error], 0 ; reset the last error code
                                             mov
2169 0000CF55 0000
                                  <1>
2170 0000CF57 C3
                                  <1>
                                            retn
2171
                                  <1>
2172
                                  <1> rw2:
2173 0000CF58 B80A000000
                                                   eax, ERR FILE NOT OPEN; file not open!
                                  <1>
                                            mov
2174 0000CF5D A3[C8030300]
                                 <1>
                                                   dword [u.error], eax
2175 0000CF62 C3
                                  <1>
                                            retn
2176
                                  <1> rw3:
2177 0000CF63 B80B000000
                                                   eax, ERR FILE ACCESS; permission denied!
                                 <1>
                                            mov
2178 0000CF68 A3[C8030300]
                                  <1>
                                                   dword [u.error], eax
                                            mov
2179 0000CF6D F9
                                  <1>
                                            stc
2180 0000CF6E C3
                                  <1>
                                            retn
2181
                                  <1>
2182
                                  <1> device read:
                                            \frac{1}{7} 11/10/2016 (TRDOS 386 = TRDOS v2.0)
2183
                                  <1>
2184
                                  <1>
                                            ; cl = DEV_OPENMODE ; open mode
                                            ; ch = DEV_ACCESS ; access flags ; al = DEV_DRIVER ; device number (eax)
2185
                                  <1>
2186
                                  <1>
                                  <1>
2188 0000CF6F F6C101
                                            test cl, 1; 1 = read, 2 = write, 3 = read&write
                                  <1>
2189 0000CF72 74EF
                                  <1>
                                                   short rw3
                                            jz
2190
                                  <1>
2191 0000CF74 89C3
                                  <1>
                                            mov
                                                   ebx, eax
2192 0000CF76 66C1E302
                                  <1>
                                            shl
                                                  bx, 2 ; *4
2193
                                  <1>
2194 0000CF7A F6C580
                                  <1>
                                            test ch, 80h; bit 7, installable device driver flag
2195 0000CF7D 7406
                                  <1>
                                            jz short d_read_2 ; Kernel device
                                            ; installable device
2196
                                  <1>
2197
                                  <1> d read 1:
2198 0000CF7F FFA3[34660100]
                                  <1>
                                             jmp dword [ebx+IDEV_RADDR-4]
2199
                                  <1> d read 2:
2200 0000CF85 FFA3[DE150100]
                                  <1>
                                                  dword [ebx+KDEV_RADDR-4]
                                            jmp
2201
                                  <1>
2202
                                  <1> device write:
                                           \frac{1}{7} 11/10/2016 (TRDOS 386 = TRDOS v2.0)
2203
                                  <1>
2204
                                  <1>
                                            ; cl = DEV_OPENMODE ; open mode
                                            ; ch = DEV_ACCESS ; access flags
; al = DEV_DRIVER ; device number (eax)
2205
                                  <1>
2206
                                  <1>
                                  <1>
2208 0000CF8B F6C102
                                            test cl, 2; 1 = read, 2 = write, 3 = read&write
                                  <1>
2209 0000CF8E 74D3
                                  <1>
                                            jz
                                                   short rw3
2210
                                  <1>
                                            mov
2211 0000CF90 89C3
                                  <1>
                                                  ebx, eax
2212 0000CF92 66C1E302
                                  <1>
                                            shl
                                                  bx, 2 ; *4
2213
                                  <1>
2214 0000CF96 F6C580
                                  <1>
                                            test ch, 80h; bit 7, installable device driver flag
2215 0000CF99 7406
                                  <1>
                                            jz
                                                  short d_write_2 ; Kernel device
2216
                                  <1>
                                            ; installable device
2217
                                  <1> d_write_1:
2218 0000CF9B FFA3[54660100]
                                           jmp dword [ebx+IDEV_WADDR-4]
                                  <1>
2219
                                  <1> d write 2:
2220 0000CFA1 FFA3[2E160100]
                                           jmp dword [ebx+KDEV WADDR-4]
                                  <1>
2221
                                  <1>
2222
2223
                                  <1> sysemt: ; enable (or disable) multi tasking -time sharing-
2224
                                  <1>
2225
                                  <1>
                                            ; 23/05/2016 - TRDOS 386 (TRDOS v2.0)
2226
                                  <1>
                                            ; 14/05/2015 (Retro UNIX 386 v1)
                                            ; 10/12/2013 - 20/04/2014 (Retro UNIX 8086 v1)
2227
                                  <1>
2228
                                  <1>
2229
                                  <1>
                                            ; Retro UNIX 8086 v1 modification:
2230
                                  <1>
                                            ; 'Enable Multi Tasking' system call instead
2231
                                  <1>
                                                   of 'Emulator Trap' in original UNIX v1 for PDP-11.
2232
                                  <1>
2233
                                  <1>
                                            ; Retro UNIX 8086 v1 feature only!
2234
                                  <1>
                                                Using purpose: Kernel will start without time-out
2235
                                  <1>
                                                   (internal clock/timer) functionality.
2236
                                  <1>
                                                   Then etc/init will enable clock/timer for
2237
                                  <1>
                                                  multi tasking.
                                            ;
2238
                                  <1>
                                            ; INPUT ->
2239
                                  <1>
2240
                                            ; BL = 0 \rightarrow disable multi tasking
                                  <1>
2241
                                  <1>
                                                   BL > 1 -> enable multi tasking (time sharing)
2242
                                   <1>
                                             ; OUTPUT ->
2243
                                  <1>
                                                   none
2244
                                  <1>
2245
                                  <1>
                                               Note: Multi tasking is disabled during system
                                                    initialization, it must be enabled by using
2246
                                  <1>
2247
                                  <1>
                                                    this system call. (Otherwise, running proces
2248
                                  <1>
                                                    will not be changed by another process within
2249
                                  <1>
                                                    run time sequence/schedule, if running process
                                                    will not 'release' itself. Only 'wakeup' procedure
2250
                                  <1>
                                                    for waiting processes and programmed timer events
2251
                                  <1>
2252
                                   <1>
                                                    for other processes can change running process
                                                    while multi tasking is disabled.) ** 23/05/2016 **
2253
                                  <1>
2254
                                  <1>
2255 0000CFA7 803D[B0030300]00
                                  <1>
                                                   byte [u.uid], 0 ; root ?
                                            cmp
2256
                                  <1>
                                            ;ja
                                                   error
                                                   badsys ; 14/05/2015
2257 0000CFAE 0F87D3F8FFFF
                                   <1>
                                            jа
                                  <1>
2258
2259 0000CFB4 FA
                                  <1>
                                            cli
2260 0000CFB5 881D[52650100]
                                  <1>
                                                   [multi_tasking], bl ; 0 to disable, >0 to enable
                                            mov
2261 0000CFBB E9A2F7FFFF
                                  <1>
                                            jmp
2262
                                  <1>
                                  <1> systimer:
2263
2264
                                  <1>
                                            ; 02/01/2017
2265
                                  <1>
                                            ; 21/12/2016
```

```
; 19/12/2016
2266
2267
                                                                  <1>
                                                                                   ; 10/12/2016 (callback)
                                                                                   ; 10/06/2016
2268
                                                                 <1>
2269
                                                                  <1>
                                                                                 ; 07/06/2016
                                                                                 ; 06/06/2016
2270
                                                                 <1>
2271
                                                                 <1>
                                                                                   ; 21/05/2016
2272
                                                                                   ; 19/05/2016
                                                                                 ; 18/05/2016 - TRDOS 386 (TRDOS v2.0)
2273
                                                                 <1>
2274
                                                                  <1>
                                                                                   ; (TRDOS 386 feature only!)
2275
                                                                 <1>
2276
                                                                 <1>
                                                                                   ; (start or stop timer event(s))
2277
                                                                  <1>
                                                                                    ; INPUT ->
2278
                                                                 <1>
2279
                                                                 <1>
                                                                                                BL = Signal return byte (response byte)
2280
                                                                 <1>
                                                                                                          (Any requested value between 0 and 255)
2281
                                                                  <1>
                                                                                                          (Kernel will put it at the requested address)
                                                                                                BH = Time count unit
2282
                                                                  <1>
                                                                                                0 = Stop timer event
1 = 18.2 ticks per second
2283
                                                                 <1>
2284
                                                                  <1>
                                                                                                       2 = 10 milliseconds
2285
                                                                 <1>
2286
                                                                 <1>
                                                                                                        3 = 1 second (for real time clock interrupt)
2287
                                                                  <1>
                                                                                                          4 = time/tick count in current time count unit
                                                                                                       // 10/12/2016
2288
                                                                 <1>
2289
                                                                  <1>
                                                                                                     80h = Stop timer event (callback method)
2290
                                                                 <1>
                                                                                                        81h = 18.2 ticks per second, callback method
2291
                                                                  <1>
                                                                                                        82h = 10 milliseconds, callback method
2292
                                                                  <1>
                                                                                                        83h = 1 second (for RTC int), callback method
2293
                                                                 <1>
                                                                                                        84h = current time count unit, callback method
2294
                                                                  <1>
2295
                                                                                                         Note: Only 03h or 83h will set real time clock
                                                                 <1>
2296
                                                                  <1>
                                                                                                                  (RTC) events (Others are for PIT events)!
2297
                                                                  <1>
                                                                                                 NOTE: If callback (user service) method is used,
2298
                                                                 <1>
2299
                                                                 <1>
                                                                                                        EDX will point to the return address (of service
2300
                                                                  <1>
                                                                                                         procedure) in user's space instead of signal
2301
                                                                 <1>
                                                                                                         response byte address. (TRDOS 386 kernel will
2302
                                                                  <1>
                                                                                                        direct the cpu to that address -in user's space-
2303
                                                                 <1>
                                                                                                        at the return of system call or interrupt
                                                                                                         just after the adjusted count/time is elapsed.)
2304
                                                                  <1>
2305
                                                                 <1>
                                                                                                     User's sevice routine must be ended with a
2306
                                                                 <1>
                                                                                                        'iret'. Normal return addresses from system
2307
                                                                  <1>
                                                                                                        calls or and interrupts will be kept same except
2308
                                                                 <1>
                                                                                                        the timer returns.
2309
                                                                 <1>
                                                                                                 BH = 0 \rightarrow Stop timer event
2310
                                                                 <1>
2311
                                                                 <1>
                                                                                                 BL = Timer event number (1 to 255) if <math>BH = 0
                                                                                                   If BL = 0, all timer events (which are belongs
2312
                                                                 <1>
                                                                                                           to running process) will be stopped
2313
                                                                 <1>
2314
                                                                  <1>
                                                                                                 ECX = Time/Tick count (depending on time count unit)
2315
                                                                  <1>
                                                                                                EDX = Signal return (Response) byte address
                                                                                    ;
                                                                                                            (virtual address in user's memory space)
2316
                                                                 <1>
                                                                                    ; OUTPUT ->
2317
                                                                  <1>
                                                                                           AL = Timer event number (1 to 255) (max. value = 16)
2318
                                                                 <1>
2319
                                                                 <1>
                                                                                                IF BH Input = 0 & CF = 0 & AL = 0 ->
2320
                                                                  <1>
                                                                                                         timer event(s) has/have been stopped/finished
2321
                                                                  <1>
                                                                                                CF = 1 & AL = 0 -> no timer setting space to set
                                                                                                CF = 1 & AL > 0 -> timer count unit is not usable
2322
                                                                  <1>
2323
                                                                 <1>
2324
                                                                  <1>
                                                                                                 NOTE: To modify a time count for a user function,
2325
                                                                 <1>
                                                                                                             at first, current timer event must be stopped
2326
                                                                 <1>
                                                                                                             then a new timer event (which is related with
2327
                                                                  <1>
                                                                                                             same user function) must be started.
2328
                                                                 <1>
2329
                                                                  <1>
                                                                                                             Signal return (response) byte may be used for
2330
                                                                  <1>
                                                                                                             several purposes. Kernel will put this value
2331
                                                                  <1>
                                                                                                             to requested address during timer interrupt,
2332
                                                                  <1>
                                                                                                             program/user can check this value to understand
2333
                                                                 <1>
                                                                                                             which event has been occurred and what is changed.
2334
                                                                  <1>
                                                                                                             (Multi timer events can share same signal address)
2335
                                                                 <1>
2336
                                                                  <1>
                                                                                                 NOTE: If the process is running while the time count
2337
                                                                  <1>
                                                                                                             is reached, kernel will put signal return (response)
2338
                                                                 <1>
                                                                                                             byte value at requested address during timer
2339
                                                                  <1>
                                                                                                             interrupt and the process will continue to run.
2340
                                                                  <1>
                                                                                                             Program/process must call (jump to) it's timer event
2341
                                                                  <1>
                                                                                                             function as required, for checking the timer event
2342
                                                                  <1>
                                                                                                             status via signal return (response) byte address.
2343
                                                                  <1>
2344
                                                                  <1>
                                                                                                             If the process is not running (waiting or sleeping
2345
                                                                                                             or released) while the time count is reached,
                                                                  <1>
2346
                                                                  <1>
                                                                                                             it is restarted from where it left, to ensure
2347
                                                                  <1>
                                                                                                             proper multi media (video, audio, clock, timer)
2348
                                                                  <1>
                                                                                                             functionality.
2349
                                                                  <1>
2350
                                                                  <1>
                                                                                                             (It is better to use 'syswait' or 'syssleep',
                                                                                                             or 'sysrele' system call just after the timer % \left( 1\right) =\left( 1\right) +\left( 1
2351
                                                                  <1>
2352
                                                                  <1>
                                                                                                             function. Otherwise, timer events may block other
2353
                                                                 <1>
                                                                                                             processes which are not using timer events.)
2354
2355
                                                                                    ; Timer Event Structure: (max. 16 timer events, 16*16 bytes)
                                                                 <1>
2356
                                                                                                                                        resb 1 ; 0 = free
                                                                  <1>
                                                                                                 Owner:
                                                                                                                                        ;>0 = process number (u.uno)
2357
                                                                 <1>
2358
                                                                 <1>
                                                                                                Calback: resb 1 ; 1 = callback, 0 = response byte
                                                                                                Interrupt: resb 1 ; 0 = Timer interrupt (or none)
2359
                                                                  <1>
2360
                                                                 <1>
                                                                                                                                   ; 1 = Real Time Clock interrupt
                                                                                                                             resb 1; 0 to 255, signal return value
2361
                                                                  <1>
                                                                                                Response:
2362
                                                                 <1>
                                                                                                 Count Limit: resd 1 ; count of ticks (total/set)
2363
                                                                 <1>
                                                                                                  Current Count: resd 1 ; count of ticks (current)
2364
                                                                 <1>
                                                                                                 Response Addr: resd 1 ; response byte (pointer) address
2365
                                                                 <1>
2366
                                                                  <1>
2367
                                                                 <1>
                                                                                    ; 19/12/2016 (timer callback)
                                                                                    mov byte [tcallback], 0
2368 0000CFC0 C605[906A0100]00
                                                                 <1>
2369 0000CFC7 C605[916A0100]00
                                                                 <1>
                                                                                               byte [trtc], 0
                                                                                    mov
```

```
dword [u.tcb], 0 ; this is not necessary...
2370 0000CFD6 0000
2371
                                <1>
2372 0000CFD8 80FF80
                                <1>
                                          cmp
                                                bh, 80h
2373 0000CFDB 7225
                                <1>
                                          jb
                                                 short systimer cb2
2374 0000CFDD 7704
                                <1>
                                          jа
                                                 short systimer cb0
                                <1>
2376 0000CFDF 31D2
                                <1>
                                                 edx, edx; 0, reset callback address
                                          xor
2377 0000CFE1 EB0B
                                <1>
                                                short systimer cb1
                                          jmp
2378
                                <1>
2379
                                <1> systimer_cb0:
2380 0000CFE3 80FF84
                                <1>
                                          cmp
                                                bh, 84h
2381 0000CFE6 7764
                                <1>
                                                 short systimer_5 ; undefined, error
                                          jа
2382
                                <1>
2383
                                <1>
                                          ;mov byte [tcallback], 1; 19/12/2016
2384 0000CFE8 FE05[906A0100]
                                                byte [tcallback]
                                <1>
                                          inc
                                <1>
2386
                                <1> systimer_cb1:
2387 0000CFEE 0FB635[B3030300]
                                <1>
                                          movzx esi, byte [u.uno] ; process number
2388 0000CFF5 66C1E602
                                <1>
                                          shl si, 2
2389 0000CFF9 8996[0C010300]
                                <1>
                                                 [esi+p.tcb-4], edx; set process timer callback address
                                 <1>
                                                               ; (overwrite prev value if it is set!)
2391 0000CFFF 80E77F
                                <1>
                                                bh, 7Fh
                                          and
2392
                                <1>
                                <1> systimer cb2:
2393
2394 0000D002 80FF02
                                <1>
                                          cmp bh, 2
2395 0000D005 7445
                                <1>
                                                    short systimer 5 ; only 18.2 ticks per second is usable
                                          jе
2396
                                <1>
                                                             ; 10 milliseconds (100 Hertz) timer
2397
                                <1>
                                                               ; will be set later (18/05/2016)
2398 0000D007 774B
                                <1>
                                          jа
                                                    short systimer 6
2399
                                <1>
2400 0000D009 20FF
                                <1>
                                          and
                                                bh, bh
2401 0000D00B 0F84BA000000
                                <1>
                                                    systimer_9
                                                                    ; stop timer event(s)
                                           jz
2402
                                <1>
2403
                                 <1>
                                          ; bh = 1 (timer interrupt, 18.2 Hz, IBM PC/AT ROMBIOS default)
2404
                                <1>
                                <1> systimer 19:
2406 0000D011 B00A
                                <1>
                                          mov al, 10; (*)
2407
                                <1>
2408
                                <1> systimer 0:
2409 0000D013 B710
                                                bh, 16
                                <1>
                                          mov
2410
                                <1>
                                          ;
                                                 [timer_events], bh ; 16 ; 07/06/2016
2411 0000D015 383D[53650100]
                                <1>
                                          cmp
2412 0000D01B 7319
                                <1>
                                                 short systimer_3 ; max. 16 timer events
2413
                                 <1>
2414 0000D01D 50
                                          push eax; (*)
                                <1>
2415
                                <1>
2416 0000D01E BF[60040300]
                                                 edi, timer_set ; beginning address of timer events
                                <1>
                                          mov
2417
                                <1>
                                                             ; setting space
2418 0000D023 30C0
                                                al, al ; 0
                                <1>
                                          xor
2419
                                <1> systimer_1:
2420 0000D025 FEC0
                                <1>
                                          inc
2421 0000D027 803F00
                                                byte [edi], 0
                                                                 ; is it free space ?
                                <1>
                                          cmp
2422 0000D02A 7639
                                <1>
                                                short systimer_7 ; yes
2423 0000D02C FECF
                                <1>
                                          dec
                                                bh
2424 0000D02E 7405
                                <1>
                                          jг
                                                short systimer_2
2425 0000D030 83C710
                                <1>
                                          add
                                                edi, 16
2426 0000D033 EBF0
                                <1>
                                                short systimer_1; next event space
                                          jmp
2427
                                <1>
                                <1> systimer_2:
2428
2429 0000D035 58
                                <1>
                                                 eax ; (*) discard
                                          pop
2430
                                 <1> systimer 3:
2431 0000D036 C605[64030300]00
                                <1> mov byte [u.r0], 0
2432
                                 <1> systimer 4:
2433 0000D03D C705[C8030300]1B00- <1>
                                                    dword [u.error], ERR MISC
                                       mov
2433 0000D045 0000
                                <1>
                                 <1>
                                                                    ; one of miscellaneous/other errors
2435 0000D047 E9F6F6FFFF
                                <1>
                                                 error ; cf -> 1
                                          jmp
2436
                                <1>
2437
                                <1> systimer_5:
2438 0000D04C 883D[64030300]
                                <1>
                                          mov
                                                 [u.r0], bh; Time count unit (=2 or >3)
                                                short systimer_4 ; 07/06/2016
2439 0000D052 EBE9
                                <1>
                                          jmp
2440
                                <1>
2441
                                <1> systimer_6:
2442 0000D054 80FF04
                                <1>
                                          cmp
                                                bh, 4
                                                   short systimer 5 ; undefined time count unit
2443 0000D057 77F3
                                <1>
                                          jа
                                 <1>
                                          ;jb
                                                short systimer 16
2445
                                 <1>
2446
                                 <1>
                                          ;mov al, 1 ; default (use current timer unit)
2447
                                 <1>
                                                      ; countdown value is in ECX !
                                                       ; max. value of ecx = 4294967296/10
2448
                                 <1>
2449
                                 <1>
                                            ;jmp
                                                    short systimer 0
2450
                                <1>
                                          ;jmp short systimer_19
2451 0000D059 74B6
                                <1>
                                                short systimer_19
2452
                                <1>
                                <1> systimer_16:
2453
                                <1>
                                        ; bh = 3
2455
                                <1>
                                          ; timer event via real time clock interrupt
                                          ; interrupt/update frequency: 1 Hz (1 tick per second)
2456
                                <1>
2457
                                <1>
2458 0000D05B B0B6
                                          mov al, 182; (*); 18.2 * 10
                                <1>
2459 0000D05D FE05[916A0100]
                                <1>
                                          inc byte [trtc]; timer event via real time clock
2460 0000D063 EBAE
                                <1>
                                          jmp short systimer 0
                                <1>
                                <1> systimer 7:
2462
                                          mov [u.r0], al; timer event number
2463 0000D065 A2[64030300]
                                <1>
                                <1>
2465
                                <1>
                                          ; edi = address of empty timer event area
2466 0000D06A A0[B3030300]
                                <1>
                                          mov al, [u.uno]
                                          cli ; disable interrupts
2467 0000D06F FA
                                <1>
2468 0000D070 AA
                                <1>
                                          stosb ; process number
                               <1>
2469 0000D071 A0[906A0100]
2470 0000D076 AA
                                         stosb ; 1= callback method, 0= signal response byte method
                               <1> mov al, [trtc]; timer interrupt type
<1> stosb; 1= real time clock, 0= programmable interval timer
2471 0000D077 A0[916A0100]
2472 0000D07C AA
```

2370 0000CFCE C705[D0030300]0000- <1>

mov

```
2473 0000D07D 88D8
                                <1>
                                          mov al, bl ; Signal return (Response) value
2474 0000D07F AA
                                <1>
                                          stosb ; response byte
2475 0000D080 58
                                          pop eax; (*); 10 or 182
                                <1>
2476 0000D081 89D3
                                <1>
                                          mov ebx, edx; virtual address for response/signal byte
2477 0000D083 F7E1
                                <1>
                                          mul ecx
2478
                                <1>
                                          ; (eax = 10 * count of 18.2 Hz timer ticks)
2479
                                <1>
                                          ; (count down step = 10)
2480 0000D085 AB
                                <1>
                                          stosd ; count limit (reset value)
2481 0000D086 AB
                                <1>
                                          stosd ; current count value
2482
                                <1>
                                          ; 19/12/2016
2483
                                 <1>
2484 0000D087 803D[906A0100]00
                                <1>
                                          cmp byte [tcallback], 0 ; timer callback method ?
2485 0000D08E 7604
                                 <1>
                                                short systimer_17 ; no
                                          jna
2486 0000D090 89D8
                                 <1>
                                          mov eax, ebx; virtual address for callback routine
2487 0000D092 EB0D
                                 <1>
                                                short systimer 18
                                          jmp
2488
                                 <1>
                                 <1> systimer 17: ; signal response byte method
2489
2490
                                 <1>
                                          ; ebx = virtual address
2491
                                 <1>
                                          ; [u.pgdir] = page directory's physical address
2492
                                 <1>
                                          ; 20/02/2017
2493 0000D094 FE05[926A0100]
                                 <1>
                                                 byte [no_page_swap] ; 1
2494
                                 <1>
                                                       ; Do not add this page to swap queue
2495
                                 <1>
                                                       ; and remove it from swap queue if it is
2496
                                 <1>
                                                       ; on the queue.
2497 0000D09A E84D82FFFF
                                 <1>
                                          call get_physical_addr
2498 0000D09F 721A
                                 <1>
                                          jc short systimer_8; 07/06/2016
                                 <1>
                                          ; eax = physical address of the virtual address in user's space
2500
                                <1> systimer_18:
2501 0000D0A1 AB
                                 <1>
                                          stosd ; response addr (physical) or callback addr (virtual)
2502 0000D0A2 FE05[53650100]
                                <1>
                                          inc byte [timer_events]; 07/06/201
2503
                                 <1>
                                          ; 02/01/2017
2504 0000D0A8 0FB605[B3030300]
                                 <1>
                                          movzx eax, byte [u.uno]
2505 0000D0AF FE80[FF000300]
                                <1>
                                          inc byte [eax+p.timer-1]
2506
                                 <1>
2507 0000D0B5 FB
                                 <1>
                                          sti
                                                ; enable interrupts
2508 0000D0B6 E9A7F6FFFF
                                <1>
                                                sysret
                                          jmp
                                 <1>
2510
                                <1> systimer_8:
2511
                                <1>
                                       ; 10/06/2016
2512
                                          ; 07/06/2016
                                <1>
2513 0000D0BB 28C0
                                <1>
                                        sub
2514 0000D0BD 8847F4
                                <1>
                                          mov
                                                 [edi-12], al ; clear process number (free timer event)
2515
                                <1>
                                          ; mov
                                                dword [edi], eax ; 0
2516 0000D0C0 FB
                                <1>
                                          sti
2517 0000D0C1 A2[64030300]
                                <1>
                                          mov
                                                [u.r0], al ; 0
2518 0000D0C6 E977F6FFFF
                                <1>
                                          jmp
                                                error
                                <1>
2520
                                 <1> systimer_9:
                                          ; 10/06/2016
2521
                                <1>
                                          ; 07/06/2016
2522
                                <1>
2523 0000D0CB 28C0
                                <1>
                                          sub
                                                al, al
2524 0000D0CD A2[64030300]
                                 <1>
                                          mov
                                                byte [u.r0], al ; 0
2525 0000D0D2 3805[53650100]
                                <1>
                                          cmp
                                                 byte [timer_events], al ; 0
2526 0000D0D8 7631
                                <1>
                                                short systimer_12
2527
                                 <1>
2528
                                 <1>
                                          ; Note: ecx and edx are undefined here
2529
                                 <1>
                                                 (for stop timer function)
2530
                                 <1>
2531 0000D0DA BE[60040300]
                                 <1>
                                                 esi, timer_set ; beginning address of timer events
2532
                                 <1>
                                                             ; setting space
2533 0000D0DF A0[B3030300]
                                 <1>
                                                 al, [u.uno]
                                          mov
2534
                                 <1>
2535 0000D0E4 B710
                                 <1>
                                          mov
                                                bh, 16
2536
                                 <1>
2537 0000D0E6 08DB
                                 <1>
                                                bl, bl
                                          or
2538 0000D0E8 7544
                                <1>
                                                short systimer_15
2539
                                <1>
2540
                                <1>
                                          ; clear timer event areas belong to current process
2541
                                <1>
                                          ; (for stopping all timer events belong to current process)
2542 0000D0EA FA
                                <1>
                                          cli ; disable interrupts
                                <1> systimer 10:
2543
2544
                                 <1>
                                          ; 10/06/2016
2545
                                          ; 07/06/2016
                                <1>
2546 0000D0EB 8A26
                                <1>
                                          mov ah, [esi]
                                <1>
                                                ah, ah; 0?
2547 0000D0ED 08E4
                                          or
                                                short systimer 11
2548 0000D0EF 7411
                                <1>
                                          jΖ
2549 0000D0F1 38C4
                                <1>
                                          cmp ah, al; is the process number (owner) same?
2550 0000D0F3 750D
                                <1>
                                          jne short systimer_11 ; no
2551
                                 <1>
2552
                                          ;mov byte [esi], 0
                                <1>
2553 0000D0F5 66C7060000
                                 <1>
                                          mov word [esi], 0; clear
                                 <1>
                                          ;mov
                                                 dword [esi+12], 0; clear
                                <1>
2556 0000D0FA FE0D[53650100]
                                <1>
                                                byte [timer_events]
2557 0000D100 7409
                                <1>
                                          jz
                                                short systimer_12
2558
                                <1>
                                <1> systimer 11:
2559
                                <1>
2560 0000D102 FECF
                                          dec bh
2561 0000D104 7405
                                <1>
                                                 short systimer_12
                                          jΖ
2562 0000D106 83C610
                                <1>
                                          add
                                                esi, 16
2563 0000D109 EBE0
                                <1>
                                                short systimer_10
2564
                                <1>
2565
                                <1> systimer_12:
2566 0000D10B 0FB635[B3030300] <1>
                                       movzx esi, byte [u.uno]
2567 0000D112 08DB
                                <1>
                                          or
                                                bl, bl; all timer events or one timer event ?
2568 0000D114 740C
                                <1>
                                          jΖ
                                                short systimer_13
2569 0000D116 8A9E[FF000300]
                                <1>
                                          mov
                                                bl, [esi+p.timer-1]
                                          and bl, bl; previous number of timer events for the process
2570 0000D11C 20DB
                                <1>
2571 0000D11E 7408
                                <1>
                                          jz
                                                 short systimer_14
                                          dec
2572 0000D120 FECB
                                <1>
                                                bl ; previous number of timer events for the process - 1
                                <1> systimer 13:
2573
2574 0000D122 889E[FF000300]
                                <1>
                                          mov [esi+p.timer-1], bl ; 0 ; no timer events for process
                                <1> systimer_14:
2575
2576 0000D128 FB
                                <1>
                                          sti ; enable interrupts
2577 0000D129 E934F6FFFF
                                <1>
                                          jmp sysret
```

```
2579
                                  <1> systimer 15:
                                           cmp bl, bh ; 16
2580 0000D12E 38FB
                                 <1>
2581 0000D130 0F8707FFFFF
                                 <1>
                                                      systimer 4
                                                                   ; max. 16 timer events !
                                            jа
2582
                                 <1>
                                           ;
2583 0000D136 88DA
                                 <1>
                                           mov
                                                  dl, bl
                                         dec dl ; 16 -> 15 ... 1 -> 0
2584 0000D138 FECA
                                 <1>
2585 0000D13A C0E204
                                 <1>
                                           shl dl, 4; * 16
2586 0000D13D 0FB6FA
                                 <1>
                                           movzx edi, dl
2587 0000D140 01F7
                                 <1>
                                           add edi, esi; timer_set
2588
                                 <1>
2589 0000D142 3A07
                                           cmp al, [edi] ; process number
                                  <1>
2590 0000D144 0F85F3FEFFFF
                                 <1>
                                            jne systimer_4
2591
                                 <1>
2592
                                  <1>
                                           ; same process ID
2593 0000D14A FA
                                  <1>
                                           cli ; disable interrupts
                                           ; 10/06/2016 ; 02/01/2017
                                  <1>
2595
                                  <1>
                                           ;mov byte [edi], 0
2596 0000D14B 66C7070000
                                  <1>
                                                 word [edi], 0 ; clear
                                           mov
                                           ;mov dword [edi+12], 0; clear
2597
                                  <1>
2598 0000D150 FE0D[53650100]
                                 <1>
                                            dec byte [timer_events]
2599 0000D156 EBB3
                                  <1>
                                           jmp
                                                 short systimer 12
2600
                                  <1>
2601
                                  <1> sysvideo: ; VIDEO DATA TRANSFER FUNCTIONS
                                        ; 12/05/2017
2602
                                  <1>
                                           ; 11/07/2016
2603
                                  <1>
2604
                                  <1>
                                           ; 13/06/2016
2605
                                  <1>
                                          ; 16/05/2016 - TRDOS 386 (TRDOS v2.0)
2606
                                  <1>
2607
                                  <1>
2608
                                  <1>
                                           ; VIDEO DATA TRANSFER FUNCTIONS:
2609
                                  <1>
2610
                                  <1>
                                           ; Inputs:
2611
                                  <1>
                                                  BH = 0 = VIDEO BIOS Mode 3, tty/text mode data transfers
2612
                                  <1>
                                                       BL =
2613
                                  <1>
                                                        Bits 0&1, Transfer direction
2614
                                  <1>
                                                               0 - System to system
2615
                                  <1>
                                                               1 - User to system
                                                               2 - System to user
2616
                                  <1>
                                                               3 - User to user
2617
                                  <1>
2618
                                  <1>
                                                         Bits 2&3, Transfer Type
2619
                                  <1>
                                                               0 - Display page transfer
                                                               1 - Display page window transfer
2620
                                  <1>
2621
                                  <1>
                                                               2 - Frame/Viewport/Window address transfer
2622
                                  <1>
                                                                3 - Window handle transfer
2623
                                  <1>
                                                       /// BL = 0 -> System to system (display page) transfer
2624
                                  <1>
                                                          CL = Source page
2625
                                  <1>
2626
                                  <1>
                                                          DL = Destination page
                                                       /// BL = 1&2 -> user to system & system to user transfer
2627
                                  <1>
                                                          ECX = User buffer
2628
                                  <1>
2629
                                  <1>
                                                          DL = Video page
                                                       /// BL = 5\&6 -> user to system, system to user transfer
2630
                                  <1>
2631
                                  <1>
                                                       (window in current display page and in current mode)
2632
                                  <1>
                                                          ESI = User's buffer address
2633
                                  <1>
                                                          ECX Low 16 bits = Top left column (X1 position)
                                                          ECX High 16 bits = Top row (Y1 position)
2634
                                  <1>
                                                          EDX Low 16 bits = Bottom right column (X2 position)
2635
                                  <1>
2636
                                  <1>
                                                          EDX High 16 bits = Bottom row (Y2 position)
2637
                                  <1>
                                                              If BL = 5 \rightarrow
2638
                                  <1>
                                                          EDI = Swap address (in user's memory space)
2639
                                  <1>
                                                          (If swap address > 0, previous content of the window
2640
                                  <1>
                                                          will be saved into swap area in user's memory space)
2641
                                  <1>
                                                       /// BL = 4 -> system to system transfer
2642
                                  <1>
                                                          ESI = System's source buffer (video page) address
2643
                                  <1>
                                                          ECX Low 16 bits = Top left column (X1 position)
                                                          ECX High 16 bits = Top row (Y1 position)
2644
                                  <1>
                                                          EDX Low 16 bits = Bottom right column (X2 position)
2645
                                  <1>
2646
                                  <1>
                                                          EDX High 16 bits = Bottom row (Y2 position)
                                                          EDI = System's destination buffer (video page) address
2647
                                  <1>
2648
                                  <1>
2649
                                  <1>
                                                  BH = 1 = CGA Graphics (0B8000h) data transfers
2650
                                  <1>
                                                     BL =
2651
                                  <1>
                                                       0 = Fill color (color in CL] (32K)
2652
                                  <1>
                                                        1 = User to system display page transfer
2653
                                  <1>
                                                        2 = System to user display page transfer
2654
                                  <1>
                                                        3 = NOT bits in window (ECX, EDX)
                                                        4 = Window copy (system to system)
2655
                                  <1>
2656
                                  <1>
                                                        5 = User to system window transfer
2657
                                                         6 = System to user window transfer
                                  <1>
2658
                                  <1>
                                                         7 = AND display page bytes with CL
2659
                                  <1>
                                                         8 = OR display page bytes with CL
                                                         9 = XOR display page bytes with CL
2660
                                  <1>
2661
                                  <1>
2662
                                  <1>
                                                       /// BL = 0 -> Fill color (all screen pixels)
                                                          CL = Color value
2663
                                  <1>
                                                       /// BL = 1\&2 -> user to system & system to user transfer
2664
                                  <1>
2665
                                  <1>
                                                          ECX = User buffer
                                                       /// BL = 5&6 -> user to system, system to user transfer
2666
                                  <1>
                                                         (window in current display page and in current mode)
                                  <1>
2667
                                                          ESI = User's buffer address
2668
                                  <1>
2669
                                  <1>
                                                          ECX Low 16 bits = Top left column (X1 position)
                                                          ECX High 16 bits = Top row (Y1 position)
2670
                                  <1>
2671
                                  <1>
                                                          EDX Low 16 bits = Bottom right column (X2 position)
                                                          EDX High 16 bits = Bottom row (Y2 position)
2672
                                  <1>
2673
                                  <1>
                                                       /// BL = 4 -> system to system (window) transfer
2674
                                  <1>
                                                          ESI = System's source buffer (video page) address
2675
                                                          ECX Low 16 bits = Top left column (X1 position)
                                  <1>
2676
                                  <1>
                                                          ECX High 16 bits = Top row (Y1 position)
                                                          EDX Low 16 bits = Bottom right column (X2 position)
2677
                                  <1>
                                                          EDX High 16 bits = Bottom row (Y2 position)
2678
                                  <1>
2679
                                  <1>
                                                          EDI = System's destination buffer (video page) address
                                                       /// BL = 3 \rightarrow NOT byte in display page/memory
2680
                                  <1>
2681
                                  <1>
                                                          ECX Low 16 bits = Top left column (X1 position)
2682
                                  <1>
                                                          ECX High 16 bits = Top row (Y1 position)
```

2578

EDX Low 16 bits = Bottom right column (X2 position)

```
2684
                                   <1>
                                                            EDX High 16 bits = Bottom row (Y2 position)
                                             ;
2685
                                   <1>
2686
                                   <1>
                                                   BH = 2 = VGA Graphics (0A0000h) data transfers
                                                      BL =
2687
                                   <1>
2688
                                   <1>
                                                          x0h = Fill color (color in CL] (64K)
2689
                                   <1>
                                                          x1h = User to system display page transfer
                                                         x2h = System to user display page transfer
2690
                                   <1>
2691
                                   <1>
                                                          x3h = NOT bits in window (ECX, EDX)
                                                         x4h = Window copy (system to system)
2692
                                   <1>
                                             ;
                                                        x5h = User to system window transfer
2693
                                   <1>
                                                        x6h = System to user window transfer
x7h = AND display page bytes with CL
2694
                                   <1>
2695
                                   <1>
2696
                                   <1>
                                                         x8h = OR display page bytes with CL
                                                         x9h = XOR display page bytes with CL
2697
                                   <1>
2698
                                   <1>
                                                          x = 0 \rightarrow screen width = 320
                                                         x = 1 \rightarrow screen width = 640
2699
                                   <1>
                                                         x = 2 \rightarrow screen width = 800
2700
                                   <1>
2701
                                   <1>
2702
                                   <1>
                                                        /// BL = 0 -> Fill color (all screen pixels)
                                                           CL = Color value
2703
                                   <1>
2704
                                   <1>
                                                         /// BL = 1\&2 -> user to system & system to user transfer
2705
                                                           ECX = User buffer
                                   <1>
2706
                                   <1>
                                                         /// BL = 5\&6 -> user to system, system to user transfer
                                                          (window in current display page and in current mode)
2707
                                   <1>
2708
                                   <1>
                                                           ESI = User's buffer address
2709
                                   <1>
                                                            ECX Low 16 bits = Top left column (X1 position)
                                                           ECX High 16 bits = Top row (Y1 position)
2710
                                   <1>
2711
                                   <1>
                                                            EDX Low 16 bits = Bottom right column (X2 position)
2712
                                                           EDX High 16 bits = Bottom row (Y2 position)
                                   <1>
2713
                                   <1>
                                                                /// BL = 4 -> system to system (window) transfer
2714
                                   <1>
                                                            ESI = System's source buffer (video page) address
                                             ;
2715
                                   <1>
                                                            ECX Low 16 bits = Top left column (X1 position)
2716
                                   <1>
                                                            ECX High 16 bits = Top row (Y1 position)
2717
                                   <1>
                                                            EDX Low 16 bits = Bottom right column (X2 position)
                                                            EDX High 16 bits = Bottom row (Y2 position)
2718
                                   <1>
2719
                                   <1>
                                                            EDI = System's destination buffer (video page) address
                                                         /// BL = 3 -> NOT byte in display page/memory
2720
                                   <1>
2721
                                   <1>
                                                            ECX Low 16 bits = Top left column (X1 position)
                                                            ECX High 16 bits = Top row (Y1 position)
2722
                                   <1>
2723
                                   <1>
                                                            EDX Low 16 bits = Bottom right column (X2 position)
2724
                                   <1>
                                                            EDX High 16 bits = Bottom row (Y2 position)
2725
                                   <1>
2726
                                   <1>
                                                   BH = 3 = Super VGA, LINEAR FRAME BUFFER data transfers
                                                      BL =
2727
                                   <1>
2728
                                   <1>
                                                          0 = Fill color (color in ECX] (Frame buffer size)
                                                         1 = User to system display page transfer
2729
                                   <1>
                                                         2 = System to user display page transfer
3 = NOT bits in window (ECX, EDX)
2730
                                   <1>
2731
                                   <1>
2732
                                   <1>
                                                         4 = Window copy (system to system)
                                             ;
                                                         5 = User to system window transfer
2733
                                   <1>
2734
                                   <1>
                                                          6 = System to user window transfer
                                                          7 = AND display page bytes with ECX
2735
                                   <1>
2736
                                   <1>
                                                         8 = OR display page bytes with ECX
2737
                                   <1>
                                                          9 = XOR display page bytes with ECX
2738
                                   <1>
2739
                                   <1>
                                                       /// BL = 0 -> Fill color (all screen pixels)
                                                           CL = Color value
2740
                                   <1>
2741
                                   <1>
                                                        /// BL = 1&2 -> user to system & system to user transfer
2742
                                                           ECX = User buffer
                                   <1>
2743
                                   <1>
                                                         /// BL = 5\&6 -> user to system, system to user transfer
2744
                                   <1>
                                                           (window in current display page and in current mode)
2745
                                   <1>
                                                           ESI = User's buffer address
2746
                                   <1>
                                                           ECX Low 16 bits = Top left column (X1 position)
2747
                                   <1>
                                                           ECX High 16 bits = Top row (Y1 position)
2748
                                   <1>
                                                            EDX Low 16 bits = Bottom right column (X2 position)
                                                            EDX High 16 bits = Bottom row (Y2 position)
2749
                                   <1>
2750
                                   <1>
                                                         /// BL = 4 -> system to system (window) transfer
2751
                                   <1>
                                                            ESI = System's source buffer (video page) address
                                                            ECX Low 16 bits = Top left column (X1 position)
2752
                                   <1>
2753
                                   <1>
                                                            ECX High 16 bits = Top row (Y1 position)
2754
                                   <1>
                                                            EDX Low 16 bits = Bottom right column (X2 position)
                                                           EDX High 16 bits = Bottom row (Y2 position)
2755
                                   <1>
2756
                                   <1>
                                                            EDI = System's destination buffer (video page) address
2757
                                   <1>
                                                         /// BL = 3 -> NOT byte in display page/memory
2758
                                   <1>
                                                            ECX Low 16 bits = Top left column (X1 position)
2759
                                   <1>
                                                            ECX High 16 bits = Top row (Y1 position)
2760
                                   <1>
                                                            EDX Low 16 bits = Bottom right column (X2 position)
2761
                                   <1>
                                                            EDX High 16 bits = Bottom row (Y2 position)
2762
                                   <1>
2763
                                   <1>
                                             ; Outputs:
                                                    EAX = transfer/byte count
2764
                                   <1>
2765
                                   <1>
2766
                                   <1>
                                                    NOTE: If the source or destination address passes out of
                                                    video pages (display memory limits), data will not be transferred
2767
                                   <1>
2768
                                   <1>
                                                    and EAX will return as 0.
2769
                                   <1>
2770
                                   <1>
2771
                                   <1>
                                             ; DIRECT (STANDARD VGA/CGA) DISPLAY MEMORY ACCESS FUNCTIONS:
2772
                                   <1>
                                                    BH = 4 = CGA direct video memory (OB8000h, 32K) access
2773
                                   <1>
2774
                                   <1>
                                                          Page directory & page tables of the user's
2775
                                   <1>
                                                           program will be updated to direct access to
2776
                                   <1>
                                                           OB8000h (32K) video (CGA, color) memory; if
2777
                                   <1>
                                                          there is not a permission conflict or lock!
2778
                                   <1>
                                                            (User's program/process will have permission to
2779
                                   <1>
                                                           access locked display memory if the owner is
                                             ;
2780
                                   <1>
                                                          it's parent.)
2781
                                   <1>
                                                        Screen width = 320
2782
                                   <1>
2783
                                   <1>
2784
                                   <1>
                                                    BH = 5 = VGA direct video memory (0A0000h, 64K) access
2785
                                   <1>
                                                          Page directory & page tables of the user's
                                                           program will be updated to direct access to
2786
                                   <1>
2787
                                   <1>
                                                           OA0000h (64K) video (VGA) memory; if there is not
```

2683

```
2788
                                   <1>
                                                          a permission conflict or lock!
2789
                                   <1>
                                                            (User's program/process will have permision to
2790
                                   <1>
                                                           access locked display memory if the owner is
2791
                                   <1>
                                                          it's parent.)
2792
                                   <1>
2793
                                   <1>
                                                        BL = Screen width (320, 640, 800)
2794
                                   <1>
2795
                                   <1>
                                             ; Outputs:
2796
                                   <1>
                                                   EAX = Display mmory address for direct access
2797
                                                          0A0000h for VGA, 0B8000h for CGA
                                   <1>
2798
                                   <1>
                                                    (Display memory size: 32K for CGA, 64K for VGA)
2799
                                   <1>
                                                    EAX = 0 if display page access permission has been denied.
2800
                                   <1>
                                                          (Locked!)
2801
                                   <1>
2802
                                   <1>
                                            ; LINEAR FRAME BUFFER ACCESS FUNCTIONS:
2803
                                   <1>
2804
                                   <1>
                                                    BH = 6 = Linear Frame Buffer direct video memory access
2805
                                   <1>
                                                          Page directory & page tables of the user's
2806
                                   <1>
2807
                                   <1>
                                                          program will be updated to direct access to
2808
                                   <1>
                                                          the configured LFB (Linear Frame Buffer) address,
2809
                                   <1>
                                                          if there is not a permission conflict or lock!
2810
                                   <1>
                                                           (User's program/process will have permission to
2811
                                   <1>
                                                          access locked display memory if the owner is
2812
                                   <1>
                                                          it's parent.)
2813
                                   <1>
2814
                                   <1>
                                                          Return: EAX = Linear Frame Buffer address
2815
                                                                 EDX = Frame Buffer Size in bytes
                                   <1>
2816
                                   <1>
2817
                                   <1>
                                                   BH = 7 = Get Linear Frame Buffer info (for current mode)
2818
                                   <1>
2819
                                   <1>
                                                          Return:
                                                          EAX = Frame Buffer Address (0 = is not in use)
2820
                                   <1>
                                                         EDX = Frame Buffer Size in bytes
2821
                                   <1>
2822
                                   <1>
                                                          BL = Current Video Mode
2823
                                   <1>
                                                               BL = 0FFh -> Super VGA (Extended VGA)
2824
                                   <1>
                                                               If BL = OFFh,
2825
                                   <1>
                                                                      BH = 0 = 16 \text{ colors}
2826
                                   <1>
                                                                 BH = 1 = 256 \text{ colors}
2827
                                   <1>
                                                                 BH = 2 = 66536 \text{ colors}
2828
                                   <1>
                                                                 BH = 3 = 24 bits TRUE (16M) colors
2829
                                   <1>
                                                                 BH = 4 = 32 bits TRUE (16M) colors
                                                          ECX = Pixel resolution
2830
                                   <1>
2831
                                   <1>
                                                                 CX = Width (640, 800, 1024, 1366, 1920)
2832
                                   <1>
                                                                 High 16 bits of ECX = Height
2833
                                   <1>
2834
                                   <1>
                                                   NOTE: Each process will have it's own frame buffer
2835
                                   <1>
                                                          address and resolution parameters in 'u' area.
2836
                                   <1>
                                                          Then, if the current frame buffer & resolution
2837
                                   <1>
                                                          is different, frame buffer r/w functions
                                             ;
2838
                                   <1>
                                                          will use scale factor to convert process's
2839
                                   <1>
                                                            pixel coordinates to actual screen coordinates.
                                                         resolution -> dimensional scale
2840
                                   <1>
2841
                                   <1>
                                                          color size -> color scale
2842
                                   <1>
                                                         * RGB (TRUE) colors to 256 colors conversion:
2843
                                   <1>
                                                             TRUE Colors \rightarrow 8,8,8 (R,G,B; byte 0 is R)
2844
                                   <1>
                                                          256 colors -> 2,2,2,2 (R,G,B,L; bit 0&1 is R)
                                                            bit 6\&7 \rightarrow luminosity base level (0,1,2,3)
2845
                                   <1>
2846
                                   <1>
                                                            bit 4\&5 -> blue level (0,1,2,3)
2847
                                                            bit 2%3 -> green level (0,1,2,3)
                                   <1>
2848
                                   <1>
                                                            bit 0\&1 -> \text{ red level } (0,1,2,3)
2849
                                   <1>
                                                          Example: total red level : luminosity + red level
                                                          Luminosity base level: 0 \rightarrow 16
2850
                                   <1>
2851
                                   <1>
                                                                             1 -> 32
2852
                                   <1>
                                                                             2 -> 64
2853
                                   <1>
                                                                             3 -> 128
2854
                                   <1>
                                                          Color level:
                                                                            0 -> 0
2855
                                   <1>
2856
                                   <1>
                                                                            1 -> luminosity level
                                                                            2 -> luminosity level + 64
2857
                                   <1>
                                                                            3 -> 255
2858
                                   <1>
2859
                                   <1>
                                                         Luminosity base level = min (R,G,B)
                                                               if it is <16, it will be set to 16
2860
                                   <1>
2861
                                   <1>
                                                         Color levels: Color values are fixed to (nearest)
2862
                                   <1>
                                                              one of all possible set level (step) values
2863
                                   <1>
                                                              (according to luminosity base level); then
2864
                                   <1>
                                                             color levels are set to R-L, G-L, B-L.
2865
                                                    For example: If luminosity base level is 32
                                   <1>
2866
                                   <1>
                                                             all possible set values are 0, 32, 96, 255.
2867
                                   <1>
                                                        * RGB (TRUE) colors to 16 colors conversion:
2868
                                   <1>
2869
                                   <1>
                                                        16 colors: R, B,G, L bits (4 bits)
                                                              If any one of R,G,B >= 128 L = 1
2870
                                   <1>
2871
                                   <1>
                                                              If max. value of (R,G,B) >= 32, it is 1
2872
                                   <1>
                                                                 else all color bits (R&G&B&L) are 0
                                                              If the second value >= max. value / 2
2873
                                   <1>
2874
                                   <1>
                                                                it is 1
                                                              If third value value >= max. value / 2
2875
                                   <1>
2876
                                   <1>
                                                                 it is 1
2877
                                   <1>
                                                        Example: R = 132, G = 64, B = 78
2878
                                   <1>
                                                               L = 1, R = 1
                                                                G < 66 --> G = 0
2879
                                   <1>
                                                                B >= 66 --> B = 1
                                   <1>
2880
2881
                                   <1>
2882
                                   <1>
                                             ; 16/05/2016
2883 0000D158 31C0
                                   <1>
                                             xor eax, eax
2884 0000D15A A3[64030300]
                                                   [u.r0], eax
                                   <1>
2885
                                   <1>
2886 0000D15F 20FF
                                  <1>
                                                   bh, bh
                                             and
                                                   sysvideo_13 ; 11/07/2016
2887 0000D161 0F8572020000
                                  <1>
2888
                                   <1>
                                             ; Video mode 0, 80*25 text mode, CGA 16 colors ; [CRT MODE] = 3
2889
                                  <1>
2890 0000D167 88DF
                                   <1>
2891 0000D169 C0EF02
                                   <1>
                                                   bh, 2
                                             shr
```

```
and bh, bh
jnz syst
   2892 0000D16C 20FF
                                                    <1>
   2893 0000D16E 0F8598000000
                                                    <1>
                                                                   jnz sysvideo 4
   2894 0000D174 BF00800B00
                                                                    mov edi, 0B8000h
                                                    <1>
   2895 0000D179 20D2
                                                    <1>
                                                                    and dl, dl
### Comparison of Comparison o
                                                    <1>
   2896 0000D17B 7413
                                                                    jz
                                                                             short sysvideo 1
  2896 0000D17B 7413
2897 0000D17D 80FA07
2898 0000D180 0F87DCF5FFFF
                                                   2925 0000D1D2 89CF
  2926 0000D1D4 B9A00F0000
2927 0000D1D9 E81F160000
2928 0000D1DE 0F827EF5FFFF
   2929 0000D1E4 890D[64030300] <1>
   2930 0000D1EA E973F5FFFF
                                                    <1>
                                                                    jmp
                                                                             sysret
  2931
                                                    <1> sysvideo_3:
   2932
                                                    <1> ; user to system video/display page transfer (mode 0)
   2933 0000D1EF 89CE
                                                    <1>
                                                                   mov esi, ecx; user buffer
  2934
2935 0000D1F1 B9A00F0000
2000D1F6 E84C160000
                                                   ; edi = video page address
                                                                  call transfer_from_user_buffer ; fast transfer
   2937 0000D1FB 0F8261F5FFFF
                                                    <1>
                                                                    jс
                                                                              sysret
   2938 0000D201 890D[64030300]
                                                    <1>
                                                                   mov
                                                                            [u.r0], ecx
   2939 0000D207 E956F5FFFF
                                                    <1>
                                                                    jmp sysret
  2940
                                                     <1> sysvideo_4:
   2941 0000D20C 80E303
                                                    <1> and bl, 3
   2942 0000D20F 0F85F6000000
                                                    <1>
                                                                   jnz sysvideo_9
                                                                   cmp cl, 7
   2943 0000D215 80F907
                                                    <1>
   2944 0000D218 0F8744F5FFFF
                                                    <1>
                                                                             sysret
   2945
                                                    <1>
                                                                   ; system to system video/display page window transfer (mode 0)
   2946 0000D21E 81FE00800B00
                                                    <1>
                                                                   cmp esi, 0B8000h
   2947 0000D224 0F8238F5FFFF
                                                    <1>
                                                                    jb
                                                                             sysret
                                                                            esi, 0B8000h+(80*25*2*8)
   2948 0000D22A 81FE00FD0B00
                                                    <1>
                                                                   cmp
   2949 0000D230 0F832CF5FFFF
                                                    <1>
                                                                   jnb
                                                                            sysret
   2950 0000D236 81FF00800B00
                                                     <1>
                                                                             edi, 0B8000h
                                                                    cmp
   2951 0000D23C 0F8220F5FFFF
                                                    <1>
                                                                   jb
                                                                             sysret
                                                                               edi, 0B8000h+(80*25*2*8)
   2952 0000D242 81FF00FD0B00
                                                    <1>
                                                                    cmp
   2953 0000D248 0F8314F5FFFF
                                                     <1>
                                                                    jnb
                                                                             sysret
  2954
                                                     <1>
                                                                   push ecx
   2955 0000D24E 51
                                                    <1>
                                                                    push edx
   2956 0000D24F 52
                                                    <1>
   2957 0000D250 0FB7C1
                                                    <1>
                                                                    movzx eax, cx; top left column
   2958 0000D253 50
                                                    <1>
                                                                   push eax
                                                   <1>
   2959 0000D254 C1E910
                                                                             ecx, 16; top row
                                                                   shr
   2960 0000D257 66B8A000
                                                    <1>
                                                                             ax, 80*2; 80 colums, 160 bytes per row
                                                                    mov
                                                    <1>
  2961 0000D25B F7E1
                                                                   mul
                                                                             ecx
                                                <1>
   2962 0000D25D 01C6
                                                                    add
                                                                             esi, eax
   2963 0000D25F 01C7
                                                    <1>
                                                                    add
                                                                             edi, eax
   2964 0000D261 58
                                                    <1>
                                                                    pop
                                                                             eax
   2965 0000D262 66D1E0
                                                   <1>
                                                                   shl
                                                                             ax, 1 ; *2
   2966 0000D265 01C6
                                                    <1>
                                                                    add
                                                                             esi, eax
  2967 0000D267 01C7
                                                    <1>
                                                                    add
                                                                             edi, eax
   2968 0000D269 5A
                                                    <1>
                                                                             edx
                                                                   pop
   2969 0000D26A 59
                                                    <1>
                                                                    pop
                                                                             ecx
                                                                             eax, 0B8000h+(80*25*2*8)
   2970 0000D26B B800FD0B00
                                                     <1>
                                                                    mov
   2971 0000D270 39C6
                                                    <1>
                                                                    cmp
                                                                             esi, eax
   2972 0000D272 0F83EAF4FFF
                                                    <1>
                                                                    jnb
                                                                             sysret
   2973 0000D278 39C6
                                                     <1>
                                                                    cmp
                                                                              esi, eax
   2974 0000D27A 0F83E2F4FFFF
                                                    <1>
                                                                    jnb
                                                                              sysret
   2975
                                                     <1>
                                                                    push esi; ****
   2976 0000D280 56
                                                     <1>
                                                                            edi ; ***
   2977 0000D281 57
                                                    <1>
                                                                    push
   2978 0000D282 52
                                                                   push edx; **
                                                    <1>
                                                   <1>
                                                                   push ecx; *
   2979 0000D283 51
   2980 0000D284 C1E910
                                                    <1>
                                                                              ecx, 16; top row
                                                                    shr
   2981 0000D287 C1EA10
                                                   <1>
                                                                    shr
                                                                             edx, 16; bottom row
   2982 0000D28A 83F918
                                                    <1>
                                                                    cmp
                                                                             ecx, 24 ; max. 25 rows
   2983 0000D28D 7773
                                                    <1>
                                                                    jа
                                                                              short sysvideo_6
   2984 0000D28F 83FA18
                                                                              edx, 24 ; max. 25 rows
                                                    <1>
                                                                    cmp
   2985 0000D292 776E
                                                    <1>
                                                                              short sysvideo_6
                                                                   jа
   2986 0000D294 28CA
                                                    <1>
                                                                   sub
                                                                              dl, cl
  2987 0000D296 726A
                                                    <1>
                                                                    jс
                                                                              short sysvideo_6
                                                                            eax ; *****
   2988 0000D298 50
                                                                   push
                                                    <1>
   2989 0000D299 89D3
                                                    <1>
                                                                              ebx, edx; row count - 1
                                                                    mov
   2990 0000D29B B8A000000
                                                    <1>
                                                                    mov
                                                                              eax, 80*2
   2991 0000D2A0 F7E0
                                                    <1>
                                                                   mul
                                                                              eax
   2992 0000D2A2 01C6
                                                   <1>
                                                                   add
                                                                              esi, eax
                                                    <1>
   2993 0000D2A4 01C7
                                                                   add
                                                                              edi, eax
                                                   <1>
                                                                             eax ; ****
  2994 0000D2A6 58
                                                                    pop
                                                                              esi, eax
   2995 0000D2A7 39C6
                                                    <1>
                                                                    cmp
   2996 0000D2A9 7757
                                                     <1>
                                                                              short sysvideo_6
                                                                    jа
```

```
<1>
<1>
<1>
  2997 0000D2AB 39C7
                                               edi, eax
                                          cmp
2998 0000D2AD 7753
                                          jа
                                                short sysvideo 6
  2999 0000D2AF 59
                                                ecx ; *
                                          pop
                                               edx ; **
                                               ecx, 79; max. 80 columns
                                                short sysvideo_7
                                                edx, 79; max. 80 columns
                                                short sysvideo_7
                                               short sysvideo 7
                                          ; edx = column count (width) - 1
                                                short sysvideo_7
                                                short sysvideo_7
                                                dl ; column count
                                               al, dl ; (80 - columns) * 2
                                                esi, eax ; next row
                                                edi, eax ; next row
                                                short sysvideo_5
  3036 0000D302 59
                                <1> pop ecx; *
                                          pop edx; **
  3037 0000D303 5A
                                <1>
                                <1> sysvideo 7:
  3038
                                <1> pop edi; ***
  3039 0000D304 5F
                                          pop esi; ****
  3040 0000D305 5E
                                <1>
                                          jmp
  3041 0000D306 E957F4FFFF
                                <1>
                                               sysret
                                <1>
  3042
  3043
                                <1> sysvideo_9:
                                <1> cmp bl, 2
  3044 0000D30B 80FB02
  3045 0000D30E 0F874EF4FFFF
                                <1>
                                                sysret
                                <1>
  3047 0000D314 56
                                <1>
                                         push esi; ****
  3048 0000D315 57
                                <1>
                                         push edi; ***
                                          push edx; **
  3049 0000D316 52
                                <1>
                             <1>
                                          push ecx; *
  3050 0000D317 51
  3051
                                <1>
                          3052 0000D318 C1E910
                                          shr
                                                ecx, 16; top row
  3053 0000D31B C1EA10
                                          shr edx, 16; bottom row
  3054 0000D31E 83F918
                                               ecx, 24 ; max. 25 rows
                                          cmp
  3055 0000D321 77DF
                                          jа
                                                short sysvideo_6
  3056 0000D323 83FA18
                                                edx, 24 ; max. 25 rows
                                          cmp
                                                short sysvideo_6
  3057 0000D326 77DA
                                          jа
  3058 0000D328 28CA
                                                dl, cl
                                <1>
                                          sub
  3059 0000D32A 72D6
                                <1>
                                                short sysvideo_6
                                          jс
  3060
                                <1>
  3061 0000D32C 88CD
                                <1>
                                                ch, cl ; top row
                                          mov
 3062 0000D32E 8A0D[EE580100] <1>
3063 0000D334 BFA00F0000 <1>
3064 0000D339 D3E7 <1>
                                                cl, [ACTIVE_PAGE]
                                          mov
                                                edi, 80*25*\overline{2}
                                          mov
                                                edi, cl
edi, 0B8000h - 80*25*2
  3064 0000D339 D3E7
                                <1>
                                          shl
 3065 0000D33B 81C760700B00
                                <1>
                                          add
  3066
                                <1>
  3067 0000D341 88D7
                                <1>
                                          mov
                                                bh, dl ; row count - 1
  3068 0000D343 88EA
                                                dl, ch; top row
                                <1>
                                          mov
  3069 0000D345 B8A0000000
                                <1>
                                                eax, 80*2
                                          mov
  3070 0000D34A F7E2
                                <1>
                                          mul
                                                edx
  3071 0000D34C 01C7
                                         add
                                <1>
                                                edi, eax
  3072
                                <1>
  3073 0000D34E 59
                                <1>
                                                ecx ; *
                                          pop
  3074 0000D34F 5A
                                                edx ; **
                                <1>
                                          pop
  3075 0000D350 81E1FFFF0000
                                <1>
                                                ecx, OFFFFh
                                          and
  3076 0000D356 81E2FFFF0000
                                <1>
                                               edx, OFFFFh
                                          and
                                               ecx, 79; max. 80 columns
  3077 0000D35C 83F94F
                                 <1>
                                          cmp
  3078 0000D35F 77A3
                                                short sysvideo_7
                                 <1>
                                          ja
  3079 0000D361 83FA4F
                                                edx, 79; max. 80 columns
                                <1>
                                          cmp
  3080 0000D364 779E
                                <1>
                                                short sysvideo_7
                                          jа
  3081
                                <1>
  3082 0000D366 28CA
                                                dl, cl
                                <1>
                                          sub
  3083 0000D368 769A
                                               short sysvideo 7
                                <1>
                                          jna
  3084
                                <1>
                                          movzx eax, cl ; left column
  3085 0000D36A 0FB6C1
                                <1>
  3086 0000D36D D0E0
                                          shl al, 1 ; column * 2
                                <1>
  3087 0000D36F 01C7
                                                edi, eax
                                <1>
                                          add
                                <1>
  3088
  3089 0000D371 FEC2
                                <1>
                                                dl ; column count
                                          inc
  3090 0000D373 D0E2
                                <1>
                                          shl
                                                dl, 1
  3091 0000D375 88D1
                                <1>
                                          mov
                                                cl, dl ; column count * 2
                                <1>
  3092 0000D377 B2A0
                                          mov
                                                dl, 80*2
                                                eax ; *** (swap address)
  3093 0000D379 58
                                <1>
                                          pop
                                                esi ; ****
  3094 0000D37A 5E
                                <1>
                                          pop
  3095 0000D37B FEC7
                                <1>
                                          inc
                                                bh
  3096
                                <1>
  3097
                                               edx, 80*2
                                <1>
                                          ;mov
  3098 0000D37D B2A0
                                <1>
                                          mov
                                                dl, 80*2
  3099
                                <1>
                                          ;
  3100 0000D37F 80FB01
                                <1>
                                               bl, 1
                                          cmp
  3101 0000D382 7735
                                 <1>
                                                short sysvideo 11
                                          jа
```

```
3102
                               <1>
3103
                          <1>
                                        ; user to system video/display page window transfer (mode 0)
3104 0000D384 21C0
                                       and eax, eax; swap address
3105 0000D386 7413
                                             short sysvideo_10 ; no window swap
                                     ; save previous window content in user's buffer (swap address)
                             <1>
<1>
3106
3107 0000D388 56
                                       push esi ; user buffer
                                      push edi; beginning address of the window
                          <1>
<1>
<1>
3108 0000D389 57
<1>
                                     mov esi, edi
                                             edi, eax
                                       mov
                                       call transfer_to_user_buffer ; fast transfer
                                       pop edi
                                       pop
                                             esi
3114 0000D395 0F82C7F3FFFF
                              <1>
                                        jс
                                             sysret
                              <1> sysvideo 10:
3115
                              ; user to system video/display page window transfer (mode 0)
3116
3117
                              <1>
                                                   user buffer
                                   call transfer_from_user_buffer; fast transfer jc sysret add [u.r0], ecx add edi, edx; next row
3122 0000D3AE 01CE
                              <1>
                                     add esi, ecx
                                            bh
short sysvideo_10
3123 0000D3B0 FECF
                              <1>
                                       dec
                              <1>
3124 0000D3B2 75E7
                                        jnz
                       <1>
3125 0000D3B4 E9A9F3FFFF
                                       jmp
                                             sysret
                              <1>
3126
                              <1> sysvideo_11:
3127
3128
                              <1> ; system to user video/display page window transfer (mode 0)
3129 0000D3B9 87FE
                              <1>
                                        xchg edi, esi
3130
                              <1> sysvideo_12:
                              <1> ; esi = beginning address of the window
3131
3132
                              <1>
                                       ; edi =
                                                  user buffer
                                   call transfer_to_user_buffer ; fast transfer
jc sysret
add [u.r0], ecx
3133 0000D3BB E83D140000
                              <1>
3134 0000D3C0 0F829CF3FFFF
                              <1>
3135 0000D3C6 010D[64030300]
                              <1>
                                     add esi, edx; next row add edi, ecx
3136 0000D3CC 01D6
                              <1>
3137 0000D3CE 01CF
                              <1>
3138 0000D3D0 FECF
                             <1>
                                        dec bh
3139 0000D3D2 /5E/
3140 0000D3D4 E989F3FFFF
                              <1>
                                       jnz short sysvideo_12
                              <1>
                                       jmp
                                             sysret
3141
                              <1>
                              <1> sysvideo 13:
3142
3143 0000D3D9 80FF01
3144 0000D3DC 0F871F030000
3142
                                       cmp bh, 1
ja sysvideo_38
                              <1>
                              <1>
3145
                              <1>
                                       ; BH = 1 = CGA Graphics (OB8000h) data transfers
3146
                              <1>
3147 0000D3E2 20DB
                              <1>
                                        and bl, bl
3148 0000D3E4 751A
                              <1>
                                      jnz short sysvideo_14
3149
                              <1>
3150
                              <1>
                                       ; BL = 0 = Fill color (color in CL] (32K)
3151
                              <1>
3153 0000D3E8 B900800000
3154 0000D3EP 655
                              <1>
                                       mov al, cl
                              <1>
                                       mov
                                             ecx, 32768
3154 0000D3ED 66890D[64030300] <1>
                                       mov
                                             [u.r0], cx
3155 0000D3F4 BF00800B00
                              <1>
                                       mov edi, 0B8000h
3156 0000D3F9 F3AB
                              <1>
                                       rep stosd
3157 0000D3FB E962F3FFFF
                              <1>
                                       jmp
                                             sysret
                              <1>
3159
                              <1> sysvideo_14:
                                   cmp bl, 1
ja short sysvideo_16
3160 0000D400 80FB01
                              <1>
3161 0000D403 7723
                             <1>
3162
                              <1>
3163 0000D405 89CE
                              <1>
                                       mov esi, ecx; user buffer
                              <1>
                                       ; BL = 1 = user to system video/display page transfer
3164
3165
                              <1> sysvideo_15:
3166 0000D407 BF00800B00
                              <1> mov edi, 0B8000h
3167
                              <1>
                                       ; edi = video page address
3172 0000D423 E93AF3FFFF
                              <1>
                                      jmp sysret
3173
                               <1>
3174
                              <1> sysvideo_16:
3175 0000D428 80FB02
                              <1> cmp bl, 2
3176 0000D42B 7723
                              <1>
                                             short sysvideo 18
3177
                              <1>
3178 0000D42D 89CF
                              <1>
                                       mov edi, ecx ; user buffer
3179
                                      ; BL = 2 = system to user video/display page transfer
                              <1>
                               <1> sysvideo 17:
3180
3181 0000D42F BE00800B00
                              <1> mov esi, 0B8000h
3182 0000D434 B900800000
                              <1>
                                        mov ecx, 32768
                                            transfer_to_user_buffer ; fast transfer
3183 0000D439 E8BF130000
                               <1>
                                        call
                                             sysret ; [u.r0] = 0
3184 0000D43E 0F821EF3FFFF
                              <1>
                                        jс
3185 0000D444 66890D[64030300]
                               <1>
                                             [u.r0], cx
3186 0000D44B E912F3FFFF
                               <1>
                                             sysret
                                        jmp
3187
                               <1>
                               <1> sysvideo 18:
3189 0000D450 80FB03
                              <1>
                                        cmp bl, 3
3190 0000D453 777E
                              <1>
                                             short sysvideo_23
3191
                              <1>
3192
                                      ; BL = 3 = NOT bits in window (ECX, EDX)
                              <1>
3193
                               <1>
3194 0000D455 BF00800B00
                              <1>
                                             edi, 0B8000h
                                       mov
3195 0000D45A 89FE
                              <1>
                                             esi, edi
                                      mov
3196
                               <1>
3197 0000D45C 39CA
                                              edx, ecx; bottom-right > top-left?
                              <1>
                                        cmp
3198 0000D45E 7716
                               <1>
                                              short sysvideo 20 ; window
                                        jа
3199
                                        ; full screen (update)
                              <1>
3200 0000D460 B900800000
                               <1>
                                        mov ecx, 32768
3201 0000D465 66890D[64030300] <1>
                                        mov
                                             [u.r0], cx
                              <1> sysvideo_19:
3202
                                        not byte [esi] ; NOT operation inc esi
3203 0000D46C F616
                              <1>
3204 0000D46E 46
                              <1>
3205 0000D46F E2FB
                              <1>
                                        loop sysvideo_19
3206 0000D471 E9ECF2FFFF
                              <1>
                                        jmp sysret
```

```
<1> sysvideo_20:
                  <1>
3208 0000D476 0FB7C2
                                        movzx eax, dx; bottom right column
3209 0000D479 6629C8
                                        sub ax, cx ; - top left column
                   3210 0000D47C 0F82E0F2FFFF
                                        jb sysret; invalid
                                              ax ; same column no == 1 column
3211 0000D482 6640
3212 0000D484 50
                                        push eax; byte count per window row
3213 0000D485 52
3214 0000D486 BB40010000
                                              ebx, 320 ; screen width
3215 0000D48B 89C8
                                              eax, ecx
                                              eax, 16 ; top row
3216 0000D48D C1E810
3217 0000D490 F7E3
3218 0000D492 6689CA
                                              dx, cx; top left column
3219 0000D495 01D0
                                              eax, edx
3220 0000D497 01C6
                                              esi, eax ; start address
3221 0000D499 59
                                              ecx ; edx
3222 0000D49A 89C8
                                              eax, ecx
3223 0000D49C C1E810
                                              eax, 16 ; bottom row
3224 0000D49F F7E3
3225 0000D4A1 6689CA
                                              dx, cx; bottom right column
3226 0000D4A4 01D0
                                              eax, edx
3227 0000D4A6 01C7
                                              edi, eax ; stop address (included)
3228 0000D4A8 5A
                                              edx ; byte count per window row
3229 0000D4A9 81FFFFF0B00
3230 0000D4AF 0F87ADF2FFFF
                                              edi, OBFFFFh
                                              sysret
3231 0000D4B5 56
3232 0000D4B6 4E
3234 0000D4B7 89D1
                                        mov ecx, edx
                              <1>
3235
                               <1> sysvideo_22:
3236 0000D4B9 46
                              <1> inc esi
3237 0000D4BA F616
                              <1>
                                        not byte [esi]
3238 0000D4BC E2FB
                              <1>
                                        loop sysvideo 22
                              <1>
3239 0000D4BE 01DE
                                        add esi, ebx; bytes per screen row
3240
                              <1>
                                      cmp
3241 0000D4C0 39FE
                              <1>
                                              esi, edi ; stop address (included in loop)
3242 0000D4C2 76F3
                                              short sysvideo 21
                              <1>
                                        jna
                                        pop
3243 0000D4C4 5E
                              <1>
                                              esi
3244 0000D4C5 29F7
                               <1>
                                              edi, esi
                                        sub
3245 0000D4C7 66893D[64030300]
                              <1>
                                              [u.r0], di
                                        mov
3246 0000D4CE E98FF2FFFF
                              <1>
                                              sysret
                                        jmp
3247
                               <1>
                               <1> sysvideo 23:
3248
3249 0000D4D3 80FB04
                                        cmp bl, 4
                               <1>
3250 0000D4D6 0F87A7000000
                              <1>
                                                sysvideo_26
3251
                               <1>
                                        ; BL = 4 = window copy (system to system)
3252
                               <1>
3253
                              <1>
3254 0000D4DC B800800B00
                              <1>
                                              eax, 0B8000h
                                        mov
3255 0000D4E1 39C6
                               <1>
                                        cmp
                                              esi, eax
3256 0000D4E3 0F8279F2FFFF
                              <1>
                                        jb
                                              sysret
                                              edi, eax
3257 0000D4E9 39C7
                              <1>
                                        cmp
3258 0000D4EB 0F8271F2FFFF
                               <1>
                                        jb
                                              sysret
3259 0000D4F1 6605FF7F
                                              ax, 7FFFh ; 32767
                              <1>
                                        add
3260 0000D4F5 39C6
                              <1>
                                              esi, eax
                                        cmp
3261 0000D4F7 0F8765F2FFFF
                              <1>
                                        jа
                                              sysret
3262 0000D4FD 39C7
                               <1>
                                        cmp
                                              edi, eax
3263 0000D4FF 0F875DF2FFFF
                              <1>
                                        jа
                                              sysret
3264
                               <1>
3265 0000D505 39CA
                               <1>
                                             edx, ecx; bottom-right > top-left?
                                        cmp
                                              short sysvideo_24 ; window
3266 0000D507 7714
                              <1>
                                        jа
3267
                              <1>
                                        ; full screen copy
3268 0000D509 89C1
                               <1>
                                        mov ecx, eax
                                             ecx, edi
3269 0000D50B 29F9
                              <1>
                                        sub
3270 0000D50D 6641
                               <1>
                                        inc
                                              [u.r0], cx
3271 0000D50F 66890D[64030300] <1>
                                        mov
3272 0000D516 F3A4
                               <1>
                                        rep
                                              movsb
3273 0000D518 E945F2FFFF
                              <1>
                                        jmp
                                              sysret
                       3274
                              <1> sysvideo_24:
                              <1>
3275 0000D51D 0FB7C2
                                        movzx eax, dx ; bottom right column
3276 0000D520 6629C8
                                        sub ax, cx ; - top left column
3277 0000D523 0F8239F2FFFF
                                        jb sysret; invalid
3278 0000D529 6640
                                        inc
                                              ax ; same column no == 1 column
3279 0000D52B 50
                                        push eax; byte count per window row
3280
                              <1>
3281 0000D52C 52
                               <1>
                                        push edx
                                              ebx, 320 ; screen width
3282 0000D52D BB40010000
                              <1>
                                        mov
3283 0000D532 89C8
                              <1>
                                              eax, ecx
                                        mov
3284 0000D534 C1E810
                              <1>
                                        shr
                                              eax, 16
                                                         ; top row
3285 0000D537 F7E3
                              <1>
                                        mul
                                              ebx
                                              dx, cx; top left column
3286 0000D539 6689CA
                              <1>
                                        mov
3287 0000D53C 01D0
                              <1>
                                        add
                                              eax, edx
3288 0000D53E 01C7
                               <1>
                                        add
                                               edi, eax ; start address
                          <1>
3289 0000D540 01C6
                                        add
                                              esi, eax
3290 0000D542 59
                               <1>
                                              ecx ; edx
                                        pop
                              <1>
3291 0000D543 89C8
                                              eax, ecx
                                        mov
                              <1>
3292 0000D545 C1E810
                                        shr
                                              eax, 16 ; bottom row
3293 0000D548 F7E3
                              <1>
                                        mul
                                              ebx
                             <1>
                                      mov
3294 0000D54A 6689CA
                                              dx, cx; bottom right column
3295 0000D54D 01D0
                              <1>
                                       add
3296 0000D54F 5A
                              <1>
                                       pop
                                              edx ; byte count per window row
3297 0000D550 0500800B00
                              <1>
                                        add
                                              eax, 0B8000h
3298 0000D555 3DFFFF0B00
                                              eax, OBFFFFh
                               <1>
                                        cmp
3299 0000D55A 0F8702F2FFFF
                              <1>
                                        jа
                                              sysret
                                        push edi ; start address
3300 0000D560 57
                              <1>
3301 0000D561 50
                               <1>
                                        push eax ; stop address (included)
                              <1> sysvideo_25:
3302
3303 0000D562 89D1
                              <1>
                                        mov ecx, edx
3304 0000D564 F3A4
                              <1>
                                        rep
                                              movsb
3305 0000D566 4F
                              <1>
                                        dec
                                              edi
3306 0000D567 4E
                              <1>
                                        dec
                                              esi
3307 0000D568 01DF
                              <1>
                                        add
                                              edi, ebx ; bytes per screen row
3308 0000D56A 01DE
                              <1>
                                        add
                                              esi, ebx
                              <1>
3309
                                        ;
3310 0000D56C 3B3C24
                              <1>
                                               edi, [esp] ; stop addr(included in loop)
                                        cmp
3311 0000D56F 76F1
                               <1>
                                              short sysvideo 25
                                        jna
```

3207

```
3312 0000D571 5B
                                                       <1>
                                                                      pop
                                                                                ebx ; stop address
      3313 0000D572 5F
                                                        <1>
                                                                       pop
                                                                                edi ; start address
      3314 0000D573 29FB
                                                       <1>
                                                                                ebx, edi
                                                                       sub
      3315 0000D575 6643
                                                       <1>
                                                                       inc
                                                                                bx
      3316 0000D577 66891D[64030300] <1>
                                                                       mov
                                                                                [u.r0], bx
      3317 0000D57E E9DFF1FFFF
                                                       <1>
                                                                       jmp
                                                                                sysret
                                                       <1>
      3319
                                                        <1> sysvideo 26:
      3320 0000D583 80FB05
                                                       <1>
                                                                       cmp bl, 5
      3321 0000D586 0F8795000000
                                                       <1>
                                                                       ja sysvideo_29
      3322
                                                       <1>
      3323
                                                        <1>
                                                                      ; BL = 5 = window copy (user to system)
      3324
                                                       <1>
     3324

3325 0000D58C B800800B00

3326 0000D591 39C7

3327 0000D593 0F82C9F1FFFF

3328 0000D599 6605FF7F
                                                                               eax, 0B8000h
                                                      <1>
                                                                       mov
                                                                               edi, eax
                                                       <1>
                                                                       cmp
                                                                                 sysret
                                                       <1>
                                                                       jb
                                                                               ax, 7FFFh ; 32767
                                                      <1>
                                                                       add
      3329 0000D59D 39C7
                                                       <1>
                                                                               edi, eax
                                                                       cmp
      3330 0000D59F 0F87BDF1FFFF
                                                       <1>
                                                                                svsret
                                                                       jа
     3331
                                                       <1>
      3332
                                                       <1>
                                                                       ; esi = user buffer (in user's memory space)
      3333 0000D5A5 39CA
                                                                       cmp edx, ecx; bottom-right > top-left ?
                                                       <1>
      3334 0000D5A7 0F865AFEFFFF
                                                      <1>
                                                                       jna sysvideo_15 ; full screen copy
                                                       <1>
      3336 0000D5AD 0FB7C2
                                                       <1>
                                                                       movzx eax, dx ; bottom right column
      3337 0000D5B0 6629C8
                                                                       sub ax, cx ; - top left column
                                                       <1>
      3338 0000D5B3 0F82A9F1FFFF
                                                      <1>
                                                                               sysret ; invalid
                                                                       jb
      3339 0000D5B9 6640
                                                       <1>
                                                                      inc ax; same column no == 1 column
      3340 0000D5BB 50
                                                       <1>
                                                                      push eax; byte count per window row
      3341
                                                       <1>
      3342 0000D5BC 52
                                             <1>
                                                                      push edx
| Mov | Shr | Mov | Shr 
      3343 0000D5BD BB40010000
                                                                                ebx, 320 ; screen width
                                                                       mov
                                                                                eax, ecx
                                                                                eax, 16
                                                                                                  ; top row
                                                                                ebx
                                                                                dx, cx; top left column
                                                                               eax, edx
                                                                                edi, eax ; start address
                                                                                ecx ; edx
                                                                                eax, ecx
                                                                                eax, 16 ; bottom row
                                                                                ebx
                                                                                dx, cx; bottom right column
                                                                                eax, edx
                                                                                edx ; byte count per window row
                                                                                eax, 0B8000h
                                                                                eax, OBFFFFh
                                                                                 sysret
                                                                                edi ; start address
                                                                     push eax ; stop address (included)
                                                       <1> mov ecx, edx; byte count
     <1>
      3364
                                                                      ; user to system video/display page window transfer
      3376 0000D60C 6643
                                                        <1>
                                                                       inc
                                                                                bx
      3377 0000D60E 66891D[64030300] <1>
                                                                       mov
                                                                                 [u.r0], bx
      3378 0000D615 E948F1FFFF
                                                                       jmp
                                                       <1>
                                                                                sysret
      3379
                                                       <1> sysvideo_28:
      3380 0000D61A 58
                                                       <1>
                                                                       pop
      3381 0000D61B 5A
                                                       <1>
                                                                                edx
                                                                       pop
      3382 0000D61C E941F1FFFF
                                                       <1>
                                                                       jmp
                                                                                sysret
      3383
                                                        <1>
                                                        <1> sysvideo_29:
      3384
      3385 0000D621 80FB06
                                                       <1>
                                                                       cmp
                                                                             bl, 6
                                                                                 sysvideo_32
      3386 0000D624 0F8797000000
                                                        <1>
                                                                       jа
      3387
                                                        <1>
      3388
                                                        <1>
                                                                      ; BL = 6 = window copy (system to user)
      3389
                                                        <1>
                                                                       mov edi, esi ; user buffer
      3390 0000D62A 89F7
                                                        <1>
      3391
                                                        <1>
      3392 0000D62C B800800B00
                                                                                eax, 0B8000h
                                                        <1>
                                                                       mov
      3393 0000D631 39C6
                                                                                 esi, eax
                                                        <1>
                                                                       cmp
      3394 0000D633 0F8229F1FFFF
                                                       <1>
                                                                       jb
                                                                                 sysret
      3395 0000D639 6605FF7F
                                                        <1>
                                                                                ax, 7FFFh ; 32767
      3396 0000D63D 39C6
                                                        <1>
                                                                       cmp
                                                                               esi, eax
      3397 0000D63F 0F871DF1FFFF
                                                       <1>
                                                                       jа
                                                                                 sysret
                                                       <1>
      3399
                                                       <1>
                                                                       ; edi = user buffer (in user's memory space)
      3400 0000D645 39CA
                                                        <1>
                                                                       cmp edx, ecx; bottom-right > top-left ?
                                                                       jna sysvideo 17 ; full screen copy
      3401 0000D647 0F86E2FDFFFF
                                                       <1>
      3402
                                                        <1>
      3403 0000D64D 0FB7C2
                                                        <1>
                                                                       movzx eax, dx; bottom right column
      3404 0000D650 6629C8
                                                       <1>
                                                                       sub ax, cx ; - top left column
      3405 0000D653 0F8209F1FFFF
                                                       <1>
                                                                                sysret ; invalid
      3406 0000D659 6640
                                                                                ax ; same column no == 1 column
                                                        <1>
                                                                       inc
      3407 0000D65B 50
                                                       <1>
                                                                       push
                                                                               eax ; byte count per window row
                                                       <1>
      3409 0000D65C 52
                                                       <1>
                                                                       push edx
                                                                                 ebx, 320 ; screen width
      3410 0000D65D BB40010000
                                                       <1>
                                                                       mov
      3411 0000D662 89C8
                                                      <1>
                                                                                 eax, ecx
                                                                       mov
      3412 0000D664 C1E810
                                                      <1>
                                                                       shr
                                                                                eax, 16
                                                                                                   ; top row
      3413 0000D667 F7E3
                                                       <1>
                                                                       mul
                                                                                ebx
                                                      <1>
      3414 0000D669 6689CA
                                                                       mov
                                                                                dx, cx; top left column
      3415 0000D66C 01D0
                                                      <1>
                                                                       add
                                                                                eax, edx
      3416 0000D66E 01C6
                                                        <1>
                                                                       add
                                                                               esi, eax ; start address
```

```
3417 0000D670 59
                               <1>
                                         pop
                                               ecx ; edx
3418 0000D671 89C8
                               <1>
                                         mov
                                               eax, ecx
                              <1>
3419 0000D673 C1E810
                                         shr
                                               eax, 16 ; bottom row
3420 0000D676 F7E3
                              <1>
                                         mul
                                               ebx
3421 0000D678 6689CA
                               <1>
                                         mov
                                               dx, cx; bottom right column
                               <1>
3422 0000D67B 01D0
                                         add
                                               eax, edx
                              <1> pop
<1> add
<1> cmp
<1> ja
<1> push
3423 0000D67D 5A
                                               edx ; byte count per window row
3424 0000D67E 0500800B00 <1>
3425 0000D688 3DFFFF0B00 <1>
3426 0000D688 0F87D4F0FFF <1>
                                               eax, 0B8000h
                                               eax, OBFFFFh
                                               sysret
3427 0000D68E 56
                                         push esi ; start address
3428 0000D68F 50
                               <1>
                                         push eax ; stop address (included)
3429
                               <1> sysvideo_30:
3430 0000D690 89D1
                               <1> mov ecx, edx ; byte count
                                         ; user to system video/display page window transfer
3431
                               <1>
                          3435 0000D699 010D[64030300]
3436 0000D69F 01DF
3437 0000D6A1 01CE
                                               edi, [esp]; stop addr(included in loop)
3438 0000D6A3 3B3C24
3439 0000D6A6 76E8
3440 0000D6A8 5B
3441 0000D6A9 5F
3442 0000D6AA 29FB
3443 0000D6AC 6643
                               <1>
                                         inc
                                               bx
3444 0000D6AE 66891D[64030300] <1>
                                               [u.r0], bx
                                         mov
3445 0000D6B5 E9A8F0FFFF
                               <1>
                                         jmp
                                               sysret
3446
                               <1> sysvideo_31:
3447 0000D6BA 58
                               <1>
                                         pop eax
3448 0000D6BB 5A
                               <1>
                                               edx
                                         pop
3449 0000D6BC E9A1F0FFFF
                               <1>
                                         jmp
                                               sysret
3450
                               <1>
3451
                               <1> sysvideo_32:
                                        cmp bl, 7
ja short sysvideo_34
3452 0000D6C1 80FB07
                               <1>
3453 0000D6C4 770F
                               <1>
3454
                               <1>
                                        ; BL = 7 = AND display page bytes with CL
3455
                               <1>
3456
                               <1>
3457 0000D6C6 BE00800B00
                               <1>
                                         mov esi, OB8000h
3458 0000D6CB B900800000
                               <1>
                                         mov
                                              ecx, 32768
                               <1> sysvideo_33:
                                         and byte [esi], cl
3460 0000D6D0 200E
                               <1>
3461 0000D6D2 46
                                         inc esi
                               <1>
3462 0000D6D3 E2FB
                               <1>
                                         loop sysvideo_33
3463
                               <1>
3464
                               <1> sysvideo_34:
                                         cmp bl, 8
ja short sysvideo_36
3465 0000D6D5 80FB08
                               <1>
3466 0000D6D8 770F
                               <1>
3467
                               <1>
3468
                                <1>
                                        ; BL = 8 = OR display page bytes with CL
3469
                               <1>
3470 0000D6DA BE00800B00
                               <1>
                                         mov esi, 0B8000h
3471 0000D6DF B900800000
                               <1>
                                         mov
                                              ecx, 32768
3472
                               <1> sysvideo_35:
3473 0000D6E4 080E
                               <1> or byte [esi], cl
3474 0000D6E6 46
                               <1>
                                         inc esi
3475 0000D6E7 E2FB
                               <1>
                                         loop sysvideo_35
3476
                               <1>
3477
                               <1> sysvideo_36:
                                     cmp bl, 9
3478 0000D6E9 80FB09
                                <1>
3479 0000D6EC 0F8770F0FFFF
                               <1>
                                         jа
                                               sysret ; nothing to do
3480
                               <1>
3481
                                <1>
                                        ; BL = 9 = XOR display page bytes with CL
3482
                               <1>
3483 0000D6F2 BE00800B00
                                             esi, 0B8000h
                               <1>
3484 0000D6F7 B900800000
                               <1>
                                         mov ecx, 32768
3485
                               <1> sysvideo_37:
                                         xor byte [esi], cl
3486 0000D6FC 300E
                               <1>
3487 0000D6FE 46
                               <1>
                                         inc esi
3488 0000D6FF E2FB
                               <1>
                                         loop sysvideo_37
3489
                               <1>
3490
                               <1> sysvideo_38:
                                     cmp bh, 2
ja sysvideo_64
3491 0000D701 80FF02
                               <1>
3492 0000D704 0F8733030000
                               <1>
                               <1>
                                         ; BH = 2 = VGA Graphics (0A0000h) data transfers
3494
                               <1>
3495 0000D70A 88DC
                               <1>
                                              ah, bl
                                         mov
3496 0000D70C 80E30F
                               <1>
                                         and bl, 0Fh
3497 0000D70F C0EC04
                                <1>
                                         shr
                                               ah, 4
3498 0000D712 C1E310
                                <1>
                                         shl
                                               ebx, 16
3499 0000D715 66BB4001
                            <1>
                                               bx, 320; 320*200, 320*240
                                         mov
3500 0000D719 20E4
                               <1>
                                               ah, ah
                                               short sysvideo 39
3501 0000D71B 7413
                               <1>
                                         iΖ
                                               bx, 1; 640*20\overline{0}, 640*400, 640*480
                               <1>
3502 0000D71D 66D1E3
                                         shl
3503 0000D720 80FC02
                              <1>
                                         cmp
                                              ah, 2
3504 0000D723 720B
                               <1>
                                         jb
                                               short sysvideo_39
3505 0000D725 0F8737F0FFFF
                               <1>
                                               sysret ; invalid
                                         jа
                               <1>
                                         ; 800*600
3507 0000D72B 6681C3A000
                                         add bx, 160; 800
                               <1>
3508
                                <1> sysvideo 39:
3509 0000D730 C1CB10
                                              ebx, 16
                               <1>
                                         ror
                               <1>
3511 0000D733 20DB
                                <1>
                                         and
                                               bl, bl
3512 0000D735 7519
                               <1>
                                         jnz
                                               short sysvideo_40
3513
                                <1>
3514
                                        ; BL = 0 = Fill color (color in CL] (64K)
                                <1>
3515
                                <1>
3516 0000D737 88C8
                               <1>
                                               al, cl
                                         mov
3517 0000D739 B900000100
                               <1>
                                         mov
                                               ecx, 65536
3518 0000D73E 890D[64030300]
                               <1>
                                         mov
                                               [u.r0], ecx
3519 0000D744 BF00000A00
                               <1>
                                         mov
                                               edi, 0A0000h
3520 0000D749 F3AB
                                <1>
                                         rep
                                               stosd
3521 0000D74B E912F0FFFF
                                <1>
                                               sysret
                                         jmp
```

```
3522
                                                        <1>
  3523
                                                        <1> sysvideo 40:
  3524 0000D750 80FB01
                                                       <1> cmp bl, 1
  3525 0000D753 7722
                                                      <1>
                                                                                 short sysvideo_42
                                                      <1>
  3526
  3527 0000D755 89CE
                                                      <1>
                                                                       mov esi, ecx; user buffer
                                                                     ; BL = 1 = user to system video/display page transfer
  3528
                                                      <1>
  3529
                                                      <1> sysvideo_41:
                                                                mov edi, 0A0000h
  3530 0000D757 BF00000A00
                                                      <1>
                                                                       ; edi = video page address
  3531
                                                      <1>
 3532 0000D75C B900000100
3533 0000D761 E8E1100000
3534 0000D766 0F82F6EFFFFF
  call transfer_from_user_buffer ; fast transfer
jc sysret ; [u.r0] = 0
  3536 0000D772 E9EBEFFFFF
                                                                    jmp sysret
                                                      <1>
                                                       <1>
  3537
                                                      <1> sysvideo 42:
  3538
  3539 0000D777 80FB02
                                                      <1> cmp bl, 2
  3540 0000D77A 7722
                                                      <1>
                                                                                 short sysvideo_44
  3541
                                                      <1>
  3542 0000D77C 89CF
                                                      <1>
                                                                       mov edi, ecx ; user buffer
                                                                      ; BL = 2 = system to user video/display page transfer
  3543
                                                       <1>
  3544
                                                      <1> sysvideo_43:
 3544
3545 0000D77E BE00000A00
3546 0000D783 B900000100
3547 0000D788 E870100000
3548 0000D78D 0F82CFEFFFFF
3549 0000D793 890D[64030300]
3548 0000D793 890D[64030300]
3549 0000D793 890D[64030300]
3549 0000D793 890D[64030300]
3549 0000D793 890D[64030300]
3549 0000D793 890D[64030300]
                                                                       call transfer_to_user_buffer; fast transfer
                                                                       jc sysret ; [u.r0] = 0
  3550 0000D799 E9C4EFFFFF
                                                      <1>
                                                                      jmp
                                                                                 sysret
  3551
                                                       <1>
  3552
                                                       <1> sysvideo_44:
  3553 0000D79E 80FB03
                                                                cmp bl, 3
                                                       <1>
  3554 0000D7A1 777A
                                                       <1>
                                                                       jа
                                                                                 short sysvideo_49
  3555
                                                       <1>
                                                                       ; BL = 3 = NOT bits in window (ECX, EDX)
  3556
                                                        <1>
  3557
                                                       <1>
  3558 0000D7A3 BF00000A00
                                                                                edi, 0A0000h
                                                      <1>
                                                                      mov
  3559 0000D7A8 89FE
                                                       <1>
                                                                    mov
                                                                                 esi, edi
  3560
                                                       <1>
  3561 0000D7AA 39CA
                                                      <1>
                                                                       cmp edx, ecx ; bottom-right > top-left ?
                                                                       ja short sysvideo_45 ; window
  3562 0000D7AC 770B
                                                      <1>
D7AE BS
0D7B3 890D1

000D7B9 F616
0000D7BB 46
0000D7BC E2FB
J 0000D7C3 0FB7C2 <1>
3573 0000D7C6 6629C8 <1>
3574 0000D7C9 0F8293EFFFFF <1>
3575 0000D7CF 6640 <1>
3576 0000D7D1 50 <1>
push
77 0000D7D2 52 <1>
push
000D7D3 C1EB10 <1>
push
00D7D6 89C8 <1>
mc
7D8 C1E810 <1>
F7E3 <1>
F7E3 <1>
<1>
<1>
<1>

<a href="https://www.example.com/repairs/balance/">
<a href="http
                                                                       ; full screen (update)
  3563
3564 0000D7AE B900000100
3565 0000D7B3 890D[64030300]
<1> (1) 8
  3563
                                                       <1>
                                                                       mov ecx, 65536
                                                                       mov
                                                                                [u.r0], ecx
                                                      <1> sysvideo_45:
                                                                       not byte [esi]; NOT operation inc esi
                                                                       loop sysvideo_45
                                                                        jmp
                                                                                 sysret
                                                      <1> sysvideo 46:
                                                     mul
                                                                                 ebx
                                                                 mov
                                                                                 dx, cx; top left column
                                                                                 eax, edx
                                                                                 esi, eax ; start address
                                            ecx ; edx
                                                                       pop
                                                                                 eax, ecx
                                                                                 eax, 16; bottom row
                                                                    mul
  3588 0000D7EA F7E3
                                                                                 ebx
  3589 0000D7EC 6689CA
                                                      <1> mov
<1> add
<1> add
<1> pop
                                                                                 dx, cx; bottom right column
  3590 0000D7EF 01D0
  3591 0000D7F1 01C7
                                                                                 edi, eax ; stop address (included)
  3592 0000D7F3 5A
                                                                                 edx ; byte count per window row
  3593 0000D7F4 81FFFFFF0A00
                                                       <1>
                                                                       cmp
                                                                                 edi, OAFFFFh
  3594 0000D7FA 0F8762EFFFFF
                                                       <1>
                                                                        jа
                                                                                  sysret
                                                                       push esi
  3595 0000D800 56
                                                       <1>
  3596 0000D801 4E
                                                       <1>
                                                                        dec
                                                                                 esi
                                                       <1> sysvideo_47:
  3597
  3598 0000D802 89D1
                                                       <1> mov
                                                                              ecx, edx
  3599
                                                       <1> sysvideo_48:
  3600 0000D804 46
                                                       <1>
                                                                 inc esi
                                                                       not byte [esi]
  3601 0000D805 F616
                                                      <1>
  3602 0000D807 E2FB
                                                       <1>
                                                                       loop sysvideo_48
  3603 0000D809 01DE
                                                        <1>
                                                                        add
                                                                                  esi, ebx ; bytes per screen row
                                                        <1>
  3605 0000D80B 39FE
                                                        <1>
                                                                                  esi, edi ; stop address (included in loop)
  3606 0000D80D 76F3
                                                        <1>
                                                                        jna
                                                                                  short sysvideo 47
  3607 0000D80F 5E
                                                       <1>
                                                                        pop
                                                                                  esi
                                                                                  edi, esi
  3608 0000D810 29F7
                                                       <1>
                                                                        sub
  3609 0000D812 893D[64030300]
                                                      <1>
                                                                                  [u.r0], edi
                                                                        mov
  3610 0000D818 E945EFFFFF
                                                       <1>
                                                                                  sysret
                                                                        jmp
  3611
                                                       <1>
  3612
                                                        <1> sysvideo 49:
  3613 0000D81D 80FB04
                                                                        cmp bl, 4
                                                        <1>
  3614 0000D820 0F87A1000000
                                                       <1>
                                                                        jа
                                                                                  sysvideo_52
  3615
                                                       <1>
  3616
                                                        <1>
                                                                       ; BL = 4 = window copy (system to system)
  3617
                                                       <1>
  3618 0000D826 B800000A00
                                                       <1>
                                                                                  eax, 0A0000h
  3619 0000D82B 39C6
                                                       <1>
                                                                        cmp
                                                                                  esi, eax
  3620 0000D82D 0F822FEFFFFF
                                                       <1>
                                                                        jb
                                                                                  svsret
  3621 0000D833 39C7
                                                      <1>
                                                                                 edi, eax
                                                                        cmp
  3622 0000D835 0F8227EFFFFF
                                                      <1>
                                                                        jb
                                                                                  sysret
                                                                                  ax, 0FFFFh ; 65535
  3623 0000D83B 6683C0FF
                                                       <1>
                                                                        add
  3624 0000D83F 39C6
                                                       <1>
                                                                        cmp
                                                                                 esi, eax
  3625 0000D841 0F871BEFFFFF
                                                       <1>
                                                                                  svsret
                                                                        jа
  3626 0000D847 39C7
                                                        <1>
                                                                                 edi, eax
                                                                        cmp
```

```
3627 0000D849 0F8713EFFFFF <1>
                                       jа
                                             sysret
3628
                              <1>
3629 0000D84F 39CA
                              <1>
                                       cmp edx, ecx ; bottom-right > top-left ?
                          <1>
<1>
<1>
3630 0000D851 7712
                                             short sysvideo_50 ; window
                                       jа
3631
                                       ; full screen copy
                              <1>
3632 0000D853 89C1
                                       mov ecx, eax
                    <1>
3633 0000D855 29F9
                                      sub ecx, edi
inc ecx
mov
                                             [u.r0], ecx
                                             movsb
                                             sysret
                              <1> sysvideo 50:
                              <1> movzx eax, dx; bottom right column
                                       sub ax, cx ; - top left column
                                       jb sysret; invalid
                                       inc ax; same column no == 1 column
                                       push eax ; byte count per window row
3644
                              <1>
3645 0000D874 52
                                       push edx
                         <1>
                                             ebx, 16; 320,640,800 : screen width
3646 0000D875 C1EB10
                                       shr
3647 0000D878 89C8
                                       mov
                                             eax, ecx
3648 0000D87A C1E810
                                             eax, 16
                                       shr
                                                        ; top row
                        3649 0000D87D F7E3
                                             ebx
3650 0000D87F 6689CA
                                             dx, cx; top left column
3651 0000D882 01D0
                                             eax, edx
3652 0000D884 01C7
                                             edi, eax ; start address
3653 0000D886 01C6
                                             esi, eax
3654 0000D888 59
                                             ecx ; edx
3655 0000D889 89C8
                                             eax, ecx
3656 0000D88B C1E810
                                             eax, 16; bottom row
3657 0000D88E F7E3
                                             ebx
3658 0000D890 6689CA
                                             dx, cx; bottom right column
3659 0000D893 01D0
                                             eax, edx
3660 0000D895 5A
                                             edx ; byte count per window row
eax, 0A0000h
                                             eax, OAFFFFh
                                             sysret
3664 0000D8A6 57
                                       push edi ; start address
3665 0000D8A7 50
                              <1>
                                       push eax ; stop address (included)
                              <1> sysvideo_51:
3666
                          <1>
<1>
<1>
3667 0000D8A8 89D1
                                       mov ecx, edx
3668 0000D8AA F3A4
                                       rep
                                             movsb
                              <1>
3669 0000D8AC 4F
                                             edi
                                       dec
3670 0000D8AD 4E
                            <1>
                                       dec
                                             esi
3671 0000D8AE 01DF
                              <1>
                                       add
                                             edi, ebx ; bytes per screen row
3672 0000D8B0 01DE
                              <1>
                                       add
                                             esi, ebx
3673
                              <1>
                                       ;
                       3674 0000D8B2 3B3C24
                                   cmp
jna
pop
                                             edi, [esp] ; stop addr(included in loop)
3675 0000D8B5 76F1
                              <1>
                                             short sysvideo 51
3676 0000D8B7 5B
                                             ebx ; stop address
                                       pop
                                       pop
3677 0000D8B8 5F
                                             edi ; start address
3678 0000D8B9 29FB
                                       sub
                                             ebx, edi
3679 0000D8BB 43
                              <1>
                                       inc
                                             ebx
3680 0000D8BC 891D[64030300] <1>
                                             [u.r0], ebx
                                       mov
3681 0000D8C2 E99BEEFFFF
                              <1>
                                       jmp
                                             sysret
3682
                              <1>
                              <1> sysvideo_52:
3684 0000D8C7 80FB05
                                       cmp bl, 5
                              <1>
3685 0000D8CA 0F8791000000
                              <1>
                                       ja sysvideo_55
                              <1>
3686
3687
                              <1>
                                       ; BL = 5 = window copy (user to system)
3688
                              <1>
3689 0000D8D0 B800000A00
3690 0000D8D5 39C7
                              <1>
                                             eax, 0A0000h
                                       mov
3690 0000D8D5 39C7
                              <1>
                                             edi, eax
                                        cmp
3692 0000D8DD 6683C0FF
3693 0000D8E1 39C7
                              <1>
                                        jb
                                             sysret
                              <1>
                                       add
                                             ax, 0FFFFh ; 65535
                              <1>
                                       cmp
                                            edi, eax
3694 0000D8E3 0F8779EEFFFF
                              <1>
                                       jа
                                             sysret
3695
                              <1>
3696
                              <1>
                                       ; esi = user buffer (in user's memory space)
3697 0000D8E9 39CA
                              <1>
                                       cmp edx, ecx ; bottom-right > top-left ?
                                       jna sysvideo_41 ; full screen copy
3698 0000D8EB 0F8666FEFFFF
                              <1>
                              <1>
3699
3700 0000D8F1 0FB7C2
                              <1>
                                       movzx eax, dx ; bottom right column
                                       sub ax, cx ; - top left column
3701 0000D8F4 6629C8
                              <1>
3702 0000D8F7 0F8265EEFFFF
                                             sysret ; invalid
                              <1>
                                       jb
3703 0000D8FD 6640
                              <1>
                                       inc ax; same column no == 1 column
3704 0000D8FF 50
                              <1>
                                       push eax; byte count per window row
3705
                              <1>
3706 0000D900 52
                              <1>
                                       push edx
3707 0000D901 C1EB10
                              <1>
                                        shr
                                             ebx, 16; 320,640,800: screen width
3708 0000D904 89C8
                                              eax, ecx
                               <1>
                                        mov
                         <1>
3709 0000D906 C1E810
                                             eax, 16
                                       shr
                                                        ; top row
3710 0000D909 F7E3
                              <1>
3711 0000D90B 6689CA
                              <1>
                                       mov
                                             dx, cx; top left column
                              <1>
                                             eax, edx
3712 0000D90E 01D0
                                       add
3713 0000D910 01C7
                             <1>
                                       add
                                             edi, eax ; start address
                             <1>
<1>
3714 0000D912 59
                           <1> pop
<1> mov
<1> shr
                                             ecx ; edx
3715 0000D913 89C8
                                             eax, ecx
3716 0000D915 C1E810
                                             eax, 16 ; bottom row
                             <1>
3717 0000D918 F7E3
                                   mul
                                             ebx
                             <1>
<1>
3718 0000D91A 6689CA
                                       mov
                                             dx, cx; bottom right column
3719 0000D91D 01D0
                                             eax, edx
                                       add
                             <1> pop
3720 0000D91F 5A
                                             edx ; byte count per window row
3721 0000D91F 3A
3721 0000D920 0500000A00
3722 0000D925 3DFFFF0A00
3723 0000D92A 0F8732EEFFFF
                             <1> add 
<1> cmp
                                             eax, 0A0000h
                                       cmp
                                             eax, OAFFFFh
                                     ja -
                             <1>
                                             sysret
3724 0000D930 57
                              <1>
                                       push edi ; start address
                                       push eax ; stop address (included)
3725 0000D931 50
                              <1>
                              <1> sysvideo_53:
3726
                              <1>
3727 0000D932 89D1
                                       mov ecx, edx; byte count
3728
                              <1>
                                       ; user to system video/display page window transfer
                              <1>
3729
                                       ; esi = user buffer
                                     call transfer_from_user_buffer ; fast transfer
                              <1>
3730 0000D934 E80E0F0000
3731 0000D939 721F
                              <1>
                                     jc short sysvideo_54
```

```
add
add
add
cm'
-> j'
3732 0000D93B 010D[64030300] <1>
                                             [u.r0], ecx
3733 0000D941 01DF
                               <1>
                                              edi, ebx ; next row
                              <1>
3734 0000D943 01CE
                                              esi, ecx
                            <1>
3735 0000D945 3B3C24
                                              edi, [esp] ; stop addr(included in loop)
3736 0000D948 76E8
                              <1>
                                              short sysvideo 53
3737 0000D94A 5B
                              <1>
                                              ebx ; stop address
3738 0000D94B 5F
                                              edi ; start address
                              <1>
3739 0000D94C 29FB
                              <1>
                                              ebx, edi
                                    sub
3740 0000D94E 43
                               <1>
                                        inc
                                              ebx
3742 0000D955 E908EEFFFF <1>
3743 3744 0000T05
                                        mov
                                              [u.r0], ebx
                                        jmp
                                              sysret
                               <1> sysvideo_54:
                              <1>
3744 0000D95A 58
                                        pop eax
3745 0000D95B 5A
                              <1>
                                              edx
                                        pop
3746 0000D95C E901EEFFFF
                              <1>
                                        jmp
                                              sysret
3747
                               <1>
3748
                               <1> sysvideo 55:
3749 0000D961 80FB06
                               <1>
                                        cmp bl, 6
3750 0000D964 0F8793000000
                               <1>
                                              sysvideo_58
3751
                               <1>
3752
                               <1>
                                        ; BL = 6 = window copy (system to user)
3753
                               <1>
3754 0000D96A 89F7
                               <1>
                                        mov edi, esi ; user buffer
3755
                               <1>
3756 0000D96C B800000A00
                               <1>
                                              eax, 0A0000h
                                        mov
3757 0000D971 39C6
                               <1>
                                        cmp
                                              esi, eax
3758 0000D973 0F82E9EDFFFF
                              <1>
                                        jb
                                              sysret
3759 0000D979 6683C0FF
                                              ax, 0FFFFh ; 65535
                                        add
                               <1>
                                        cmp esi, eax ja sysret
3760 0000D97D 39C6
                               <1>
3761 0000D97F 0F87DDEDFFFF
                              <1>
3762
                               <1>
3763
                               <1>
                                        ; edi = user buffer (in user's memory space)
3764 0000D985 39CA
                                        cmp edx, ecx ; bottom-right > top-left ?
                              <1>
3765 0000D987 0F86A2FAFFFF
                              <1>
                                        jna sysvideo 17 ; full screen copy
3766
                               <1>
3767 0000D98D 0FB7C2
                              <1>
                                        movzx eax, dx; bottom right column
3768 0000D990 6629C8
                                        sub ax, cx ; - top left column
                              <1>
3769 0000D993 0F82C9EDFFFF
                                        jb sysret; invalid
                              <1>
3770 0000D999 6640
                               <1>
                                              ax ; same column no == 1 column
3771 0000D99B 50
                              <1>
                                        push eax; byte count per window row
3772
                              <1>
3773 0000D99C 52
                               <1>
                                        push edx
                           <1>
<1>
<1>
                                              ebx, 16; 320, 640,800; screen width
3774 0000D99D C1EB10
                                        shr
3775 0000D9A0 89C8
                                              eax, ecx
                                        mov
3776 0000D9A2 C1E810
                              <1>
                                        shr
                                              eax, 16
                                                          ; top row
3777 0000D9A5 F7E3
                              <1>
                                        mul
                                              ebx
                            <1>
3778 0000D9A7 6689CA
                                              dx, cx; top left column
                                        mov
                          <1>
<1>
<1>
3779 0000D9AA 01D0
                                        add
                                              eax, edx
3780 0000D9AC 01C6
                                        add
                                              esi, eax ; start address
3781 0000D9AE 59
                                              ecx ; edx
                                        pop
                            <1>
3782 0000D9AF 89C8
                                        mov
                                              eax, ecx
3783 0000D9B1 C1E810
                              <1>
                                        shr
                                              eax, 16; bottom row
                          <1><1><1>
3784 0000D9B4 F7E3
                                        mul
                                              ebx
3785 0000D9B6 6689CA
                                              dx, cx; bottom right column
                                        mov
                                   add
pop
add
cmp
3786 0000D9B9 01D0
                              <1>
                                              eax, edx
3787 0000D9BB 5A
                              <1>
                                              edx ; byte count per window row
eax, 0A0000h
                                              eax, OAFFFFh
                                        jа
                                              sysret
                                        push esi ; start address
3791 0000D9CC 56
                               <1>
3792 0000D9CD 50
                               <1>
                                        push eax ; stop address (included)
3793
                               <1> sysvideo 56:
3794 0000D9CE 89D1
                               <1>
                                        mov ecx, edx; byte count
3795
                               <1>
                                        ; user to system video/display page window transfer
                                        ; esi = user buffer
3796
                               <1>
                                   ; €
cal.
jc
add
add
3797 0000D9D0 E8280E0000
                               <1>
                                        call transfer_to_user_buffer ; fast transfer
3798 0000D9D5 721F
                                        jc short sysvideo_57
                              <1>
3799 0000D9D7 010D[64030300]
                                              [u.r0], ecx
                              <1>
3800 0000D9DD 01DF
                               <1>
                                              edi, ebx ; next row
3801 0000D9DF 01CE
                               <1>
                                              esi, ecx
3802 0000D9E1 3B3C24
                              <1>
                                        cmp
                                              edi, [esp] ; stop addr(included in loop)
3803 0000D9E4 76E8
                               <1>
                                        jna
                                              short sysvideo 56
3804 0000D9E6 5B
                              <1>
                                              ebx ; stop address
                                        pop
3805 0000D9E7 5F
                              <1>
                                        pop
                                              edi ; start address
                                              ebx, edi
3806 0000D9E8 29FB
                               <1>
                                        sub
3807 0000D9EA 43
                               <1>
                                        inc
                                              ebx
3808 0000D9EB 891D[64030300] <1>
                                        mov
                                              [u.r0], ebx
3809 0000D9F1 E96CEDFFFF
                               <1>
                                        jmp
                                              sysret
                               <1> sysvideo 57:
3810
3811 0000D9F6 58
                                    pop eax
                               <1>
3812 0000D9F7 5A
                               <1>
                                              edx
                                        pop
                                        jmp
3813 0000D9F8 E965EDFFFF
                               <1>
                                               sysret
3814
                               <1>
3815
                               <1> sysvideo_58:
                                     cmp bl, 7
3816 0000D9FD 80FB07
                               <1>
3817 0000DA00 770F
                               <1>
                                              short sysvideo_60
                                        jа
                               <1>
3819
                               <1>
                                        ; BL = 7 = AND display page bytes with CL
3820
                               <1>
3821 0000DA02 BE00000A00
                               <1>
                                              esi, 0A0000h
                                        mov
3822 0000DA07 B900000100
                               <1>
                                        mov
                                              ecx, 65536
                               <1> sysvideo 59:
3824 0000DA0C 200E
                               <1>
                                        and byte [esi], cl
3825 0000DA0E 46
                               <1>
                                             esi
3826 0000DA0F E2FB
                                        loop sysvideo_59
                               <1>
3827
                               <1>
                               <1> sysvideo_60:
                                        cmp bl, 8
3829 0000DA11 80FB08
                               <1>
3830 0000DA14 770F
                                              short sysvideo 62
                               <1>
                                        jа
                               <1>
3831
                                        ; BL = 8 = OR display page bytes with CL
3832
                               <1>
3833
                               <1>
3834 0000DA16 BE00000A00
                               <1>
                                        mov esi, 0A0000h
                                        mov ecx, 65536
3835 0000DA1B B900000100
                               <1>
                               <1> sysvideo 61:
```

```
3837 0000DA20 080E
                                          or byte [esi], cl inc esi
                                <1>
3838 0000DA22 46
                                 <1>
                                           loop sysvideo_61
3839 0000DA23 E2FB
                                <1>
3840
                                 <1>
3841
                                 <1> sysvideo 62:
                                           cmp bl, 9
3842 0000DA25 80FB09
                                 <1>
3843 0000DA28 0F8734EDFFFF
                                 <1>
                                                   sysret ; nothing to do
                                           jа
3844
                                 <1>
3845
                                 <1>
                                           ; BL = 9 = XOR display page bytes with CL
3846
                                 <1>
                                           mov esi, 0A0000h
3847 0000DA2E BE00000A00
                                 <1>
3848 0000DA33 B900000100
                                 <1>
                                           mov
                                                 ecx, 65536
                                 <1> sysvideo_63:
3849
3850 0000DA38 300E
                                <1>
                                           xor byte [esi], cl
                                                esi
3851 0000DA3A 46
                                 <1>
                                           inc
3852 0000DA3B E2FB
                                           loop sysvideo 63
                                <1>
                                 <1>
3854
                                 <1> sysvideo 64:
                                          cmp bh, 3
je short sysvideo_68
3855 0000DA3D 80FF03
                                 <1>
3856 0000DA40 7464
                                <1>
3857 0000DA42 80FF04
                                <1>
                                           cmp bh, 4
3858 0000DA45 7721
                                 <1>
                                          ja short sysvideo_65
3859
                                 <1>
                                         ; BH = 4
3860
                                 <1>
                                          ; Direct User Access for CGA video memory.
3861
                                 <1>
3862
                                 <1>
                                           ; Setup user's page tables for direct access to OB8000h.
3863
                                 <1>
3864
                                 <1>
                                          ; Permission checks are not implemented yet !
3865
                                 <1>
                                           ; (11/07/2016)
3866
                                 <1>
3867 0000DA47 B800800B00
                                 <1>
                                           mov eax, 0B8000h
                                           mov ecx, 8; 8 pages (8*4K=32K)
mov ebx, eax; 12/05/2017; virtual = physical
3868 0000DA4C B908000000
                                 <1>
3869 0000DA51 89C3
                                 <1>
3870 0000DA53 E8A67CFFFF
                                 <1>
                                           call direct_memory_access
                                         jc sysret
3871 0000DA58 0F8204EDFFFF
                                 <1>
                                           ; eax = 0B8000h if there is not an error
3872
                                 <1>
3873 0000DA5E A3[64030300]
                                           mov [u.r0], eax
                                 <1>
3874 0000DA63 E9FAECFFFF
                                 <1>
                                           jmp sysret
3875
                                 <1>
3876
                                 <1> sysvideo_65:
3877 0000DA68 80FF05
                                 <1>
                                           cmp bh, 5
3878 0000DA6B 7721
                                 <1>
                                                 short sysvideo 66
3879
                                 <1>
3880
                                 <1>
                                         ; BH = 5
                                         ; Direct User Access for VGA video memory.
3881
                                 <1>
                                           ; Setup user's page tables for direct access to 0A0000h.
3882
                                 <1>
3883
                                 <1>
3884
                                 <1>
                                          ; Permission checks are not implemented yet !
3885
                                 <1>
                                           ; (11/07/2016)
3886
                                 <1>
                                        mov eax, 0A0000h
3887 0000DA6D B800000A00
                                 <1>
                                          mov ecx, 16; 16 pages (16*4K=64K)
mov ebx, eax; 12/05/2017; virtual = physical
3888 0000DA72 B910000000
                                 <1>
3889 0000DA77 89C3
                                 <1>
                                       jc sysret; eax = 0 pc f
3890 0000DA79 E8807CFFFF
                                <1>
                                           call direct_memory_access
3891 0000DA7E 0F82DEECFFFF
                                 <1>
3892
                                 <1>
                                           ; eax = 0A0000h if there is not an error
3893 0000DA84 A3[64030300]
                                 <1>
                                          mov [u.r0], eax
3894 0000DA89 E9D4ECFFFF
                                 <1>
                                           jmp sysret
3895
                                 <1>
3896
                                 <1> sysvideo_66:
3897 0000DA8E 80FF06
                                 <1>
                                           cmp bh, 6
3898 0000DA91 7705
                                 <1>
                                           jа
                                                 short sysvideo 67
                                           ; BH = 6
3899
                                 <1>
3900
                                 <1>
                                         ; Direct User Access for (Super VGA) Linear Frame Buffer.
3901
                                 <1>
                                           ; Setup user's page tables for direct access to LFB.
3902
                                 <1>
                                           ; Not implemented yet !
                                 <1>
3904
                                           ; (11/07/2016)
                                 <1>
3905 0000DA93 E9CAECFFFF
                                 <1>
                                           jmp
                                                 sysret
3906
                                 <1>
3907
                                 <1> sysvideo_67:
                                           cmp bh, 7
3908 0000DA98 80FF07
                                 <1>
3909 0000DA9B 0F87C1ECFFFF
                                                 sysret ; invalid !
                                 <1>
                                           jа
3910
                                 <1>
                                           ; BH = 7
3911
                                 <1>
3912
                                           ; Get (Super/Extended VGA) Linear Frame Buffer info.
                                 <1>
3913
                                 <1>
                                           ; Not implemented yet !
3914
                                 <1>
                                           ; (11/07/2016)
3915
                                 <1>
3916 0000DAA1 E9BCECFFFF
                                 <1>
                                           jmp sysret
3917
                                 <1>
3918
                                  <1> sysvideo_68:
3919
                                 <1>
3920
                                  <1>
                                           ; Super VGA, LINEAR FRAME BUFFER data transfers
3921
                                  <1>
                                           ; Not implemented for yet ! (11/07/2016)
3922 0000DAA6 E9B7ECFFFF
                                 <1>
                                                sysret
3923
                                 <1>
3924
                                 <1> mkdir:
3925
                                 <1>
                                           ; 04/12/2015 (14 byte directory names)
3926
                                 <1>
                                           ; 12/10/2015
                                           ; 17/06/2015 (Retro UNIX 386 v1 - Beginning)
3927
                                 <1>
                                           ; 29/04/2013 - 01/08/2013 (Retro UNIX 8086 v1)
3928
                                  <1>
3929
                                 <1>
3930
                                 <1>
                                           ; 'mkdir' makes a directory entry from the name pointed to
3931
                                  <1>
                                           ; by u.namep into the current directory.
3932
                                 <1>
                                           ; INPUTS ->
3933
                                  <1>
3934
                                 <1>
                                               u.namep - points to a file name
                                                            that is about to be a directory entry.
3935
                                 <1>
                                                ii - current directory's i-number.
3936
                                 <1>
                                           ; OUTPUTS ->
3937
                                 <1>
3938
                                  <1>
                                                u.dirbuf+2 - u.dirbuf+10 - contains file name.
3939
                                 <1>
                                                u.off - points to entry to be filled
3940
                                  <1>
                                                     in the current directory
3941
                                 <1>
                                                u.base - points to start of u.dirbuf.
```

```
3942
                                 <1>
                                               r1 - contains i-number of current directory
3943
                                  <1>
3944
                                 <1>
                                           ; ((AX = R1)) output
3945
                                 <1>
3946
                                 <1>
                                               (Retro UNIX Prototype : 11/11/2012, UNIXCOPY.ASM)
3947
                                 <1>
                                                ((Modified registers: eAX, eDX, eBX, eCX, eSI, eDI, eBP))
3948
                                 <1>
3949
                                 <1>
3950
                                 <1>
                                           ; 17/06/2015 - 32 bit modifications (Retro UNIX 386 v1)
3951 0000DAAB 31C0
                                 <1>
                                           xor eax, eax
3952 0000DAAD BF[9A030300]
                                 <1>
                                           mov
                                                 edi, u.dirbuf+2
3953 0000DAB2 89FE
                                 <1>
                                           mov
                                                 esi, edi
3954 0000DAB4 AB
                                 <1>
                                           stosd
3955 0000DAB5 AB
                                 <1>
                                           stosd
3956
                                 <1>
                                           ; 04/12/2015 (14 byte directory names)
3957 0000DAB6 AB
                                 <1>
                                           stosd
3958 0000DAB7 66AB
                                 <1>
                                           stosw
3959
                                                 ; jsr r0,copyz; u.dirbuf+2; u.dirbuf+10. / clear this
                                 <1>
3960 0000DAB9 89F7
                                 <1>
                                                  edi, esi ; offset to u.dirbuf
                                           ; 12/10/2015 ([u.namep] -> ebp)
3961
                                 <1>
3962
                                 <1>
                                           ;mov ebp, [u.namep]
3963 0000DABB E80D030000
                                           call trans addr nmbp; convert virtual address to physical
                                 <1>
3964
                                 <1>
                                                 ; esi = physical address (page start + offset)
3965
                                 <1>
                                                 ; ecx = byte count in the page (1 - 4096)
3966
                                 <1>
                                           ; edi = offset to u.dirbuf (edi is not modified in trans addr nm)
                                             ; mov u.namep,r2 / r2 points to name of directory entry
3967
                                 <1>
3968
                                 <1>
                                                  ; mov $u.dirbuf+2,r3 / r3 points to u.dirbuf+2
                                 <1> mkdir_1: ; 1:
3969
3970 0000DAC0 45
                                 <1>
                                           inc ebp; 12/10/2015
3971
                                 <1>
3972
                                 <1>
                                           ; / put characters in the directory name in u.dirbuf+2 - u.dirbuf+10
3973
                                 <1>
                                            ; 01/08/2013
3974 0000DAC1 AC
                                 <1>
                                           lodsb
3975
                                 <1>
                                                 ; movb (r2)+,r1 / move character in name to r1
3976 0000DAC2 20C0
                                 <1>
                                           and
                                                al, al
3977 0000DAC4 7427
                                 <1>
                                           jz
                                                 short mkdir_3
                                 <1>
                                                 ; beq 1f / if null, done
3979 0000DAC6 3C2F
                                           cmp al, '/'
                                 <1>
                                                 ; cmp r1,$'/ / is it a "/"?
3980
                                 <1>
                                           je
                                                 short mkdir_err
3981 0000DAC8 7414
                                 <1>
                                           ;je error
3982
                                 <1>
                                                  ; beg error9 / yes, error
3983
                                 <1>
                                           ; 12/10/2015
3984
                                 <1>
3985 0000DACA 6649
                                           dec cx
                                 <1>
3986 0000DACC 7505
                                 <1>
                                           jnz
                                                 short mkdir 2
3987
                                 <1>
                                           ; 12/10/2015 ([u.namep] -> ebp)
3988 0000DACE E800030000
                                 <1>
                                           call trans addr nm; convert virtual address to physical
3989
                                 <1>
                                                 ; esi = physical address (page start + offset)
3990
                                 <1>
                                                  ; ecx = byte count in the page
3991
                                 <1>
                                           ; edi = offset to u.dirbuf (edi is not modified in trans_addr_nm)
3992
                                 <1> mkdir_2:
3993 0000DAD3 81FF[A8030300]
                                 <1>
                                                   edi, u.dirbuf+16;; 04/12/2015 (10 -> 16)
                                                  ; cmp r3,$u.dirbuf+10. / have we reached the last slot for
3994
                                 <1>
3995
                                 <1>
                                                                    ; / a char?
3996 0000DAD9 74E5
                                 <1>
                                           jе
                                                 short mkdir 1
3997
                                 <1>
                                                 ; beq 1b / yes, go back
3998 0000DADB AA
                                 <1>
3999
                                 <1>
                                                 ; movb r1, (r3) + / no, put the char in the u.dirbuf
4000 0000DADC EBE2
                                 <1>
                                                 short mkdir_1
                                                  ; br 1b / get next char
4001
                                 <1>
4002
                                 <1> mkdir_err:
                                 <1>
                                           ; 17/06/2015
4004 0000DADE C705[C8030300]1300- <1>
                                                 dword [u.error], ERR_NOT_DIR ; 'not a valid directory !'
4004 0000DAE6 0000
                                 <1>
4005 0000DAE8 E955ECFFFF
                                 <1>
                                           jmp
                                                 error
4006
                                 <1>
                                 <1> mkdir 3: ; 1:
4008 0000DAED A1[78030300]
                                 <1>
                                           mov eax, [u.dirp]
4009 0000DAF2 A3[80030300]
                                 <1>
                                           mov
                                                 [u.off], eax
                                                 ; mov u.dirp, u.off / pointer to empty current directory
4010
                                 <1>
4011
                                 <1>
                                                               ; / slot to u.off
                                 <1> wdir: ; 29/04/2013
4012
4013 0000DAF7 C705[84030300]-
                                            mov dword [u.base], u.dirbuf
                                 <1>
4013 0000DAFD [98030300]
                                 <1>
                                                 ; mov $u.dirbuf, u.base / u.base points to created file name
                                 <1>
4015 0000DB01 C705[88030300]1000- <1>
                                           mov
                                                   dword [u.count], 16; 04/12/2015 (10 -> 16)
4015 0000DB09 0000
                                 <1>
4016
                                                 ; mov $10., u.count / u.count = 10
                                 <1>
4017 0000DB0B 66A1[51040300]
                                 <1>
                                                 ax, [ii]
                                                  ; mov ii,r1 / r1 has i-number of current directory
4018
                                 <1>
4019 0000DB11 B201
                                 <1>
                                           mov
                                                  dl, 1 ; owner flag mask ; RETRO UNIX 8086 v1 modification !
4020 0000DB13 E8331D0000
                                  <1>
                                           call
                                                  access
                                                  ; jsr r0,access; 1 / get i-node and set its file up
4021
                                 <1>
4022
                                  <1>
                                                              ; / for writing
4023
                                  <1>
                                           ; AX = i-number of current directory
4024
                                 <1>
                                            ; 01/08/2013
4025 0000DB18 FE05[C6030300]
                                                  byte [u.kcall]; the caller is 'mkdir' sign
                                 <1>
                                           inc
4026 0000DB1E E83C0F0000
                                 <1>
                                           call writei
                                                  ; jsr r0, writei / write into directory
                                 <1>
4027
4028 0000DB23 C3
                                 <1>
                                                  ; rts r0
4029
                                 <1>
4030
                                 <1>
                                 <1> sysexec:
4031
                                          ; 18/11/2017
4032
                                 <1>
                                           ; 14/11/2017
4033
                                 <1>
4034
                                 <1>
                                           ; 13/11/2017
                                           ; 24/10/2016, 04/01/2017
4035
                                 <1>
                                           ; 24/04/2016 - TRDOS 386 (TRDOS v2.0)
; 23/06/2015 - 23/10/2015 (Retro UNIX 386 v1)
                                 <1>
4036
4037
                                 <1>
                                           ; 03/06/2013 - 06/12/2013 (Retro UNIX 8086 v1)
4038
                                 <1>
4039
                                 <1>
4040
                                  <1>
                                           ; 'sysexec' initiates execution of a file whose path name if
                                           ; pointed to by 'name' in the sysexec call.
4041
                                 <1>
                                           ; 'sysexec' performs the following operations:
4042
                                 <1>
4043
                                 <1>
                                           ; 1. obtains i-number of file to be executed via 'namei'.
```

```
4044
                                 <1>
                                                2. obtains i-node of file to be exceuted via 'iget'.
4045
                                 <1>
                                                3. sets trap vectors to system routines.
4046
                                 <1>
                                               4. loads arguments to be passed to executing file into
4047
                                 <1>
                                                 highest locations of user's core
4048
                                 <1>
                                               5. puts pointers to arguments in locations immediately
4049
                                 <1>
                                                 following arguments.
                                                6. saves number of arguments in next location.
4050
                                 <1>
                                                7. intializes user's stack area so that all registers
4051
                                 <1>
4052
                                 <1>
                                                 will be zeroed and the PS is cleared and the PC set
                                               to core when 'sysret' restores registers
4053
                                 <1>
4054
                                 <1>
                                                 and does an rti.
4055
                                 <1>
                                               8. inializes u.r0 and u.sp
4056
                                 <1>
                                                9. zeros user's core down to u.r0
4057
                                 <1>
                                           ; 10. reads executable file from storage device into core
                                                starting at location 'core'.
4058
                                 <1>
4059
                                 <1>
                                               11. sets u.break to point to end of user's code with
                                                data area appended.
4060
                                 <1>
                                           ;
                                               12. calls 'sysret' which returns control at location
4061
                                 <1>
4062
                                 <1>
                                                'core' via 'rti' instruction.
4063
                                 <1>
                                          ; Calling sequence:
4064
                                 <1>
4065
                                 <1>
                                          ;
                                                 sysexec; namep; argp
4066
                                 <1>
                                           ; Arguments:
4067
                                 <1>
                                                 namep - points to pathname of file to be executed
                                                 argp - address of table of argument pointers
4068
                                 <1>
                                                 argp1... argpn - table of argument pointers
argp1:<...0> ... argpn:<...0> - argument strings
4069
                                 <1>
4070
                                 <1>
                                          ; Inputs: (arguments)
4071
                                 <1>
4072
                                 <1>
                                           ; Outputs: -
4073
                                 <1>
                                           ; .......
4074
                                 <1>
4075
                                 <1>
                                           ; Retro UNIX 386 v1 modification:
                                                 User application runs in it's own virtual space
4076
                                 <1>
4077
                                 <1>
                                                  which is izolated from kernel memory (and other
4078
                                 <1>
                                                 memory pages) via 80386 paging in ring 3
                                                  privilige mode. Virtual start address is always 0.
4079
                                 <1>
4080
                                 <1>
                                                  User's core memory starts at linear address 400000h
4081
                                 <1>
                                                 (the end of the 1st 4MB).
4082
                                 <1>
4083
                                 <1>
                                           ; Retro UNIX 8086 v1 modification:
4084
                                 <1>
                                                 user/application segment and system/kernel segment
4085
                                 <1>
                                                  are different and sysenter/sysret/sysrele routines
                                                  are different (user's registers are saved to
4086
                                 <1>
4087
                                 <1>
                                                 and then restored from system's stack.)
4088
                                 <1>
                                                  NOTE: Retro UNIX 8086 v1 'arg2' routine gets these
4089
                                 <1>
                                                       arguments which were in these registers;
4090
                                 <1>
4091
                                 <1>
                                                       but, it returns by putting the 1st argument
4092
                                 <1>
                                                       in 'u.namep' and the 2nd argument
4093
                                 <1>
                                                       on top of stack. (1st argument is offset of the
4094
                                 <1>
                                                       file/path name in the user's program segment.)
4095
                                 <1>
4096
                                 <1>
                                           ;call arg2
4097
                                 <1>
                                           ; * name - 'u.namep' points to address of file/path name
4098
                                 <1>
                                                     in the user's program segment ('u.segmnt')
4099
                                 <1>
                                                      with offset in BX register (as sysopen argument 1).
4100
                                 <1>
                                           ; * argp - sysexec argument 2 is in CX register
4101
                                 <1>
                                                     which is on top of stack.
4102
                                 <1>
4103
                                                  ; jsr r0,arg2 / arg0 in u.namep,arg1 on top of stack
                                 <1>
4104
                                 <1>
4105
                                 <1>
                                           ; 23/06/2015 (32 bit modifications)
                                 <1>
4106
4107
                                 <1>
                                           ;; 13/11/2017
4108
                                 <1>
                                           ;;mov [u.namep], ebx; argument 1
4109
                                 <1>
                                            ; 18/10/2015
4110 0000DB24 890D[4C040300]
                                           mov
                                 <1>
                                                 [argv], ecx ; * ; argument 2
4111
                                 <1>
4112
                                 <1>
                                           ; 13/11/2017
4113 0000DB2A 89DE
                                 <1>
                                           mov esi, ebx
4114 0000DB2C E84E210000
                                 <1>
                                           call set_working_path_x
4115 0000DB31 7319
                                 <1>
                                           jnc short sysexec_0
4116
                                 <1>
4117
                                 <1>
                                           ;; 'bad command or file name'
4118
                                 <1>
                                           ;mov eax, ERR_BAD_CMD_ARG ; 01h ; TRDOS 8086
4119
                                 <1>
4120
                                 <1>
                                           ; 'file not found !' error
4121 0000DB33 B802000000
                                           mov eax, ERR NOT FOUND; 02h; TRDOS 8086
                                 <1>
4122
                                 <1> sysexec_not_found_err:
4123
                                 <1> sysexec access error:
4124
                                 <1> sysexec_ext_error:
4125 0000DB38 A3[64030300]
                                  <1>
                                           mov [u.r0], eax
4126 0000DB3D A3[C8030300]
                                 <1>
                                                  [u.error], eax
                                           mov
4127 0000DB42 E80D220000
                                 <1>
                                           call reset_working_path
4128 0000DB47 E9F6EBFFFF
                                 <1>
                                           jmp
                                                 error
4129
                                 <1>
4130
                                 <1> sysexec 0:
                                          ; 13/11/2017
4131
                                 <1>
                                           ;mov esi, FindFile Name
4132
                                 <1>
4133 0000DB4C 66B80018
                                           mov ax, 1800h; Only files
                                 <1>
4134 0000DB50 E892A7FFFF
                                 <1>
                                           call find_first_file
4135 0000DB55 72E1
                                 <1>
                                          jc short sysexec_not_found_err ; eax = 2
                                 <1>
4136
4137
                                 <1>
                                          ; check file attributes
4138
                                 <1>
                                           ; (attribute bits = 00ADVSHR) ; 18h = Directory+Volume
4139
                                 <1>
                                           ; BL = Attributes byte
4140
                                 <1>
4141 0000DB57 F6C306
                                            test bl, 6 ; system file or hidden file (S+H)
                                 <1>
4142
                                 <1>
                                           ;jz short sysexec_0ext
4143 0000DB5A 7417
                                 <1>
                                                 short sysexec_1 ; yes
4144
                                 <1>
4145
                                 <1>
                                           ; 13/11/2017
4146
                                 <1>
                                           ; /// TRDOS386 permission check for multiuser mode ///
4147
                                 <1>
                                           ; SYSTEM file or HIDDEN file !!
                                 <1>
4148
                                           ; (Only super user has permission to run this file.)
```

```
4149
                                 <1>
4150
                                 <1>
                                          ; ([u.uid]=0 for super user or root in multiuser mode)
4151
                                          ; ([u.uid]=0 for any users in singleuser mode)
                                 <1>
4152 0000DB5C 803D[B0030300]00
                                 <1>
                                          cmp byte [u.uid], 0 ; Super User ([u.uid]=0) ?
                                 <1>
                                          ;jna short sysexec_0ext
4153
4154 0000DB63 760E
                                 <1>
                                               short sysexec_1 ; yes
                                          jna
4155
                                 <1>
4156
                                          ; 'permission denied !' error
                                 <1>
4157 0000DB65 B80B000000
                                 <1>
                                            mov eax, ERR_FILE_ACCESS ; 11 = ERR_PERM_DENIED
4158 0000DB6A EBCC
                                 <1>
                                            jmp short sysexec_access_error
4159
                                 <1>
4160
                                 <1> sysexec_not_exf:
                                       ; 'not executable file !' error
4161
                                <1>
4162 0000DB6C B816000000
                                <1>
                                          mov eax, ERR_NOT_EXECUTABLE
4163 0000DB71 EBC5
                                 <1>
                                          jmp
                                                sysexec_ext_error
4164
                                 <1>
4165
                                 <1> ;sysexec 0ext:
                                 <1> sysexec_1:
4166
4167
                                 <1>
                                          ; 18/11/2017
                                          mov esi, FindFile_Name
4168 0000DB73 BE[68620100]
                                <1>
4169
                                <1>
                                        ; 13/11/2017
                                          ; check program file name extension
                                 <1>
                                          ; ('.PRG' for current TRDOS version)
4171
                                <1>
4172 0000DB78 E80DC2FFFF
                                <1>
                                          call check_prg_filename_ext
4173 0000DB7D 72ED
                                <1>
                                          jc short sysexec_not_exf
4174
                                <1>
4175
                                <1>
                                          ; 18/11/2017
4176 0000DB7F 3C50
                                          cmp al, 'P'
                                <1>
4177 0000DB81 75E9
                                 <1>
                                          jne
                                                short sysexec_not_exf
4178
                                 <1>
4179
                                 <1>
                                          ; '.PRG' extension is OK.
4180
                                 <1>
                                          ; Only '.PRG' files are valid program files
                                          ; for current TRDOS 386 version.
4181
                                 <1>
4182
                                 <1>
                                                 edx, [FindFile DirEntry+DirEntry FileSize]
4183 0000DB83 8B15[94620100]
                                 <1>
                                          mov
4184 0000DB89 66A1[8C620100]
                                 <1>
                                          mov
                                                 ax, [FindFile_DirEntry+DirEntry_FstClusHI]
4185 0000DB8F C1E010
                                 <1>
                                                eax, 16
4186 0000DB92 66A1[92620100]
                                 <1>
                                                ax, [FindFile_DirEntry+DirEntry_FstClusLO]
                                          mov
                                          ; EAX = First Cluster number
4187
                                 <1>
                                          ; EDX = File Size
4188
                                 <1>
4189
                                 <1>
4190 0000DB98 A3[51040300]
                                 <1>
                                          mov
                                                 [ii], eax
4191 0000DB9D 8915[55040300]
                                 <1>
                                                [i.size], edx
                                          mov
4192
                                 <1>
                                 <1> ; sysexec_1:
4193
                                       ; \overline{13}/11/2017 - TRDOS 386 (TRDOS v2.0)
4194
                                 <1>
                                          ; 24/06/2015 - 23/10/2015 (Retro UNIX 386 v1)
4195
                                 <1>
4196
                                 <1>
                                          ; Moving arguments to the end of [u.upage]
4197
                                 <1>
                                          ; (by regarding page borders in user's memory space)
                                 <1>
4198
                                          ; 10/10/2015
4199
                                 <1>
4200
                                 <1>
                                          ; 21/07/2015
                                          mov ebp, esp; (**)
4201 0000DBA3 89E5
                                 <1>
4202
                                <1>
                                          ; 18/10/2015
4203 0000DBA5 89EF
                                 <1>
                                          mov edi, ebp
4204 0000DBA7 B900010000
                                <1>
                                          mov
                                                ecx, MAX_ARG_LEN; 256
                                          ; sub edi, MAX ARG LEN ; 256
                                 <1>
4206 0000DBAC 29CF
                                <1>
                                          sub edi, ecx
4207 0000DBAE 89FC
                                <1>
                                                esp, edi ; *!*
                                          mov
                                          xor eax, eax
4208 0000DBB0 31C0
                                <1>
4209 0000DBB2 A3[8C030300]
                                <1> mov
<1> mov
<1> dec
                                                [u.nread], eax ; 0
4210 0000DBB7 66A3[4A040300]
                                                [argc], ax ; 0 ; 13/11/2017
4211 0000DBBD 49
                                                 ecx ; 256 - 1
                                          dec
4212 0000DBBE 890D[88030300]
                                <1>
                                                [u.count], ecx; MAX_ARG_LEN - 1; 255
                                          mov
4213
                                 <1>
                                          ;mov dword [u.count], MAX_ARG_LEN - 1; 255
4214
                                 <1> sysexec_2:
                                 <1> mov
4215 0000DBC4 8B35[4C040300]
                                                esi, [argv] ; 18/10/2015
                                          call get_argp
4216 0000DBCA E866000000
                                <1>
4217 0000DBCF B90400000
                                <1>
                                                 ecx, 4 ; mov ecx, 4
4218
                                 <1> sysexec_3:
4219 0000DBD4 21C0
                                <1> and
                                                 eax, eax
4220 0000DBD6 0F8429050000
                                 <1>
                                          jz
                                                    sysexec_6
                                <1>
                                          ; 18/10/2015
4221
                                       add [argv], ecx; 4
4222 0000DBDC 010D[4C040300]
                                 <1>
                                        inc
                                                word [argc]
4223 0000DBE2 66FF05[4A040300]
                                 <1>
4224
                                 <1>
                                        mov
4225 0000DBE9 A3[84030300]
                                 <1>
                                                 [u.base], eax
                                        ; 23/10/2015
4226
                                 <1>
4227 0000DBEE 66C705[C4030300]00- <1>
                                         mov word [u.pcount], 0
4227 0000DBF6 00
                                 <1>
4228
                                 <1> sysexec 4:
                                                 cpass ; get a character from user's core memory
4229 0000DBF7 E8A10B0000
                                 <1>
                                          call
4230 0000DBFC 750E
                                             jnz short sysexec_5
                                 <1>
4231
                                 <1>
                                               ; (max. 255 chars + null)
4232
                                 <1>
                                          ; 18/10/2015
4233 0000DBFE 28C0
                                 <1>
                                          sub al, al
4234 0000DC00 AA
                                 <1>
                                          stosb
4235 0000DC01 FF05[8C030300]
                                <1>
                                          inc dword [u.nread]
4236 0000DC07 E9F9040000
                                <1>
                                                 sysexec_6 ; 24/04/2016
                                          qmŗ
                                <1> sysexec 5:
4238 0000DC0C AA
                                <1>
                                          stosb
4239 0000DC0D 20C0
                                          and al, al
                                 <1>
4240 0000DC0F 75E6
                                <1>
                                                short sysexec 4
                                          jnz
4241 0000DC11 B904000000
                                <1>
                                                 ecx, 4
                                          mov
                                                [ncount], ecx ; 4
4242 0000DC16 390D[48040300]
                                <1>
                                          cmp
4243 0000DC1C 72A6
                                                 short sysexec 2
                                <1>
                                          jb
                                        mov
4244 0000DC1E 8B35[44040300]
                                                 esi, [nbase]
                                 <1>
4245 0000DC24 010D[44040300]
                                        add
                                 <1>
                                                [nbase], ecx ; 4
4246 0000DC2A 66290D[48040300]
                                 <1>
                                          sub
                                                 [ncount], cx
4247 0000DC31 8B06
                                 <1>
                                          mov
                                                eax, [esi]
4248 0000DC33 EB9F
                                 <1>
                                          jmp
                                                 short sysexec_3
4249
                                 <1>
4250
                                 <1> get_argp:
                                         ; 14/11/2017 - TRDOS 386 (TRDOS v2.0)
4251
                                 <1>
4252
                                 <1>
                                          ; 18/10/2015 (nbase, ncount)
```

```
; 21/07/2015
4253
                                 <1>
4254
                                 <1>
                                          ; 24/06/2015 (Retro UNIX 386 v1)
4255
                                 <1>
                                          ; Get (virtual) address of argument from user's core memory
4256
                                 <1>
4257
                                 <1>
                                          ; INPUT:
4258
                                 <1>
                                                esi = virtual address of argument pointer
                                 <1>
                                                eax = virtual address of argument
4260
4261
                                 <1>
4262
                                 <1>
                                          ; Modified registers: EAX, EBX, ECX, EDX, ESI
4263
                                 <1>
4264 0000DC35 833D[BC030300]00
                                                  dword [u.ppgdir], 0 ; /etc/init ?
                                 <1>
                                          cmp
4265
                                <1>
                                                                ; (the caller is kernel)
4266 0000DC3C 7667
                                                    short get_argpk
                                <1>
                                <1>
4267
                                          ;
4268 0000DC3E 89F3
                                                ebx, esi
                                <1>
                                          mov
4269 0000DC40 E8A776FFFF
                                <1>
                                          call get_physical_addr ; get physical address
4270 0000DC45 0F8289000000
                                <1>
                                          jс
                                                  get_argp_err
4271 0000DC4B A3[44040300]
                                <1>
                                                 [nbase], eax ; physical address
4272 0000DC50 66890D[48040300]
                               <1>
                                                [ncount], cx; remain byte count in page (1-4096)
                                          mov
4273 0000DC57 B804000000
                                <1>
                                          mov
                                                eax, 4 ; 21/07/2015
4274 0000DC5C 6639C1
                                <1>
                                          cmp
                                                cx, ax ; 4
4275 0000DC5F 735D
                                <1>
                                          jnb
                                                short get_argp2
4276 0000DC61 89F3
                                <1>
                                          mov
                                                ebx, esi
4277 0000DC63 01CB
                                <1>
                                          add
                                                ebx, ecx
4278 0000DC65 E88276FFFF
                                <1>
                                          call get_physical_addr ; get physical address
4279 0000DC6A 7268
                                <1>
                                          jс
                                                short get_argp_err
                                        ;push esi
4280
                                <1>
4281 0000DC6C 89C6
                                <1>
                                          mov esi, eax
4282 0000DC6E 66870D[48040300]
                                          xchg cx, [ncount]
                               <1>
4283 0000DC75 8735[44040300]
                                <1>
                                          xchg esi, [nbase]
4284 0000DC7B B504
                                <1>
                                          mov
                                                ch, 4
4285 0000DC7D 28CD
                                <1>
                                          sub
                                                ch, cl
4286
                                <1> get_argp0:
4287 0000DC7F AC
                                <1>
                                          lodsb
4288 0000DC80 6650
                                <1>
                                          push ax
4289 0000DC82 FEC9
                                <1>
4290 0000DC84 75F9
                                <1>
                                         jnz short get_argp0
4291 0000DC86 8B35[44040300]
                                         mov esi, [nbase]
                                <1>
                                         ; 21/07/2015
                                <1>
4293 0000DC8C 0FB6C5
                                <1>
                                         movzx eax, ch
4294 0000DC8F 0105[44040300]
                                <1>
                                          add [nbase], eax
                                       sub
4295 0000DC95 662905[48040300] <1>
                                                [ncount], ax
4296
                                <1> get_argp1:
4297 0000DC9C AC
                                <1>
                                         lodsb
4298 0000DC9D FECD
                                <1>
                                          dec ch
                                          jz
4299 0000DC9F 7447
                                <1>
                                                short get_argp3
4300 0000DCA1 6650
                                           push ax
                                <1>
4301 0000DCA3 EBF7
                                <1>
                                          jmp
                                                 short get_argp1
                                <1> get_argpk:
4303
                                <1>
                                          ; Argument is in kernel's memory space
4304 0000DCA5 66C705[48040300]00- <1>
                                          mov
                                                word [ncount], PAGE_SIZE; 4096
4304 0000DCAD 10
                               <1>
4305 0000DCAE 8935[44040300]
                                <1>
                                                [nbase], esi
                                          mov
4306 0000DCB4 8305[44040300]04
                                <1>
                                          add
                                                dword [nbase], 4
4307 0000DCBB 8B06
                                <1>
                                          mov
                                                eax, [esi] ; virtual addr. = physcal addr.
4308 0000DCBD C3
                                 <1>
                                         retn
4309
                                <1> get_argp2:
                                      ; 21/07/2015
4310
                                 <1>
4311
                                <1>
                                          ;mov eax, 4
4312 0000DCBE 8B15[44040300]
                                <1>
                                          mov
                                                edx, [nbase] ; 18/10/2015
4313 0000DCC4 0105[44040300]
                                <1>
                                          add
                                                [nbase], eax
4314 0000DCCA 662905[48040300]
                                <1>
                                                [ncount], ax
                                          sub
4315
                                <1>
4316 0000DCD1 8B02
                                <1>
                                          mov
                                                eax, [edx]
4317 0000DCD3 C3
                                <1>
                                          retn
                                <1> get_argp_err:
4319 0000DCD4 A3[C8030300]
                                      mov [u.error], eax
                                <1>
4320
                                <1>
                                          ; 14/11/2017
4321 0000DCD9 B801000000
                                <1>
                                               eax, ERR_BAD_CMD_ARG ; 01h ; TRDOS 8086
                                          mov
4322 0000DCDE A3[64030300]
                                <1>
                                          mov
                                                [u.r0], eax
4323 0000DCE3 E95AEAFFFF
                                <1>
                                          jmp
                                                error
                                <1> get_argp3:
4324
4325 0000DCE8 B103
                                <1>
                                                cl, 3
                                <1> get_argp4:
4326
                                          shl
4327 0000DCEA C1E008
                                <1>
                                                 eax, 8
4328 0000DCED 665A
                                <1>
                                          pop
                                                dx
4329 0000DCEF 88D0
                                                al, dl
                                <1>
                                          mov
4330 0000DCF1 E2F7
                                          loop get_argp4
                                <1>
4331
                                <1>
                                          ;pop esi
4332 0000DCF3 C3
                                 <1>
                                          retn
                                 <1>
4334
                                 <1> sysstat:
                                         ; 13/01/2017 - TRDOS 386 (TRDOS v2.0)
4335
                                 <1>
4336
                                          ; temporary !
                                          mov eax, ERR_INV_FNUMBER; 'invalid function number!'
4337 0000DCF4 B801000000
                                 <1>
4338 0000DCF9 A3[C8030300]
                                 <1>
                                          mov
                                                 [u.error], eax
                                <1>
                                                   [u.r0], eax
4339 0000DCFE A3[64030300]
                                           mov
4340 0000DD03 E93AEAFFFF
                                 <1>
                                               error
                                          jmp
4341
                                 <1>
4342
                                 <1> sysfstat:
                                          ; 13/01/2017 - TRDOS 386 (TRDOS v2.0)
4343
                                 <1>
4344
                                 <1>
                                          ; temporary !
4345 0000DD08 B801000000
                                          mov eax, ERR_INV_FNUMBER; 'invalid function number!'
                                 <1>
4346 0000DD0D A3[C8030300]
                                 <1>
                                                   [u.error], eax
                                           mov
4347 0000DD12 A3[64030300]
                                <1>
                                            mov
                                                    [u.r0], eax
4348 0000DD17 E926EAFFFF
                                 <1>
                                          jmp error
4349
                                 <1>
4350
                                 <1> fclose:
                                         ; 06/10/2016 (TRDOS 386 = TRDOS v2.0)
4351
                                 <1>
4352
                                 <1>
4353
                                 <1>
                                          ; 18/06/2015 (Retro UNIX 386 v1 - Beginning)
4354
                                 <1>
                                                       (32 bit offset pointer modification)
4355
                                 <1>
                                          ; 19/04/2013 - 12/01/2014 (Retro UNIX 8086 v1)
4356
                                 <1>
```

```
4357
                                 <1>
                                           ; Given the file descriptor (index to the u.fp list)
4358
                                 <1>
                                           ; 'fclose' first gets the i-number of the file via 'getf'.
4359
                                 <1>
                                           ; If i-node is active (i-number > 0) the entry in
4360
                                 <1>
                                           ; u.fp list is cleared. If all the processes that opened
4361
                                 <1>
                                           ; that file close it, then fsp etry is freed and the file
4362
                                 <1>
                                           ; is closed. If not a return is taken.
4363
                                           ; If the file has been deleted while open, 'anyi' is called
4364
                                 <1>
                                           ; to see anyone else has it open, i.e., see if it is appears
4365
                                 <1>
                                           ; in another entry in the fsp table. Upon return from 'anyi'
                                           ; a check is made to see if the file is special.
4366
                                 <1>
4367
                                 <1>
                                           ; INPUTS ->
4368
                                 <1>
                                           ; r1 - contains the file descriptor (value=0,1,2...)
4369
                                 <1>
                                               u.fp - list of entries in the fsp table
4370
                                 <1>
4371
                                 <1>
                                               fsp - table of entries (4 words/entry) of open files.
                                           ; OUTPUTS ->
4372
                                 <1>
4373
                                 <1>
                                           ; r1 - contains the same file descriptor
                                               r2 - contains i-number
4374
                                 <1>
4375
                                 <1>
4376
                                 <1>
                                           ; ((AX = R1))
                                           ; ((Modified registers: eDX, eBX, eCX, eSI, eDI, eBP))
4377
                                 <1>
4378
                                 <1>
                                           ; Retro UNIX 8086 v1 modification : CF = 1
4379
                                 <1>
4380
                                 <1>
                                                          if i-number of the file is 0. (error)
4381
                                 <1>
                                           ; TRDOS 386 (06/10/2016)
4382
                                 <1>
4383
                                 <1>
                                          ; INPUT:
4384
                                 <1>
4385
                                 <1>
                                                 EAX = File Handle (File Descriptor, File Index)
                                           ;
4386
                                 <1>
4387
                                 <1>
                                           ; OUTPUT:
                                                 CF = 1 \rightarrow File not open !
4388
                                 <1>
                                                  CF = 0 \rightarrow OK!
4389
                                 <1>
4390
                                 <1>
                                                      EBX = File Number (System)
                                                      [cdev] = Logical DOS Drive Number
4391
                                 <1>
4392
                                 <1>
                                                      EAX = File Handle/Number (user)
4393
                                 <1>
4394
                                           ; Modified Registers: EBX
                                 <1>
4395
                                 <1>
4396 0000DD1C 50
                                           push eax ; File handle
                                 <1>
4397
                                 <1>
4398 0000DD1D E846000000
                                 <1>
                                           call getf
4399 0000DD22 0F8205240000
                                                  device_close ; eax = device number
                                 <1>
                                           jс
                                 <1>
4400
4401 0000DD28 80BB[E6680100]01
                                                 byte [ebx+OF\_MODE], 1; open mode; 0 = empty entry
                                 <1>
                                           cmp
4402 0000DD2F 722E
                                                  short fclose 1
                                                                           ; 1 = \text{read}, 2 = \text{write}
                                 <1>
                                           jb
4403
                                 <1>
4404 0000DD31 83F801
                                 <1>
                                                  eax, 1; is the first cluster number > 0
                                           cmp
4405 0000DD34 7229
                                 <1>
                                                  short fclose_1; no, this is empty entry
4406
                                 <1>
4407
                                 <1> fclose_0:
4408 0000DD36 FE8B[FA680100]
                                 <1>
                                           dec
                                                  byte [ebx+OF_OPENCOUNT] ; decrement the number of processes
4409
                                 <1>
                                                                       ; that have opened the file
4410 0000DD3C 7921
                                 <1>
                                                  short fclose_1 ; jump if not negative (jump if bit 7 is 0)
4411
                                 <1>
                                                       ; if all processes haven't closed the file, return
4412
                                 <1>
                                 <1>
                                           ; eax ; First cluster
4414 0000DD3E 31C0
                                 <1>
                                           xor eax, eax; 0
4415 0000DD40 8883[E6680100]
                                 <1>
                                                  [ebx+OF_MODE], al ; 0 = \text{empty entry}
                                           ;mov [ebx+OF_STATUS], al ; 0 = empty entry
4416
                                 <1>
4417 0000DD46 66C1E302
                                 <1>
                                          shl bx, 2
4418 0000DD4A 8983[B4680100]
                                 <1>
                                                 [ebx+OF_FCLUSTER], eax ; 0
                                           mov
                                                  [ebx+OF_CCLUSTER], eax; 0
4419 0000DD50 8983[CC690100]
                                 <1>
                                           mov
                                           ;mov [ebx+OF_CCINDEX], eax ; 0
                                 <1>
4421 0000DD56 A3[74030300]
                                 <1>
                                           mov
                                                 [u.fofp], eax; 0
4422 0000DD5B 66C1EB02
                                 <1>
                                           shr
                                                 bx, 2
                                 <1> fclose 1: ; 1:
4424 0000DD5F 58
                                                 eax ; File handle (File Descriptor, File Index)
                                 <1>
                                           pop
4425 0000DD60 C680[6A030300]00
                                 <1>
                                                 byte [eax+u.fp], 0 ; clear that entry in the u.fp list
4426 0000DD67 C3
                                 <1>
                                           retn
4427
                                 <1>
4428
                                 <1> getf:
                                           ; 12/10/2016
4429
                                 <1>
4430
                                 <1>
                                           ; 11/10/2016
4431
                                 <1>
                                           ; 08/10/2016
4432
                                 <1>
                                           ; 06/10/2016 (TRDOS 386 = TRDOS v2.0)
4433
                                 <1>
                                           ; / get the device number and the i-number of an open file
4434
                                 <1>
                                           ; 13/05/2015
                                           ; 11/05/2015 (Retro UNIX 386 v1 - Beginning)
4435
                                 <1>
                                           ; 19/04/2013 - 18/11/2013 (Retro UNIX 8086 v1)
4436
                                 <1>
4437
                                 <1>
4438 0000DD68 89C3
                                                  ebx, eax
                                  <1>
                                           mov
4439
                                 <1> getf1:
4440 0000DD6A 83FB0A
                                         cmp ebx, 10
                                 <1>
4441 0000DD6D 730A
                                 <1>
                                           jnb short getf2
4442 0000DD6F 8A9B[6A030300]
                                 <1>
                                           mov bl, [ebx+u.fp]
4443 0000DD75 08DB
                                <1>
                                         or
                                                 bl, bl
4444 0000DD77 7503
                                 <1>
                                          jnz short getf3
4445
                                 <1> getf2:
4446
                                 <1>
                                         ; 'File not open !' error (ax=0)
4447 0000DD79 29C0
                                 <1>
                                           sub
                                                 eax, eax
4448 0000DD7B C3
                                 <1>
                                           retn
                                 <1> getf3:
4449
4450 0000DD7C F6C380
                                <1>
                                                 bl, 80h
                                         test
4451 0000DD7F 7530
                                 <1>
                                           jnz
                                                 short getf5 ; device
4452 0000DD81 FECB
                                 <1>
                                           dec
                                                 bl ; 0 based
                                                 al, [ebx+OF DRIVE]
4453 0000DD83 8A83[DC680100]
                                <1>
                                           mov
4454 0000DD89 A2[46030300]
                                                 [cdev], al
                                <1>
                                          mov
4455 0000DD8E C0E302
                                 <1>
                                          shl
                                                 bl, 2; *4 (dword offset)
                                        mov
4456 0000DD91 8B83[2C690100]
                                <1>
                                                 eax, [ebx+OF SIZE]
                                <1> mov
<1> lea
<1> mov
4457 0000DD97 A3[55040300]
                                                 [i.size], eax ; file size
4458 0000DD9C 8D83[04690100]
                                                 eax, [ebx+OF POINTER] ;12/10/2016
4459 0000DDA2 A3[74030300]
                                          mov
                                                 [u.fofp], eax
                                 <1>
                                                  eax, [ebx+OF FCLUSTER]
4460 0000DDA7 8B83[B4680100]
                                           mov
4461 0000DDAD C0EB02
                                 <1>
                                           shr
                                                 bl, 2; /4 (byte offset)
```

```
4462
                                  <1> getf4:
4463 0000DDB0 C3
                                  <1>
                                           retn
                                  <1> getf5:
4464
4465
                                  <1>
                                          ; get device number
                                            and bl, 7Fh; 1 to 7Fh
4466 0000DDB1 80E37F
                                  <1>
                                                   bl ; 0 based (0 to 7Eh)
4467 0000DDB4 FECB
                                  <1>
                                            dec
                                 c1> dec b1; 0 based (0 to /En,
c1> mov al, [ebx+DEV_DRIVER]
c1> mov ch, [ebx+DEV_ACCESS]
c1> mov cl, [ebx+DEV_OPENMODE]
c1> and ch, 0FEh; reset bit 0; dev_close
4468 0000DDB6 8A83[0E670100]
4469 0000DDBC 8AAB[78660100]
4470 0000DDC2 8A8B[2C670100]
4471 0000DDC8 80E5FE
                                        stc ; cf = 1
4472 0000DDCB F9
                                  <1>
4473 0000DDCC C3
                                  <1>
                                            retn
4474
                                  <1>
4475
                                  <1> trans_addr_nmbp:
                                        ; 18/10/2015
4476
                                  <1>
4477
                                  <1>
                                            ; 12/10/2015
4478 0000DDCD 8B2D[7C030300]
                                  <1>
                                          mov ebp, [u.namep]
                                  <1> trans_addr_nm:
4479
                                         ; Convert virtual (pathname) address to physical address
4480
                                  <1>
4481
                                  <1>
                                            ; (Retro UNIX 386 v1 feature only !)
4482
                                  <1>
                                          ; 18/10/2015
                                          ; 12/10/2015 (u.pnbase & u.pncount has been removed from code) ; 02/07/2015
4483
                                   <1>
4484
                                  <1>
4485
                                  <1>
                                          ; 17/06/2015
                                  <1>
                                           ; 16/06/2015
4486
4487
                                  <1>
                                            ; INPUTS:
4488
                                  <1>
4489
                                  <1>
                                                   ebp = pathname address (virtual) ; [u.namep]
                                          ;
4490
                                  <1>
                                                    [u.pgdir] = user's page directory
                                            ; OUTPUT:
4491
                                  <1>
4492
                                  <1>
                                                   esi = physical address of the pathname
4493
                                   <1>
                                                   ecx = remain byte count in the page
                                            ;
4494
                                  <1>
                                            ;
4495
                                  <1>
                                            ; (Modified registers: EAX, EBX, ECX, EDX, ESI)
4496
                                  <1>
                                          ;
4497 0000DDD3 833D[BC030300]00
                                  <1>
                                                      dword [u.ppgdir], 0 ; /etc/init ? (sysexec)
                                            jna short trans_addr_nmk; the caller is os kernel;
4498 0000DDDA 7618
                                  <1>
4499
                                  <1>
                                                                     ; it is already physical address
4500 0000DDDC 50
                                  <1>
4501 0000DDDD 89EB
                                            mov ebx, ebp ; [u.namep] ; pathname address (virtual)
                                 <1>
4502 0000DDDF E80875FFFF
                                                   call get_physical_addr ; get physical address
                                  <1>
                                           jс
4503 0000DDE4 7204
                                  <1>
                                                  short tr_addr_nm_err
                                            ; 18/10/2015
4504
                                  <1>
4505
                                  <1>
                                          ; eax = physical address
4506
                                  <1>
                                            ; cx = remain byte count in page (1-4096)
4507
                                  <1>
                                                  ; 12/10/2015 (cx = [u.pncount])
                                                  esi, eax ; 12/10/2015 (esi=[u.pnbase])
4508 0000DDE6 89C6
                                  <1>
4509 0000DDE8 58
                                  <1>
                                            pop eax
4510 0000DDE9 C3
                                  <1>
                                            retn
4511
                                  <1>
4512
                                  <1> tr_addr_nm_err:
                                         mov [u.error], eax;pop eax
4513 0000DDEA A3[C8030300]
                                  <1>
4514
                                  <1>
4515 0000DDEF E94EE9FFFF
                                  <1>
                                            jmp error
4516
                                  <1>
4517
                                  <1> trans_addr_nmk:
                                  <1> ; 12/10/2015
4519
                                  <1>
                                            ; 02/07/2015
                                          mov esi, [u.namep] ; [u.pnbase]
mov cx, PAGE_SIZE ; 4096 ; [u.pncount]
4520 0000DDF4 8B35[7C030300]
                                  <1>
4521 0000DDFA 66B90010
                                  <1>
4522 0000DDFE C3
                                  <1>
                                            retn
4523
                                  <1>
4524
                                  <1>
4525
                                  <1> sysbreak:
                                        ; 18/10/2015
4526
                                  <1>
4527
                                  <1>
                                            ; 07/10/2015
                                          ; 23/06/2015 (Retro UNIX 386 v1 - Beginning)
4528
                                  <1>
                                           ; 20/06/2013 - 24/03/2014 (Retro UNIX 8086 v1)
                                  <1>
4529
4530
                                  <1>
                                            ; 'sysbreak' sets the programs break points.
4531
                                  <1>
                                            ; It checks the current break point (u.break) to see if it is
4532
                                  <1>
4533
                                   <1>
                                            ; between "core" and the stack (sp). If it is, it is made an
                                            ; even address (if it was odd) and the area between u.break
4534
                                  <1>
                                           ; and the stack is cleared. The new breakpoint is then put
4535
                                  <1>
                                           ; in u.break and control is passed to 'sysret'.
4536
                                  <1>
4537
                                  <1>
4538
                                   <1>
                                            ; Calling sequence:
4539
                                  <1>
                                          ; sysbreak; addr
4540
                                   <1>
                                            ; Arguments: .
4541
                                  <1>
                                             ; Inputs: u.break - current breakpoint
4542
                                   <1>
4543
                                   <1>
                                             ; Outputs: u.break - new breakpoint
4544
                                   <1>
                                                   area between old u.break and the stack (sp) is cleared.
4545
                                   <1>
4546
                                   <1>
                                             ; Retro UNIX 8086 v1 modification:
4547
                                   <1>
4548
                                   <1>
                                                   The user/application program puts breakpoint address
                                                    in BX register as 'sysbreak' system call argument.
4549
                                  <1>
4550
                                   <1>
                                                    (argument transfer method 1)
4551
                                   <1>
4552
                                  <1>
                                            ; NOTE: Beginning of core is 0 in Retro UNIX 8086 v1 !
4553
                                   <1>
                                                   ((!'sysbreak' is not needed in Retro UNIX 8086 v1!))
4554
                                  <1>
                                               NOTE:
4555
                                   <1>
                                                   'sysbreak' clears extended part (beyond of previous
                                                    'u.break' address) of user's memory for original unix's
4556
                                   <1>
4557
                                   <1>
                                                    'bss' compatibility with Retro UNIX 8086 v1 (19/11/2013)
4558
                                   <1>
                                                   ; mov u.break,r1 / move users break point to r1 \,
                                  <1>
4559
4560
                                   <1>
                                                   ; cmp r1, $core / is it the same or lower than core?
4561
                                  <1>
                                                   ; blos 1f / yes, 1f
4562
                                  <1>
                                             ; 23/06/2015
4563 0000DDFF 8B2D[90030300]
                                   <1>
                                            mov
                                                  ebp, [u.break] ; virtual address (offset)
4564
                                  <1>
                                            ; and ebp, ebp
                                            jz short sysbreak 3;
4565
                                  <1>
                                  <1>
                                            ; Retro UNIX 386 v1 NOTE: u.break points to virtual address !!!
4566
```

```
4567
                                           ; (Even break point address is not needed for Retro UNIX 386 v1)
                                 <1>
4568 0000DE05 8B15[5C030300]
                                 <1>
                                                edx, [u.sp]; kernel stack at the beginning of sys call
4569 0000DE0B 83C20C
                                                edx, 12; EIP -4-> CS -4-> EFLAGS -4-> ESP (user)
                                 <1>
                                           add
                                 <1>
                                           ; 07/10/2015
4571 0000DE0E 891D[90030300]
                                 <1>
                                                [u.break], ebx ; virtual address !!!
                                          mov
4572
                                 <1>
4573 0000DE14 3B1A
                                 <1>
                                           cmp
                                                 ebx, [edx] ; compare new break point with
4574
                                 <1>
                                                          ; with top of user's stack (virtual!)
4575 0000DE16 7323
                                 <1>
                                                 short sysbreak 3
                                                 ; cmp r1,sp / is it the same or higher
4576
                                 <1>
                                                        ; / than the stack?
4577
                                 <1>
4578
                                                 ; bhis 1f / yes, 1f
                                 <1>
4579 0000DE18 89DE
                                 <1>
                                                esi, ebx
                                           mov
4580 0000DE1A 29EE
                                 <1>
                                           sub
                                                esi, ebp ; new break point - old break point
4581 0000DE1C 761D
                                 <1>
                                                short sysbreak 3
                                           jna
4582
                                <1>
                                           ;push ebx
                                 <1> sysbreak 1:
4584 0000DE1E 89EB
                                 <1>
                                                ebx, ebp
                                           mov
4585 0000DE20 E8C774FFFF
                                 <1>
                                           call get_physical_addr ; get physical address
                                          jс
                                                 tr addr_nm_err
4586 0000DE25 72C3
                                <1>
                                          ; 18/10/2015
4587
                                <1>
4588 0000DE27 89C7
                                 <1>
                                          mov
                                                edi, eax
4589 0000DE29 29C0
                                <1>
                                                eax, eax ; 0
                                          sub
                                <1>
                                                 ; ECX = remain byte count in page (1-4096)
4591 0000DE2B 39CE
                                <1>
                                                esi, ecx
                                          cmp
4592 0000DE2D 7302
                                <1>
                                           jnb
                                                short sysbreak_2
4593 0000DE2F 89F1
                                <1>
                                          mov
                                                ecx, esi
4594
                                <1> sysbreak_2:
                                <1>
4595 0000DE31 29CE
                                           sub
                                                esi, ecx
4596 0000DE33 01CD
                                           add ebp, ecx
                                <1>
4597 0000DE35 F3AA
                                <1>
                                           rep
                                                stosb
4598 0000DE37 09F6
                                <1>
                                          or
                                                 esi, esi
4599 0000DE39 75E3
                                <1>
                                                short sysbreak_1
                                          jnz
4600
                                 <1>
                                                 ; bit $1,r1 / is it an odd address
4601
                                 <1>
                                                 ; beq 2f / no, its even
4602
                                 <1>
                                                 ; clrb (r1) + / yes, make it even
4603
                                 <1>
                                          ; 2: / clear area between the break point and the stack
4604
                                 <1>
4605
                                 <1>
                                                ; cmp r1,sp / is it higher or same than the stack
4606
                                 <1>
                                                 ; bhis 1f / yes, quit
4607
                                 <1>
                                                 ; clr (r1) + / clear word
4608
                                 <1>
                                                 ; br 2b / go back
4609
                                 <1>
                                          ;pop ebx
4610
                                 <1> sysbreak 3: ; 1:
4611
                                 <1>
                                          ;mov [u.break], ebx; virtual address!!!
                                                 ; jsr r0,arg; u.break / put the "address"
4612
                                 <1>
                                                     ; / in u.break (set new break point)
4613
                                 <1>
4614
                                                 ; br sysret4 / br sysret
                                 <1>
4615 0000DE3B E922E9FFFF
                                 <1>
                                          jmp
                                                sysret
4616
                                 <1>
4617
                                 <1> sysseek: ; / moves read write pointer in an fsp entry
4618
                                 <1>
                                        ; 06/11/2016 - TRDOS 386 (TRDOS v2.0)
                                           ; 22/06/2015 (Retro UNIX 386 v1 - Beginning)
4619
                                 <1>
4620
                                 <1>
                                          ; 07/07/2013 - 05/08/2013 (Retro UNIX 8086 v1)
4621
                                 <1>
                                           ; 'sysseek' changes the \ensuremath{\mathrm{r/w}} pointer of (3rd word of in an
4622
                                 <1>
                                          ; fsp entry) of an open file whose file descriptor is in u.r0.
4623
                                          ; The file descriptor refers to a file open for reading or
4624
                                 <1>
4625
                                 <1>
                                          ; writing. The read (or write) pointer is set as follows:
                                          ; * if 'ptrname' is 0, the pointer is set to offset.
4626
                                 <1>
                                                * if 'ptrname' is 1, the pointer is set to its
4627
                                 <1>
4628
                                 <1>
                                                   current location plus offset.
                                                 \mbox{\ensuremath{\star}} if 'ptrname' is 2, the pointer is set to the
4629
                                 <1>
4630
                                 <1>
                                                  size of file plus offset.
4631
                                 <1>
                                          ; The error bit (e-bit) is set for an undefined descriptor.
4632
                                 <1>
4633
                                 <1>
                                          ; Calling sequence:
4634
                                 <1>
                                         ; sysseek; offset; ptrname
4635
                                 <1>
                                          ; Arguments:
4636
                                 <1>
                                          ; offset - number of bytes desired to move
                                                       the r/w pointer
4637
                                 <1>
4638
                                 <1>
                                                ptrname - a switch indicated above
4639
                                 <1>
4640
                                 <1>
                                          ; Inputs: r0 - file descriptor
4641
                                 <1>
                                           ; Outputs: -
                                           f ......
4642
                                 <1>
4643
4644
                                 <1>
                                           ; Retro UNIX 8086 v1 modification:
4645
                                 <1>
                                                  'sysseek' system call has three arguments; so,
4646
                                 <1>
                                                 * 1st argument, file descriptor is in BX (BL) register
                                                 * 2nd argument, offset is in CX register
4647
                                 <1>
4648
                                 <1>
                                                 * 3rd argument, ptrname/switch is in DX (DL) register
                                 <1>
4650 0000DE40 E821000000
                                 <1>
                                           call seektell
4651
                                 <1>
                                          ; EAX = Current R/W pointer of the file
4652
                                 <1>
                                           ; EBX = [u.fofp]
                                 <1>
                                          ; [u.base] = offset (ECX input)
4654
                                 <1>
4655 0000DE45 0305[84030300]
                                 <1>
                                                 eax, [u.base]
4656 0000DE4B 8903
                                 <1>
                                                [ebx], eax
                                          mov
4657 0000DE4D E910E9FFFF
                                 <1>
                                           jmp
                                                sysret
                                 <1>
                                 <1> systell: ; / get the r/w pointer
4659
4660
                                 <1>
                                          ; 06/11/2016 - TRDOS 386 (TRDOS v2.0) - temporary !-
4661
                                 <1>
                                           ; 22/06/2015 (Retro UNIX 386 v1 - Beginning)
4662
                                 <1>
                                           ; 07/07/2013 - 05/08/2013 (Retro UNIX 8086 v1)
4663
                                 <1>
4664
                                 <1>
                                          ; Retro UNIX 8086 v1 modification:
                                           ; ! 'systell' does not work in original UNIX v1,
4665
                                 <1>
                                 <1>
4666
                                                it returns with error !
4667
                                 <1>
                                          ; Inputs: r0 - file descriptor
                                          ; Outputs: r0 - file r/w pointer
4668
                                 <1>
4669
                                 <1>
                                 <1>
4670
                                           ;xor ecx, ecx; 0
4671 0000DE52 BA01000000
                                 <1>
                                          mov edx, 1; 05/08/2013
```

```
4672
                                                                                         ;call seektell
                                                                        <1>
4673 0000DE57 E810000000
                                                                        <1>
                                                                                           call seektell0 ; 05/08/2013
                                                                                            ;; 06/11/2016
4674
                                                                        <1>
4675
                                                                        <1>
                                                                                            ;; mov eax, [ebx]
4676 0000DE5C A3[64030300]
                                                                        <1>
                                                                                            mov [u.r0], eax
4677 0000DE61 E9FCE8FFFF
                                                                        <1>
                                                                                             jmp
                                                                                                         sysret
                                                                        <1>
                                                                        <1>; Original unix v1 'systell' system call:
4679
4680
                                                                        <1>
                                                                                                          ; jsr r0, seektell
4681
                                                                        <1>
                                                                                                           ; br error4
4682
                                                                        <1>
4683
                                                                         <1> seektell:
                                                                                      ; 06/11/2016 - TRDOS 386 (TRDOS v2.0)
4684
                                                                        <1>
4685
                                                                        <1>
                                                                                            ; 03/01/2016
                                                                                         ; 22/06/2015 (Retro UNIX 386 v1 - Beginning)
4686
                                                                        <1>
4687
                                                                        <1>
                                                                                            ; 07/07/2013 - 05/08/2013 (Retro UNIX 8086 v1)
4688
                                                                        <1>
                                                                                            ; 'seektell' puts the arguments from sysseek and systell % \left( 1\right) =\left( 1\right) \left( 
4689
                                                                        <1>
4690
                                                                        <1>
                                                                                             ; call in u.base and u.count. It then gets the i-number of
4691
                                                                        <1>
                                                                                             ; the file from the file descriptor in u.rO and by calling
4692
                                                                        <1>
                                                                                            ; getf. The i-node is brought into core and then u.count
4693
                                                                         <1>
                                                                                            ; is checked to see it is a 0, 1, or 2.
4694
                                                                        <1>
                                                                                             ; If it is 0 - u.count stays the same
4695
                                                                        <1>
                                                                                                                   1 - u.count = offset (u.fofp)
4696
                                                                        <1>
                                                                                                                 2 - u.count = i.size (size of file)
4697
                                                                        <1>
4698
                                                                        <1>
                                                                                             ; !! Retro UNIX 8086 v1 modification:
4699
                                                                        <1>
                                                                                                          Argument 1, file descriptor is in BX;
4700
                                                                        <1>
                                                                                                           Argument 2, offset is in CX;
4701
                                                                        <1>
                                                                                                           Argument 3, ptrname/switch is in DX register.
                                                                                             ;
4702
                                                                        <1>
4703
                                                                         <1>
                                                                                             ; ((Return -> eax = base for offset (position= base+offset))
4704
                                                                        <1>
4705 0000DE66 890D[84030300]
                                                                        <1>
                                                                                                          [u.base], ecx; offset
                                                                        <1> seektell0:
4706
4707 0000DE6C 8915[88030300]
                                                                        <1>
                                                                                                            [u.count], edx
                                                                                             ; EBX = file descriptor (file number)
4708
                                                                        <1>
4709 0000DE72 E8F3FEFFFF
                                                                        <1>
                                                                                             call getf1
4710
                                                                        <1>
                                                                                            ; EAX = First cluster of the file
4711
                                                                        <1>
                                                                                            ; EBX = File number (Open file number)
4712
                                                                        <1>
                                                                                            ; [u.fofp] = Pointer to File pointer
                                                                                            ; [i.size] = File size
4713
                                                                        <1>
4714
                                                                        <1>
4715 0000DE77 09C0
                                                                        <1>
                                                                                                            eax, eax
4716 0000DE79 7514
                                                                        <1>
                                                                                            jnz
                                                                                                         short seektell1
4717
                                                                        <1>
4718 0000DE7B B80A000000
                                                                        <1>
                                                                                            mov
                                                                                                           eax, ERR_FILE_NOT_OPEN
4719 0000DE80 A3[64030300]
                                                                        <1>
                                                                                             mov
                                                                                                           [u.r0], eax
4720 0000DE85 A3[C8030300]
                                                                        <1>
                                                                                                           dword [u.error], eax ; 'file not open !'
4721 0000DE8A E9B3E8FFFF
                                                                        <1>
                                                                                             jmp
                                                                                                          error
4722
                                                                        <1>
4723
                                                                        <1> seektell1:
                                                                        <1> mov
4724 0000DE8F 8B1D[74030300]
                                                                                                                ebx, [u.fofp]
4725 0000DE95 803D[88030300]01
                                                                      <1>
                                                                                             cmp byte [u.count], 1
4726 0000DE9C 7705
                                                                        <1>
                                                                                                          short seektell2
                                                                                             jа
4727 0000DE9E 7409
                                                                        <1>
                                                                                             jе
                                                                                                          short seektell3
4728 0000DEA0 31C0
                                                                        <1>
                                                                                                         eax, eax
                                                                                            xor
4729 0000DEA2 C3
                                                                        <1>
                                                                                            retn
4730
                                                                        <1>
4731
                                                                        <1> seektell2:
4732 0000DEA3 A1[55040300]
                                                                        <1>
                                                                                                                         eax, [i.size]
                                                                                             mov
4733 0000DEA8 C3
                                                                        <1>
                                                                                             retn
4734
                                                                        <1>
4735
                                                                        <1> seektell3:
4736 0000DEA9 8B03
                                                                        <1>
                                                                                    mov
                                                                                                           eax, [ebx]
4737 0000DEAB C3
                                                                        <1>
                                                                                             retn
4738
                                                                        <1>
                                                                        <1> sysintr: ; / set interrupt handling
4739
                                                                                            ; 22/06/2015 (Retro UNIX 386 v1 - Beginning)
4740
                                                                        <1>
4741
                                                                        <1>
                                                                                             ; 07/07/2013 (Retro UNIX 8086 v1)
4742
                                                                        <1>
4743
                                                                         <1>
                                                                                             ; 'sysintr' sets the interrupt handling value. It puts
4744
                                                                        <1>
                                                                                             ; argument of its call in u.intr then branches into 'sysquit'
4745
                                                                        <1>
                                                                                             ; routine. u.tty is checked if to see if a control tty exists.
                                                                                            ; If one does the interrupt character in the tty buffer is
4746
                                                                        <1>
                                                                                             ; cleared and 'sysret'is called. If one does not exits
4747
                                                                        <1>
4748
                                                                         <1>
                                                                                             ; 'sysret' is just called.
4749
                                                                        <1>
4750
                                                                         <1>
                                                                                            ; Calling sequence:
4751
                                                                         <1>
                                                                                                     sysintr; arg
4752
                                                                         <1>
                                                                                             ; Argument:
                                                                                                            arg - if 0, interrupts (ASCII DELETE) are ignored.
4753
                                                                         <1>
                                                                                                                    - if 1, intterupts cause their normal result
4754
                                                                         <1>
4755
                                                                         <1>
                                                                                                                           i.e force an exit.
4756
                                                                         <1>
                                                                                                                    - if arg is a location within the program,
                                                                                                                         control is passed to that location when
4757
                                                                         <1>
4758
                                                                         <1>
                                                                                                                         an interrupt occurs.
4759
                                                                        <1>
                                                                                             ; Inputs: -
4760
                                                                        <1>
                                                                                             ; Outputs: -
4761
                                                                        <1>
                                                                                             4762
                                                                        <1>
4763
                                                                         <1>
                                                                                             ; Retro UNIX 8086 v1 modification:
                                                                                                              'sysintr' system call sets u.intr to value of {\tt BX}
4764
                                                                        <1>
4765
                                                                        <1>
                                                                                                            then branches into sysquit.
4766
                                                                        <1>
                                                                                             ;
4767 0000DEAC 66891D[AA030300]
                                                                        <1>
                                                                                             mov
                                                                                                         [u.intr], bx
                                                                                                           ; jsr r0, arg; u.intr / put the argument in u.intr
4768
                                                                         <1>
4769
                                                                        <1>
                                                                                                            ; br 1f / go into quit routine
4770 0000DEB3 E9AAE8FFFF
                                                                        <1>
                                                                                             jmp
                                                                                                          sysret
4771
                                                                        <1>
4772
                                                                        <1> sysquit:
4773
                                                                        <1>
                                                                                             ; 22/06/2015 (Retro UNIX 386 v1 - Beginning)
                                                                                             ; 07/07/2013 (Retro UNIX 8086 v1)
4774
                                                                        <1>
4775
                                                                        <1>
4776
                                                                        <1>
                                                                                             ; 'sysquit' turns off the quit signal. it puts the argument of
```

```
4777
                                 <1>
                                           ; the call in u.quit. u.tty is checked if to see if a control
4778
                                  <1>
                                           ; tty exists. If one does the interrupt character in the tty
                                           ; buffer is cleared and 'sysret'is called. If one does not exits
4779
                                 <1>
4780
                                 <1>
                                           ; 'sysret' is just called.
4781
                                 <1>
4782
                                 <1>
                                           ; Calling sequence:
4783
                                 <1>
                                               sysquit; arg
4784
                                           ; Argument:
                                 <1>
4785
                                 <1>
                                                 arg - if 0, this call diables quit signals from the
                                                        typewriter (ASCII FS)
4786
                                 <1>
                                                      - if 1, quits are re-enabled and cause execution to
4787
                                 <1>
4788
                                  <1>
                                                       cease and a core image to be produced.
4789
                                 <1>
                                                         i.e force an exit.
4790
                                 <1>
                                                      - if arg is an addres in the program,
4791
                                 <1>
                                                        a quit causes control to sent to that
                                                        location.
4792
                                 <1>
4793
                                  <1>
                                           ; Inputs: -
4794
                                           ; Outputs: -
                                 <1>
4795
                                 <1>
                                           i ......
4796
                                 <1>
                                          ; Retro UNIX 8086 v1 modification:
4797
                                 <1>
                                                   'sysquit' system call sets u.quit to value of BX
4798
                                  <1>
4799
                                 <1>
                                                  then branches into 'sysret'.
                                           ;
4800
                                  <1>
4801 0000DEB8 66891D[AC030300]
                                 <1>
                                           mov
                                                 [u.quit], bx
4802 0000DEBF E99EE8FFFF
                                 <1>
                                           jmp
                                                 sysret
                                  <1>
                                                  ; jsr r0, arg; u.quit / put argument in u.quit
                                           ;1:
4804
                                 <1>
4805
                                  <1>
                                                  ; mov u.ttyp,r1 / move pointer to control tty buffer
                                                              ; / to r1
4806
                                 <1>
4807
                                  <1>
                                                  ; beq sysret4 / return to user
4808
                                                  ; clrb 6(r1) / clear the interrupt character
                                  <1>
                                                          ; / in the tty buffer
4809
                                 <1>
4810
                                 <1>
                                                  ; br sysret4 / return to user
4811
                                 <1>
4812
                                 <1> anyi:
                                           ; 06/10/2016 (TRDOS 386 = TRDOS v2.0)
4813
                                  <1>
4814
                                 <1>
                                           ; Major Modification!
4815
                                 <1>
                                           ; TRDOS 386 does not permit to delete a file while it is open
                                           ; The role of 'anyi' procedure has beeen changed to ensure that.
4816
                                 <1>
4817
                                 <1>
                                           ; 22/06/2015 (Retro UNIX 386 v1 - Beginning)
4818
                                  <1>
                                           ; 25/04/2013 (Retro UNIX 8086 v1)
4819
                                 <1>
4820
                                 <1>
                                           ; 'anyi' is called if a file deleted while open.
4821
                                 <1>
                                           ; "anyi" checks to see if someone else has opened this file.
4822
                                 <1>
4823
                                 <1>
                                           ; INPUTS ->
4824
                                 <1>
                                               r1 - contains an i-number
4825
                                 <1>
4826
                                 <1>
                                                fsp - start of table containing open files
4827
                                 <1>
4828
                                  <1>
                                           ; OUTPUTS ->
                                                "deleted" flag set in fsp entry of another occurrence of
4829
                                 <1>
4830
                                 <1>
                                                    this file and r2 points 1st word of this fsp entry.
4831
                                 <1>
                                                if file not found - bit in i-node map is cleared
4832
                                 <1>
                                                                (i-node is freed)
4833
                                  <1>
                                                           all blocks related to i-node are freed
                                                          all flags in i-node are cleared
4834
                                 <1>
4835
                                 <1>
                                           ; ((AX = R1)) input
4836
                                 <1>
4837
                                 <1>
                                                 (Retro UNIX Prototype : 02/12/2012, UNIXCOPY.ASM)
4838
                                  <1>
                                                  ((Modified registers: eDX, eCX, eBX, eSI, eDI, eBP))
4839
                                 <1>
4840
                                 <1>
                                           ; / r1 contains an i-number
4841
                                 <1>
                                           ; TRDOS 386 (06/10/2016)
4842
                                 <1>
4843
                                  <1>
                                           ; INPUT:
4844
                                 <1>
4845
                                 <1>
                                                 EAX = First Cluster
                                                  DL = Logical DOS Drive Number
4846
                                 <1>
                                           ;
4847
                                 <1>
                                           ; OUTPUT:
4848
                                  <1>
                                                 CF = 1 -> EBX = File Handle/Number/Index
4849
                                 <1>
                                                  CF = 0 \rightarrow EBX = 0
4850
                                 <1>
4851
                                  <1>
                                           ; Modified Registers: EBX
4852
                                 <1>
4853
                                  <1>
4854 0000DEC4 31DB
                                 <1>
                                           xor
                                                  ebx, ebx
4855
                                  <1> anyi 0:
4856 0000DEC6 80BB[E6680100]00
                                                  byte [ebx+OF MODE], 0 ; 0 = empty entry
                                 <1>
                                           cmp
4857 0000DECD 770A
                                 <1>
                                                  short anyi_2; 1 (r), 2 (w) or 3 (r&w)
4858
                                  <1> anyi 1:
4859 0000DECF FEC3
                                 <1>
                                           inc
4860 0000DED1 80FB0A
                                 <1>
                                                  bl, OPENFILES; max. count of open files
                                           cmp
4861 0000DED4 72F0
                                 <1>
                                                  short anyi_0
                                           jb
4862 0000DED6 31C0
                                 <1>
                                           xor
                                                  eax, eax
4863 0000DED8 C3
                                 <1>
                                           retn
4864
                                 <1> anyi 2:
                                                  dl, [ebx+OF DRIVE]
4865 0000DED9 3A93[DC680100]
                                <1>
                                           cmp
4866 0000DEDF 75EE
                                 <1>
                                                  short anyi 1
                                           jne
                                                  bx, 2; *4 (dword offset)
4867 0000DEE1 66C1E302
                                 <1>
                                           shl
4868 0000DEE5 3B83[B4680100]
                                 <1>
                                           cmp
                                                  eax, [ebx+OF FCLUSTER]
4869 0000DEEB 7406
                                 <1>
                                           jе
                                                  short anyi_3
4870 0000DEED 66C1EB02
                                 <1>
                                                  bx, 2 ; /4 (byte offset)
                                           shr
4871 0000DEF1 EBDC
                                 <1>
                                           jmp
                                                 short anyi_1
                                 <1> anyi_3:
4872
                                                 bx, 2; /4 (bytes offset) (index)
4873 0000DEF3 66C1EB02
                                 <1>
4874 0000DEF7 F9
                                 <1>
                                           stc
4875 0000DEF8 C3
                                 <1>
                                 <1>
4876
                                 <1>; Retro UNIX 386 v1 Kernel (v0.2) - SYS9.INC
4877
4878
                                 <1> ; Last Modification: 09/12/2015
4879
                                 <1>
4880
                                 <1> syssleep:
4881
                                 <1>
                                       ; 29/06/2015 - (Retro UNIX 386 v1)
```

```
<1>
4882
                                           ; 11/06/2014 - (Retro UNIX 8086 v1)
4883
                                 <1>
4884
                                           ; Retro UNIX 8086 v1 feature only
                                 <1>
4885
                                 <1>
                                           ; (INPUT -> none)
4886
                                 <1>
4887 0000DEF9 0FB61D[B3030300]
                                 <1>
                                           movzx ebx, byte [u.uno] ; process number
4888 0000DF00 8AA3[7F000300]
                                           mov ah, [ebx+p.ttyc-1] ; current/console tty
                                 <1>
4889 0000DF06 E841190000
                                 <1>
                                           call sleep
4890 0000DF0B E952E8FFFF
                                 <1>
                                           jmp
                                                 sysret
4891
                                 <1>
4892
                                 <1> _vp_clr:
4893
                                 <1>
                                          ; Reset/Clear Video Page
4894
                                 <1>
                                          ; 30/06/2015 - (Retro UNIX 386 v1)
4895
                                 <1>
                                           ; 21/05/2013 - 30/10/2013 (Retro UNIX 8086 v1) (U0.ASM)
4896
                                 <1>
4897
                                 <1>
4898
                                 <1>
                                           ; Retro UNIX 8086 v1 feature only !
4899
                                 <1>
                                           ; INPUTS ->
4900
                                 <1>
4901
                                 <1>
                                           ; BH = video page number
4902
                                 <1>
4903
                                 <1>
                                          ; OUTPUT ->
4904
                                 <1>
                                           ; none
4905
                                 <1>
                                           ; ((Modified registers: eAX, BH, eCX, eDX, eSI, eDI))
                                 <1>
4906
4907
                                 <1>
                                           ; 04/12/2013
4908 0000DF10 28C0
                                 <1>
                                           sub al, al
4909
                                 <1>
                                          ; al = 0 (clear video page)
4910
                                 <1>
                                           ; bh = video page ; 13/05/2016
4911 0000DF12 B407
                                           mov ah, 07h
                                 <1>
4912
                                 <1>
                                           ; ah = 7 (attribute/color)
                                          xor cx, cx; 0, left upper column (cl) & row (cl) mov dx, 184Fh; right lower column & row (dl=24, dh=79)
4913 0000DF14 6631C9
                                 <1>
4914 0000DF17 66BA4F18
                                 <1>
4915 0000DF1B E8F33AFFFF
                                 <1>
                                           call scroll up
                                           ; bh = video page
4916
                                 <1>
                                           xor dx, dx ; 0 (cursor position)
4917 0000DF20 6631D2
                                 <1>
                                                _set_cpos
4918 0000DF23 E9293EFFFF
                                 <1>
                                           jmp
4919
                                 <1>
                                 <1> sysmsg:
4920
                                         ; 13/05/2016
4921
                                 <1>
4922
                                 <1>
                                           ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
                                           ; 01/07/2015 - 11/11/2015 (Retro UNIX 386 v1)
4923
                                 <1>
4924
                                 <1>
                                           ; Print user-application message on user's console tty
4925
                                 <1>
4926
                                 <1>
                                           ; Input -> EBX = Message address
                                                    ECX = Message length (max. 255)
4927
                                 <1>
4928
                                 <1>
                                                    DL = Color (IBM PC Rombios color attributes)
4929
                                 <1>
                                                 ecx, MAX MSG LEN; 255
4930 0000DF28 81F9FF000000
                                 <1>
                                           cmp
4931 0000DF2E 0F872EE8FFFF
                                 <1>
                                                 sysret; nothing to do with big message size
                                           jа
4932 0000DF34 08C9
                                 <1>
                                                 cl, cl
4933 0000DF36 0F8426E8FFFF
                                 <1>
                                           jΖ
                                                 sysret
4934 0000DF3C 20D2
                                 <1>
                                           and
                                                 dl, dl
4935 0000DF3E 7502
                                 <1>
                                                 short sysmsg0
                                           jnz
4936 0000DF40 B207
                                 <1>
                                                 dl, 07h ; default color
                                           mov
4937
                                 <1>
                                                  ; (black background, light gray character)
                                 <1> sysmsg0:
4939 0000DF42 891D[84030300]
                                 <1>
                                           mov
                                                 [u.base], ebx
4940 0000DF48 8815[EF580100]
                                 <1>
                                                  [ccolor], dl ; color attributes
                                           mov
4941 0000DF4E 89E5
                                 <1>
                                                 ebp, esp
                                           mov
4942 0000DF50 31DB
                                 <1>
                                                  ebx, ebx; 0
                                           xor
4943 0000DF52 891D[8C030300]
                                 <1>
                                                 [u.nread], ebx; 0
                                           mov
4944
                                 <1>
4945 0000DF58 381D[C6030300]
                                 <1>
                                                 [u.kcall], bl; 0
                                           cmp
                                                 short sysmsgk ; Temporary (01/07/2015)
4946 0000DF5E 7769
                                 <1>
                                           jа
4947
                                 <1>
4948 0000DF60 890D[88030300]
                                 <1>
                                           mov
                                                 [u.count], ecx
                                                 ecx ; + 00h ; ASCIIZ
4949 0000DF66 41
                                 <1>
                                          inc
4950 0000DF67 29CC
                                 <1>
                                           sub
                                                 esp, ecx
4951 0000DF69 89E7
                                 <1>
                                           mov
                                                 edi, esp
4952 0000DF6B 89E6
                                 <1>
                                           mov
                                                 esi, esp
4953 0000DF6D 66891D[C4030300]
                                                  [u.pcount], bx ; reset page (phy. addr.) counter
                                 <1>
                                           mov
4954
                                 <1>
                                           ; 11/11/2015
4955 0000DF74 8A25[94030300]
                                 <1>
                                           mov ah, [u.ttyp]; recent open tty
                                           ; 0 = none
4956
                                 <1>
4957 0000DF7A FECC
                                 <1>
                                           dec
4958 0000DF7C 790C
                                 <1>
                                                 short sysmsg1
                                           jns
4959 0000DF7E 8A1D[B3030300]
                                 <1>
                                           mov
                                                 bl, [u.uno] ; process number
                                                 ah, [ebx+p.ttyc-1]; user's (process's) console tty
4960 0000DF84 8AA3[7F000300]
                                 <1>
                                           mov
4961
                                 <1> sysmsg1:
4962 0000DF8A 8825[96030300]
                                 <1>
                                           mov
                                                 [u.ttyn], ah
                                  <1> sysmsq2:
4963
4964 0000DF90 E808080000
                                <1>
                                           call cpass
4965 0000DF95 7416
                                 <1>
                                           jΖ
                                                 short sysmsg5
4966 0000DF97 AA
                                 <1>
                                           stosb
4967 0000DF98 20C0
                                 <1>
                                           and
                                                 al, al
                                                 short sysmsg2
4968 0000DF9A 75F4
                                <1>
                                           jnz
4969
                                <1> sysmsg3:
4970 0000DF9C 80FC07
                                <1>
                                                 ah, 7 ; tty number
                                           cmp
4971 0000DF9F 7711
                                <1>
                                                 short sysmsg6 ; serial port
                                           jа
4972 0000DFA1 E83E000000
                                <1>
                                           call print_cmsg
                                 <1> sysmsg4:
4974 0000DFA6 89EC
                                <1>
                                           mov
                                                  esp, ebp
4975 0000DFA8 E9B5E7FFFF
                                <1>
                                                 sysret
                                           jmp
                                 <1> sysmsg5:
4977 0000DFAD C60700
                                <1>
                                           mov
                                                 byte [edi], 0
4978 0000DFB0 EBEA
                                <1>
                                                 short sysmsg3
                                           qmj
4979
                                <1> sysmsg6:
4980 0000DFB2 8A06
                                 <1>
                                           mov
                                                 al, [esi]
4981 0000DFB4 E891180000
                                           call sndc
                                <1>
                                                 short sysmsg4
4982 0000DFB9 72EB
                                <1>
                                           jс
4983 0000DFBB 803E00
                                <1>
                                           cmp
                                                 byte [esi], 0 ; 0 is stop character
4984 0000DFBE 76E6
                                <1>
                                           jna
                                                 short sysmsg4
4985 0000DFC0 46
                                 <1>
                                           inc
                                                 esi
4986 0000DFC1 8A25[96030300]
                                <1>
                                          mov
                                                 ah, [u.ttyn]
```

```
4987 0000DFC7 EBE9
                                <1>
                                         jmp short sysmsg6
4988
                                <1>
4989
                                <1> sysmsgk: ; Temporary (01/07/2015)
4990
                                <1>
                                       ; The message has been sent by Kernel (ASCIIZ string)
4991
                                <1>
                                         ; (ECX -character count- will not be considered)
4992 0000DFC9 8B35[84030300]
                                <1>
                                         mov esi, [u.base]
4993 0000DFCF 8A25[EE580100]
                                       mov ah, [ptty]; present/current screen (video page)
                                <1>
4994 0000DFD5 8825[96030300]
                                <1>
                                         mov [u.ttyn], ah
4995 0000DFDB C605[C6030300]00
                                <1>
                                         mov
                                               byte [u.kcall], 0
4996 0000DFE2 EBB8
                                <1>
                                              short sysmsg3
                                         jmp
4997
                                <1>
4998
                                <1> print cmsg:
                                      ; 18/11/2017
4999
                                <1>
5000
                                <1>
                                         ; 13/05/2016 - TRDOS 386 (TRDOS v2.0)
                                        ; 01/07/2015 (Retro UNIX 386 v1)
5001
                                <1>
5002
                                <1>
5003
                                <1>
                                         ; print message (on user's console tty)
                                               with requested color
5004
                                <1>
                                       ;
5005
                                <1>
                                         ; INPUTS:
5006
                                <1>
5007
                                <1>
                                               esi = message address
                                                [u.ttyn] = tty number (0 to 7)
5008
                                <1>
                                         ;
5009
                                <1>
                                               [ccolor] = color attributes (IBM PC BIOS colors)
                                         ;
5010
                                <1>
                                         ;mov bh, ah
                                <1>
5011
5012 0000DFE4 8A3D[96030300]
                                <1>
                                         mov
                                               bh, [u.ttyn]
5013
                                <1>
                                         ;mov bl, [ccolor]; *
5014
                                <1> pcmsg1:
5015 0000DFEA AC
                                <1>
                                      lodsb
5016 0000DFEB 20C0
                                <1>
                                         and al, al ; 0
5017 0000DFED 740F
                                <1>
                                         jz short pcmsg2
5018 0000DFEF 56
                                <1>
                                         push esi
5019 0000DFF0 8A1D[EF580100]
                                         mov bl, [ccolor] ; * (video.s 'ull'&'beep' change BL)
                                <1>
5020
                                <1>
                                         ;mov bh, [u.ttyn]
5021 0000DFF6 E8C03CFFFF
                                <1>
                                         call _write_tty
                                <1>
5022 0000DFFB 5E
                                         pop
                                                esi
5023 0000DFFC EBEC
                                <1>
                                               short pcmsg1
                                         jmp
5024
                                <1> pcmsg2:
5025 0000DFFE C3
                                <1>
5026
                                <1>
5027
                                <1> sysgeterr:
                                     ; 09/12/2015
5028
                                <1>
                                         ; 21/09/2015 - (Retro UNIX 386 v1 feature only!)
5029
                                <1>
5030
                                <1>
                                        ; Get last error number or page fault count
5031
                                <1>
                                         ; (for debugging)
5032
                                <1>
                                         ; Input -> EBX = return type
5033
                                <1>
5034
                                <1>
                                                  0 = last error code (which is in 'u.error')
                                       ;
5035
                                <1>
                                                  FFFFFFFF = page fault count for running process
                                                  FFFFFFEh = total page fault count
5036
                                <1>
                                         ;
                                                  1 .. FFFFFFFDh = undefined
5037
                                <1>
5038
                                <1>
                                         ;
                                <1>
                                         ; Output -> EAX = last error number or page fault count
5039
5040
                                <1>
                                                  (depending on EBX input)
5041
                                <1>
                                         ;
5042 0000DFFF 21DB
                                <1>
                                         and ebx, ebx
5043 0000E001 750B
                                <1>
                                         jnz short glerr 2
5044
                                <1> glerr_0:
5045 0000E003 A1[C8030300]
                                <1>
                                               eax, [u.error]
                                         mov
                                <1> glerr_1:
5046
                                <1>
5047 0000E008 A3[64030300]
                                         mov
                                               [u.r0], eax
5048 0000E00D C3
                                <1>
                                         retn
                               <1> glerr_2:
5049
5050 0000E00E 43
                               <1> inc ebx ; FFFFFFFF -> 0, FFFFFFFEh -> FFFFFFFFh
5051 0000E00F 74FD
                                         jz
                               <1>
                                               short glerr_2 ; page fault count for process
                               <1>
<1>
5052 0000E011 43
                                               ebx ; FFFFFFFF -> 0
                                         inc
5053 0000E012 75EF
                               5054 0000E014 A1[80050300]
                               <1>
5055 0000E019 EBED
                                        jmp short glerr_1
                                <1> glerr_3:
5056
5057 0000E01B A1[CC030300]
                                       mov eax, [u.pfcount]
                               <1>
5058 0000E020 EBE6
                                <1>
                                         jmp short glerr_1
5059
                                <1>
5060
                                <1> load_and_run_file:
                                     - ; <u>18/11/2017</u>
5061
                                <1>
                                         ; 22/01/2017
5062
                                <1>
5063
                                <1>
                                         ; 04/01/2017, 07/01/2017
                                        ; 24/10/2016
5064
                                <1>
5065
                                <1>
                                         ; 24/04/2016, 02/05/2016, 03/05/2016, 06/05/2016
                                         ; 23/04/2016 (TRDOS 386 = TRDOS v2.0)
5066
                                <1>
                                         ; 23/10/2015 (Retro UNIX 386 v1, 'sysexec')
5067
                                <1>
                                         ; 23/06/2015 (Retro UNIX 386 v1 - Beginning)
5068
                                <1>
                                         ; 03/06/2013 - 06/12/2013 (Retro UNIX 8086 v1)
5069
                                <1>
5070
                                <1>
                                         ; EAX = First Cluster number
5071
                                <1>
                                         ; EDX = File Size
5072
                                <1>
                                          ; ESI = Argument list address
                                         ; [argc] = argument count
5073
                                <1>
5074
                                <1>
                                         ; [u.nread] = argument list length
5075
                                <1>
                                         ; [esp] = return address to the caller (*)
5076
                                <1>
5077 0000E022 8935[4C040300]
                                                [argv], esi
                                <1>
                                         mov
5078 0000E028 8915[55040300]
                                <1>
                                         mov
                                                [i.size], edx
5079 0000E02E A3[51040300]
                                <1>
                                                [ii], eax
                                         mov
                                <1>
5081
                                <1>
                                         ;sti ; 07/01/2017
                                         ;mov
5082
                                                eax, [k page dir]
                                <1>
5083
                                <1>
                                              [u.pgdir], eax
                                         ;mov
                                                eax, eax ; clc ; *** ; 04/01/2017
5084 0000E033 31C0
                                <1>
                                          xor
5085
                                <1>
                                         ; mov
                                               [u.r0], eax ; 0 ; 07/01/2017
5086
                                <1>
                                         ; 06/05/2016
5087
                                <1>
5088
                                <1>
                                         ; Set 'sysexit' return order to MainProg
5089
                                <1>
5090 0000E035 58
                                         pop eax; * 'loc load and run file 8:' address
                                <1>
5091
                                <1>
                                          ;; 22/01/2017
```

```
;;cli ; 07/01/2017
5093 0000E036 8B25[5C580100]
                                 <1>
                                          mov esp, [tss.esp0]
5094
                                 <1>
5095
                                 <1>
                                           ; 'loc_load_run_file_8' address has
                                           ; 'jmp loc file rw_restore_retn' instruction
5096
                                 <1>
5097
                                 <1>
                                           ; 'loc_file_rw_restore_retn:' will return to
5098
                                 <1>
                                           ; [mainprog return addr]
                                           ; just after 'call command interpreter'
5099
                                 <1>
5100
                                 <1>
5101 0000E03C 68[AB630000]
                                 <1>
                                           push _end_of_mainprog ; we must not return to here !
5102 0000E041 FF35[40650100]
                                 <1>
                                           push dword [mainprog_return_addr]
5103 0000E047 89E5
                                 <1>
                                           mov ebp, esp; **
5104
                                 <1>
5105 0000E049 9C
                                 <1>
                                          pushfd ; EFLAGS ; IRETD ; ***
                                          push KCODE ; cs ; IRETD
push eax ; * (eip) ; IRETD
5106 0000E04A 6A08
                                 <1>
5107 0000E04C 50
                                 <1>
                                          mov [u.sp], esp
5108 0000E04D 8925[5C030300]
                                <1>
5109
                                 <1>
                                           ;mov byte [u.quant], time_count
5110 0000E053 1E
                                 <1>
                                           push ds
5111 0000E054 06
                                           push es
                                <1>
5112 0000E055 0FA0
                                <1>
                                          push fs
5113 0000E057 0FA8
                                 <1>
                                           push gs
5114
                                 <1>
                                           ;mov eax, [u.r0]
5115 0000E059 29C0
                                 <1>
                                                 eax, eax
5116 0000E05B 60
                                 <1>
                                           pushad
5117 0000E05C 68[62C70000]
                                 <1>
                                           push sysret
                                 <1>
                                           ;push sysrel1 ; 07/01/2017
5119 0000E061 8925[60030300]
                                 <1>
                                           mov [u.usp], esp
5120
                                 <1>
5121 0000E067 E845060000
                                           call wswap; Save MainProg (process 1) 'u' structure
                                 <1>
5122
                                 <1>
                                                     ; and registers for return (from program)
5123 0000E06C 89EC
                                 <1>
                                                 esp, ebp ; **
                                           mov
                                           ;;22/01/2017
5124
                                 <1>
                                           ;;sti ; 07/01/2017
5125
                                 <1>
5126 0000E06E 50
                                 <1>
                                           push eax ; * 'loc_load_and_run_file_8:' address
5127
                                 <1>
5128
                                 <1>
                                           ;;; 02/05/2016
5129
                                 <1>
                                           ;;; Create a new process (parent: MainProg)
                                           xor esi, esi
5130 0000E06F 31F6
                                 <1>
5131
                                 <1> cnpm_1: ; search p.stat table for unused process number
5132 0000E071 46
                                 <1>
                                           inc esi
5133 0000E072 80BE[AF000300]00
                                 <1>
                                           cmp
                                                 byte [esi+p.stat-1], 0 ; SFREE
                                                             ; is process active, unused, dead
5134
                                 <1>
                                                 short cnpm_2 ; it's unused so branch
5135 0000E079 760B
                                 <1>
                                                 si, nproc ; all processes checked short cnpm_1 ; no, branch back
5136 0000E07B 6683FE10
                                 <1>
                                           cmp
5137 0000E07F 72F0
                                 <1>
                                           jb
5138 0000E081 E99B83FFFF
                                 <1>
                                                 panic
                                           jmp
5139
                                 <1> cnpm_2:
5140 0000E086 A1[B8030300]
                                 <1>
                                                  eax, [u.pgdir]; page directory of MainProg
5141 0000E08B A3[BC030300]
                                 <1>
                                                 [u.ppgdir], eax ; parent's page directory
                                           mov
5142 0000E090 E8426BFFFF
                                 <1>
                                           call allocate_page
5143 0000E095 0F828683FFFF
                                 <1>
                                           jс
                                                 panic
5144
                                 <1>
                                           ; EAX = UPAGE (user structure page) address
5145 0000E09B A3[B4030300]
                                 <1>
                                           mov [u.upage], eax; memory page for 'user' struct (child)
5146 0000E0A0 89F7
                                 <1>
                                          mov
                                                 edi, esi
5147 0000E0A2 66C1E702
                                 <1>
                                           shl
                                                 di, 2
5148 0000E0A6 8987[BC000300]
                                 <1>
                                           mov [edi+p.upage-4], eax; memory page for 'user' struct
5149 0000E0AC E8A06BFFFF
                                 <1>
                                           call clear_page ; 03/05/2016
5150
                                 <1>
                                           ;movzx eax, byte [p.ttyc] ; console tty (for MainProg)
                                           sub ax, ax; 0
5151 0000E0B1 6629C0
                                 <1>
5152 0000E0B4 668986[7F000300]
                                <1>
                                                 [esi+p.ttyc-1], ax ; al - set child's console tty
                                 <1>
                                                                 ; ah - reset child's wait channel
5154 0000E0BB 89F0
                                 <1>
                                           mov
                                                 eax, esi
5155 0000E0BD A2[B3030300]
                                 <1>
                                           mov [u.uno], al; child process number
5156 0000E0C2 FE86[AF000300]
                                 <1>
                                           inc byte [esi+p.stat-1] ; 1, SRUN
5157 0000E0C8 66D1E6
                                 <1>
                                           shl si, 1; multiply si by 2 to get index into p.pid table
5158 0000E0CB 66FF05[4E030300]
                                 <1>
                                           inc word [mpid]; increment m.pid; get a new process name
5159 0000E0D2 66A1[4E030300]
                                 <1>
                                           mov ax, [mpid]
5160 0000E0D8 668986[1E000300]
                                 <1>
                                                 [esi+p.pid-2], ax ; put new process name
5161
                                 <1>
                                                                ; in child process' name slot
                                                ax, [p.pid] ; get process name of MainProg
5162
                                 <1>
5163 0000E0DF 66B80100
                                 <1>
                                           mov
                                                  ax, 1
5164 0000E0E3 668986[3E000300]
                                 <1>
                                           mov
                                                  [esi+p.ppid-2], ax ; put parent process name
                                                                  ; in parent process slot for child
5165
                                 <1>
5166 0000E0EA 6648
                                                 ax ; 0
                                 <1>
                                           dec
5167 0000E0EC 66A3[94030300]
                                 <1>
                                           mov
                                                 [u.ttyp], ax ; 0
                                 <1>
                                           ;;;
5169 0000E0F2 A1[51040300]
                                 <1>
                                           mov
                                                 eax, [ii]
                                           ; Retro UNIX 386 v1, 'sysexec' (u2.s)
5170
                                 <1>
5171 0000E0F7 E84C170000
                                           call iopen
                                 <1>
5172
                                 <1>
                                           ; 06/06/2016
5173 0000E0FC C605[A9030300]01
                                  <1>
                                           mov byte [u.pri], 1 ; normal priority
                                 <1>
5175 0000E103 EB16
                                 <1>
                                                  short sysexec_7 ; 02/05/2016
                                           jmp
5176
                                 <1>
                                 <1> sysexec 6:
5177
                                          ; 19/11/2017
5178
                                 <1>
5179
                                 <1>
                                           ; 18/11/2017
5180
                                 <1>
                                           ; 14/11/2017
5181
                                 <1>
                                           ; 13/11/2017
5182 0000E105 8925[4C040300]
                                           mov [argv], esp; *!*; start address of argument list
                                 <1>
5183
                                 <1>
                                           ; 04/01/2017
5184
                                 <1>
5185
                                 <1>
                                           ; 24/10/2016
5186
                                 <1>
                                           ;;02/05/2016
5187
                                 <1>
                                           ; 23/04/2016 (TRDOS 386)
5188
                                 <1>
                                           ; 18/10/2015 ('sysexec_6')
5189
                                 <1>
                                           ; 23/06/2015
5190 0000E10B A1[B8030300]
                                 <1>
                                                 eax, [u.pgdir]; physical address of page directory
5191
                                 <1>
                                           ;cmp eax, [k_page_dir] ; TRDOS MainProg ?
5192
                                 <1>
                                           ;je
                                                 short sysexec_7
5193
                                 <1>
                                           ; 19/11/2017
                                           mov
5194 0000E110 8B1D[BC0303001
                                 <1>
                                                  ebx, [u.ppgdir] ; phy addr of the parent's page dir
5195 0000E116 E8F56BFFFF
                                 <1>
                                           call
                                                 deallocate_page_dir
                                 <1> sysexec_7:
5196
```

5092

```
5197 0000E11B E8256BFFFF
                                <1>
                                        call make_page_dir
5198 0000E120 0F82FB82FFFF
                                <1>
                                        jc panic ; allocation error
                                                       ; after a deallocation would be nonsence !?
5199
                                <1>
                                          ; 24/07/2015
5200
                                 <1>
5201
                                 <1>
                                          ; map kernel pages (1st 4MB) to PDE 0
5202
                                 <1>
                                          ; of the user's page directory
5203
                                                (It is needed for interrupts!)
                                          ; 18/10/2015
5204
                                 <1>
                                          mov edx, [k_page_dir] ; Kernel's page directory
mov eax, [edx] ; physical address of
5205 0000E126 8B15[C0580100]
                                <1>
5206 0000E12C 8B02
                                <1>
5207
                                <1>
                                                         ; kernel's first page table (1st 4 MB)
5208
                                <1>
                                                          ; (PDE 0 of kernel's page directory)
5209 0000E12E 8B15[B8030300]
                                <1>
                                                edx, [u.pgdir]
                                          mov
5210 0000E134 8902
                                <1>
                                                [edx], eax ; PDE 0 (1st 4MB)
                                        mov
5211
                                <1>
                                          ; 20/07/2015
5212
                                <1>
5213 0000E136 BB00004000
                                <1>
                                          mov
                                               ebx, CORE; start address = 0 (virtual) + CORE
                                          ; 18/10/2015
5214
                                <1>
5215 0000E13B BE[3C040300]
                                <1>
                                                esi, pcore ; physical start address
5216
                                <1> sysexec 8:
5217 0000E140 B907000000
                                <1>
                                          mov ecx, PDE_A_USER + PDE_A_WRITE + PDE_A_PRESENT
5218 0000E145 E8196BFFFF
                                <1>
                                          call make_page_table
5219 0000E14A 0F82D182FFFF
                                          jс
                                <1>
                                                panic
                                <1>
                                          ;mov ecx, PTE_A_USER + PTE_A_WRITE + PTE_A_PRESENT
5221 0000E150 E81C6BFFFF
                                          call make_page; make new page, clear and set the pte
                                <1>
5222 0000E155 0F82C682FFFF
                                <1>
                                          jс
                                <1>
5224 0000E15B 8906
                                <1>
                                          mov [esi], eax ; 24/06/2015
5225
                                <1>
                                          ; ebx = virtual address (24/07/2015)
5226 0000E15D E8B470FFFF
                                         call add_to_swap_queue
                                <1>
                                         cmp esi, ecore ; user's stack (last) page ? je short sysexec 9 · vos
5227
                                <1>
5228 0000E162 81FE[40040300]
                                <1>
5229 0000E168 740C
                                <1>
                                        mov esi, ecore ; physical address of the last page
5230 0000E16A BE[40040300]
                                <1>
                                       ; 20/07/2015
5231
                                <1>
                                          mov ebx, (ECORE - PAGE_SIZE) + CORE
5232 0000E16F BB00F0FFFF
                                <1>
                                ; ebx = virtual end address + segment base address - 4K
5233
5234 0000E174 EBCA
                                                    short sysexec_8
                                <1> sysexec 9:
5235
                                <1> ; 19/11/2017
5236
5237
                                <1>
                                         ; 24/04/2016 (TRDOS 386 = TRDOS v2.0)
                                        ; 25/06/2015, 26/08/2015, 18/10/2015
; move arguments from kernel stack to [ecore]
5238
                                 <1>
5239
                                 <1>
                                        ; (argument list/line will be copied from kernel stack
5240
                                 <1>
                                        ; frame to the last (stack) page of user's core memory)
5241
                                 <1>
5242
                                 <1>
                                          ; 18/10/2015
                                        mov edi, [ecore]
5243 0000E176 8B3D[40040300]
                                <1>
                                        add
5244 0000E17C 81C700100000
                                <1>
                                               edi, PAGE SIZE
5245
                                <1>
                                          ; 19/11/2017
                                       sub edi, 4
5246 0000E182 83EF04
                                <1>
5247 0000E185 C7070000000
                                <1>
                                        mov dword [edi], 0
5248 0000E18B 89FB
                                <1>
                                          mov
                                               ebx, edi
5249
                                <1>
5250 0000E18D 0FB705[4A040300] <1>
                                        movzx eax, word [argc]
5251 0000E194 09C0
                                <1>
                                         or eax, eax
5252 0000E196 7445
                                <1>
                                          jг
                                                short sysexec_13 ; 19/11/2017
                                 <1>
                                          ;jnz short sysexec 10
5254
                                 <1>
                                          ;mov ebx, edi
5255
                                 <1>
                                          ; sub ebx, 4
                                          ;mov [ebx], eax ; 0
5256
                                 <1>
5257
                                 <1>
                                          ;jmp short sysexec_13
5258
                                 <1> sysexec 10:
5259 0000E198 8B0D[8C030300]
                                <1>
                                          mov
                                                ecx, [u.nread]
5260
                                <1>
                                          ; 13/11/2017
5261
                                <1>
                                          ;mov esi, TextBuffer ; 'load_and_execute_file'
                                          ;mov esi, esp ; 'sysexec'
5262
                                <1>
                                          mov esi, [argv]; 24/04/2016 (TRDOS 386 = TRDOS v2.0)
5263 0000E19E 8B35[4C040300]
                                <1>
                                          ; sub edi, ecx ; page end address - argument list length
5264
                                <1>
5265 0000E1A4 29CB
                                <1>
                                                ebx, ecx ; 19/11/2017
5266 0000E1A6 89C2
                                <1>
                                          mov
                                                edx, eax
5267 0000E1A8 FEC2
                                <1>
                                          inc
                                                dl ; argument count + 1 for argc value
5268 0000E1AA C0E202
                                <1>
                                          shl
                                                dl, 2 ; 4 * (argument count + 1)
                                <1>
5269
                                          ;mov ebx, edi
5270 0000E1AD 89DF
                                <1>
                                                edi, ebx ; 19//11/2017
5271 0000E1AF 80E3FC
                                <1>
                                                bl, OFCh; 32 bit (dword) alignment
                                          and
5272 0000E1B2 29D3
                                <1>
                                          sub
                                                ebx, edx
5273 0000E1B4 89FA
                                <1>
                                                edx, edi
                                          mov
5274 0000E1B6 F3A4
                                <1>
                                          rep
                                                movsb
5275 0000E1B8 89D6
                                <1>
                                                esi, edx
                                          mov
5276 0000E1BA 89DF
                                <1>
                                          mov
                                                edi, ebx
5277 0000E1BC BA00F0BFFF
                            <1>
                                                 edx, ECORE - PAGE_SIZE; virtual addr. of the last page
                                          mov
5278 0000E1C1 2B15[40040300]
                                <1>
                                          sub
                                                 edx, [ecore] ; difference (virtual - physical)
5279 0000E1C7 AB
                                          stosd ; eax = argument count
                                 <1> sysexec_11:
5281 0000E1C8 89F0
                                <1>
                                       mov
                                                eax, esi
5282 0000E1CA 01D0
                                <1>
                                          add
                                                eax, edx
5283 0000E1CC AB
                                <1>
                                          stosd ; eax = virtual address
5284
                                <1>
                                          ;dec byte [argc]
5285 0000E1CD 66FF0D[4A040300]
                                <1>
                                                word [argc] ; 14/11/2017
                                          dec
5286 0000E1D4 7407
                                        jz
                                <1>
                                                short sysexec_13
                                <1> sysexec_12:
5287
5288 0000E1D6 AC
                                <1>
                                          lodsb
5289 0000E1D7 20C0
                                <1>
                                          and
                                                al, al
5290 0000E1D9 75FB
                                <1>
                                                short sysexec_12
5291 0000E1DB EBEB
                                <1>
                                          jmp short sysexec 11
5292
                                <1> sysexec_13:
5293
                                 <1>
                                          ; 24/10/2016
                                          ; 24/04/2016 - TRDOS 386 (TRDOS v2.0)
; 23/06/2015 - 19/10/2015 (Retro UNIX 386 v1, 'sysexec_13')
5294
                                 <1>
5295
                                 <1>
5296
                                 <1>
5297
                                 <1>
                                         ; moving arguments to [ecore] is OK here..
5298
                                 <1>
5299
                                 <1>
                                          ; ebx = beginning addres of argument list pointers
                                 <1>
                                             ; in user's stack
5301 0000E1DD 2B1D[40040300]
                                 <1>
                                          sub
                                                ebx, [ecore]
```

```
5302 0000E1E3 81C300F0BFFF
                                                  ebx, (ECORE - PAGE SIZE)
                                <1>
                                           add
                                                   ; end of core - 4096 (last page)
                                 <1>
5304
                                 <1>
                                                       ; (virtual address)
5305 0000E1E9 891D[4C040300]
                                 <1>
                                                 [argv], ebx
                                           mov
5306 0000E1EF 891D[90030300]
                                 <1>
                                                 [u.break], ebx ; available user memory
                                          mov
5307
                                 <1>
5308 0000E1F5 29C0
                                 <1>
                                           sub
                                                 eax, eax
5309 0000E1F7 C705[88030300]2000- <1>
                                                 dword [u.count], 32 ; Executable file header size
                                          mov
5309 0000E1FF 0000
                                 <1>
5310 0000E201 C705[74030300]-
                                 <1>
                                                 dword [u.fofp], u.off
                                          mov
5310 0000E207 [80030300]
                                 <1>
5311 0000E20B A3[80030300]
                                 <1>
                                                 [u.off], eax; 0
                                          mov
5312 0000E210 A3[84030300]
                                 <1>
                                                 [u.base], eax ; 0, start of user's core (virtual)
                                          mov
5313
                                 <1>
                                          ; 24/10/2016
5314 0000E215 A0[86590100]
                                 <1>
                                          mov
                                                al, [Current Drv]
                                                 [cdev], al
5315 0000E21A A2[46030300]
                                 <1>
                                          mov
                                 <1>
5317 0000E21F A1[51040300]
                                                eax, [ii] ; Fist Cluster of the Program (PRG) file
                                 <1>
                                          mov
                                           ; EAX = First cluster of the executable file
5318
                                 <1>
5319 0000E224 E80A010000
                                 <1>
                                          call readi
5320
                                 <1>
5321 0000E229 8B0D[90030300]
                                 <1>
                                                 ecx, [u.break]; top of user's stack (physical addr.)
                                          mov
5322 0000E22F 890D[88030300]
                                 <1>
                                          mov
                                                 [u.count], ecx; save for overrun check
                                 <1>
5324 0000E235 8B0D[8C030300]
                                 <1>
                                                 ecx, [u.nread]
                                          mov
5325 0000E23B 890D[90030300]
                                 <1>
                                           mov
                                                 [u.break], ecx ; virtual address (offset from start)
5326 0000E241 80F920
                                 <1>
                                                 cl, 32
                                           cmp
                                                     short sysexec_15
5327 0000E244 7540
                                 <1>
                                           jne
5328
                                 <1>
                                          ;:
5329
                                 <1>
                                          ; Retro UNIX 386 v1 (32 bit) executable file header format
5330 0000E246 8B35[3C040300]
                                 <1>
                                           mov esi, [pcore]; start address of user's core memory
5331
                                 <1>
                                                             ; (phys. start addr. of the exec. file)
5332 0000E24C AD
                                           lodsd
                                 <1>
5333 0000E24D 663DEB1E
                                <1>
                                           cmp ax, 1EEBh; EBH, 1Eh -> jump to +32
5334 0000E251 7533
                                 <1>
                                                short sysexec 15
                                           jne
5335 0000E253 AD
                                 <1>
                                           lodsd
                                           mov
                                                 ecx, eax; text (code) section size
5336 0000E254 89C1
                                <1>
5337 0000E256 AD
                                 <1>
                                           lodsd
5338 0000E257 01C1
                                 <1>
                                           add ecx, eax; + data section size (initialized data)
5339 0000E259 89CB
                                <1>
                                           mov
                                                ebx, ecx
5340 0000E25B AD
                                <1>
                                          lodsd
5341 0000E25C 01C3
                                 <1>
                                           add
                                                 ebx, eax; + bss section size (for overrun checking)
5342 0000E25E 3B1D[88030300]
                                 <1>
                                           cmp
                                                 ebx, [u.count]
5343 0000E264 7711
                                 <1>
                                                 short sysexec_14 ; program overruns stack !
                                           jа
5344
                                 <1>
                                          ;
5345
                                 <1>
                                           ; add bss section size to [u.break]
5346 0000E266 0105[90030300]
                                 <1>
                                          add [u.break], eax
5347
                                 <1>
                                                ecx, 32 ; header size (already loaded)
5348 0000E26C 83E920
                                 <1>
                                          sub
5349
                                 <1>
                                          ;cmp ecx, [u.count]
5350
                                 <1>
                                          ;jnb short sysexec_16
5351 0000E26F 890D[88030300]
                                 <1>
                                          mov
                                                 [u.count], ecx; required read count
5352 0000E275 EB29
                                 <1>
                                          jmp
                                                 short sysexec_16
5353
                                 <1> sysexec_14:
5354
                                 <1>
                                          ; insufficient (out of) memory
5355 0000E277 C705[C8030300]0400- <1>
                                                dword [u.error], ERR_MINOR_IM ; 1
5355 0000E27F 0000
                                <1>
5356 0000E281 E9BCE4FFFF
                                 <1>
                                          jmp
                                                 error
5357
                                 <1> sysexec_15:
5358 0000E286 8B15[55040300]
                                      mov edx, [i.size]; file size
                                <1>
5359 0000E28C 29CA
                                <1>
                                           sub edx, ecx; file size - loaded bytes
5360 0000E28E 7626
                                 <1>
                                                 short sysexec_17 ; no need to next read
                                          jna
5361 0000E290 01D1
                                                ecx, edx ; [i.size]
                                <1>
                                           add
5362 0000E292 3B0D[88030300]
                                <1>
                                                 ecx, [u.count]; overrun check (!)
                                           cmp
5363 0000E298 77DD
                                 <1>
                                                 short sysexec_14
                                           jа
5364 0000E29A 8915[88030300]
                                 <1>
                                          mov
                                                 [u.count], edx
5365
                                 <1> sysexec_16:
5366 0000E2A0 A1[51040300]
                                 <1>
                                          mov
                                                 eax, [ii] ; first cluster
5367 0000E2A5 E889000000
                                 <1>
                                           call
5368 0000E2AA 8B0D[8C030300]
                                 <1>
                                          mov
                                                 ecx, [u.nread]
                                                 [u.break], ecx
5369 0000E2B0 010D[90030300]
                                 <1>
                                          add
                                 <1> sysexec 17:
5370
5371 0000E2B6 A1[51040300]
                                 <1>
                                         mov
                                                 eax, [ii] ; first cluster
5372 0000E2BB E889150000
                                 <1>
                                          call iclose
5373 0000E2C0 31C0
                                 <1>
                                          xor
                                                 eax, eax
5374 0000E2C2 FEC0
                                 <1>
                                          inc
                                                 al
5375 0000E2C4 66A3[AA030300]
                                 <1>
                                                [u.intr], ax ; 1 (interrupt/time-out is enabled)
                                          mov
5376 0000E2CA 66A3[AC030300]
                                 <1>
                                          mov
                                                [u.quit], ax ; 1 ('crtl+brk' signal is enabled)
5377 0000E2D0 833D[BC030300]00
                                 <1>
                                           cmp dword [u.ppgdir], 0 ; is the caller MainProg (kernel) ?
                                           ja short sysexec_18; no, the caller is user process
5378 0000E2D7 770C
                                 <1>
                                          ; If the caller is kernel (MainProg), 'sysexec' will come here
5379
                                 <1>
5380 0000E2D9 8B15[C0580100]
                                 <1>
                                           mov edx, [k_page_dir] ; kernel's page directory
5381 0000E2DF 8915[BC030300]
                                 <1>
                                          mov [u.ppgdir], edx; next time 'sysexec' must not come here
                                 <1> sysexec_18:
                                        ; 02/05/2016
5383
                                 <1>
                                           ; 24/04/2016 (TRDOS 386 = TRDOS v2.0)
5384
                                 <1>
                                          ; 18/10/2015 (Retro UNIX 386 v1)
5385
                                 <1>
                                         ; 05/08/2015
5386
                                 <1>
5387
                                 <1>
                                          ; 29/07/2015
5388
                                 <1>
                                          ; **** arguments list test start - 19/11/2017
5389
                                 <1>;
5390
                                 <1>;
                                          mov ebp, [argv]
                                                 ebp, ECORE - 4096
                                 <1>;
5391
                                           sub
5392
                                 <1>;
                                                ebp, [ecore]
5393
                                 <1>;
5394
                                 <1>;
                                          mov
                                                 ebx, [ebp]
5395
                                 <1>;
                                                 [argc], bx
                                          mov
                                          add
5396
                                 <1>;
                                                 ebp, 4
5397
                                 <1>;
                                           mov
                                                 byte [ccolor], 1Fh
                                 <1> ;_zx0:
5398
                                          cmp
5399
                                 <1> ;
                                                 word [argc], 0
                                                 short _zx2
5400
                                 <1>;
                                           jna
                                 <1> ;_zx1:
5401
                                 <1> ; push ebp
5402
5403
                                 <1>;
                                          mov
                                                esi, [ebp]
```

```
<1>;
5404
                                 <1>;
5405
                                           sub
                                                  esi, ECORE - 4096
                                 <1>;
5406
                                           add
                                                 esi, [ecore]
5407
                                 <1>;
5408
                                 <1>;
                                           call print_cmsg
5409
                                 <1>;
5410
                                 <1>;
                                           dec
                                                  word [argc]
                                 <1>;
5411
                                           jz
                                                 short _zx2
5412
                                 <1>;
                                                 al, '.'
5413
                                 <1>;
                                           mov
5414
                                 <1>;
                                           mov
                                                 bl, 07h
5415
                                 <1>;
                                           mov
                                                 bh, [u.ttyn]
5416
                                 <1>;
                                           call
                                                 _write_tty
5417
                                 <1>;
5418
                                 <1>;
                                                 ebp
                                           pop
                                                 ebp, 4
5419
                                 <1>;
                                           add
5420
                                 <1>;
                                           jmp
                                                 short _zx1
                                 <1> ;_zx2:
5421
5422
                                 <1>;
                                           pop
                                                  ebp
5423
                                 <1>;
                                                 byte [ccolor], 07h
                                           mov
                                                 eax, 1
5424
                                 <1>;
                                           ; **** arguments list test stop
5425
                                 <1>;
                                 <1>;
5426
                                           Test result is OK! (there is not a wrong thing) - 19/11/2017
5427
                                 <1>
5428 0000E2E5 8B2D[4C040300]
                                 <1>
                                                  ebp, [argv]; user's stack pointer must point to argument
5429
                                 <1>
                                                            ; list pointers (argument count)
5430 0000E2EB FA
                                 <1>
5431 0000E2EC 8B25[5C580100]
                                 <1>
                                           mov
                                                     esp, [tss.esp0]; ring 0 (kernel) stack pointer
5432
                                 <1>
                                           ;mov
                                                        esp, [u.sp] ; Restore Kernel stack
                                                           ; for this process
5433
                                 <1>
                                           ;add esp, 20; --> EIP, CS, EFLAGS, ESP, SS
5434
                                 <1>
5435
                                 <1>
                                           ;xor eax, eax; 0
5436 0000E2F2 FEC8
                                 <1>
                                           dec
                                                 al; eax = 0
5437
                                 <1>
                                           ;mov edx, UDATA
                                           ; 18/11/2017
5438
                                 <1>
5439 0000E2F4 6A23
                                 <1>
                                           push UDATA; user's stack segment
5440
                                 <1>
                                           ;push edx
                                           push ebp ; user's stack pointer
5441 0000E2F6 55
                                 <1>
                                                     ; (points to number of arguments)
5442
                                 <1>
5443
                                 <1>
5444
                                 <1>
                                           ; 04/01/2017
5445
                                 <1>
                                           ; MainProg comes here while [sysflg] = OFFh
5446
                                 <1>
                                           ; (but sysexec comes here while [sysflg] = 0)
5447 0000E2F7 C605[5B030300]00
                                 <1>
                                           mov byte [sysflg], 0; 04/01/2017
5448
                                 <1>
                                                               ; (timer_int sysflg control)
5449 0000E2FE FB
                                 <1>
                                           pushfd ; EFLAGS
5450 0000E2FF 9C
                                 <1>
                                                 ; Set IF for enabling interrupts in user mode
5451
                                 <1>
5452
                                 <1>
                                                 dword [esp], 200h
5453
                                 <1>
5454
                                 <1>
                                           ;mov bx, UCODE
5455
                                 <1>
                                           ; push bx ; user's code segment
5456 0000E300 6A1B
                                 <1>
                                           push UCODE
5457
                                 <1>
                                           ;push 0
                                           push eax ; EIP (=0) - start address -
5458 0000E302 50
                                 <1>
5459 0000E303 8925[5C030300]
                                 <1>
                                           mov
                                                 [u.sp], esp; 29/07/2015
5460
                                 <1>
                                           ; Remedy of a General Protection Fault during 'iretd' is here !
5461
                                 <1>
5462
                                 <1>
                                           ; ('push dx' would cause to general protection fault,
5463
                                 <1>
                                           ; after 'pop ds' etc.)
5464
                                 <1>
5465
                                 <1>
                                           ;; push dx ; ds (UDATA)
5466
                                 <1>
                                           ;; push dx ; es (UDATA)
5467
                                 <1>
                                           ;; push dx ; fs (UDATA)
5468
                                 <1>
                                           ;; push dx ; gs (UDATA)
5469
                                 <1>
5470
                                 <1>
                                           ; This is a trick to prevent general protection fault
5471
                                           ; during 'iretd' intruction at the end of 'sysrele' (in ul.s):
                                 <1>
5472 0000E309 66BA2300
                                 <1>
                                                 dx, UDATA ; 19/11/2017
5473 0000E30D 8EC2
                                 <1>
                                                 es, dx ; UDATA
                                           mov
5474 0000E30F 06
                                 <1>
                                           push es ; ds (UDATA)
                                           push es ; es (UDATA)
5475 0000E310 06
                                 <1>
5476 0000E311 06
                                 <1>
                                           push es; fs (UDATA)
                                           push es ; gs (UDATA)
5477 0000E312 06
                                 <1>
5478 0000E313 66BA1000
                                 <1>
                                                 dx, KDATA
                                           mov
5479 0000E317 8EC2
                                 <1>
                                           mov
                                                 es, dx
5480
                                 <1>
5481
                                 <1>
                                           ;; pushad simulation
5482 0000E319 89E5
                                 <1>
                                                 ebp, esp; esp before pushad
5483 0000E31B 50
                                           push eax ; eax (0)
                                 <1>
5484 0000E31C 50
                                 <1>
                                           push eax; ecx (0)
                                           push
5485 0000E31D 50
                                 <1>
                                                  eax ; edx (0)
5486 0000E31E 50
                                 <1>
                                           push eax; ebx (0)
5487 0000E31F 55
                                 <1>
                                           push ebp; esp before pushad
5488 0000E320 50
                                 <1>
                                           push eax ; ebp (0)
5489 0000E321 50
                                 <1>
                                           push
                                                 eax ; esi (0)
5490 0000E322 50
                                 <1>
                                                 eax ; edi (0)
                                           push
5491
                                 <1>
5492 0000E323 A3[64030300]
                                 <1>
                                                  [u.r0], eax ; eax = 0
5493 0000E328 8925[60030300]
                                 <1>
                                                  [u.usp], esp
                                           mov
5494
                                 <1>
5495
                                 <1>
                                           ; 14/11/2017
5496 0000E32E E931E4FFFF
                                 <1>
                                           jmp sysret0
5497
                                 <1>
                                           ; 02/05/2016
5498
                                 <1>;
5499
                                           ;inc byte [sysflg] ; OFFh -> 0
                                 <1>;
5500
                                 <1>;
                                           ;mov byte [sysflg], 0 ; 04/01/2017
                                 <1>;
5501
                                           movzx ebx, byte [u.uno]
5502
                                 <1>;
                                           shl
                                                 bl, 1 ; 13/11/2017
                                                word [ebx+p.ppid-2], 1 ; MainProg
5503
                                 <1>;
                                           cmp
5504
                                 <1>;
                                           jа
                                                 sysret0 ; 03/05/2016
                                 <1>;
5505
                                           push sysret; *
5506
                                 <1>;
                                           mov
                                                 [u.usp], esp
5507
                                 <1>;
                                           call wswap; save child process 'u' structure and
5508
                                 <1>;
                                                     ; registers
```

```
5509
                                  <1>;
                                         add dword [u.usp], 4; 03/05/2016
5510
                                  <1> ;sysexec 19: ; 02/05/2016
                                           retn ; * 'sysret' ; byte [sysflg] -> OFFh
5511
                                  <1>;
5512
                                  <1>
5513
                                  <1> readi:
5514
                                  <1>
                                           ; 01/05/2016
                                           ; 25/04/2016 - TRDOS 386 (TRDOS v2.0)
5515
                                          ; 20/05/2015 - Retro UNIX 386 v1
5516
                                  <1>
5517
                                  <1>
                                           ; 11/03/2013 - 31/07/2013 (Retro UNIX 8086 v1)
5518
                                  <1>
5519
                                  <1>
                                           ; Reads from a file whose the first cluster number in EAX
5520
                                  <1>
                                           ; INPUTS ->
5521
                                  <1>
                                           ; EAX - First cluster number of the file
5522
                                  <1>
5523
                                  <1>
                                                u.count - byte count user desires
                                                u.base - points to user buffer
5524
                                  <1>
                                               u.fofp - points to dword with current file offset
5525
                                  <1>
                                            ; i.size - file size
                                  <1>
5526
5527
                                  <1>
                                                cdev - logical dos drive number of the file
5528
                                  <1>
                                            ; OUTPUTS ->
                                                u.count - cleared
5529
                                  <1>
5530
                                  <1>
                                                u.nread - accumulates total bytes passed back
                                           ;
                                  <1>
5531
5532
                                  <1>
                                           ; ((EAX)) input/output
                                            ; (Retro UNIX Prototype : 14/12/2012 - 01/03/2013, UNIXCOPY.ASM)
5533
                                  <1>
5534
                                  <1>
                                             ; ((Modified registers: edx, ebx, ecx, esi, edi))
                                  <1>
5536 0000E333 31D2
                                  <1>
                                           xor
                                                   edx, edx; 0
5537 0000E335 8915[8C030300]
                                  <1>
                                                   [u.nread], edx; 0
                                            mov
5538 0000E33B 668915[C4030300]
                                                   [u.pcount], dx ; 19/05/2015
                                  <1>
                                            mov
5539 0000E342 3915[88030300]
                                  <1>
                                            cmp
                                                   [u.count], edx; 0
5540 0000E348 7701
                                  <1>
                                            jа
                                                  short readi 1
5541 0000E34A C3
                                  <1>
                                            retn
5542
                                  <1> readi 1:
5543
                                  <1> dskr:
5544
                                  <1>
                                            ; 01/05/2016
                                            ; 25/04/2016 - TRDOS 386 (TRDOS v2.0)
5545
                                  <1>
                                           ; 24/05/2015 - 12/10/2015 (Retro UNIX 386 v1)
; 26/04/2013 - 03/08/2013 (Retro UNIX 8086 v1)
5546
                                  <1>
5547
                                  <1>
                                  <1> dskr_0:
5548
5549 0000E34B 8B15[55040300]
                                 <1>
                                            mov edx, [i.size]
                                                 ebx, [u.fofp] edx, [ebx]
5550 0000E351 8B1D[74030300]
                                  <1>
                                           mov
5551 0000E357 2B13
                                  <1>
                                            sub
5552 0000E359 7647
                                  <1>
                                                  short dskr_4
                                            jna
5553
                                  <1>
                                           ;
5554 0000E35B 50
                                  <1>
                                            push
                                                  eax ; 01/05/2016
5555 0000E35C 3B15[88030300]
                                  <1>
                                            cmp
                                                  edx, [u.count]
5556 0000E362 7306
                                                  short dskr_1
                                  <1>
                                            jnb
5557 0000E364 8915[88030300]
                                  <1>
                                            mov
                                                  [u.count], edx
                                  <1> dskr_1:
5558
                                          ; EAX = First Cluster
5559
                                  <1>
5560
                                  <1>
                                           ; [Current Drv] = Physical drive number
5561 0000E36A E83B000000
                                           call mget_r
                                  <1>
                                  <1>
                                           ; NOTE: in 'mget_r', relevant sector will be read in buffer
5562
5563
                                  <1>
                                           ; if it is not already in buffer !
5564 0000E36F BB[8C050300]
                                  <1>
                                            mov ebx, readi_buffer
5565 0000E374 803D[C6030300]00
                                                  byte [u.kcall], 0; the caller is 'namei' sign (=1)
                                  <1>
                                            cmp
5566 0000E37B 770F
                                  <1>
                                                  short dskr_3 ; zf=0 -> the caller is 'namei'
                                            jа
5567 0000E37D 66833D[C4030300]00
                                 <1>
                                                  word [u.pcount], 0
                                            cmp
5568 0000E385 7705
                                  <1>
                                            jа
                                                  short dskr_3
5569
                                  <1> dskr_2:
                                  <1>
                                            ; [u.base] = virtual address to transfer (as destination address)
5571 0000E387 E894010000
                                  <1>
                                            call trans_addr_w ; translate virtual address to physical (w)
5572
                                  <1> dskr 3:
                                           ; EBX (r5) = system (I/O) buffer address -physical-
5573
                                  <1>
5574 0000E38C E8F7010000
                                  <1>
                                            call sioreg
5575 0000E391 87F7
                                  <1>
                                           xchg esi, edi
5576
                                  <1>
                                           ; EDI = file (user data) offset
5577
                                  <1>
                                           ; ESI = sector (I/O) buffer offset
5578
                                  <1>
                                           ; ECX = byte count
5579 0000E393 F3A4
                                  <1>
                                            rep movsb
                                           ; eax = remain bytes in buffer
5580
                                  <1>
5581
                                  <1>
                                                    (check if remain bytes in the buffer > [u.pcount])
                                             ;
5582 0000E395 09C0
                                  <1>
                                            or
                                                  short dskr_2; (page end before system buffer end!)
5583 0000E397 75EE
                                  <1>
                                            jnz
                                                  eax ; (first cluster number)
5584 0000E399 58
                                  <1>
                                            pop
5585 0000E39A 390D[88030300]
                                  <1>
                                                  [u.count], ecx; 0
                                            cmp
5586 0000E3A0 77A9
                                  <1>
                                            jа
                                                  short dskr_0
5587
                                  <1> dskr 4:
5588 0000E3A2 C605[C6030300]00
                                                  byte [u.kcall], 0
                                  <1>
                                           mov
5589 0000E3A9 C3
                                  <1>
                                            retn
                                  <1>
                                  <1> mget_r:
5591
                                           ; 24/10/2016
5592
                                  <1>
                                           ; 22/10/2016
5593
                                  <1>
5594
                                  <1>
                                            ; 12/10/2016
5595
                                  <1>
                                           ; 29/04/2016
5596
                                  <1>
                                           ; 25/04/2016 - TRDOS 386 (TRDOS v2.0)
                                           ; 03/06/2015 (Retro UNIX 386 v1, 'mget', u.5s)
5597
                                  <1>
5598
                                  <1>
                                           ; 22/03/2013 - 31/07/2013 (Retro UNIX 8086 v1)
5599
                                  <1>
5600
                                  <1>
                                           ; Get existing or (allocate) a new disk block for file
5601
                                  <1>
                                           ; INPUTS ->
5602
                                  <1>
                                                [u.fofp] = file offset pointer
5603
                                  <1>
5604
                                  <1>
                                                 EAX = First Cluster
                                                 [cdev] = Logical dos drive number
5605
                                  <1>
                                                 ([u.off] = file offset)
                                  <1>
5606
5607
                                  <1>
                                            ; OUTPUTS ->
                                            ; EAX = logical sector number
                                  <1>
5608
5609
                                  <1>
                                                 ESI = Logical Dos Drive Description Table address
5610
                                  <1>
5611
                                  <1>
                                            ; Modified registers: EDX, EBX, ECX, ESI, EDI
                                  <1>
5612
5613 0000E3AA 8B35[74030300]
                                  <1>
                                                    esi, [u.fofp]
                                            mov
```

```
5614 0000E3B0 8B1E
                                                 ebx, [esi]; (u.off)
                                 <1>
                                           mov
5615
                                 <1>
5616 0000E3B2 29C9
                                 <1>
                                           sub
                                                  ecx, ecx
5617 0000E3B4 8A2D[46030300]
                                 <1>
                                                 ch, [cdev]
5618
                                 <1>
5619 0000E3BA BE00010900
                                 <1>
                                                  esi, Logical_DOSDisks
                                           mov
5620 0000E3BF 01CE
                                 <1>
                                           add
                                                 esi, ecx
5621
                                 <1>
5622 0000E3C1 380D[F4640100]
                                 <1>
                                                  [readi.valid], cl ; 0
                                           cmp
5623 0000E3C7 7649
                                 <1>
                                           jna
                                                 short mget r 0
5624
                                 <1>
5625 0000E3C9 3A2D[F5640100]
                                 <1>
                                                 ch, [readi.drv]
                                           cmp
5626 0000E3CF 7541
                                 <1>
                                                 short mget_r_0
                                           jne
5627
                                 <1>
5628 0000E3D1 3B05[08650100]
                                 <1>
                                                 eax, [readi.fclust]
                                           cmp
                                                 short mget r 3
5629 0000E3D7 7565
                                 <1>
                                           jne
                                 <1>
5631 0000E3D9 89D8
                                 <1>
                                                 eax, ebx ; file offset
                                           mov
5632 0000E3DB 668B0D[FC640100]
                                 <1>
                                                 cx, [readi.bpc]
                                           mov
5633 0000E3E2 41
                                 <1>
                                                 ecx ; <= 65536
                                           inc
5634 0000E3E3 29D2
                                 <1>
                                           sub
                                                 edx, edx
5635 0000E3E5 F7F1
                                 <1>
                                           div
5636
                                 <1>
5637 0000E3E7 8B3D[04650100]
                                 <1>
                                                 edi, [readi.c_index] ; cluster index
5638
                                 <1>
5639 0000E3ED 39F8
                                 <1>
                                           cmp
                                                eax, edi
5640 0000E3EF 757A
                                 <1>
                                           jne short mget_r_4 ; (*)
5641
                                 <1>
5642
                                           ; edx = byte offset in cluster (<= 65535)
                                 <1>
5643 0000E3F1 668915[FE640100]
                                          mov [readi.offset], dx
                                 <1>
5644 0000E3F8 66C1EA09
                                 <1>
                                           shr dx, 9; / 512
5645 0000E3FC 8815[F7640100]
                                 <1>
                                          mov
                                                 [readi.s index], dl; sector index in cluster (0 to spc -1)
5646
                                 <1>
5647 0000E402 A1[00650100]
                                 <1>
                                                 eax, [readi.cluster] ; > 0 if [readi.valid] = 1
                                           mov
                                           mov edx, [readi.fs_index]
5648 0000E407 8B15[0C650100]
                                 <1>
                                           jmp
5649 0000E40D E99A000000
                                 <1>
                                                  mget_r_7
5650
                                 <1>
                                 <1> mget_r_0:
5651
5652 0000E412 882D[F5640100]
                                 <1>
                                                 [readi.drv], ch ; physical drive number
                                           mov
                                                 byte [esi+LD_FATType], 0
5653 0000E418 807E0300
                                <1>
                                           cmp
5654 0000E41C 7707
                                 <1>
                                                 short mget_r_1
                                                 cl, [esi+LD_FS_BytesPerSec+1]
cl, 1; ; 1 for 512 bytes, 4 for 2048 bytes
5655 0000E41E 8A4E12
                                 <1>
                                           mov
5656 0000E421 D0E9
                                <1>
                                           shr
5657 0000E423 EB03
                                 <1>
                                                 short maet r 2
                                           jmp
                                 <1> mget_r_1:
5658
5659 0000E425 8A4E13
                                 <1>
                                           mov
                                                 cl, [esi+LD_BPB+BPB_SecPerClust]
                                 <1> mget r 2:
5660
5661 0000E428 880D[F6640100]
                                                 [readi.spc], cl ; sectors per cluster
                                 <1>
                                           mov
5662
                                 <1>
                                           ; NOTE: readi bytes per sector value is always 512 !
5663 0000E42E 66C1E109
                                           shl cx, 9; * 512
                                 <1>
5664 0000E432 6649
                                 <1>
                                           dec
                                                 cx ; bytes per cluster - 1
5665 0000E434 66890D[FC640100]
                                 <1>
                                           mov
                                                 [readi.bpc], cx
5666 0000E43B 6629C9
                                 <1>
                                          sub
                                                 CX, CX
5667
                                 <1> mget_r_3:
5668 0000E43E A3[08650100]
                                 <1> mov
                                                 [readi.fclust], eax ; first cluster (or FDT address)
5669 0000E443 880D[F4640100]
                                 <1>
                                                  [readi.valid], cl ; 0
                                           mov
5670
                                 <1>
                                                [readi.s index], cl ; 0
                                          ;mov
5671
                                 <1>
                                                [readi.offset], cx ; 0
                                          ;mov
5672 0000E449 890D[04650100]
                                 <1>
                                                  [readi.c_index], ecx ; 0
                                          mov
5673 0000E44F 890D[00650100]
                                 <1>
                                                 [readi.cluster], ecx; 0
                                          mov
5674 0000E455 890D[F8640100]
                                 <1>
                                                 [readi.sector], ecx ; 0
                                          mov
                                 <1>
5676 0000E45B 89D8
                                 <1>
                                                 eax, ebx ; file offset
                                          mov
5677 0000E45D 668B0D[FC640100]
                                 <1>
                                                 cx, [readi.bpc]
                                           mov
5678 0000E464 41
                                 <1>
                                                 ecx ; <= 65536
                                           inc
5679 0000E465 29D2
                                 <1>
                                           sub
                                                 edx, edx
5680 0000E467 F7F1
                                 <1>
                                           div
                                                 ecx
5681
                                 <1>
                                                 edi, [readi.c_index] ; previous cluster index
                                           ; mov
5682 0000E469 29FF
                                 <1>
                                           sub
                                                 edi, edi
5683
                                 <1> mget_r_4:
5684 0000E46B A3[04650100]
                                 <1>
                                                 [readi.c_index], eax ; cluster index
                                           ; edx = byte offset in cluster (<= 65535)
5685
                                 <1>
5686 0000E470 668915[FE640100]
                                           mov [readi.offset], dx
                                 <1>
5687 0000E477 66C1EA09
                                 <1>
                                                 dx, 9; / 512
                                                 [readi.s_index], dl ; sector index in cluster (0 to spc -1)
5688 0000E47B 8815[F7640100]
                                 <1>
                                          mov
5689
                                 <1>
5690 0000E481 89C1
                                 <1>
                                                 ecx, eax; current cluster index
                                          mov
5691 0000E483 A1[08650100]
                                 <1>
                                           mov
                                                 eax, [readi.fclust]
5692 0000E488 09C9
                                 <1>
                                                 ecx, ecx; cluster index
                                           or
5693 0000E48A 741B
                                 <1>
                                                 short mget_r_6
                                           jΖ
5694
                                 <1>
5695 0000E48C 39CF
                                 <1>
                                           cmp
                                                 edi, ecx
5696 0000E48E 7710
                                                  short mget_r_5; old cluster index is higher
                                <1>
                                           jа
5697 0000E490 8B15[00650100]
                                 <1>
                                                  edx, [readi.cluster]
5698 0000E496 21D2
                                 <1>
                                                 edx, edx
                                           and
5699 0000E498 7406
                                 <1>
                                           jz
                                                 short mget_r_5
                                          ; valid 'readi' parameters (*)
                                <1>
5701 0000E49A 89D0
                                <1>
                                          mov eax, edx
5702 0000E49C 29F9
                                 <1>
                                                 ecx, edi
                                           sub
5703 0000E49E 740C
                                 <1>
                                                 short mget_r_7
                                          iΖ
5704
                                 <1> mget_r_5:
                                          ; EAX = Beginning cluster
5705
                                 <1>
                                           ; EDX = Sector index in disk/file section
5706
                                 <1>
                                                 (Only for SINGLIX file system!)
5707
                                 <1>
                                         ; ECX = Cluster sequence number after the beginning cluster
5708
                                 <1>
5709
                                 <1>
                                           ; ESI = Logical DOS Drive Description Table address
5710 0000E4A0 E836E1FFFF
                                 <1>
                                           call get_cluster_by_index
5711 0000E4A5 724E
                                 <1>
                                           jc short mget_r_err
5712
                                 <1>
                                           ; EAX = Cluster number
5713
                                 <1> mget_r_6:
5714 0000E4A7 A3[00650100]
                                <1>
                                           mov
                                                  [readi.cluster], eax; FDT number for Singlix File System
5715
                                 <1> mget_r_7:
5716 0000E4AC 807E0300
                                <1>
                                           cmp
                                                 byte [esi+LD_FATType], 0
5717 0000E4B0 765F
                                 <1>
                                                 short mget r 12
                                           jna
5718
                                 <1>
```

```
5719 0000E4B2 83E802
                                       sub eax, 2
                                <1>
5720 0000E4B5 0FB615[F6640100]
                               <1>
                                         movzx edx, byte [readi.spc]
5721 0000E4BC F7E2
                                <1>
                                         mul edx
5722
                                <1>
                                               eax, [esi+LD_DATABegin]
5723 0000E4BE 034668
                                <1>
                                         add
5724 0000E4C1 8A15[F7640100]
                                <1>
                                         mov
                                                dl, [readi.s_index]
5725 0000E4C7 01D0
                                               eax, edx
                                <1>
                                         add
5726
                                <1> mget_r_8:
5727
                                <1>
                                         ; eax = logical sector number
5728 0000E4C9 803D[F4640100]00
                                               byte [readi.valid], 0
                               <1>
                                         cmp
5729 0000E4D0 7608
                                <1>
                                               short mget_r_9
5730 0000E4D2 3B05[F8640100]
                                <1>
                                          cmp
                                               eax, [readi.sector]
                               <1>
5731 0000E4D8 7436
                                               short mget_r_11 ; sector is already in 'readi' buffer
                                         jе
                                <1> mget r 9:
5732
5733 0000E4DA A3[F8640100]
                               <1> mov
                                               [readi.sector], eax
                                               ebx, readi buffer; buffer address
5734 0000E4DF BB[8C050300]
                                <1>
                                         mov
5735 0000E4E4 B901000000
                                <1>
                                         mov ecx, 1
                                      ; 29/04/2016
                                <1>
5736
5737
                                <1>
                                         ;xor dl, dl
5738
                                <1>
5739
                                <1>
                                        ; EAX = Logical sector number
5740
                                <1>
                                         ; ECX = Sector count
                                         ; EBX = Buffer address
5741
                                <1>
                                         ; (EDX = 0)
5742
                                <1>
                                         ; ESI = Logical DOS drive description table address
5743
                                <1>
5744
                                <1>
5745 0000E4E9 E86E130000
                                <1>
                                        call disk read
5746 0000E4EE 7314
                                <1>
                                        jnc short mget_r_10
5747
                                <1>
5748
                                         ; 22/10/2016 (15h -> 17)
                                <1>
5749 0000E4F0 B811000000
                                <1>
                                        mov
                                               eax, 17; Drive not ready or read error!
5750
                                <1> mget r err:
5751 0000E4F5 A3[C8030300]
                                                [u.error], eax
                                <1>
                                         mov
5752
                                <1>
                                         ; 12/10/2016
5753 0000E4FA A3[64030300]
                                <1>
                                     jmp
                                         mov
                                               [u.r0], eax
5754 0000E4FF E93EE2FFFF
                                <1>
                                                error
                                <1> mget_r_10:
                               <1>
                                               byte [readi.valid], 1 ; 24/10/2016
5756 0000E504 C605[F4640100]01
                                         mov
5757 0000E50B A1[F8640100]
                                <1>
                                         mov
                                               eax, [readi.sector]
                                <1> mget_r_11:
5758
5759 0000E510 C3
                                <1>
                                        retn
5760
                                <1> mget_r_12:
                                <1> ; EAX = FDT number
<1> : EDX = Sector index
5761
5762
                                <1>
                                         ; EDX = Sector index from FDT sector (0,1,2,3,4...)
                                         inc eax; the first data sector in FS disk section
5763 0000E511 40
                                <1>
5764 0000E512 8915[0C650100]
                                <1>
                                         mov
                                               [readi.fs_index], edx
5765 0000E518 01D0
                                         add eax, edx
                               <1>
5766 0000E51A EBAD
                                <1>
                                               short mget_r_8
                                        jmp
5767
                                <1>
                                <1> trans_addr_r:
5768
                                      ; 12/10/2016
5769
                                <1>
5770
                                <1>
                                         ; 02/05/2016 - TRDOS 386 (TRDOS v2.0)
5771
                                <1>
                                         ; Translate virtual address to physical address
5772
                                <1>
                                        ; for reading from user's memory space
5773
                                <1>
                                        ; 04/06/2015 - 18/10/2015 (Retro UNIX 386 v1)
5774
                                <1>
5775 0000E51C 31D2
                                <1>
                                         xor edx, edx; 0 (read access sign)
5776 0000E51E EB04
                                <1>
                                        jmp short trans_addr_rw
5777
                                <1>
                                <1> trans_addr_w:
5778
                                       ; 12/10/2016
5779
                                <1>
5780
                                <1>
                                         ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
5781
                                <1>
                                         ; Translate virtual address to physical address
5782
                                <1>
                                       ; for writing to user's memory space
5783
                                <1>
                                        ; 04/06/2015 - 18/10/2015 (Retro UNIX 386 v1)
5784
                                <1>
5785 0000E520 29D2
                                <1>
                                         sub
                                               edx, edx
5786 0000E522 FEC2
                               <1>
                                         inc dl; 1 (write access sign)
5787
                                <1> trans_addr_rw:
5788 0000E524 50
                               <1>
                                         push eax
5789 0000E525 53
                               <1>
                                         push ebx
5790 0000E526 52
                                         push edx ; r/w sign (in DL)
                                <1>
5791
                                <1>
5792 0000E527 8B1D[84030300]
                               <1>
                                       mov
                                               ebx, [u.base]
5793 0000E52D E8BA6DFFFF
                                         call get_physical_addr ; get physical address
                                <1>
                                               short passc 0
5794 0000E532 730F
                                <1>
                                         jnc
5795 0000E534 A3[C8030300]
                                <1>
                                               [u.error], eax
                                         mov
                                               [u.r0], eax ; 12/10/2016
5796 0000E539 A3[64030300]
                                <1>
                                         mov
                                         ;pop edx
5797
                                <1>
5798
                                <1>
                                         ;pop ebx
5799
                                <1>
                                          ;pop
                                               eax
5800 0000E53E E9FFE1FFFF
                                          jmp
                                <1>
                                                error
                               <1> passc_0:
5802 0000E543 F6C202
                                         test dl, PTE_A_WRITE; writable page
                               <1>
5803 0000E546 5A
                               <1>
                                         pop
                                               edx
5804 0000E547 751C
                               <1>
                                         jnz
                                               short passc_1
                               <1>
                            <1>
<1>
5806 0000E549 20D2
                                        and dl, dl
                                       ...a
j z
5807 0000E54B 7418
                                                short passc_1
                                         ; read only (duplicated) page -must be copied to a new page-
5808
                               <1>
5809
                               <1>
                                       ; EBX = linear address
5810 0000E54D 51
5811 0000E54E E8326AFFFF
                                       push ecx
                               <1>
                               <1>
                                         call copy_page
                               <1>
                                       pop
                                               ecx
                                     jc short passc_2
push eax; physical address of the new/allocated page
call add_to_swap_queue
5813 0000E554 721E
                               <1>
                               <1>
5814 0000E556 50
5815 0000E557 E8BA6CFFFF <1> 5816 0000E55C 58 <1>
                                     pop
                                               eax
                                               ebx, PAGE OFF ; OFFFh
5817 0000E55D 81E3FF0F0000
                               <1>
                                         and
                                         ;mov ecx, PAGE SIZE
5818
                               <1>
5819
                                       ; sub ecx, ebx
                               <1>
5820 0000E563 01D8
                                <1>
                                         add
                                               eax, ebx
5821
                               <1> passc_1:
5822 0000E565 A3[C0030300]
                               <1>
                                                [u.pbase], eax ; physical address
5823 0000E56A 66890D[C4030300] <1>
                                               [u.pcount], cx; remain byte count in page (1-4096)
                                         mov
```

```
ebx
                                           pop
5825 0000E572 58
                                 <1>
                                           pop
                                                  eax
5826 0000E573 C3
                                 <1>
                                           retn
5827
                                 <1> passc_2:
5828 0000E574 B804000000
                                                  eax, ERR MINOR IM ; "Insufficient memory !" error
                                 <1>
                                           mov
5829 0000E579 A3[64030300]
                                 <1>
                                           mov
                                                  [u.r0], eax; 12/10/2016
5830 0000E57E A3[C8030300]
                                 <1>
                                           mov
                                                  dword [u.error], eax
5831
                                 <1>
                                           ;pop
                                                ebx
5832
                                 <1>
                                           ;pop
                                                  eax
5833 0000E583 E9BAE1FFFF
                                 <1>
                                                  error
                                           jmp
5834
                                 <1>
5835
                                 <1> sioreg:
                                          ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
5836
                                 <1>
5837
                                 <1>
                                           ; 19/05/2015 - 25/07/2015 (Retro UNIX 386 v1)
5838
                                 <1>
                                           ; 12/03/2013 - 22/07/2013 (Retro UNIX 8086 v1)
5839
                                 <1>
                                           ; INPUTS ->
5840
                                  <1>
                                                 EBX = system buffer (data) address (r5)
                                 <1>
5841
                                                 [u.fofp] = pointer to file offset pointer
5842
                                 <1>
                                                 [u.base] = virtual address of the user buffer
                                                 [u.pbase] = physical address of the user buffer
5843
                                 <1>
5844
                                 <1>
                                                 [u.count] = byte count
                                                 [u.pcount] = byte count within page frame
5845
                                  <1>
                                           ;
                                 <1>
                                           ; OUTPUTS ->
5846
5847
                                 <1>
                                                 ESI = user data offset (r1)
                                 <1>
                                                 EDI = system (I/O) buffer offset (r2)
5848
                                                 ECX = byte count (r3)
5849
                                 <1>
5850
                                 <1>
                                                 EAX = remain bytes after byte count within page frame
                                 <1>
                                                  (If EAX > 0, transfer will continue from the next page)
5851
5852
                                 <1>
                                 <1>
                                           ; ((Modified registers: EDX))
5853
5854
                                 <1>
5855 0000E588 8B35[74030300]
                                 <1>
                                                     esi, [u.fofp]
                                             mov
                                                     edi, [esi]
5856 0000E58E 8B3E
                                 <1>
                                            mov
5857 0000E590 89F9
                                 <1>
                                           mov ecx, edi
                                                 ecx, OFFFFFE00h
5858 0000E592 81C900FEFFFF
                                 <1>
                                           or
5859 0000E598 81E7FF010000
                                                 edi, 1FFh
                                 <1>
                                           and
5860 0000E59E 01DF
                                 <1>
                                           add
                                                 edi, ebx ; EBX = system buffer (data) address
5861 0000E5A0 F7D9
                                 <1>
                                           neg
                                                 ecx
                                                  ecx, [u.count]
5862 0000E5A2 3B0D[88030300]
                                 <1>
                                           cmp
5863 0000E5A8 7606
                                                 short sioreg 0
                                 <1>
                                           jna
5864 0000E5AA 8B0D[88030300]
                                 <1>
                                           mov
                                                  ecx, [u.count]
5865
                                 <1> sioreg 0:
5866 0000E5B0 803D[C6030300]00
                                                 byte [u.kcall], 0
                                 <1>
                                           cmp
5867 0000E5B7 7613
                                 <1>
                                                 short sioreg_1
                                           ; the caller is 'mkdir' or 'namei'
5868
                                 <1>
5869 0000E5B9 A1[84030300]
                                 <1>
                                           mov eax, [u.base]
5870 0000E5BE A3[C0030300]
                                 <1>
                                                  [u.pbase], eax ; physical address = virtual address
                                           mov
5871 0000E5C3 66890D[C4030300]
                                 <1>
                                                 word [u.pcount], cx ; remain bytes in buffer (1 sector)
                                           mov
5872 0000E5CA EB0B
                                 <1>
                                           jmp
                                                  short sioreg 2
                                 <1> sioreg_1:
5874 0000E5CC 0FB715[C4030300]
                                 <1>
                                           movzx edx, word [u.pcount]
5875 0000E5D3 39D1
                                 <1>
                                           cmp
                                                 ecx, edx
5876 0000E5D5 772A
                                 <1>
                                           jа
                                                  short sioreg_4 ; transfer count > [u.pcount]
5877
                                 <1> sioreg_2: ; 2:
5878 0000E5D7 31C0
                                 <1>
                                                 eax, eax
                                           xor
5879
                                 <1> sioreg_3:
5880 0000E5D9 010D[8C030300]
                                 <1> add
                                                 [u.nread], ecx
5881 0000E5DF 290D[88030300]
                                           sub
                                 <1>
                                                 [u.count], ecx
5882 0000E5E5 010D[84030300]
                                 <1>
                                                  [u.base], ecx
                                           add
5883 0000E5EB 010E
                                 <1>
                                           add [esi], ecx
5884 0000E5ED 8B35[C0030300]
                                 <1>
                                          mov esi, [u.pbase]
5885 0000E5F3 66290D[C4030300]
                                 <1>
                                           sub
                                                  [u.pcount], cx
5886 0000E5FA 010D[C0030300]
                                 <1>
                                                 [u.pbase], ecx
                                           add
                                         retn
5887 0000E600 C3
                                 <1>
5888
                                 <1> sioreg_4:
5889
                                 <1>
                                         ; transfer count > [u.pcount]
5890
                                 <1>
                                           ; (ecx > edx)
5891 0000E601 89C8
                                 <1>
                                           mov
                                                 eax, ecx
5892 0000E603 29D0
                                 <1>
                                                  eax, edx; remain bytes for 1 sector (block) transfer
5893 0000E605 89D1
                                 <1>
                                                 ecx, edx ; current transfer count = [u.pcount]
                                           mov
                                                short sioreg 3
5894 0000E607 EBD0
                                 <1>
                                           jmp
5895
                                 <1>
                                 <1> tswitch: ; Retro UNIX 386 v1
5896
                                 <1> tswap:
5897
                                           ; 16/01/2017
5898
                                 <1>
                                           ; 21/05/2016 - TRDOS 386 (TRDOS v2.0)
5899
                                 <1>
5900
                                  <1>
                                           ; 10/05/2015 - 01/09/2015 (Retro UNIX 386 v1)
                                           ; 14/04/2013 - 14/02/2014 (Retro UNIX 8086 v1)
5901
                                 <1>
5902
                                  <1>
                                           ; time out swap, called when a user times out.
5903
                                           ; the user is put on the low priority queue.
                                  <1>
5904
                                  <1>
                                           ; This is done by making a link from the last user
                                           ; on the low priority queue to him via a call to 'putlu'.
5905
                                  <1>
5906
                                  <1>
                                           ; then he is swapped out.
5907
                                  <1>
5908
                                  <1>
                                           ; TRDOS 386 (TRDOS v2.0) modification -> ** 21/05/2016 **
5909
                                  <1>
                                                  * when a high priority (event) process will be stopped
5910
                                  <1>
                                                  (swapped out, swithched out/off), 'tswap/tswitch' will
5911
                                  <1>
                                                  not add it to a run queue.
5912
                                  <1>
                                                  /// What for: Process may be already in a run queue,
5913
                                  <1>
                                                  it is unspeficied state because process might be started
5914
                                  <1>
                                                  by a timer event which does not regard previous priority
5915
                                  <1>
                                                  level and run queue of the process (for fast executing!).
                                                  After the 'run for event', process will be sequenced
5916
                                 <1>
5917
                                  <1>
                                                  to run by it's actual run queue. ///
5918
                                  <1>
5919
                                  <1>
                                           ; Retro UNIX 386 v1 modification ->
                                                   swap (software task switch) is performed by changing
5920
                                  <1>
5921
                                 <1>
                                                  user's page directory (u.pgdir) instead of segment change
5922
                                  <1>
                                                  as in Retro UNIX 8086 v1.
5923
                                 <1>
5924
                                  <1>
                                           ; RETRO UNIX 8086 v1 modification ->
5925
                                  <1>
                                                   'swap to disk' is replaced with 'change running segment'
5926
                                 <1>
                                                  according to 8086 cpu (x86 real mode) architecture.
                                                  pdp-11 was using 64KB uniform memory while IBM PC
5927
                                  <1>
5928
                                  <1>
                                                  compatibles was using 1MB segmented memory
```

5824 0000E571 5B

```
in 8086/8088 times.
5929
                                  <1>
5930
                                   <1>
                                            ; INPUTS ->
5931
                                  <1>
5932
                                  <1>
                                            ; u.uno - users process number
5933
                                  <1>
                                                 runq+4 - lowest priority queue
                                            ; OUTPUTS ->
5934
                                  <1>
5935
                                  <1>
                                            ; r0 - users process number
5936
                                  <1>
                                                 r2 - lowest priority queue address
5937
                                  <1>
5938
                                            ; ((AX = R0, BX = R2)) output
                                  <1>
5939
                                  <1>
                                             ; ((Modified registers: EDX, EBX, ECX, ESI, EDI))
5940
                                   <1>
5941
                                  <1>
5942
                                  <1>
                                            NOTE:
5943
                                  <1>
                                            ;* [u.pri] priority level is specified by run queue which is process
5944
                                  <1>
                                             ; comes to run from.
                                            ;* Initial [u.pri] is 1 ('normal/regular') for programs
5945
5946
                                  <1>
                                            ; (which are launched by MainProg or 'sysexec'), it is changed
5947
                                  <1>
                                            ; to 2 ('high') by timer event, if program uses 'systimer' system call.
5948
                                  <1>
                                            ;* Program (Process) also can change it's running priority
5949
                                  <1>
                                            ; from 1 to 0 or up to 2 by using 'syspri' system call; but,
                                            ; if program selects priority level 2 (high) for running, next time
5950
                                   <1>
5951
                                  <1>
                                            ; it is reduced to 1 (normal/regular) because 'syspri' adds this
5952
                                  <1>
                                            ; program to 'run for normal' queue while running duration is a bit
                                            ; protected from swap/switch out immediate, behalf of other high
5953
                                  <1>
5954
                                  <1>
                                             ; priority process in sequence. Program (with high priority) will not
5955
                                  <1>
                                            ; be swapped/switched out (by timer event) before it's time quantum
5956
                                  <1>
                                            ; will be elapsed, but, this will be temporary if program is not using
5957
                                  <1>
                                            ; timer event function.
5958
                                  <1>
5959
                                  <1>
                                             ; For example:
5960
                                   <1>
                                             ; If a process frequently gets a timer event, it runs at high priority
                                             ; level but when it returns from running it returns to actual run queue,
5961
                                  <1>
5962
                                  <1>
                                             ;not to 'run for event' queue again.
5963
                                  <1>
                                             ;'tswap' will not change the sequence at return/stop(swap out) stage.
                                             ;But if priority level not high (=2, 'run for event'), 'tswap/tswitch'
5964
                                  <1>
                                             ; will add the stopping process to relevant run queue according to
5965
                                   <1>
5966
                                  <1>
                                            ;[u.pri] priority level.
5967
                                  <1>
5968
                                  <1>
                                            ; 16/01/2017
5969 0000E609 BB[54030300]
                                            mov ebx, runq+2 ; 'runq normal' ; normal/regular priority
                                  <1>
5970
                                   <1>
                                            ; 21/05/2016
                                            ;cmp byte [u.pri], 2 ; high priority (run for event) ?
5971
                                  <1>
5972
                                  <1>
                                            ; jnb short swap
                                            ; 16/01/2017
5973
                                   <1>
5974
                                  <1>
                                            ; (Normal and also high/event priority processes will be added to
5975
                                   <1>
                                            ; normal priority run queue for ensuring circular running sequence!)
5976
                                            ; (Timer interrupt or 'syspri' system call may change priority and run
                                  <1>
5977
                                  <1>
                                            ; queue to high/event level.)
5978 0000E60E 803D[A9030300]00
                                            cmp byte [u.pri], 0
                                  <1>
5979 0000E615 7702
                                  <1>
                                            ja
                                                   short tswap_1; normal priority run queue
5980
                                   <1>
5981 0000E617 43
                                  <1>
                                            inc
                                                   ebx
5982 0000E618 43
                                  <1>
                                            inc
                                                                ; runq+4, 'runq_background', low priority
                                                   ebx
5983
                                  <1> tswap_1:
5984 0000E619 A0[B3030300]
                                  <1>
                                                   al, [u.uno]
                                            mov
5985
                                   <1>
                                                        ; movb u.uno,r1 / move users process number to r1
5986
                                  <1>
                                                    ; mov $runq+4,r2
5987
                                  <1>
                                                          ; / move lowest priority queue address to r2
5988
                                  <1>
                                                   ; ebx = run queue
5989 0000E61E E8FE000000
                                  <1>
                                             call putlu
                                                   ; jsr r0, putlu / create link from last user on Q to
5990
                                   <1>
                                                                ; / u.uno's user
5991
                                  <1>
5992
                                  <1>
5993
                                  <1> switch: ; Retro UNIX 386 v1
5994
                                  <1> swap:
                                          ; 02/01/2017
5995
                                  <1>
                                            ; 21/05/2016
5996
                                  <1>
5997
                                  <1>
                                            ; 20/05/2016
5998
                                  <1>
                                            ; 02/05/2016
5999
                                  <1>
                                            ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
                                            ; 10/05/2015 - 02/09/2015 (Retro UNIX 386 v1)
; 14/04/2013 - 08/03/2014 (Retro UNIX 8086 v1)
6000
                                   <1>
6001
                                  <1>
6002
                                  <1>
                                            ; 'swap' is routine that controls the swapping of processes
6003
                                  <1>
6004
                                  <1>
                                            ; in and out of core.
6005
                                   <1>
                                            ; TRDOS 386 (TRDOS v2.0) modification -> ** 20/05/2016 **
6006
                                  <1>
                                                   * 3 different priority level is applied
6007
                                   <1>
6008
                                                    (just as original unix v1)
                                   <1>
6009
                                   <1>
                                                   1) high priority (event) run queue, 'runq_event'
                                                   2) normal priority (regular) run queue, 'runq_normal'3) low priority (background) run queue, 'runq_backgroud
6010
                                   <1>
6011
                                   <1>
6012
                                   <1>
                                                    'swap' code will run a process which has max. priority
6013
                                   <1>
                                                       (for earliest event at first)
6014
                                   <1>
6015
                                   <1>
                                            ; Retro UNIX 386 v1 modification ->
6016
                                  <1>
                                                    swap (software task switch) is performed by changing
6017
                                   <1>
                                                   user's page directory (u.pgdir) instead of segment change
                                                   as in Retro UNIX 8086 v1.
6018
                                   <1>
6019
                                   <1>
6020
                                   <1>
                                            ; RETRO UNIX 8086 v1 modification ->
                                                    'swap to disk' is replaced with 'change running segment'
6021
                                  <1>
                                                    according to 8086 cpu (x86 real mode) architecture.
6022
                                   <1>
                                                   pdp-11 was using 64KB uniform memory while IBM PC
6023
                                   <1>
6024
                                   <1>
                                                   compatibles was using 1MB segmented memory
6025
                                   <1>
                                                   in 8086/8088 times.
6026
                                  <1>
                                            ; INPUTS ->
6027
                                   <1>
6028
                                            ; runq table - contains processes to run.
                                  <1>
6029
                                  <1>
                                                  p.link - contains next process in line to be run.
6030
                                   <1>
                                                 u.uno - process number of process in core
6031
                                  <1>
                                                 s.stack - swap stack used as an internal stack for swapping.
6032
                                   <1>
                                            ; OUTPUTS ->
6033
                                   <1>
                                             ; (original unix v1 -> present process to its disk block)
```

```
6034
                                                (original unix v1 -> new process into core ->
6035
                                 <1>
                                                     Retro Unix 8086 v1 -> segment registers changed
6036
                                 <1>
                                                     for new process)
6037
                                 <1>
                                                u.quant = 3 (Time quantum for a process)
6038
                                 <1>
                                                 ((INT 1Ch count down speed -> 18.2 times per second)
6039
                                 <1>
                                                RETRO UNIX 8086 v1 will use INT 1Ch (18.2 times per second)
                                                 for now, it will swap the process if there is not
6040
                                 <1>
                                                  a keyboard event (keystroke) (Int 15h, function 4Fh)
6041
                                 <1>
6042
                                 <1>
                                                  or will count down from 3 to 0 even if there is a
6043
                                 <1>
                                                   keyboard event locking due to repetitive key strokes.
6044
                                 <1>
                                                   u.quant will be reset to 3 for RETRO UNIX 8086 v1.
6045
                                 <1>
6046
                                 <1>
                                           ; ((Modified registers: EAX, EDX, EBX, ECX, ESI, EDI))
6047
                                 <1>
6048
                                 <1>
6049
                                 <1>
                                           ; High priority queue is the first for selecting a process to run.
6050
                                 <1>
                                           ; If there is not a process in high priority level run queue,
6051
                                 <1>
                                           ;a process in normal priority run queue will be selected
6052
                                 <1>
                                           ;or a proces in low priority run queue will be selected if normal
6053
                                 <1>
                                           ;priority level run queue is empty.
6054
                                 <1>
                                 <1>
                                           ; 21/05/2016 -(3 priority levels, 3 run queues)
                                                 esi, runq; 'runq_event'; high priority, 'run for event'
6056 0000E623 BE[52030300]
                                 <1>
                                           mov
6057 0000E628 C605[50650100]03
                                 <1>
                                                 byte [priority], 3 ; high priority + 1
                                                 ebx, ebx ; 02/01/2017
6058 0000E62F 31DB
                                 <1>
                                           xor
6059
                                 <1> swap_0: ; 1: / search runq table for highest priority process
6060 0000E631 66AD
                                 <1>
                                          lodsw ; mov ax, [esi], add esi+2
                                           ;xor ebx, ebx; 02/05/2016
6061
                                 <1>
6062 0000E633 6621C0
                                 <1>
                                                 ax, ax; are there any processes to run in this Q entry
                                           jnz
                                                 short swap_2
6063 0000E636 750E
                                 <1>
6064
                                 <1>
                                           ; 21/05/2026
6065
                                 <1>
                                           ; runq_normal = runq+2, runq_background = runq+4
6066 0000E638 FE0D[50650100]
                                 <1>
                                                 byte [priority] ; 3 -> 3, 2 -> 1, 1-> 0
                                           dec
6067 0000E63E 75F1
                                 <1>
                                           jnz short swap_0
                                           ; cmp esi, runq+6 ; if zero compare address to end of table
6068
                                 <1>
                                                 short swap_0; if not at end, go back
6069
                                 <1>
                                           ;jb
                                 <1> swap_1:
6070
                                         ; 02/05/2016
6071
                                 <1>
6072
                                 <1>
                                           ; 29/04/2016 (TRDOS 386 = TRDOS v2.0)
6073
                                 <1>
                                           ; No user process to run...
6074
                                 <1>
                                           ; Run the kernel process... MainProg: Internal Command Interpreter
                                           inc al; mov al, 1; process number of MainProg
inc bl; mov bl, al; 1
6075 0000E640 FEC0
                                 <1>
6076 0000E642 FEC3
                                 <1>
6077 0000E644 EB1E
                                 <1>
                                           jmp
                                                 short swap_4
6078
                                 <1> swap 2:
6079
                                 <1>
                                           ; 21/05/2016
6080 0000E646 FE0D[50650100]
                                 <1>
                                           dec byte [priority] ; priority level of present user/process
6081
                                 <1>
                                                               ; 0, 1, 2
6082 0000E64C 4E
                                 <1>
                                           dec esi
6083 0000E64D 4E
                                 <1>
                                           dec esi
6084
                                 <1>
6085 0000E64E 88C3
                                 <1>
                                           mov
                                                 bl, al
                                                 al, ah; is there only 1 process in the queue to be run
6086 0000E650 38E0
                                 <1>
                                           cmp
6087 0000E652 740A
                                 <1>
                                                 short swap_3 ; yes
                                           je
6088 0000E654 8AA3[9F000300]
                                 <1>
                                           mov
                                                 ah, [ebx+p.link-1]
6089 0000E65A 8826
                                 <1>
                                                  mov [esi], ah; move next process in line into run queue
6090 0000E65C EB06
                                 <1>
                                           jmр
                                                 short swap 4
                                 <1> swap_3:
6091
6092 0000E65E 6631D2
                                 <1>
                                           xor
6093 0000E661 668916
                                 <1>
                                           mov
                                                 [esi], dx; zero the entry; no processes on the Q
6094
                                 <1> swap_4:
6095 0000E664 8A25[B3030300]
                                 <1>
                                           mov
                                                 ah, [u.uno]
6096 0000E66A 38C4
                                                  ah, al; is this process the same as the process in core?
                                 <1>
                                           cmp
6097 0000E66C 743B
                                 <1>
                                                       short swap_8 ; yes, don't have to swap
6098 0000E66E 08E4
                                 <1>
                                                 ah, ah ; is the process \# = 0
                                          or
6099 0000E670 740D
                                 <1>
                                                       short swap_6 ; 'sysexit'
6100
                                 <1>
                                           ;cmp ah, al ;is this process the same as the process in core?
                                                 ;je short swap_8; yes, don't have to swap
6101
                                 <1>
6102 0000E672 8925[60030300]
                                 <1>
                                                  [u.usp], esp; return address for 'syswait' & 'sleep'
6103 0000E678 E834000000
                                 <1>
                                           call wswap ; write out core to disk
                                                  short swap_7
6104 0000E67D EB1C
                                 <1>
6105
                                 <1> swap 6:
6106
                                           ; Deallocate memory pages belong to the process
                                 <1>
6107
                                 <1>
                                           ; which is being terminated.
                                 <1>
                                           ; (Retro UNIX 386 v1 modification !)
6108
6109
                                 <1>
6110 0000E67F 53
                                 <1>
                                           push ebx
6111 0000E680 A1[B8030300]
                                 <1>
                                           mov
                                                  eax, [u.pgdir] ; page directory of the process
6112 0000E685 8B1D[BC030300]
                                 <1>
                                                  ebx, [u.ppgdir] ; page directory of the parent process
6113 0000E68B E88066FFFF
                                 <1>
                                           call deallocate_page_dir
                                                  eax, [u.upage]; 'user' structure page of the process
6114 0000E690 A1[B4030300]
                                 <1>
                                           mov
                                                  deallocate_page
6115 0000E695 E81B67FFFF
                                 <1>
                                           call
6116 0000E69A 5B
                                 <1>
                                                  ebx
                                           pop
6117
                                 <1> swap_7:
6118 0000E69B C0E302
                                 <1>
                                           shl
                                                 bl, 2 ; * 4
6119 0000E69E 8B83[BC000300]
                                                  eax, [ebx+p.upage-4]; the 'u' page of the new process
                                 <1>
                                           mov
6120 0000E6A4 E840000000
                                 <1>
                                           call rswap; read new process into core
6121
                                 <1> swap_8:
6122
                                 <1>
                                           ; Retro UNIX 8086 v1 modification !
6123 0000E6A9 C605[A8030300]04
                                           mov byte [u.quant], time_count
                                 <1>
6124 0000E6B0 C3
                                 <1>
6125
                                 <1>
                                 <1> wswap: ; < swap out, swap to disk >
6126
                                          ; 28/02/2017 (fnsave)
6127
                                 <1>
6128
                                 <1>
                                           ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
6129
                                 <1>
                                           ; 09/05/2015 (Retro UNIX 386 v1)
6130
                                 <1>
                                           ; 26/05/2013 - 08/03/2014 (Retro UNIX 8086 v1)
                                          ; 'wswap' writes out the process that is in core onto its
                                 <1>
6131
6132
                                 <1>
                                           ; appropriate disk area.
6133
                                 <1>
                                           ; Retro UNIX 386 v1 modification ->
6134
                                 <1>
6135
                                 <1>
                                                  User (u) structure content and the user's register content
6136
                                 <1>
                                                  will be copied to the process's/user's UPAGE (a page for
6137
                                 <1>
                                                  saving 'u' structure and user registers for task switching).
                                 <1>
                                                 u.usp - points to kernel stack address which contains
6138
```

```
user's registers while entering system call.
6140
                                  <1>
                                                   u.sp - points to kernel stack address
                                            ;
6141
                                                         to return from system call -for IRET-.
                                  <1>
6142
                                  <1>
                                                   [u.usp] + 32 + 16 = [u.sp]
6143
                                  <1>
                                                   [u.usp] \rightarrow edi, esi, ebp, esp (= [u.usp]+32), ebx,
6144
                                  <1>
                                                         edx, ecx, eax, gs, fs, es, ds, \rightarrow [u.sp].
6145
                                  <1>
                                          ; Retro UNIX 8086 v1 modification ->
6146
                                  <1>
6147
                                  <1>
                                                    'swap to disk' is replaced with 'change running segment'
6148
                                  <1>
                                                   according to 8086 cpu (x86 real mode) architecture.
6149
                                  <1>
                                                   pdp-11 was using 64KB uniform memory while IBM PC
6150
                                  <1>
                                                   compatibles was using 1MB segmented memory
6151
                                  <1>
                                                   in 8086/8088 times.
6152
                                  <1>
                                            ; INPUTS ->
6153
                                  <1>
                                                u.break - points to end of program
6154
                                  <1>
6155
                                  <1>
                                                u.usp - stack pointer at the moment of swap
6156
                                  <1>
                                                core - beginning of process program
6157
                                  <1>
                                                 ecore - end of core
                                                user - start of user parameter area
6158
                                  <1>
6159
                                  <1>
                                            ; u.uno - user process number
                                           ;
                                                 p.dska - holds block number of process
6160
                                  <1>
                                            ; OUTPUTS ->
6161
                                  <1>
6162
                                  <1>
                                            ; swp I/O queue
                                               p.break - negative word count of process
                                  <1>
6163
6164
                                  <1>
                                                 r1 - process disk address
6165
                                  <1>
                                               r2 - negative word count
6166
                                  <1>
6167
                                  <1>
                                            ; RETRO UNIX 8086 v1 input/output:
6168
                                  <1>
                                            ; INPUTS ->
6169
                                  <1>
6170
                                  <1>
                                                u.uno - process number (to be swapped out)
                                            ;
                                            ; OUTPUTS ->
6171
                                  <1>
6172
                                  <1>
6173
                                  <1>
                                                ((Modified registers: ECX, ESI, EDI))
6174
                                  <1>
6175
                                  <1>
6176
                                  <1>
                                            ; 28/02/2017
6177
                                  <1>
6178
                                            ;cmp byte [multi_tasking], 0 ; Musti tasking mode ?
                                  <1>
6179
                                  <1>
                                            ;jna short wswp
6180 0000E6B1 803D[DA030300]00
                                  <1>
                                            cmp
                                                  byte [u.fpsave], 0 ; 28/02/2017
6181 0000E6B8 7606
                                  <1>
                                            jna
                                                  short wswp
6182 0000E6BA DD35[DC030300]
                                  <1>
                                            fnsave [u.fpregs] ; save floating point registers (94 bytes)
6183
                                  <1> wswp:
6184 0000E6C0 8B3D[B4030300]
                                  <1>
                                                   edi, [u.upage] ; process's user (u) structure page addr
6185 0000E6C6 B938000000
                                  <1>
                                                   ecx, (U SIZE + 3) / 4
                                            mov
6186 0000E6CB BE[5C030300]
                                                   esi, user ; active user (u) structure
                                  <1>
                                            mov
6187 0000E6D0 F3A5
                                  <1>
                                            rep
                                                   movsd
6188
                                  <1>
                                            ;
6189 0000E6D2 8B35[60030300]
                                                   esi, [u.usp]; esp (system stack pointer,
                                  <1>
6190
                                  <1>
                                                              ; points to user registers)
6191 0000E6D8 8B0D[5C030300]
                                  <1>
                                            mov
                                                   ecx, [u.sp] ; return address from the system call
                                                              ; (for IRET)
6192
                                  <1>
6193
                                  <1>
                                                               ; [u.sp] -> EIP (user)
6194
                                  <1>
                                                               ; [u.sp+4] \rightarrow CS (user)
                                                               ; [u.sp+8] -> EFLAGS (user)
6195
                                  <1>
6196
                                                               ; [u.sp+12] \rightarrow ESP (user)
                                  <1>
6197
                                  <1>
                                                               ; [u.sp+16] \rightarrow SS (user)
                                                              ; required space for user registers
6198 0000E6DE 29F1
                                  <1>
                                            sub
                                                   ecx, esi
                                                   ecx, 20
6199 0000E6E0 83C114
                                  <1>
                                                                     ; +5 dwords to return from system call
                                  <1>
                                                               ; (for IRET)
6201 0000E6E3 C1E902
                                  <1>
                                                   ecx, 2
                                            shr
6202 0000E6E6 F3A5
                                  <1>
                                            rep
                                                   movsd
6203 0000E6E8 C3
                                  <1>
                                            retn
6204
                                  <1>
6205
                                  <1> rswap: ; < swap in, swap from disk >
                                          ; 28/02/2017 (frstor)
6206
                                  <1>
6207
                                  <1>
                                            ; 15/01/2017
                                            ; 14/01/2017
6208
                                  <1>
6209
                                  <1>
                                           ; 21/05/2016
                                            ; 03/05/2016
6210
                                  <1>
                                            ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
                                  <1>
6211
6212
                                  <1>
                                           ; 09/05/2015 - 15/09/2015 (Retro UNIX 386 v1)
6213
                                  <1>
                                            ; 26/05/2013 - 08/03/2014 (Retro UNIX 8086 v1)
                                            ; 'rswap' reads a process whose number is in r1,
6214
                                  <1>
6215
                                  <1>
                                            ; from disk into core.
6216
                                  <1>
6217
                                  <1>
                                            ; Retro UNIX 386 v1 modification ->
6218
                                                   User (u) structure content and the user's register content
                                  <1>
                                                   will be restored from process's/user's UPAGE (a page for
6219
                                  <1>
6220
                                   <1>
                                                   saving 'u' structure and user registers for task switching).
6221
                                  <1>
                                                   u.usp - points to kernel stack address which contains
6222
                                   <1>
                                                       user's registers while entering system call.
6223
                                  <1>
                                                   u.sp - points to kernel stack address
                                                          to return from system call -for IRET-.
6224
                                  <1>
6225
                                                   [u.usp] + 32 + 16 = [u.sp]
                                   <1>
6226
                                  <1>
                                                   [u.usp] \rightarrow edi, esi, ebp, esp (= [u.usp]+32), ebx,
                                                         edx, ecx, eax, gs, fs, es, ds, -> [u.sp].
6227
                                  <1>
6228
                                  <1>
                                            ; RETRO UNIX 8086 v1 modification ->
6229
                                  <1>
6230
                                   <1>
                                                    'swap to disk' is replaced with 'change running segment'
6231
                                  <1>
                                                   according to 8086 cpu (x86 real mode) architecture.
                                                   pdp-11 was using 64KB uniform memory while IBM PC
6232
                                  <1>
6233
                                  <1>
                                                   compatibles was using 1MB segmented memory
6234
                                  <1>
                                                   in 8086/8088 times.
6235
                                  <1>
                                            ; INPUTS ->
6236
                                  <1>
6237
                                  <1>
                                                 r1 - process number of process to be read in
6238
                                                 p.break - negative of word count of process
                                  <1>
                                                 p.dska - disk address of the process
6239
                                  <1>
6240
                                   <1>
                                                 u.emt - determines handling of emt's
                                                 u.ilgins - determines handling of illegal instructions
6241
                                  <1>
6242
                                  <1>
                                            ; OUTPUTS ->
6243
                                  <1>
                                            ; 8 = (u.ilgins)
```

6139

```
6244
                                  <1>
                                                24 = (u.emt)
6245
                                  <1>
                                                swp - bit 10 is set to indicate read
                                                       (bit 15=0 when reading is done)
6246
                                  <1>
6247
                                  <1>
                                               swp+2 - disk block address
6248
                                  <1>
                                                swp+4 - negative word count
6249
                                  <1>
                                                  ((swp+6 - address of user structure))
6250
                                  <1>
6251
                                  <1>
                                          ; RETRO UNIX 8086 v1 input/output:
6252
                                  <1>
                                           ; INPUTS ->
6253
                                  <1>
6254
                                  <1>
                                               AL
                                                        - new process number (to be swapped in)
                                           ; OUTPUTS ->
6255
                                  <1>
6256
                                  <1>
                                               none
6257
                                  <1>
                                           ; ((Modified registers: EAX, ECX, ESI, EDI, ESP))
6258
                                  <1>
6259
                                  <1>
                                           ; Retro UNIX 386 v1 - modification ! 14/05/2015
6260
                                  <1>
6261 0000E6E9 89C6
                                 <1>
                                           mov esi, eax ; process's user (u) structure page addr
                                                  ecx, (U_SIZE + 3) / 4
6262 0000E6EB B938000000
                                  <1>
6263 0000E6F0 BF[5C030300]
                                 <1>
                                                  edi, user ; active user (u) structure
                                           mov
6264 0000E6F5 F3A5
                                 <1>
                                                  movsd
6265 0000E6F7 58
                                                  eax ; 'rswap' return address
                                  <1>
                                           pop
6266
                                 <1>
6267
                                 <1>
                                           ;cli
6268 0000E6F8 8B3D[60030300]
                                 <1>
                                                  edi, [u.usp] ; esp (system stack pointer,
                                           mov
6269
                                  <1>
                                                            ; points to user registers)
6270 0000E6FE 89FC
                                  <1>
                                                  esp, edi
                                                              ; 14/01/2017
                                            mov
6271 0000E700 8B0D[5C030300]
                                 <1>
                                            mov
                                                  ecx, [u.sp] ; return address from the system call
6272
                                  <1>
                                                              ; (for IRET)
6273
                                  <1>
                                                              ; [u.sp] -> EIP (user)
6274
                                  <1>
                                                              ; [u.sp+4] \rightarrow CS (user)
6275
                                  <1>
                                                              ; [u.sp+8] -> EFLAGS (user)
                                                              ; [u.sp+12] -> ESP (user)
6276
                                  <1>
6277
                                 <1>
                                                             ; [u.sp+16] -> SS (user)
                                                            ; required space for user registers
6278 0000E706 29F9
                                 <1>
                                                  ecx, edi
                                            sub
6279 0000E708 83C114
                                 <1>
                                            add
                                                  ecx, 20
                                                                ; +5 dwords to return from system call
                                 <1>
                                                              ; (for IRET)
6281 0000E70B C1E902
                                 <1>
                                           shr
                                                  ecx, 2
6282 0000E70E F3A5
                                  <1>
                                                  movsd
                                           rep
                                           ;mov esp, [u.usp] ; 15/09/2015
6283
                                 <1>
6284
                                  <1>
                                           ;sti
6285
                                  <1>
                                           ; 28/02/2017
6286
                                  <1>
                                           ;cmp byte [multi_tasking], 0 ; Musti tasking mode ?
6287
                                  <1>
                                           ;jna short rswp_retn
                                           cmp byte [u.fpsave], 0
6288 0000E710 803D[DA030300]00
                                 <1>
6289 0000E717 7606
                                                 short rswp_retn
                                  <1>
                                            jna
6290 0000E719 DD25[DC030300]
                                           frstor [u.fpregs] ; restore floating point regs (94 bytes)
                                  <1>
                                 <1> rswp_retn:
6291
6292 0000E71F 50
                                  <1>
                                           push
                                                 eax ; 'rswap' return address
6293 0000E720 C3
                                  <1>
                                            retn
6294
                                  <1>
6295
                                  <1> putlu:
                                         ; 20/05/2016
6296
                                  <1>
6297
                                  <1>
                                           ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
                                          ; 10/05/2015 - 12/09/2015 (Retro UNIX 386 v1)
; 15/04/2013 - 23/02/2014 (Retro UNIX 8086 v1)
6298
                                  <1>
6299
                                  <1>
6300
                                  <1>
                                           ; 'putlu' is called with a process number in r1 and a pointer
6301
                                  <1>
                                           ; to lowest priority Q (runq+4) in r2. A link is created from
6302
                                  <1>
                                           ; the last process on the queue to process in r1 by putting
                                           ; the process number in r1 into the last process's link.
6303
                                  <1>
6304
                                  <1>
6305
                                  <1>
                                           ; INPUTS ->
                                           ; r1 - user process number
                                  <1>
6306
6307
                                  <1>
                                               r2 - points to lowest priority queue
                                               p.dska - disk address of the process
6308
                                  <1>
6309
                                  <1>
                                                u.emt - determines handling of emt's
                                              u.ilgins - determines handling of illegal instructions
6310
                                  <1>
6311
                                  <1>
                                           ; OUTPUTS ->
6312
                                  <1>
                                           ; r3 - process number of last process on the queue upon
6313
                                  <1>
                                                  entering putlu
6314
                                  <1>
                                               p.link-1 + r3 - process number in r1
6315
                                  <1>
                                                r2 - points to lowest priority queue
                                  <1>
6316
6317
                                  <1>
                                           ; ((Modified registers: EDX, EBX))
6318
                                  <1>
6319
                                  <1>
                                           ; / r1 = user process no.; r2 points to lowest priority queue
6320
                                  <1>
                                           ; EBX = r2
6321
                                  <1>
                                           ; EAX = r1 (AL=r1b)
6322
                                  <1>
6323
                                  <1>
                                            ; 20/05/2016
6324
                                  <1>
6325
                                  <1>
                                            ; AL = process number (1 to 16) // Retro UNIX 8086, 386 v1 //
                                                  (max. 16 processes available for current kernel version)
6326
                                  <1>
6327
                                  <1>
                                            ; EBX = run queue address ; 20/05/2016 (TRDOS 386)
6328
                                  <1>
                                                  ; which is one of following addresses:
                                                  ; 1) 'runq_event' high priority run queue
6329
                                  <1>
                                                  ; 2) 'rung normal' normal/regular priority run queue
6330
                                  <1>
6331
                                  <1>
                                                   ; 3) 'runq_background' low priority run queue
6332
                                  <1>
6333
                                            ;mov ebx, runq
                                  <1>
6334 0000E721 0FB613
                                  <1>
                                            movzx
                                                         edx, byte [ebx]
6335 0000E724 43
                                  <1>
                                            inc
                                                 ebx
6336 0000E725 20D2
                                  <1>
                                            and
                                                 dl, dl
6337
                                  <1>
                                                  ; tstb (r2)+ / is queue empty?
6338 0000E727 740A
                                  <1>
                                                  jz short putlu 1
                                                  ; beq 1f / yes, branch
6339
                                  <1>
                                                  dl, [ebx] ; 12/09/2015
6340 0000E729 8A13
                                  <1>
                                                  ; movb (r2),r3 / no, save the "last user" process number
6341
                                  <1>
                                                             ; / in r3
6342
                                  <1>
6343 0000E72B 8882[9F000300]
                                                  mov [edx+p.link-1], al
                                  <1>
6344
                                  <1>
                                                  ; movb r1,p.link-1(r3) / put pointer to user on
6345
                                  <1>
                                                             ; / "last users" link
6346 0000E731 EB03
                                  <1>
                                                  short putlu_2
6347
                                  <1>
                                                  ; br 2f /
6348
                                  <1> putlu_1: ; 1:
```

```
<1>
6349 0000E733 8843FF
                                            mov
                                                   [ebx-1], al
6350
                                  <1>
                                                         ; movb r1,-1(r2) / user is only user;
6351
                                  <1>
                                                             ; / put process no. at beginning and at end
6352
                                  <1> putlu 2: ; 2:
6353 0000E736 8803
                                  <1>
                                           mov [ebx], al
6354
                                  <1>
                                                         ; movb r1, (r2) / user process in r1 is now the last entry
                                                             ; / on the queue
6355
                                  <1>
6356 0000E738 88C2
                                           mov dl, al
                                  <1>
6357 0000E73A 88B2[9F000300]
                                  <1>
                                                      [edx+p.link-1], dh; 0
6358
                                  <1>
                                              ; dec r2 / restore r2
6359 0000E740 C3
                                  <1>
                                              retn
6360
                                  <1>
                                                  ; rts r0
6361
                                  <1>
                                  <1> sysver:
6362
                                            ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
6363
                                  <1>
6364 0000E741 C705[64030300]0002- <1>
                                            mov dword [u.r0], 200h; AH = major version, AL = minor version
6364 0000E749 0000
                                  <1>
6365 0000E74B E912E0FFFF
                                  <1>
                                            jmp
                                                  sysret
6366
                                  <1>
6367
                                  <1>
6368
                                  <1> syspri: ; change running priority (of the process)
6369
                                  <1>
                                           ; 21/05/2016
                                            ; 20/05/2026 - TRDOS 386 (TRDOS v2.0)
6370
                                  <1>
6371
                                  <1>
                                            ; INPUT ->
6372
                                  <1>
                                                  BL = priority level
                                            ;
6373
                                  <1>
                                                     0 = low running priority (running on background)
6374
                                  <1>
                                                      1 = normal/regular priority (running as regular)
6375
                                                      2 = high/event priority (running for event)
                                  <1>
6376
                                  <1>
                                                      >2 = invalid, it will accepted as 2 (event)
6377
                                  <1>
                                                     OFFh = get/return current running priority only
                                            ;
                                            ; OUTPUT ->
6378
                                  <1>
6379
                                  <1>
                                                * if current [u.pri] < 2
                                            ;
                                                    if BL input < 0FFh ->
6380
                                  <1>
6381
                                  <1>
                                                        [u.pri] is updated as in BL input (0,1,2)
6382
                                  <1>
                                                    if BL input = OFFh -> AL = [u.pri] (current)
6383
                                  <1>
6384
                                  <1>
                                                   * if current [u.pri] = 2
                                                     if BL input < OFFh \rightarrow cf = 1 & AL = 2
6385
                                  <1>
6386
                                  <1>
                                                     if BL input = 0FFh \rightarrow cf = 0 & AL = 2
6387
                                  <1>
6388
                                  <1>
6389
                                  <1>
                                                   If [u.pri] = 2, it can not be changed to 1 or 0;
                                                   because, run queue of the running process is unspecified
6390
                                  <1>
6391
                                  <1>
                                                             stage. Process might be started by a timer event
                                                   or priority might be changed to high by previous
6392
                                  <1>
                                                   'syspri' system
6393
                                  <1>
                                                                      call. In both cases, the process is in
                                                   'runq_normal' or 'runq_background' queue.
6394
                                  <1>
6395
                                  <1>
                                                   As result of this fact, when the [u.quant] time quantum
6396
                                  <1>
                                                   of the process is elapsed or 'sysrele' system call is
6397
                                  <1>
                                                   instructed by the process, 'tswap' ('tswitch') procedure
6398
                                  <1>
                                                   will be called (to 'swap' or 'switch' out the procedure)
6399
                                  <1>
                                                   and it will not call 'putlu' to add the (stopping)
6400
                                  <1>
                                                   process to relevant run queue when [u.pri] = 2.
6401
                                  <1>
                                                   (Otherwise, it would be possible to add process to
                                                   a run queue while it is already in a run queue, wrongly.)
6402
                                  <1>
6403
                                  <1>
6404
                                  <1>
                                                   If [u.pri] < 2, 'tswap/tswitch' procedure will call</pre>
                                                   'putlu' to add process to relevant run queue
6405
                                  <1>
6406
                                  <1>
                                                   according to [u.pri] value. ('runq_normal' for 1,
                                                   'runq_background' for 0).
6407
                                  <1>
6408
                                  <1>
                                                   If BL input >= 2 and < OFFh while [u.pri] < 2,
6409
                                  <1>
                                            ;
6410
                                  <1>
                                                   process will be added to 'runq_normal' queue and
                                            ;
6411
                                  <1>
                                                   [u.pri] will be set to 2. (in 'syspri' system call)
6412
                                  <1>
                                            ;
6413
                                  <1>
6414 0000E750 29C0
                                  <1>
                                            sub
                                                   eax, eax; 0
6415 0000E752 A3[C8030300]
                                  <1>
                                            mov
                                                   [u.error], eax
                                  <1>
6417 0000E757 A0[A9030300]
                                  <1>
                                                   al, [u.pri]
                                            mov
6418 0000E75C A3[64030300]
                                  <1>
                                            mov
                                                   [u.r0], eax
6419
                                  <1>
6420 0000E761 FEC3
                                  <1>
                                            inc
                                                   sysret; OFFh -> 0, get priority level
6421 0000E763 0F84F9DFFFFF
                                  <1>
                                            jz
6422
                                  <1>
6423 0000E769 3C02
                                  <1>
                                            cmp
6424 0000E76B 0F83D1DFFFFF
                                  <1>
                                            jnb
                                                   error; CF = 1 \& AL = 2 (\& last error = 0)
6425
                                  <1>
6426 0000E771 FECB
                                  <1>
                                            dec
                                                   bl
6427 0000E773 80FB02
                                  <1>
                                                  bl, 2
                                            cmp
6428 0000E776 7602
                                  <1>
                                            jna
                                                   short syspri_1
6429 0000E778 B302
                                  <1>
                                                   bl, 2
                                            mov
                                  <1> syspri_1:
6430
6431 0000E77A 881D[A9030300]
                                                   [u.pri], bl
                                  <1>
6432 0000E780 80FB02
                                  <1>
                                                  bl, 2
                                            cmp
6433 0000E783 0F82D9DFFFFF
                                  <1>
                                            jb
                                                      sysret
                                  <1>
6435
                                  <1>
                                           ; here...
6436
                                  <1>
                                            ; Priority of current process has been changed to high
6437
                                  <1>
                                            ; ('run for event') but current process will be added to
                                            ; 'run as normal' queue. ('run for event' high priority
6438
                                  <1>
6439
                                  <1>
                                            ; queue is under control of timer -& RTC- interrupt only!)
6440
                                  <1>
                                            ; (Otherwise, process can fall into black hole!
6441
                                  <1>
6442
                                  <1>
                                            ; e.g. if it is not in waiting list and it has not got
6443
                                  <1>
                                            ; a timer event and it is not in a run queue!
                                            ; Because, when [u.pri] is 2, 'tswap/tswitch' will not
6444
                                  <1>
6445
                                  <1>
                                            ; add the stopping process to a run queue.)
6446
                                  <1>
6447 0000E789 A0[B3030300]
                                  <1>
                                                   al, [u.uno]
                                            mov
6448 0000E78E BB[54030300]
                                  <1>
                                                   ebx, runq_normal ; normal priority !
6449
                                  <1>
                                                                 ; [u.pri] is set to high
6450
                                  <1>
                                                                 ; but 'runq_event' queue is set
                                                                 ; only by the kernel's timer
6451
                                  <1>
6452
                                  <1>
                                                                 ; event function (timer interrupt).
```

```
6453 0000E793 E889FFFFF
                                <1>
                                          call putlu
6454 0000E798 E9C5DFFFFF
                                 <1>
                                          jmp sysret
                                 <1>
6456
                                 <1> cpass: ; / get next character from user area of core and put it in AL (r1)
                                         ; 02/05/2016 - TRDOS 386 (TRDOS v2.0)
; 19/05/2015 - 18/10/2015 (Retro UNIX 386 v1)
6457
                                 <1>
6458
                                 <1>
                                          ; 14/08/2013 - 20/09/2013 (Retro UNIX 8086 v1)
6459
                                 <1>
                                         ; INPUTS ->
                                 <1>
6460
6461
                                 <1>
                                                [u.base] = virtual address in user area
                                                [u.count] = byte count (max.)
6462
                                 <1>
                                          ;
6463
                                 <1>
                                                [u.pcount] = byte count in page (0 = reset)
6464
                                 <1>
                                          ; OUTPUTS ->
                                          ; AL = the character which is pointed by [u.base]
6465
                                 <1>
6466
                                 <1>
                                                zf = 1 -> transfer count has been completed
                                 <1>
6467
                                          ; ((Modified registers: EAX, EDX, ECX))
6468
                                 <1>
6469
                                 <1>
                                                6470 0000E79D 833D[88030300]00
                                 <1>
                                          cmp
                                 <1>
                                                                 ; i.e., u.count, # of chars. left
6471
                                                                ; to be transferred = 0?) yes, branch
6472 0000E7A4 763F
                                          jna short cpass 3
                                 <1>
6473 0000E7A6 FF0D[88030300]
                                 <1>
                                          dec dword [u.count]
                                                                      ; no, decrement u.count
6474
                                 <1>
                                           ; 19/05/2015
6475
                                          ; (Retro UNIX 386 v1 - translation from user's virtual address
                                 <1>
6476
                                 <1>
                                                            to physical address
                                                 word [u.pcount], 0 ; byte count in page = 0 (initial value)
6477 0000E7AC 66833D[C4030300]00 <1>
                                          cmp
6478
                                 <1>
                                                            ; 1-4095 --> use previous physical base address
6479
                                 <1>
                                                            ; in [u.pbase]
6480 0000E7B4 770E
                                                 short cpass_1
                                 <1>
                                          jа
6481 0000E7B6 833D[BC030300]00
                                                  dword [u.ppgdir], 0 ; is the caller os kernel
                                 <1>
                                          cmp
6482 0000E7BD 7427
                                                  short cpass_k ; (sysexec, '/etc/init') ? (MainProg)
                                 <1>
                                          iе
6483 0000E7BF E858FDFFFF
                                 <1>
                                          call
                                                trans_addr_r
6484
                                 <1> cpass 1:
6485 0000E7C4 66FF0D[C4030300]
                                                 word [u.pcount]
                                 <1>
                                          dec
6486
                                 <1> cpass 2:
                                                 edx, [u.pbase]
6487 0000E7CB 8B15[C0030300]
                                 <1>
                                          mov
6488 0000E7D1 8A02
                                 <1>
                                                             ; take the character pointed to
                                          mov
                                                 al, [edx]
                                 <1>
                                                             ; by u.base and put it in r1
6490 0000E7D3 FF05[8C030300]
                                 <1>
                                          inc
                                                 dword [u.nread] ; increment no. of bytes transferred
6491 0000E7D9 FF05[84030300]
                                 <1>
                                                 dword [u.base] ; increment the buffer address to point to the
                                          inc
                                                             ; next byte
                                 <1>
6492
6493 0000E7DF FF05[C0030300]
                                 <1>
                                          inc
                                                 dword [u.pbase]
6494
                                 <1> cpass_3:
6495 0000E7E5 C3
                                 <1>
                                          retn
                                 <1> cpass_k:
6496
                                       ; 02/07/2015
6497
                                 <1>
                                          ; The caller is os kernel
6498
                                 <1>
6499
                                 <1>
                                          ; (get sysexec arguments from kernel's memory space)
6500 0000E7E6 8B1D[84030300]
                                <1>
                                        mov ebx, [u.base]
6501 0000E7EC 66C705[C4030300]00- <1>
                                          mov
                                                  word [u.pcount], PAGE_SIZE ; 4096
6501 0000E7F4 10
                                 <1>
6502 0000E7F5 891D[C0030300]
                                 <1>
                                          mov
                                                [u.pbase], ebx
                                          jmp
6503 0000E7FB EBCE
                                 <1>
                                                short cpass 2
6504
                                 <1>
6505
                                 <1> transfer_to_user_buffer: ; fast transfer
                                       ; \overline{2}7/\overline{0}5/20\overline{1}6
6506
                                 <1>
6507
                                 <1>
                                          ; 16/05/2016 - TRDOS 386 (TRDOS v2.0)
6508
                                 <1>
                                         ; INPUT ->
                                 <1>
6509
6510
                                 <1>
                                                ESI = source address in system space
6511
                                 <1>
                                                 EDI = user's buffer address
6512
                                 <1>
                                                ECX = transfer (byte) count
6513
                                 <1>
                                                 [u.pgdir] = user's page directory
                                          ;
                                          ; OUTPUT ->
                                 <1>
6514
6515
                                 <1>
                                                ECX = actual transfer count
                                 <1>
6516
                                                 cf = 1 \rightarrow error
6517
                                 <1>
                                                 [u.count] = remain byte count
6518
                                 <1>
                                 <1>
                                          ; Modified registers: eax, ecx
6519
6520
                                 <1>
6521
                                 <1>
6522 0000E7FD 21C9
                                 <1>
                                          and
                                                 ecx, ecx
6523 0000E7FF 743B
                                 <1>
                                                 short ttub 4
                                          jΖ
6524
                                 <1>
6525 0000E801 890D[88030300]
                                 <1>
                                                 [u.count], ecx
6526
                                 <1>
6527 0000E807 57
                                 <1>
                                          push
                                                edi
6528 0000E808 56
                                 <1>
                                          push
                                                esi
6529 0000E809 53
                                 <1>
                                          push
                                                ebx
6530 0000E80A 52
                                          push
                                 <1>
                                                 edx
6531 0000E80B 51
                                 <1>
                                          push ecx
6532
                                 <1>
6533 0000E80C 89FB
                                 <1>
                                          mov
                                                 ebx, edi
6534 0000E80E 81C300004000
                                 <1>
                                          add
                                                ebx, CORE ; 27/05/2016
6535
                                 <1> ttub 1:
                                      ; ebx = virtual (linear) address
6536
                                 <1>
6537
                                <1>
                                         ; [u.pgdir] = user's page directory
6538 0000E814 E8D96AFFFF
                                <1>
                                                call get_physical_addr_x ; get physical address
                                       jc short ttub_5
; eax = physical address
; ecx = remain byte count in page (1-4096)
6539 0000E819 7222
                                <1>
6540
                                <1>
6541
                                <1>
6542 0000E81B 89C7
                                <1>
                                        mov edi, eax
                                                eax, [u.count]
6543 0000E81D A1[88030300]
                                <1>
                                          mov
6544 0000E822 39C1
                                <1>
                                          cmp
                                                ecx, eax
6545 0000E824 7602
                                <1>
                                                short ttub 2
                                          jna
6546 0000E826 89C1
                                <1>
                                                ecx, eax
                                          mov
                                <1> ttub_2:
6547
6548 0000E828 29C8
                                <1> sub
                                                eax, ecx
6549 0000E82A 01CB
                                <1>
                                          add
                                                ebx, ecx
6550 0000E82C F3A4
                                <1>
                                          rep
                                                 movsb
6551 0000E82E A3[88030300]
                                <1>
                                                [u.count], eax
                                          mov
                                       or
6552 0000E833 09C0
                                <1>
                                                eax, eax
6553 0000E835 75DD
                                <1>
                                                short ttub 1
                                         jnz
                                <1> ttub_retn:
6554
                                <1> tfub_retn:
6555
6556 0000E837 59
```

```
6557
                                 <1> ttub 3:
                                 <1>
6558 0000E838 5A
                                           pop
                                                 edx
6559 0000E839 5B
                                 <1>
                                           pop
                                                 ebx
6560 0000E83A 5E
                                 <1>
                                                  esi
                                           pop
6561 0000E83B 5F
                                 <1>
                                                  edi
                                           pop
                                 <1> ttub_4:
6562
6563 0000E83C C3
                                 <1>
                                           retn
                                 <1> ttub 5:
6564
6565 0000E83D 59
                                 <1>
                                           pop
                                                  ecx
                                                 ecx, [u.count]; actual transfer count
6566 0000E83E 2B0D[88030300]
                                 <1>
                                           sub
6567 0000E844 F9
                                 <1>
                                           stc
6568 0000E845 EBF1
                                 <1>
                                           jmp
                                                 short ttub 3
6569
                                 <1>
                                 <1> transfer from user buffer: ; fast transfer
6570
                                        ; 27/05/2016
6571
                                 <1>
                                           ; 16/05/2016 - TRDOS 386 (TRDOS v2.0)
6572
                                 <1>
6573
                                 <1>
                                         ; INPUT ->
6574
                                 <1>
                                                 ESI = user's buffer address
6575
                                 <1>
                                                 EDI = destination address in system space
6576
                                 <1>
6577
                                 <1>
                                                 ECX = transfer (byte) count
                                 <1>
                                                  [u.pgdir] = user's page directory
6578
                                           ;
                                           ; OUTPUT ->
6579
                                 <1>
6580
                                 <1>
                                                 ecx = actual transfer count
6581
                                 <1>
                                                  cf = 1 \rightarrow error
                                           ;
                                 <1>
6582
                                                  [u.count] = remain byte count
6583
                                 <1>
6584
                                 <1>
                                           ; Modified registers: eax, ecx
6585
                                 <1>
                                 <1>
6586
6587 0000E847 21C9
                                 <1>
                                           and
                                                 ecx, ecx
                                 <1>
                                                  short tfub 4
6588
                                           ;jz
6589 0000E849 74F1
                                 <1>
                                                  short ttub 4
                                           jz
6590
                                 <1>
6591 0000E84B 890D[88030300]
                                                  [u.count], ecx
                                 <1>
                                           mov
6592
                                 <1>
6593 0000E851 57
                                 <1>
                                           push
                                                 edi
                                           push
6594 0000E852 56
                                 <1>
                                                 esi
                                           push
6595 0000E853 53
                                 <1>
                                                 ebx
6596 0000E854 52
                                           push
                                 <1>
                                                 edx
6597 0000E855 51
                                 <1>
                                           push
                                                 ecx
6598
                                 <1>
6599 0000E856 89F3
                                 <1>
                                                  ebx, esi
                                           mov
6600 0000E858 81C300004000
                                 <1>
                                           add
                                                 ebx, CORE ; 27/05/2016
6601
                                 <1> tfub 1:
6602
                                 <1>
                                           ; ebx = virtual (linear) address
                                           ; [u.pgdir] = user's page directory
6603
                                 <1>
6604 0000E85E E88F6AFFFF
                                 <1>
                                                  call get_physical_addr_x ; get physical address
6605
                                 <1>
                                           ;jc
                                                 short tfub_5
6606 0000E863 72D8
                                 <1>
                                           jc short ttub 5
6607
                                 <1>
                                           ; eax = physical address
6608
                                 <1>
                                           ; ecx = remain byte count in page (1-4096)
6609 0000E865 89C6
                                 <1>
                                           mov esi, eax
6610 0000E867 A1[88030300]
                                 <1>
                                           mov
                                                 eax, [u.count]
6611 0000E86C 39C1
                                 <1>
                                           cmp
                                                 ecx, eax
6612 0000E86E 7602
                                 <1>
                                           jna
                                                 short tfub_2
6613 0000E870 89C1
                                 <1>
                                           mov
                                                 ecx, eax
                                 <1> tfub_2:
6614
6615 0000E872 29C8
                                 <1>
                                           sub
                                                  eax, ecx
6616 0000E874 01CB
                                 <1>
                                           add
                                                 ebx, ecx
6617 0000E876 F3A4
                                 <1>
                                           rep
                                                 movsb
6618 0000E878 A3[88030300]
                                 <1>
                                           mov
                                                  [u.count], eax
6619 0000E87D 09C0
                                 <1>
                                                  eax, eax
                                           or
6620 0000E87F 75DD
                                 <1>
                                                 short tfub_1
6621
                                 <1>
6622 0000E881 EBB4
                                 <1>
                                           jmp
                                                 short tfub_retn
6623
                                 <1>
6624
                                 <1> ;tfub_retn:
6625
                                 <1>;
                                                 ecx ; transfer count = actual transfer count
                                           pop
6626
                                 <1> ;tfub 3:
6627
                                 <1> ;
                                                  edx
6628
                                 <1>;
                                                  ebx
                                           pop
                                 <1>;
6629
                                           pop
                                                  esi
6630
                                 <1>;
                                           pop
                                                  edi
                                 <1> ;tfub_4:
6631
                                 <1>;
6632
                                          retn
6633
                                 <1> ;tfub 5:
6634
                                 <1>;
                                           pop
                                                  ecx
                                                  ecx, [u.count]; actual transfer count
6635
                                 <1>;
                                           sub
                                 <1>;
6636
                                           stc
6637
                                 <1>;
                                           jmp
                                                  short tfub_3
6638
                                 <1>
                                 <1> sysfff: ; <Find First File>
6639
6640
                                          ; 17/10/2016
6641
                                 <1>
                                           ; 16/10/2016
                                           ; 15/10/2016 TRDOS 386 (TRDOS v2.0) feature only !
6642
                                 <1>
6643
                                 <1>
                                                       -derived from TRDOS v1.0, INT 21H.ASM-
6644
                                 <1>
                                                        ("loc_INT21h_find_first_file")
6645
                                 <1>
                                           ; TRDOS 8086 (v1.0)
                                             ; 07/08/2011
6646
                                 <1>
6647
                                 <1>
                                                 Find First File
6648
                                 <1>
                                                  INPUT:
                                 <1>
                                                     CX= Attributes
6649
6650
                                 <1>
                                                      DS:DX= Pointer to filename
6651
                                 <1>
                                                 MSDOS OUTPUT:
                                           ;
6652
                                 <1>
                                                     DTA: (Default address: PSP offset 80h)
6653
                                 <1>
                                                     Offset Descrription
                                           ;
                                 <1>
6654
                                                     0
                                                          Reserved for use find next file
6655
                                 <1>
                                                     21
                                                            Attribute of file found
6656
                                 <1>
                                                          Time stamp of file
                                                     22
6657
                                 <1>
                                                     24
                                                           Date stamp of file
6658
                                 <1>
                                                      26
                                                            File size in bytes
6659
                                 <1>
                                                     30
                                                            Filename and extension (zero terminated)
                                                  If cf = 1:
6660
                                 <1>
                                 <1>
                                                     Error Codes: (in AX)
6661
```

```
<1>
                                                         2 - File not found
6662
6663
                                  <1>
                                                         18 - No more files
                                           ;
6664
                                  <1>
6665
                                  <1>
                                            ; TRDOS 386 (v2.0)
                                  <1>
                                            ; 15/10/2016
6666
6667
                                  <1>
                                            ; INPUT ->
6668
                                  <1>
                                            ; CL = File attributes
6669
                                  <1>
                                                     bit 0 (1) - Read only file (R)
6670
                                  <1>
                                                        bit 1 (1) - Hidden file (H)
6671
                                  <1>
                                            ;
6672
                                  <1>
                                                           bit 2 (1) - System file (R)
                                                       bit 3 (1) - Volume label/name (V)
6673
                                  <1>
                                            ;
                                                           bit 4 (1) - Subdirectory (D)
6674
                                  <1>
                                                     bit 5 (1) - File has been archived (A)
6675
                                  <1>
6676
                                  <1>
                                                     CH = 0 -> Return basic parameters (24 bytes)
                                           ;
6677
                                  <1>
                                                     CH > 0 -> Return FindFile structure/table (128 bytes)
                                                       EBX = Pointer to filename (ASCIIZ) -path-
6678
                                  <1>
                                                     EDX = File parameters buffer address
6679
                                  <1>
                                           ;
6680
                                  <1>
                                                         (buffer size = 24 bytes if CH input = 0)
                                                         (buffer size = 128 bytes if CH input > 0)
6681
                                  <1>
                                            ;
6682
                                  <1>
                                           ; OUTPUT ->
6683
                                  <1>
                                                   EAX = 0 if CH input > 0
6684
                                  <1>
6685
                                  <1>
                                                     EAX = First cluster number of file if CH input = 0
6686
                                  <1>
                                                     EDX = File parameters table/structure address
6687
                                  <1>
                                                     Basic Parameters:
6688
                                  <1>
                                                      Offset Description
6689
                                  <1>
                                                         -----
                                                         0 File Attributes
6690
                                  <1>
                                                             Ambiguous filename chars are used sign
6691
                                  <1>
                                                       1
                                            ;
6692
                                  <1>
                                                               (0 = filename fits exactly with request)
6693
                                  <1>
                                                                (>0 = ambiguous filename chars are used)
                                                         2
                                                              Time stamp of file
6694
                                  <1>
6695
                                  <1>
                                                         4 Date stamp of file
                                                               File size in bytes
6696
                                  <1>
                                                         6
6697
                                  <1>
                                                         10
                                                                Short Filename (ASCIIZ, max. 13 bytes)
6698
                                  <1>
                                                                Longname Length (1-255) if existing
6699
                                  <1>
6700
                                  <1>
                                                       cf = 1 -> Error code in AL
6701
                                  <1>
6702
                                  <1>
                                          ; Modified Registers: EAX (at the return of system call)
6703
                                  <1>
                                           ; TR-DOS FindFile (FFF) Structure (128 bytes):
6704
                                  <1>
6705
                                  <1>
                                          ; 09/10/2011 (DIR.ASM) - 10/02/2016 (trdoskx.s)
6706
                                  <1>
6707
                                  <1>
                                           ; Offset
                                                         Parameter
                                                 FindFile_Drv 1 byte
                                           ; -----
6708
                                  <1>
                                          ; 0
                                           ; 1 FindFile_Directory 65 bytes
; 66 FindFile_Name 13 bytes
; 79 FindFile_LoreName
6709
                                  <1>
6710
                                  <1>
6711
                                  <1>
6712
                                  <1>
                                                        FindFile_LongNameEntryLength 1 byte
6713
                                  <1>
                                            ;Above 80 bytes form
6714
                                  <1>
                                           ;TR-DOS Source/Destination File FullName Format/Structure
6715
                                  <1>
                                          ; 80 FindFile_AttributesMask 1 word
                                          ; 82
6716
                                  <1>
                                                         FindFile_DirEntry 32 bytes (*)
                                                  FindFile_DirEntry 32 bytes (, FindFile_DirFirstCluster 1 double word FindFile_DirCluster 1 double word FindFile_DirEntryNumber 1 word FindFile_MatchCounter 1 word FindFile_Reserved 1 word
6717
                                  <1>
                                           ; 114
6718
                                  <1>
                                           ; 118
                                           ; 122
6719
                                  <1>
6720
                                  <1>
                                           ; 124
6721
                                  <1>
                                           ; 126
6722
                                  <1>
                                           ; (*) MS-DOS, FAT 12-16-32 classic directory entry (32 bytes)
6723
                                  <1>
6724
                                  <1>
                                           ;mov [u.namep], ebx
6725
                                  <1>
                                           ; 16/10/2016
                                           mov [FFF_UBuffer], edx
6726 0000E883 8915[70650100]
                                  <1>
6727 0000E889 66890D[75650100]
                                  <1>
                                                  [FFF_Attrib], cx ; [FFF_RType] = ch
6728
                                  <1>
                                                   ; Attributes in CL, return data type in CH
6729 0000E890 89DE
                                  <1>
                                           mov esi, ebx
6730
                                  <1>
                                            ; file name is forced, change directory as temporary
6731
                                  <1>
                                           ;mov ax, 1
6732
                                  <1>
                                           ;mov [FFF_Valid], ah ; 0 ; reset ; 17/10/2016
6733
                                  <1>
                                            ;call set_working_path
6734 0000E892 E8E8130000
                                            call set_working_path_x ; 17/10/2016
                                  <1>
6735 0000E897 731D
                                 <1>
                                           jnc short sysfff_0
6736
                                  <1>
6737 0000E899 21C0
                                                 eax, eax ; 0 -> Bad Path!
                                  <1>
                                            and
6738 0000E89B 7505
                                  <1>
                                            jnz short sysfff err
6739
                                  <1>
6740
                                  <1>
6741 0000E89D B80C000000
                                            mov eax, ERR_DIR_NOT_FOUND ; Directory not found !
                                  <1>
6742
                                  <1> sysfff_err:
6743 0000E8A2 A3[64030300]
                                  <1>
                                            mov
                                                   [u.r0], eax
                                            mov [u.error], eax
6744 0000E8A7 A3[C8030300]
                                  <1>
6745 0000E8AC E8A3140000
                                  <1>
                                             call reset_working_path
6746 0000E8B1 E98CDEFFFF
                                  <1>
                                            jmp error
6747
                                  <1>
6748
                                  <1> sysfff 0:
6749
                                  <1>
                                                 ah, ah; ah = 0
                                            ;sub
6750 0000E8B6 8A0424
                                 <1>
                                            mov
                                                  al, [esp]
6751 0000E8B9 08C0
                                 <1>
                                                  al, al
                                            or
6752 0000E8BB 7412
                                                   short sysfff 2
                                 <1>
                                            jΖ
6753 0000E8BD B410
                                  <1>
                                            mov
                                                  ah, 10h
6754 0000E8BF A808
                                                  al, 08h
                                 <1>
                                            test
6755 0000E8C1 7503
                                 <1>
                                                  short sysfff_1
                                            jnz
6756 0000E8C3 80CC08
                                 <1>
                                            or
                                                  ah, 08h
6757
                                 <1> sysfff_1:
6758 0000E8C6 2410
                                 <1>
                                                  al, 10h; Directory
                                            and
6759 0000E8C8 7405
                                 <1>
                                            jz
                                                   short sysfff_2
6760 0000E8CA 80E408
                                  <1>
                                            and
                                                  ah, 08h
6761 0000E8CD 30C0
                                  <1>
                                                  al, al; When a directory is searched,
                                            xor
                                                         ; filename will be returned even if
6762
                                  <1>
6763
                                  <1>
                                                         ; it is not a directory!
                                                         ; Because: (in order to prevent
6764
                                  <1>
6765
                                  <1>
                                                         ; creating a dir with existing file name)
6766
                                  <1>
                                                          ; Dir and file names must not be same!
```

```
6767
                                 <1>
                                                       ; (return attribute must be checked)
6768
                                 <1> sysfff 2:
                                         ; AX = Attributes mask
6769
                                 <1>
6770
                                <1>
                                               ; AL = AND mask (result must be equal to AL)
6771
                                <1>
                                                ; AH = Negative AND mask (result must be ZERO)
6772
                                <1>
                                          ; ESI = FindFile_Name address
6773
                                <1>
6774 0000E8CF E8139AFFFF
                                         call find first file
                                <1>
6775 0000E8D4 72CC
                                <1>
                                          jc short sysfff_err ; eax = 2 (File not found !)
6776
                                <1>
6777
                                <1>
                                         ; ESI = Directory Entry (FindFile_DirEntry) Location
6778
                                          ; EDI = Directory Buffer Directory Entry Location
                                 <1>
6779
                                          ; EAX = File Size
                                 <1>
6780
                                 <1>
                                          ; BL = Attributes of The File/Directory
6781
                                 <1>
                                         ; BH = Long Name Yes/No Status (>0 is YES)
6782
                                 <1>
                                          ; DX > 0 : Ambiguous filename chars are used
6783
                                 <1> sysfff_3:
6784
6785
                                 <1>
                                          ; 16/10/2016
6786 0000E8D6 668B0D[75650100]
                                <1>
                                          mov cx, [FFF Attrib]
                                          ; Attribs in CL, return data type in CH
6787
                                <1>
6788
                                 <1>
6789
                                <1>
                                          ; or cl, cl
6790
                                <1>
                                          ;jz short sysfff_4 ; 0 = No filter
                                                cl, OFFh
6791 0000E8DD 80F1FF
                                <1>
                                          xor
6792 0000E8E0 20D9
                                <1>
                                          and
                                                cl, bl
6793 0000E8E2 7409
                                <1>
                                                short sysfff_4
                                          jz
6794
                                <1>
6795
                                <1>
                                          ;mov eax, 2 ; 'file not found !' error
6796
                                          ;jmp short sysfff_err_1
                                <1>
6797
                                <1>
                                          ; 16/10/2016
6798
                                <1>
                                          call find_next_file
6799 0000E8E4 E8AD9AFFFF
                                <1>
6800 0000E8E9 72B7
                                <1>
                                                short sysfff_err ; eax = 12 (no more files !)
6801 0000E8EB EBE9
                                <1>
                                                short sysfff 3
6802
                                <1>
6803
                                <1> sysfff 4:
6804 0000E8ED 20ED
                                          and
                                                ch, ch ; [FFF_RType]
                                <1>
6805 0000E8EF 7412
                                <1>
                                                 short sysfff_5
                                          jΖ
6806 0000E8F1 B980000000
                                                ecx, 128; transfer length
                                <1>
                                          mov
6807 0000E8F6 880D[74650100]
                                <1>
                                          mov
                                                [FFF_Valid], cl
                                <1> sysfnf 11:
6809 0000E8FC BE[26620100]
                                <1>
                                                 esi, FindFile_Drv
                                          mov
6810 0000E901 EB44
                                <1>
                                                short sysfff_6
                                          jmp
6811
                                <1> sysfff_5:
6812
                                <1>
                                          ; mov
                                                esi, FindFile_DirEntry
6813 0000E903 B918000000
                                <1>
                                                ecx, 24 ; transfer length
                                          mov
6814 0000E908 880D[74650100]
                                <1>
                                                [FFF_Valid], cl
                                          mov
                                <1> sysfnf 12:
                                                edi, DTA ; FFF data transfer address
6816 0000E90E BF[306A0100]
                                <1> mov
6817
                                <1>
                                          ;mov al, [esi+DirEntry_Attr] ; 11
6818 0000E913 88D8
                                <1>
                                         mov
                                                al, bl ; File/Dir Attributes
6819 0000E915 887F17
                                <1>
                                         mov
                                                [edi+23], bh ; Longname length (0= none)
6820 0000E918 AA
                                <1>
                                        stosb
6821 0000E919 88D0
                                <1>
                                         mov al, dl; DL is for '?'
                                                al, dh ; DH is for '*'
6822 0000E91B 00F0
                                <1>
                                         add
                                <1>
                                         ; AL > 0 if ambiguous file name wildcards are used
6824 0000E91D AA
                                          stosb
                                <1>
                                        mov eax, [esi+DirEntry_WrtTime] ; 22
6825 0000E91E 8B4616
                                <1>
6826 0000E921 AB
                                         stosd ; DirEntry_WrtTime & DirEntry_WrtDate
                               <1>
                                         mov eax, [esi+DirEntry_FileSize] ; 28
6827 0000E922 8B461C
                               <1>
6828 0000E925 AB
                                <1>
                                          stosd
6829 0000E926 668B4614
                                <1>
                                         mov ax, [esi+DirEntry_FstClusHI] ; 20
6830 0000E92A 66C1E010
                                <1>
                                         shl ax, 16
                                                ax, [esi+DirEntry_FstClusLO] ; 26
6831 0000E92E 668B461A
                                <1>
                                          mov
6832 0000E932 A3[64030300]
                                <1>
                                                [u.r0], eax ; First Cluster
6833
                                <1>
6834
                                          ;movesi, FindFile_DirEntry
                                <1>
6835 0000E937 E855140000
                                <1>
                                          call get_file_name
                                <1>
6837 0000E93C 8A0D[74650100]
                                <1>
                                                cl, [FFF Valid]
6838 0000E942 BE[306A0100]
                                 <1>
                                                mov esi, DTA; FFF data transfer address
                                 <1> sysfff_6:
6839
6840 0000E947 8B3D[70650100]
                                <1>
                                         mov edi, [FFF_UBuffer] ; user's buffer address (edx)
                                          call transfer_to_user_buffer
6841 0000E94D E8ABFEFFFF
                                <1>
6842
                                <1>
6843 0000E952 890D[64030300]
                                <1>
                                          mov [u.r0], ecx; actual transfer count
                                          call reset_working_path
6844 0000E958 E8F7130000
                                <1>
6845 0000E95D E900DEFFFF
                                 <1>
                                          jmp sysret
6846
                                 <1>
                                 <1> sysfnf: ; <Find Next File>
6847
                                      ; 16/10/2016 TRDOS 386 (TRDOS v2.0) feature only !
6848
6849
                                 <1>
                                          ;
                                                      -derived from TRDOS v1.0, INT_21H.ASM-
6850
                                 <1>
                                                      ("loc_INT21h_find_next_file")
6851
                                 <1>
                                          ; TRDOS 8086 (v1.0)
                                            ; 07/08/2011
6852
                                 <1>
6853
                                 <1>
                                                Find First File
                                                INPUT:
6854
                                 <1>
6855
                                 <1>
                                                MSDOS OUTPUT:
6856
                                 <1>
                                                    DTA: (Default address: PSP offset 80h)
6857
                                 <1>
6858
                                 <1>
                                                    Offset Descrription
                                                    O Reserved for use find next file
6859
                                 <1>
                                                        Attribute of file found
6860
                                 <1>
                                                    21
                                                        Time stamp of file Date stamp of file
6861
                                 <1>
                                                    22
6862
                                 <1>
                                                    24
6863
                                 <1>
                                                    26 File size in bytes
6864
                                 <1>
                                                    30
                                                         Filename and extension (zero terminated)
                                                 If cf = 1:
6865
                                 <1>
                                 <1>
                                                  Error Codes: (in AX)
6866
6867
                                 <1>
                                                      18 - No more files
6868
                                 <1>
6869
                                 <1>
                                          ; TRDOS 386 (v2.0)
                                          ; 16/10/2016
6870
                                 <1>
6871
                                 <1>
```

```
6872
                                  <1>
                                              ; INPUT ->
6873
                                  <1>
                                                             none
6874
                                            ; OUTPUT ->
                                  <1>
6875
                                  <1>
                                                      EAX = 0 if CH input of 'Find First File' > 0
6876
                                  <1>
                                                      EAX = First cluster number of file
                                                          if CH input of 'Find First File' = 0
6877
                                  <1>
6878
                                  <1>
                                                      EDX = File parameters table/structure address
6879
                                  <1>
6880
                                  <1>
                                                       cf = 1 -> Error code in AL
                                  <1>
6881
6882
                                  <1>
                                            ; Modified Registers: EAX (at the return of system call)
6883
                                  <1>
6884
                                  <1>
6885
                                  <1>
                                            ; Note: If byte [FFF_Valid] = 0
6886
                                  <1>
                                                   'sysfnf' will return with 'no more files' error.
6887
                                  <1>
                                                   If byte [FFF_Valid] = 24
                                                   'sysfnf' will return with 32 bytes basic parameters
6888
                                  <1>
                                  <1>
                                                   at the address which is in EDX.
6889
6890
                                  <1>
                                                   If byte [FFF_Valid] = 128
                                                   'sysfnf' will return with 128 bytes Find File
6891
                                  <1>
6892
                                  <1>
                                                   Structure/Table at the address which is in EDX.
                                  <1>
6894 0000E962 803D[74650100]00
                                                  byte [FFF_Valid], 0
                                  <1>
                                            cmp
6895 0000E969 7714
                                  <1>
                                            jа
                                                  short stsfnf_0
                                            ; 'no more files !' error
6896
                                  <1>
6897 0000E96B B80C000000
                                  <1>
                                                  eax, ERR_NO_MORE_FILES ; 12
                                            mov
6898 0000E970 A3[64030300]
                                  <1>
                                            mov
                                                   [u.r0], eax
6899 0000E975 A3[C8030300]
                                  <1>
                                            mov
                                                  [u.error], eax
6900 0000E97A E9C3DDFFFF
                                  <1>
                                            jmp
                                                   error
6901
                                  <1> stsfnf 0:
6902
                                  <1>
                                            ;cmp byte [FFF_Valid], 128
6903
                                  <1>
                                                   short stsfnf 1
                                            ;je
                                                  byte [FFF Valid], 24
6904
                                  <1>
                                            ;cmp
6905
                                  <1>
                                                   short stsfnf 1
                                            ;je
6906
                                  <1>
                                                  [FFF_Valid], 24 ; Default
                                            ; mov
6907
                                  <1> stsfnf_1:
6908 0000E97F 0FB61D[86590100]
                                            movzx ebx, byte [Current Drv]
                                  <1>
6909 0000E986 66891D[7A650100]
                                  <1>
                                                   [SWP_DRV], bx
                                            mov
6910 0000E98D 8A15[26620100]
                                  <1>
                                                   dl, [FindFile_Drv]
                                            mov
6911 0000E993 38DA
                                                  dl, bl
                                  <1>
                                            cmp
6912 0000E995 750B
                                  <1>
                                            jne
                                                   short stsfnf_2
6913 0000E997 86FB
                                  <1>
                                            xchg
                                                  bh, bl
6914 0000E999 BE00010900
                                  <1>
                                                   esi, Logical_DOSDisks
                                            mov
6915 0000E99E 01DE
                                  <1>
                                            add
                                                   esi, ebx
6916 0000E9A0 EB0D
                                  <1>
                                            jmp
                                                   short sysfnf_3
6917
                                  <1>
                                  <1> stsfnf 2:
6918
6919 0000E9A2 FE05[7B650100]
                                  <1>
                                                   byte [SWP_DRV_chg]
                                            inc
6920
                                  <1>
6921 0000E9A8 E89785FFFF
                                  <1>
                                            call change_current_drive
6922 0000E9AD 7245
                                  <1>
                                                   short sysfnf_err_1 ; read error !
6923
                                  <1>
                                                                   ; (do not stop, because
                                                                   ; we don't have a
6924
                                  <1>
6925
                                  <1>
                                                                   ; 'no more files'
6926
                                  <1>
                                                                   ; -file not found- error,
6927
                                  <1>
                                                                   ; next sysfnf system call
6928
                                  <1>
                                                                   ; may solve the problem,
6929
                                  <1>
                                                                   ; after re-placing the disk)
6930
                                  <1> sysfnf_3:
6931 0000E9AF A1[9C620100]
                                  <1>
                                            mov
                                                   eax, [FindFile_DirCluster]
6932 0000E9B4 21C0
                                  <1>
                                            and
                                                   eax, eax
6933 0000E9B6 7550
                                  <1>
                                                   short sysfnf 6
                                            jnz
6934
                                  <1>
6935 0000E9B8 803D[85590100]02
                                  <1>
                                                   byte [Current_FATType], 2
                                                   short \operatorname{sysfnf} \operatorname{\underline{err}} 0; invalid, we need to stop !?
6936 0000E9BF 772C
                                  <1>
                                            jа
                                                   byte [Current_FATType], 1
6937 0000E9C1 803D[85590100]01
                                  <1>
                                            cmp
6938 0000E9C8 7223
                                  <1>
                                            jb
                                                   short sysfnf_err_0; invalid, we neeed to stop !?
6939
                                  <1>
6940 0000E9CA 3805[AC600100]
                                  <1>
                                                   byte [DirBuff_ValidData], al ; 0
                                            cmp
6941 0000E9D0 7608
                                  <1>
                                                   short sysfnf 4
                                            jna
6942
                                  <1>
6943 0000E9D2 3B05[B1600100]
                                                   eax, [DirBuff Cluster]; 0 ?
                                  <1>
                                            cmp
6944 0000E9D8 745E
                                  <1>
                                            jе
                                                   short sysfnf_9
6945
                                  <1>
6946
                                  <1>
                                            ;cmp byte [Current_Dir_Level], 0
6947
                                  <1>
                                              ;ja short sysfnf_4
6948
                                  <1>
                                              ;jna short sysfnf 9
6949
                                  <1>
                                  <1> sysfnf 4:
6950
6951 0000E9DA FE05[7B650100]
                                                   byte [SWP DRV chg]
                                  <1>
                                           inc
6952 0000E9E0 E842D3FFFF
                                  <1>
                                            call
                                                  load_FAT_root_directory
6953 0000E9E5 7351
                                  <1>
                                            jnc
                                                   short sysfnf_9
                                            ; eax = error code (17, 'drv not ready or read error')
                                  <1>
6955 0000E9E7 EB0B
                                  <1>
                                            jmp short sysfnf_err_1 ; read error ! (no FNF stop)
                                                                   ; (if you want, try again,
6956
                                  <1>
6957
                                  <1>
                                                                   ; after re-placing the disk)
                                  <1> sysfnf 5:
6958
6959 0000E9E9 3C0C
                                  <1>
                                                  al, 12 ; 'no more files' error
                                            cmp
6960 0000E9EB 7507
                                  <1>
                                                  short sysfnf_err_1 ; (no FNF stop -sysfnf will try
6961
                                  <1>
                                                                  ; to read the directory again,
6962
                                  <1>
                                                                   ; if the user calls sysfnf
6963
                                  <1>
                                                                   ; just after this error return-)
                                            ; (FNF stop -sysfnf will not try
6964
                                  <1>
                                            ; to read the directory again-)
6965
                                  <1>
6966
                                  <1>
6967
                                  <1> sysfnf_err_0:
6968 0000E9ED C605[74650100]00
                                  <1>
                                      mov byte [FFF_Valid], 0 ; FNF stop sign
                                  <1> sysfnf_err_1:
6969
6970 0000E9F4 A3[64030300]
                                  <1>
                                                   [u.r0], eax
6971 0000E9F9 A3[C8030300]
                                  <1>
                                            mov
                                                   [u.error], eax
6972 0000E9FE E851130000
                                  <1>
                                            call reset_working_path
6973 0000EA03 E93ADDFFFF
                                  <1>
                                            jmp
                                                  error
6974
                                  <1>
                                  <1> sysfnf 6:
6975
6976 0000EA08 803D[AC600100]00
                                  <1> cmp
                                                  byte [DirBuff ValidData], 0
```

```
6977 0000EA0F 7608
                                                  short sysfnf 7
                                  <1>
                                           jna
                                  <1>
6979 0000EA11 3B05[B1600100]
                                                   eax, [DirBuff Cluster]
                                  <1>
                                            cmp
6980 0000EA17 741F
                                  <1>
                                                  short sysfnf 9
6981
                                  <1>
                                  <1> sysfnf 7:
6982
6983 0000EA19 FE05[7B650100]
                                                  byte [SWP DRV chg]
                                  <1>
                                            inc
6984 0000EA1F 803D[85590100]01
                                 <1>
                                                  byte [Current FATType], 1
                                            cmp
6985 0000EA26 7309
                                  <1>
                                                  short sysfnf_8
                                            jnb
6986
                                  <1>
6987
                                           ; Singlix (TRFS) File System
                                  <1>
6988
                                           ; (access via compatibility buffer)
                                  <1>
6989 0000EA28 E8C2D3FFFF
                                            call load_FS_sub_directory
                                  <1>
6990 0000EA2D 7309
                                  <1>
                                                  short sysfnf_9
6991
                                  <1>
6992 0000EA2F EBC3
                                  <1>
                                            jmp
                                                  short sysfnf_err_1 ; read error (no FNF stop)
6993
                                  <1>
                                  <1> sysfnf 8:
6994
                                            call load FAT sub directory
6995 0000EA31 E87CD3FFFF
                                  <1>
6996 0000EA36 72BC
                                 <1>
                                            jс
                                                  short sysfnf_err_1 ; read error (no FNF stop)
6997
                                 <1>
                                  <1> sysfnf 9:
6998
6999 0000EA38 E85999FFFF
                                           call find next file
                                 <1>
7000 0000EA3D 72AA
                                 <1>
                                            jс
                                                  short sysfnf_5
                                  <1>
7001
                                                  al, [FFF_Attrib]
7002 0000EA3F A0[75650100]
                                 <1>
                                            mov
7003
                                 <1>
                                           ;or
                                                  al, al
7004
                                                  short sysfnf_10 ; 0 = No filter
                                 <1>
                                           ;jz
7005 0000EA44 34FF
                                  <1>
                                                  al, OFFh
                                            xor
7006 0000EA46 20D8
                                            and
                                  <1>
                                                 al, bl
7007 0000EA48 75EE
                                  <1>
                                            jnz short sysfnf_9; search for next file until
                                                               ; an error return from
7008
                                  <1>
7009
                                  <1>
                                                                ; find_next_file procedure
7010
                                  <1> sysfnf 10:
7011 0000EA4A 0FB60D[74650100]
                                        movzx
                                 <1>
                                                        ecx, byte [FFF_Valid]
                                            cmp cl, 128; complete FindFile structure/table
7012 0000EA51 80F980
                                  <1>
                                            je sysfnf 11
7013 0000EA54 0F84A2FEFFFF
                                  <1>
                                           ;cmp cl, 24 ; basic parameters
;je sysfnf_12
7014
                                  <1>
7015
                                  <1>
                                           ;je
7016 0000EA5A E9AFFEFFFF
                                  <1>
                                           jmp sysfnf_12
7017
                                  <1>
7018
                                  <1> writei:
                                          ; 26/10/2016
7019
                                  <1>
                                           ; 25/10/2016
7020
                                  <1>
                                          ; 23/10/2016
7021
                                  <1>
7022
                                  <1>
                                           ; 22/10/2016
                                           ; 19/10/2016 - TRDOS 386 (TRDOS v2.0)
7023
                                  <1>
                                          ; 19/05/2015 - 20/05/2015 (Retro UNIX 386 v1)
; 12/03/2013 - 31/07/2013 (Retro UNIX 8086 v1)
7024
                                  <1>
7025
                                  <1>
7026
                                  <1>
                                           ; Write data to file with first cluster number in EAX
7027
                                  <1>
7028
                                  <1>
                                           ; INPUTS ->
7029
                                  <1>
                                           ; EAX - First cluster number of the file
7030
                                  <1>
7031
                                  <1>
                                                EBX - File number (Open file index number)
7032
                                  <1>
                                                u.count - byte count to be written
                                               u.base - points to user buffer
7033
                                  <1>
                                               u.fofp - points to dword with current file offset
7034
                                  <1>
7035
                                  <1>
                                                i.size - file size
                                                cdev - logical dos drive number of the file
7036
                                  <1>
                                           ;
                                           ; OUTPUTS ->
7037
                                  <1>
7038
                                  <1>
                                           ; u.count - cleared
                                                u.nread - accumulates total bytes passed back
7039
                                  <1>
                                           ;
7040
                                  <1>
                                                i.size - new file size (if file byte offset overs file size)
7041
                                  <1>
                                                u.fofp - points to u.off (with new offset value)
7042
                                  <1>
                                           ; (Retro UNIX Prototype : 11/11/2012 - 18/11/2012, UNIXCOPY.ASM)
7043
                                  <1>
7044
                                  <1>
                                           ; ((Modified registers: eax, edx, ebx, ecx, esi, edi, ebp))
7045
                                  <1>
7046 0000EA5F 31C9
                                  <1>
                                           xor
                                                  ecx, ecx
                                                  [u.nread], ecx ; 0
7047 0000EA61 890D[8C030300]
                                  <1>
                                            mov
7048 0000EA67 66890D[C4030300]
                                  <1>
                                            mov
                                                   [u.pcount], cx; 19/05/2015
7049 0000EA6E 390D[88030300]
                                  <1>
                                                  [u.count], ecx
                                            cmp
7050 0000EA74 7701
                                  <1>
                                            jа
                                                  short writei_1
7051 0000EA76 C3
                                  <1>
                                           retn
7052
                                  <1> writei_1:
7053 0000EA77 881D[34650100]
                                  <1>
                                                   [writei.ofn], bl ; Open file number
                                           mov
7054 0000EA7D 880D[6F650100]
                                                  [setfmod], cl ; 0 ; reset 'update lm date&time' sign
                                  <1>
                                           mov
7055
                                  <1> dskw_0:
7056
                                       ; 26/10/2016
                                  <1>
                                            ; 22/10/2016, 23/10/2016, 25/10/2016
7057
                                  <1>
                                            ; 19/10/2016 - TRDOS 386 (TRDOS v2.0)
; 31/05/2015 - 25/07/2015 (Retro UNIX 386 v1)
7058
                                  <1>
7059
                                  <1>
7060
                                  <1>
                                            ; 26/04/2013 - 20/09/2013 (Retro UNIX 8086 v1)
7061
                                  <1>
                                            ; 01/08/2013 (mkdir_w check)
7062
                                  <1>
7063 0000EA83 E8D7000000
                                  <1>
                                            call mget w
7064
                                  <1>
                                            ; eax = sector/block number
7065
                                  <1>
7066 0000EA88 8B1D[74030300]
                                  <1>
                                                   ebx, [u.fofp]
                                            mov
7067 0000EA8E 8B13
                                  <1>
                                            mov
                                                   edx, [ebx]
7068 0000EA90 81E2FF010000
                                  <1>
                                            and
                                                   edx, 1FFh ; / test the lower 9 bits of the file offset
                                                   short dskw_1 ; / if its non-zero, branch
7069 0000EA96 750C
                                  <1>
                                            jnz
                                  <1>
                                                              ; if zero, file offset = 0,
                                                                    ; / 512, 1024,...(i.e., start of new block)
                                  <1>
7072 0000EA98 813D[88030300]0002- <1>
                                            cmp
                                                   dword [u.count], 512
7072 0000EAA0 0000
                                  <1>
                                                                ; / if zero, is there enough data to fill
7073
                                  <1>
7074
                                  <1>
                                                                ; / an entire block? (i.e., no. of
                                                   short dskw_2; / bytes to be written greater than 512.?
7075 0000EAA2 7337
                                  <1>
                                            jnb
                                                              ; / Yes, branch. Don't have to read block
7076
                                  <1>
7077
                                  <1> dskw 1: ; in as no past info. is to be saved
                                           ; (the entire block will be overwritten).
7078
                                  <1>
7079
                                  <1>
                                            ; 23/10/2016
7080
                                  <1>
```

```
7081 0000EAA4 BB[94070300]
                               <1>
                                        mov ebx, writei_buffer
                                        ; esi = logical dos drive description table address
                                <1>
7083
                                <1>
                                         ; eax = sector number
7084
                                <1>
                                        ; ebx = buffer address (in kernel's memory space)
7085
                                <1>
                                         ; ecx = sector count
7086 0000EAA9 B901000000
                                <1>
                                          mov ecx, 1
                                         call disk read
7087 0000EAAE E8A90D0000
                                <1>
7088
                                <1>
                                         ;call dskrd ; / no, must retain old info..
7089
                                <1>
                                                             ; / Hence, read block 'r1' into an I/O buffer
7090 0000EAB3 7326
                                <1>
                                          jnc short dskw 2
7091
                                <1>
7092
                                <1>
                                         ; disk read error
7093 0000EAB5 B811000000
                                <1>
                                         mov eax, 17; drive not ready or READ ERROR!
7094
                                <1> dskw_err: ; jump from disk write error
7095 0000EABA A3[64030300]
                                <1>
                                         mov [u.r0], eax
7096 0000EABF A3[C8030300]
                                <1>
                                                [u.error], eax
                                          mov
                                <1>
7098 0000EAC4 803D[6F650100]00
                                <1>
                                               byte [setfmod], 0
                                         cmp
7099 0000EACB 0F8671DCFFFF
                                <1>
7100
                                <1>
7101 0000EAD1 E8AF030000
                                <1>
                                          call update_file_lmdt; update last modif. date&time of the file
                                <1>
                                         ;mov byte [setfmod], 0
7103
                                <1>
7104 0000EAD6 E967DCFFFF
                                <1>
                                          jmp
                                                error
7105
                                <1>
7106
                                <1> dskw_2: ; 3:
                                     ; 23/10/2016
                                <1>
7108 0000EADB C605[10650100]01
                                <1>
                                               byte [writei.valid], 1 ; writei buffer contains valid data
7109 0000EAE2 56
                                <1>
                                         push esi; logical dos drive description table address
                                         ; EAX (r1) = block/sector number
7110
                                <1>
7111
                                <1>
                                         call wslot;
                                         ; jsr r0, wslot / set write and inhibit bits in I/O queue,
7112
                                <1>
7113
                                <1>
                                                        ; / proc. status=0, r5 points to 1st word of data
7114 0000EAE3 803D[C6030300]00
                                <1>
                                         cmp byte [u.kcall], 0
7115 0000EAEA 770F
                                <1>
                                                short dskw_4 ; zf=0 -> the caller is 'mkdir'
                                         jа
                                <1>
7117 0000EAEC 66833D[C4030300]00 <1>
                                                word [u.pcount], 0
                                          cmp
7118 0000EAF4 7705
                                <1>
                                          jа
                                                short dskw_4
                                <1> dskw 3:
7119
                                         ; [u.base] = virtual address to transfer (as source address)
7120
                                <1>
7121 0000EAF6 E821FAFFFF
                                <1>
                                          call trans_addr_r ; translate virtual address to physical (r)
7122
                                <1> dskw_4:
7123 0000EAFB BB[94070300]
                                <1>
                                         mov
                                                ebx, writei_buffer
7124
                                <1>
                                          ; EBX (r5) = system (I/O) buffer address
                                        call sioreg
7125 0000EB00 E883FAFFFF
                                <1>
                                         ; ESI = file (user data) offset
7126
                                <1>
7127
                                <1>
                                         ; EDI = sector (I/O) buffer offset
                                        ; ECX = byte count
7128
                                <1>
7129
                                <1>
7130 0000EB05 F3A4
                                <1>
                                         rep movsb
                                         ; 25/07/2015
7131
                                <1>
7132
                                <1>
                                          ; eax = remain bytes in buffer
7133
                                <1>
                                           ;
                                                  (check if remain bytes in the buffer > [u.pcount])
7134 0000EB07 09C0
                                <1>
                                          or
                                         jnz short dskw_3 ; (page end before system buffer end!)
7135 0000EB09 75EB
                                <1>
7136
                                <1>
                                <1>
                                         ; 23/10/2016
7138 0000EB0B B101
                                <1>
                                         mov cl, 1
7139 0000EB0D 5E
                                <1>
                                         pop
                                                esi
                                         mov eax, [writei.sector]
7140 0000EB0E A1[14650100]
                                <1>
7141
                                <1>
                                         ; esi = logical dos drive description table address
7142
                                <1>
                                         ; eax = sector number
7143
                                <1>
                                         ; ebx = writei buffer address
7144
                                <1>
                                         ; ecx = sector count
                                         call disk_write ; / yes, write the block
7145 0000EB13 E8350D0000
                                <1>
7146 0000EB18 7307
                                <1>
                                          jnc short dskw_5
                                <1>
7148 0000EB1A B812000000
                                <1>
                                               eax, 18 ; drive not ready or WRITE ERROR !
                                          mov
7149 0000EB1F EB99
                                <1>
                                                short dskw_err
                                          jmp
7150
                                <1>
7151
                                <1> dskw_5:
                                         ; 26/10/2016
7152
                                <1>
7153 0000EB21 0FB61D[34650100]
                                          movzx ebx, byte [writei.ofn]; open file number
                               <1>
7154 0000EB28 C0E302
                                <1>
                                        shl bl, 2; *4
7155 0000EB2B 8B83[04690100]
                                               eax, [ebx+OF_POINTER]
                                <1>
                                         mov
7156 0000EB31 3B83[2C690100]
                                <1>
                                          cmp
                                                eax, [ebx+OF_SIZE]
7157 0000EB37 7606
                                <1>
                                          jna short dskw 6
7158 0000EB39 8983[2C690100]
                                <1>
                                         mov
                                                [ebx+OF_SIZE], eax
7159
                                <1> dskw 6:
7160
                                <1>
                                         ;shr bl, 2
7161 0000EB3F 833D[88030300]00
                                <1>
                                          cmp dword [u.count], 0; / any more data to write?
                                                short dskw_7
7162 0000EB46 760A
                                 <1>
                                          jna
                                               eax, [writei.fclust]
7163 0000EB48 A1[24650100]
                                <1>
                                          mov
                                <1>
7164 0000EB4D E931FFFFFF
                                          jmp
                                                dskw_0 ; / yes, branch
                                <1> dskw 7:
7165
7166
                                          ; update last modif. date&time of the file
                                <1>
7167
                                <1>
                                          ; (also updates file size as OF SIZE)
7168 0000EB52 E82E030000
                                <1>
                                          call update_file_lmdt
7169
                                <1>
                                          ;mov byte [setfmod], 0
7170
                                <1>
                                         ; 03/08/2013
7171
                                <1>
7172 0000EB57 C605[C6030300]00
                                <1>
                                         mov byte [u.kcall], 0
                                          ; 23/10/2016
                                <1>
7173
7174
                                <1>
                                          ;mov eax, [writei.fclust]
7175 0000EB5E C3
                                <1>
                                         retn
7176
                                <1>
                                <1> mget_w:
7177
                                         ; 02/11/2016
7178
                                <1>
7179
                                <1>
                                          ; 01/11/2016
7180
                                          ; 23/10/2016, 31/10/2016
                                <1>
7181
                                <1>
                                         ; 22/10/2016 - TRDOS 386 (TRDOS v2.0)
7182
                                <1>
                                         ; 03/06/2015 (Retro UNIX 386 v1, 'mget', u.5s)
                                         ; 22/03/2013 - 31/07/2013 (Retro UNIX 8086 v1)
7183
                                <1>
7184
                                <1>
7185
                                <1>
                                          ; Get existing or (allocate) a new disk block for file
```

```
7186
                                 <1>
                                          ; INPUTS ->
7187
                                 <1>
                                               [u.fofp] = file offset pointer
7188
                                 <1>
7189
                                 <1>
                                               [i.size] = file size
7190
                                 <1>
                                               [u.count] = byte count
7191
                                 <1>
                                               EAX = First cluster
                                               [cdev] = Logical dos drive number
7192
                                 <1>
7193
                                 <1>
                                               [writei.ofn] = File Number
7194
                                 <1>
                                                           (Open file index, 0 based)
7195
                                 <1>
                                                ([u.off] = file offset)
                                          ;
                                           ; OUTPUTS ->
7196
                                 <1>
7197
                                 <1>
                                               EAX = logical sector number
                                           ;
                                               ESI = Logical Dos Drive Description Table address
7198
                                 <1>
7199
                                 <1>
7200
                                 <1>
                                          ; Modified registers: EDX, EBX, ECX, ESI, EDI, EBP
7201
                                 <1>
7202 0000EB5F 8B35[74030300]
                                 <1>
                                                    esi, [u.fofp]
                                                 ebp, [esi] ; u.off (or EBX*4+OF_POINTER)
7203 0000EB65 8B2E
                                 <1>
                                           mov
7204
                                 <1>
7205 0000EB67 29C9
                                 <1>
                                           sub
                                                 ecx, ecx
7206 0000EB69 8A2D[46030300]
                                                 ch, [cdev]
                                 <1>
                                 <1>
7208 0000EB6F BE00010900
                                 <1>
                                                 esi, Logical_DOSDisks
                                           mov
7209 0000EB74 01CE
                                 <1>
                                                 esi, ecx
7210
                                 <1>
7211
                                 <1>
                                           ; 31/10/2016
7212 0000EB76 89C3
                                 <1>
                                          mov ebx, eax; First Cluster or FDT address
7213
                                 <1>
7214 0000EB78 807E0300
                                 <1>
                                                 byte [esi+LD_FATType], 0
                                           cmp
7215 0000EB7C 0F86DD010000
                                 <1>
                                           jna mget_w_14 ; Singlix FS
7216
                                 <1>
7217 0000EB82 0FB74611
                                 <1>
                                          movzx eax, word [esi+LD BPB+BytesPerSec]
                                          movzx edx, byte [esi+LD BPB+SecPerClust]
7218 0000EB86 0FB65613
                                 <1>
7219 0000EB8A 8815[12650100]
                                 <1>
                                          mov [writei.spc], dl ; sectors per cluster
7220 0000EB90 F7E2
                                 <1>
                                          mul
                                                 edx
7221
                                 <1>
                                          ; edx = 0
                                           ; eax = bytes per cluster (<= 65536)
7222
                                 <1>
7223
                                 <1>
7224
                                 <1>
                                           ; 02/11/2016
7225 0000EB92 89C1
                                 <1>
                                           mov ecx, eax
7226 0000EB94 48
                                 <1>
                                           dec
                                                 eax
7227 0000EB95 66A3[18650100]
                                                 [writei.bpc], ax
                                 <1>
                                          mov
                                 <1>
7229 0000EB9B 89E8
                                 <1>
                                                 eax, ebp
                                           mov
7230 0000EB9D 0305[88030300]
                                 <1>
                                           add
                                                 eax, [u.count] ; next file position
7231 0000EBA3 3B05[55040300]
                                 <1>
                                           cmp
                                                 eax, [i.size] ; <= file size ?</pre>
7232 0000EBA9 0F86FC000000
                                                 mget_w_4 ; no
                                 <1>
                                           jna
7233
                                 <1>
7234 0000EBAF F7F1
                                 <1>
                                          div
7235 0000EBB1 A3[20650100]
                                 <1>
                                                [writei.c_index], eax ; cluster index
                                          mov
7236
                                 <1>
                                           ; edx = byte offset in cluster (<= 65535)
7237
                                 <1>
                                          ;mov [writei.offset], dx
7238
                                           ;shr dx, 9; / 512
                                 <1>
7239
                                 <1>
                                          ;mov [writei.s_index], dl ; sector index in cluster (0 to spc -1)
7240
                                 <1>
7241 0000EBB6 29D2
                                 <1>
                                           sub
                                                  edx, edx; 01/11/2016
7242 0000EBB8 8915[14650100]
                                 <1>
                                                  [writei.sector], edx; 0
                                          mov
                                                  [writei.offset], dx; byte offset in cluster
7243 0000EBBE 668915[1A650100]
                                 <1>
                                           mov
7244 0000EBC5 8815[13650100]
                                 <1>
                                                  [writei.s_index], dl ; sector index in cluster (0 to spc -1)
                                          mov
7245
                                 <1>
7246 0000EBCB 89D8
                                 <1>
                                          mov
                                                 eax, ebx ; First Cluster
7247
                                 <1>
7248
                                 <1>
                                           ; is this the 1st mget_w or a next mget_w call ? (by 'writei')
7249 0000EBCD 3815[10650100]
                                 <1>
                                                 byte [writei.valid], dl ; 0
7250 0000EBD3 7624
                                 <1>
                                                 short mget_w_0
                                           jna
7251
                                 <1>
7252 0000EBD5 8815[10650100]
                                                 byte [writei.valid], dl ; 0 ; reset ('writei' will set it)
                                 <1>
                                          mov
7253
                                 <1>
7254 0000EBDB 3B05[24650100]
                                 <1>
                                                 eax, [writei.fclust]
                                           cmp
7255 0000EBE1 7516
                                 <1>
                                           jne
                                                 short mget_w_0
7256
                                 <1>
7257 0000EBE3 8A0D[46030300]
                                 <1>
                                                 cl, [cdev]
                                          mov
7258 0000EBE9 3A0D[11650100]
                                                 cl, [writei.drv]
                                 <1>
                                           cmp
7259 0000EBEF 7508
                                 <1>
                                           jne short mget_w_0
                                          ; [writei.l_clust] & [writei.l_index] are valid,
7260
                                 <1>
                                           ; we don't need to get last cluster & last cluster index
7261
                                 <1>
7262 0000EBF1 8B0D[30650100]
                                 <1>
                                          mov ecx, [writei.l index]
7263 0000EBF7 EB64
                                 <1>
                                          jmp
                                                 short mget_w_2
                                 <1> mget_w_0:
7264
7265 0000EBF9 A3[24650100]
                                                 [writei.fclust], eax; first cluster
                                 <1>
                                          mov
7266
                                 <1>
                                           ; edx = 0
7267 0000EBFE A3[1C650100]
                                 <1>
                                           mov
                                                  [writei.cluster], eax; first cluster; 01/11/2016
                                                  [writei.fs_index], edx ; 0 ; curret cluster index
7268 0000EC03 8915[28650100]
                                <1>
                                 <1>
7270
                                 <1>
                                           ; FAT file system (FAT12, FAT16, FAT32)
7271 0000EC09 E8B9D7FFFF
                                 <1>
                                           call get_last_cluster
7272 0000EC0E 0F822B010000
                                 <1>
                                                 mget_w_err ; eax = error code
                                           jс
                                 <1>
                                                  [writei.lclust], eax ; last cluster
7274 0000EC14 A3[2C650100]
                                 <1>
                                 <1>
7276 0000EC19 8B0D[50630100]
                                 <1>
                                          mov
                                                 ecx, [glc_index] ; last cluster index
7277 0000EC1F 890D[30650100]
                                                 [writei.l index], ecx
                                 <1>
                                          mov
                                 <1>
7278
7279 0000EC25 A0[34650100]
                                 <1>
                                                 al, [writei.ofn]
7280 0000EC2A FEC0
                                 <1>
                                           inc
                                                 al
7281 0000EC2C A2[6F650100]
                                 <1>
                                           mov
                                                 [setfmod], al ; update lm date&time sign
                                 <1>
                                 <1> mget_w_1:
7283
7284 0000EC31 3B0D[20650100]
                                 <1>
                                           cmp
                                                 ecx, [writei.c_index] ; last cluster index
7285 0000EC37 7324
                                                 short mget_w_2; 01/11/2016
                                <1>
                                           jnb
7286
                                 <1>
7287 0000EC39 A1[2C650100]
                                <1>
                                          mov
                                                eax, [writei.lclust]
7288
                                 <1>
                                          ; EAX = Last cluster
7289 0000EC3E E892D8FFFF
                                 <1>
                                          call add new cluster
7290 0000EC43 0F82F6000000
                                 <1>
                                          jc mget_w_err ; eax = error code
```

```
7291
                                 <1>
                                        ; edx = 0
7292 0000EC49 A3[2C650100]
                                 <1>
                                          mov
                                                [writei.lclust], eax; (new) last cluster
7293 0000EC4E 8B0D[30650100]
                                <1>
                                          mov
                                                ecx, [writei.l_index]
7294 0000EC54 41
                                <1>
                                                ecx ; add 1 to last cluster index
                                          inc
7295 0000EC55 890D[30650100]
                                <1>
                                          mov
                                                [writei.l_index], ecx ; current last cluster index
7296
                                <1>
7297 0000EC5B EBD4
                                <1>
                                          jmp
                                                short mget w 1
7298
                                <1>
7299
                                <1> mget_w_2:
7300 0000EC5D 89E9
                                <1>
                                                ecx, ebp
                                          mov
7301 0000EC5F 030D[88030300]
                                          add
                                <1>
                                                ecx, [u.count]
7302 0000EC65 890D[55040300]
                                <1>
                                          mov
                                                 [i.size], ecx; save new file size
7303
                                <1>
                                          ;sub edx, edx; 0
7304
                                <1>
7305 0000EC6B A0[46030300]
                                <1>
                                         mov
                                                al, [cdev]
7306 0000EC70 A2[11650100]
                                <1>
                                         mov
                                                [writei.drv], al ; physical drive number
                                      ; edx = 0
mov eax, ebp; file offset
                                 <1>
7308 0000EC75 89E8
                                <1>
7309 0000EC77 0FB70D[18650100]
                                <1>
                                          movzx ecx, word [writei.bpc] ; bytes per cluster - 1
7310 0000EC7E 41
                                <1>
                                         inc ecx; bytes per cluster
7311 0000EC7F F7F1
                                <1>
                                        div ecx
                                        ; edx = byte offset in cluster (<= 65535)
                                 <1>
7313
                                          ; eax = cluster index
                                <1>
                                          mov [writei.c index], eax
7314 0000EC81 A3[20650100]
                                <1>
                                          mov [writei.offset], dx
7315 0000EC86 668915[1A650100]
                                <1>
7316 0000EC8D 66C1EA09
                                <1>
                                          shr
                                                dx, 9 ; / 512
7317 0000EC91 8815[13650100]
                                <1>
                                          mov [writei.s index], dl; sector index in cluster (0 to spc -1)
7318
                                <1>
                                 <1> mget_w_3:
7319
7320 0000EC97 3B05[30650100]
                                <1>
                                                eax, [writei.l_index] ; last cluster index
                                          cmp
7321 0000EC9D 752A
                                <1>
                                                short mget_w_5
7322
                                <1>
7323 0000EC9F A3[28650100]
                                                 [writei.fs_index], eax ; cluster index (for next check)
                                <1>
                                          mov
7324 0000ECA4 A1[2C650100]
                                <1>
                                                eax, [writei.lclust] ; last cluster
                                          mov
7325 0000ECA9 EB60
                                <1>
                                          jmp
                                                short mget_w_10
7326
                                <1>
7327
                                <1> mget_w_4: ; 02/11/2016
7328
                                        ; eax = next file position
                                <1>
7329 0000ECAB 2B05[88030300]
                                <1>
                                          sub eax, [u.count]; current file position
                                          ; edx = 0
7330
                                <1>
7331
                                <1>
                                          ; ecx = bytes per cluster
                                         div ecx
mov [writei.c_index], eax; cluster index
7332 0000ECB1 F7F1
                                 <1>
7333 0000ECB3 A3[20650100]
                                <1>
7334 0000ECB8 668915[1A650100]
                               <1>
                                          mov [writei.offset], dx
7335 0000ECBF 66C1EA09
                                                dx, 9 ; / 512
                                <1>
                                          shr
7336 0000ECC3 8815[13650100]
                                <1>
                                          mov
                                                [writei.s_index], dl ; sector index in cluster (0 to spc -1)
7337
                                <1>
7338
                                <1> mget_w_5:
7339 0000ECC9 21C0
                                <1>
                                          and
                                                 eax, eax ; 0 = First Cluster's index number
7340 0000ECCB 750C
                                <1>
                                          jnz
                                                short mget w 6
7341
                                <1>
7342 0000ECCD A3[28650100]
                                <1>
                                                 [writei.fs index], eax; cluster index (for next check)
                                          mov
7343 0000ECD2 A1[24650100]
                                <1>
                                          mov
                                                 eax, [writei.fclust] ; first cluster
7344 0000ECD7 EB32
                                <1>
                                                 short mget_w_10
7345
                                <1>
7346
                                <1> mget_w_6:
7347 0000ECD9 3B05[28650100]
                                <1>
                                          cmp
                                                eax, [writei.fs index] ; current cluster index (>0)
7348 0000ECDF 7507
                                <1>
                                          jne
                                                short mget_w_7
7349 0000ECE1 A1[1C650100]
                                <1>
                                                 eax, [writei.cluster] ; current cluster
                                          mov
7350 0000ECE6 EB3A
                                <1>
                                          jmp
                                                short mget_w_11
7351
                                <1>
                                <1> mget_w_7:
                                <1>
7353 0000ECE8 89C1
                                         mov
                                                ecx, eax
7354 0000ECEA 2B0D[28650100]
                                <1>
                                                ecx, [writei.fs_index]
                                          jnc short mget_w 8
7355 0000ECF0 730D
                                <1>
7356
                                <1>
                                          ; get cluster by index from the first cluster
7357 0000ECF2 A1[24650100]
                                <1>
                                          mov eax, [writei.fclust]
7358 0000ECF7 8B0D[20650100]
                                <1>
                                        mov
                                                ecx, [writei.c_index]
7359 0000ECFD EB05
                                <1>
                                          jmp
                                                short mget_w_9
7360
                                <1>
7361
                                <1> mget_w_8:
7362 0000ECFF A1[1C650100]
                                          mov
                                 <1>
                                                eax, [writei.cluster] ; beginning cluster
                                          ; ecx = cluster sequence number after the beginning cluster
7363
                                <1>
7364
                                 <1>
                                          ; sub edx, edx ; 0
7365
                                 <1>
                                 <1> mget_w_9:
7366
7367
                                 <1>
                                         ; EAX = Beginning cluster
7368
                                 <1>
                                          ; EDX = Sector index in disk/file section
7369
                                 <1>
                                               (Only for SINGLIX file system!)
7370
                                          ; ECX = Cluster sequence number after the beginning cluster
                                 <1>
7371
                                 <1>
                                          ; ESI = Logical DOS Drive Description Table address
7372 0000ED04 E8D2D8FFFF
                                 <1>
                                          call get_cluster_by_index
7373 0000ED09 7234
                                          jс
                                <1>
                                                short mget_w_err ; error code in EAX
                                 <1>
                                          ; EAX = Cluster number
7374
7375
                                 <1> mget_w_10:
7376 0000ED0B A3[1C650100]
                                                 [writei.cluster], eax; FDT number for Singlix File System
                                <1>
                                          mov
                                <1>
7378 0000ED10 807E0300
                                                byte [esi+LD_FATType], 0
                                <1>
                                          cmp
7379 0000ED14 7638
                                                short mget_w_13
                                <1>
                                          jna
                                          ; 01/11/2016
                                <1>
7381 0000ED16 8B15[20650100]
                                <1>
                                          mov
                                                edx, [writei.c_index]
7382 0000ED1C 8915[28650100]
                                 <1>
                                          mov
                                                [writei.fs index], edx
                                 <1> mget_w_11:
7383
7384 0000ED22 83E802
                                 <1>
                                         sub eax, 2
                                          movzx edx, byte [writei.spc]
7385 0000ED25 0FB615[12650100]
                                <1>
7386 0000ED2C F7E2
                                <1>
                                          mul
                                                edx
                                 <1>
7388 0000ED2E 034668
                                <1>
                                         add
                                                eax, [esi+LD_DATABegin]
7389 0000ED31 8A15[13650100]
                                <1>
                                          mov
                                                dl, [writei.s index]
7390 0000ED37 01D0
                                <1>
                                         add
                                                eax, edx
7391
                                <1> mget_w_12:
7392 0000ED39 A3[14650100]
                                <1>
                                          mov
                                                 [writei.sector], eax
7393
                                <1>
                                          ;; buffer validation must be done in writei
                                          ;;mov byte [writei.valid], 1
7394
                                <1>
7395 0000ED3E C3
                                <1>
```

```
7396
                                 <1>
7397
                                 <1> mget w err:
7398 0000ED3F A3[C8030300]
                                 <1>
                                         mov
                                                  [u.error], eax
7399 0000ED44 A3[64030300]
                                 <1>
                                                 [u.r0], eax
7400 0000ED49 E9F4D9FFFF
                                 <1>
                                           jmp
                                                 error
7401
                                 <1>
                                 <1> mget w 13:
7402
                                        ; EAX = FDT number (Current Section)
7403
                                 <1>
7404
                                 <1>
                                           ; EDX = Sector index from the first section (0,1,2,3,4...)
7405 0000ED4E 2B15[28650100]
                                           sub edx, [writei.fs index]
                                 <1>
7406
                                 <1>
                                           ; EDX = Sector index from current section
7407 0000ED54 8915[28650100]
                                 <1>
                                           mov [writei.fs index], edx
                                                 eax; the \overline{\text{first}} data sector in FS disk section
7408 0000ED5A 40
                                 <1>
                                           inc
7409 0000ED5B 01D0
                                           add eax, edx
                                 <1>
7410 0000ED5D EBDA
                                 <1>
                                           jmp
                                                 short mget_w_12
7411
                                 <1>
7412
                                 <1> mget_w_14:
7413 0000ED5F 8A4E12
                                                 cl, [esi+LD_FS_BytesPerSec+1]
                                 <1>
                                           mov
7414 0000ED62 D0E9
                                 <1>
                                                 cl, 1; ; 1 for 512 bytes, 4 for 2048 bytes
                                           shr
7415 0000ED64 880D[12650100]
                                 <1>
                                               [writei.spc], cl ; sectors per cluster
                                           mov
7416
                                 <1>
                                           ; NOTE: writei bytes per sector value is always 512 !
7417 0000ED6A 66C705[18650100]00- <1>
                                           mov
                                                word [writei.bpc], 512
7417 0000ED72 02
                                 <1>
7418
                                 <1>
7419 0000ED73 89E9
                                 <1>
                                           mov
                                                  ecx, ebp
7420 0000ED75 030D[88030300]
                                 <1>
                                           add
                                                  ecx, [u.count] ; next file position
7421 0000ED7B 3B0D[55040300]
                                 <1>
                                           cmp
                                                  ecx, [i.size]; <= file size?
7422 0000ED81 0F86C8000000
                                 <1>
                                           jna
                                                  mget_w_19 ; no
                                 <1>
7424 0000ED87 29D2
                                 <1>
                                           sub
                                                  edx, edx; 0
7425 0000ED89 8915[14650100]
                                 <1>
                                           mov
                                                  [writei.sector], edx; 0
7426 0000ED8F 668915[1A650100]
                                 <1>
                                           mov
                                                  [writei.offset], dx ; byte offset in cluster
7427 0000ED96 8815[13650100]
                                                  [writei.s_index], dl ; sector index in cluster (0 to spc -1)
                                 <1>
                                           mov
7428
                                 <1>
7429 0000ED9C C1E909
                                 <1>
                                                  ecx, 9; 1 cluster = 512 bytes
                                           shr
7430 0000ED9F 890D[20650100]
                                 <1>
                                                  [writei.c_index], ecx; section/cluster index
                                           mov
                                 <1>
7432 0000EDA5 89D8
                                 <1>
                                                  eax, ebx; FDT number (First FDT address)
                                           mov
7433
                                 <1>
                                           ; is this the 1st mget_w or a next mget_w call ? (by 'writei')
7434
                                 <1>
7435 0000EDA7 3815[10650100]
                                                 byte [writei.valid], dl ; 0
                                 <1>
                                           cmp
7436 0000EDAD 7624
                                  <1>
                                           jna
                                                 short mget w 15
7437
                                 <1>
7438 0000EDAF 8815[10650100]
                                 <1>
                                                  byte [writei.valid], dl ; 0 ; reset ('writei' will set it)
7439
                                 <1>
7440 0000EDB5 3B05[24650100]
                                 <1>
                                           cmp
                                                  eax, [writei.fclust]
7441 0000EDBB 7516
                                 <1>
                                                 short mget_w_15
                                           jne
7442
                                 <1>
7443 0000EDBD 8A0D[46030300]
                                 <1>
                                           mov
                                                 cl, [cdev]
7444 0000EDC3 3A0D[11650100]
                                                 cl, [writei.drv]
                                 <1>
                                           cmp
7445 0000EDC9 7508
                                 <1>
                                           jne short mget_w_15
                                           ; [writei.l_clust] & [writei.l index] are valid,
7446
                                 <1>
7447
                                 <1>
                                           ; we don't need to get last cluster & last cluster index
7448 0000EDCB 8B0D[30650100]
                                 <1>
                                           mov ecx, [writei.l_index]
                                                 short mget_w_17
7449 0000EDD1 EB49
                                 <1>
                                           qmŗ
7450
                                 <1> mget_w_15:
                                           mov
7451 0000EDD3 A3[24650100]
                                 <1>
                                                 [writei.fclust], eax; first section (FDT number)
7452
                                 <1>
                                           ; edx = 0
7453 0000EDD8 8915[1C650100]
                                 <1>
                                                  [writei.cluster], edx ; 0 ; current section
                                           mov
7454 0000EDDE 8915[28650100]
                                 <1>
                                                  [writei.fs_index], edx ; 0 ; curret section index
                                           mov
7455
                                 <1>
                                  <1>
                                           ; eax = FDT number (section 0 header address)
7457 0000EDE4 E81CD8FFFF
                                 <1>
                                           call get_last_section
                                                  mget_w_err ; eax = error code
7458 0000EDE9 0F8250FFFFFF
                                 <1>
                                           jс
7459
                                 <1>
7460 0000EDEF 8915[28650100]
                                 <1>
                                                  [writei.fs_index], edx; sector index in last section
                                           mov
7461
                                 <1>
7462 0000EDF5 A3[2C650100]
                                 <1>
                                                  [writei.lclust], eax ; last section address
                                           mov
                                 <1>
7463
7464 0000EDFA 8B0D[50630100]
                                 <1>
                                                  ecx, [glc_index] ; last section index
                                           mov
7465 0000EE00 890D[30650100]
                                 <1>
                                                  [writei.l_index], ecx
7466
                                  <1>
7467 0000EE06 A0[34650100]
                                 <1>
                                           mov
                                                  al, [writei.ofn]
7468 0000EE0B FEC0
                                 <1>
7469 0000EE0D A2[6F650100]
                                 <1>
                                                  [setfmod], al ; update lm date&time sign
                                           mov
7470
                                 <1>
7471
                                 <1> mget_w_16:
7472
                                 <1>
                                           ; edx = (existing) last section (sector) index
7473 0000EE12 8B0D[20650100]
                                           mov ecx, [writei.c index]; final section (sector) index
                                 <1>
7474 0000EE18 29D1
                                 <1>
                                           sub ecx, edx
7475 0000EE1A 7633
                                           jna short mget_w_19
                                 <1>
7476
                                  <1>
                                           ; ecx = sector count
                                 <1> mget_w_17:
7478 0000EE1C A1[2C650100]
                                                 eax, [writei.lclust]
                                 <1>
7479
                                 <1>
                                           ; ESI = Logical dos drv desc. table address
7480
                                 <1>
                                             ; EAX = Last section
                                 <1>
                                             ; (ECX = 0 for directory)
7482
                                 <1>
                                             ; ECX = sector count (except FDT)
7483 0000EE21 E8A2CDFFFF
                                 <1>
                                           call add_new_fs_section
                                           jnc short mget_w_18
7484 0000EE26 7312
                                 <1>
7485
                                 <1>
7486
                                 <1>
                                           ; If error number = 27h (insufficient disk space)
7487
                                 <1>
                                           ; it is needed to check free consequent sectors
7488
                                 <1>
                                           ; (1 data sector at least and +1 section header sector)
7489
                                 <1>
7490 0000EE28 83F827
                                 <1>
                                           cmp
                                                  eax, 27h
                                                 mget w err ; eax = error code
7491 0000EE2B 0F850EFFFFFF
                                 <1>
                                           jne
                                 <1>
7493
                                 <1>
                                           ; ecx = count of free consequent sectors
7494
                                           ; ecx must be > 1 (1 data + 1 header sector)
                                 <1>
7495 0000EE31 49
                                 <1>
                                           dec
                                                 ecx
7496 0000EE32 0F8407FFFFFF
                                 <1>
                                           jz
                                                 mget w err
7497 0000EE38 EBE2
                                 <1>
                                           jmp
                                                 short mget_w_17
7498
                                 <1>
7499
                                 <1> mget_w_18:
```

```
7500 0000EE3A A3[2C650100]
                                 <1>
                                       ; ecx = sector count (except section header)
                                         mov [writei.lclust], eax; (new) last section
7502 0000EE3F 8B15[30650100]
                                           mov edx, [writei.l index]
                                 <1>
7503 0000EE45 01CA
                                 <1>
                                           add edx, ecx; add sector count to index
7504 0000EE47 8915[30650100]
                                 <1>
                                           mov
                                                 [writei.l index], edx
7505 0000EE4D EBC3
                                 <1>
                                           jmp
                                                  short mget w 16
7506
                                 <1>
7507
                                 <1> mget_w_19:
7508 0000EE4F 89E9
                                 <1>
                                                  ecx, ebp
                                           mov
7509 0000EE51 030D[88030300]
                                 <1>
                                           add
                                                 ecx, [u.count]
7510 0000EE57 890D[55040300]
                                 <1>
                                           mov
                                                 [i.size], ecx ; save new file size
                                  <1>
                                           ;sub edx, edx; 0
7512
7513 0000EE5D A0[46030300]
7512
                                 <1>
                                 <1>
                                          mov al, [cdev]
                                         mov [writei.drv], al ; physical drive number
7514 0000EE62 A2[11650100]
                                 <1>
7515
                                 <1>
                                           ; edx = 0
7516 0000EE67 89E8
                                 <1>
                                           mov eax, ebp ; file offset
                                      mov eun,
; 1 cluster = 512 bycc
shr eax, 9 ; / 512
edx, 1FFh
offset in
7517 0000EE69 89C2
                                 <1>
                                           ; 1 cluster = 512 bytes (for Singlix FS)
7518
                                 <1>
7519 0000EE6B C1E809
                                 <1>
7520 0000EE6E 81E2FF010000
                                 <1>
                                         ; edx = byte offset in cluster/sector (<= 511)
; eax = section (sector/cluster) index</pre>
                                  <1>
7522
                                 <1>
7523 0000EE74 A3[20650100]
                                 <1>
                                           mov [writei.c_index], eax
                                           mov [writei.offset], dx
;mov byte [writei.s_index], 0 ; sector index in cluster
7524 0000EE79 668915[1A650100]
                                 <1>
                                           mov
                                  <1>
7526 0000EE80 E912FEFFFF
                                  <1>
                                           jmp mget_w_3
7527
                                  <1>
7528
                                  <1> update_file_lmdt: ; & update file size
7529
                                       \frac{1}{2} 26/\frac{1}{1}0/2016
                                  <1>
7530
                                  <1>
                                           ; 24/10/2016
7531
                                  <1>
                                           ; 23/10/2016
                                           ; 22/10/2016 - TRDOS 386 (TRDOS v2.0)
7532
                                  <1>
7533
                                  <1>
                                           ; Update last modification date&time of file
7534
                                  <1>
                                           ; (call from syswrite -> writei)
7535
                                  <1>
7536
                                           ; ((also updates file size)) // 26/10/2016
7537
                                  <1>
7538
                                  <1>
                                           ; INPUT:
7539
                                  <1>
                                           ; byte [setfmod] = open file number
7540
                                  <1>
                                           ; OUTPUT:
                                                 cf = 0 \rightarrow success !
7541
                                  <1>
                                                  cf = 1 -> lmdt update has been failed!
7542
                                  <1>
                                           ;
7543
                                  <1>
7544
                                  <1>
                                            ; Modified registers: eax, ebx, ecx, edx, esi, edi
7545
                                  <1>
7546
                                  <1>
7547
                                  <1>
                                           ;cmp byte [setfmod], 0
                                            ; jna short uflmdt 2 ; nothing to do
7548
                                  <1>
7549
                                  <1>
7550 0000EE85 31C0
                                  <1>
                                                 eax, eax
                                            xor
                                  <1>
7552 0000EE87 0FB61D[6F650100]
                                 <1>
                                           movzx ebx, byte [setfmod]
7553 0000EE8E FECB
                                 <1>
                                           dec bl ; open file index number (0 based)
                                  <1>
7555 0000EE90 8AA3[DC680100]
                                 <1>
                                           mov
                                                  ah, [ebx+OF_DRIVE]
7556 0000EE96 BE00010900
                                  <1>
                                           mov
                                                  esi, Logical DOSDisks
                                            add
7557 0000EE9B 01C6
                                  <1>
                                                  esi, eax
7558 0000EE9D C0E302
                                  <1>
                                            shl
                                                  bl, 2 ; *4
                                                  ecx, [ebx+OF_FCLUSTER] ; first cluster
7559 0000EEA0 8B8B[B4680100]
                                 <1>
                                           mov
7560 0000EEA6 8B93[7C690100]
                                 <1>
                                                  edx, [ebx+OF_DIRCLUSTER] ; dir cluster
                                  <1>
7562 0000EEAC DOEB
                                  <1>
                                                  bl, 1 ; /2
                                           shr
7563 0000EEAE 0FB7BB[1C6A0100]
                                  <1>
                                           movzx edi, word [ebx+OF_DIRENTRY]
7564
                                  <1>
7565 0000EEB5 803D[AC600100]01
                                  <1>
                                                  byte [DirBuff_ValidData], 1
7566 0000EEBC 726E
                                  <1>
                                           jb
                                                  short uflmdt 4
7567
                                  <1>
7568 0000EEBE A0[AA600100]
                                 <1>
                                                  al, [DirBuff_DRV]
7569 0000EEC3 2C41
                                 <1>
                                                 al, 'A'
                                           sub
                                                  al, ah
7570 0000EEC5 38E0
                                 <1>
                                            cmp
7571 0000EEC7 7563
                                                  short uflmdt 4 ; different drive
                                 <1>
                                            jne
7572 0000EEC9 8A4603
                                                  al, [esi+LD_FATType]
                                 <1>
                                           mov
7573 0000EECC 3A05[AB600100]
                                 <1>
                                           cmp
                                                  al, [DirBuff_FATType]
                                                  short uflmdt_5 ; different FS type
edx, [DirBuff_Cluster]
7574 0000EED2 755B
                                  <1>
                                           jne
7575 0000EED4 3B15[B1600100]
                                 <1>
                                            cmp
                                                  short uflmdt\overline{5}; different cluster
7576 0000EEDA 7553
                                  <1>
                                           jne
7577
                                  <1>
7578
                                  <1> uflmdt_1:
7579
                                       ; Directory buffer is ready here!
                                  <1>
                                            ; OF FCLUSTER must be compared/verified
7580
                                  <1>
7581 0000EEDC BE00000800
                                  <1>
                                            mov
                                                  esi, Directory_Buffer
                                                  di, 5; dir entry index * 32
7582 0000EEE1 66C1E705
                                 <1>
                                            shl
7583 0000EEE5 01FE
                                  <1>
                                                  esi, edi ; offset
7584
                                  <1>
7585 0000EEE7 F6460B18
                                                 byte [esi+DirEntry Attr], 18h; Vol & Dir
                                 <1>
                                            test
7586 0000EEEB 750F
                                                  short uflmdt 2 ; not a valid file !
                                 <1>
                                           jnz
7587 0000EEED 668B4614
                                 <1>
                                                  ax, [esi+DirEntry_FstClusHI]
                                           mov
7588 0000EEF1 C1E010
                                 <1>
                                           shl
                                                  eax, 16
7589 0000EEF4 668B461A
                                 <1>
                                                  ax, [esi+DirEntry FstClusLO]
                                           mov
7590 0000EEF8 39C8
                                 <1>
                                            cmp
                                                  eax, ecx; same first cluster?
                                                  short uflmdt_3 ; yes, it is OK !!!
7591 0000EEFA 7407
                                  <1>
                                           jе
7592
                                 <1>
                                  <1> uflmdt 2:
7593
7594
                                  <1>
                                           ; save directory buffer if has modified/changed sign
7595
                                  <1>
                                            ; (It is good to save dir buff even if the searched
                                  <1>
                                            ; directory entry is not found !?)
7597 0000EEFC E81BBAFFFF
                                 <1>
                                            call save_directory_buffer
7598 0000EF01 F9
                                  <1>
                                            stc ; update failed
7599 0000EF02 C3
                                 <1>
                                           retn
7600
                                 <1>
7601
                                  <1> uflmdt 3:
7602
                                           ; Update directory entry
                                 <1>
7603
                                  <1>
                                            ; 26/10/2016
7604 0000EF03 D0E3
                                  <1>
                                            shl bl, 1; *2
```

```
7605 0000EF05 8B83[2C690100]
                               <1>
                                                eax, [ebx+OF_SIZE] ; file size
                                       mov
7606 0000EF0B 89461C
                                <1>
                                         mov
                                               [esi+DirEntry FileSize], eax
7607
                               <1>
7608 0000EF0E E86BB9FFFF
                               <1>
                                         call convert_current_date_time
7609
                                <1>
                                         ; OUTPUT -> DX = Date in dos dir entry format
7610
                                                  AX = Time in dos dir entry format
                                <1>
7611 0000EF13 66894616
                                     mov
                                                [esi+DirEntry WrtTime], ax
                               <1>
7612 0000EF17 66895618
                                     mov
                               <1>
                                               [esi+DirEntry_WrtDate], dx
7613 0000EF1B 66895612
                               <1>
                                               [esi+DirEntry_LastAccDate], dx
                                         mov
                                               byte [DirBuff ValidData], 2
7614 0000EF1F C605[AC600100]02
                             <1>
                                         mov
7615 0000EF26 E8F1B9FFFF
                                <1>
                                         call save_directory_buffer
7616 0000EF2B C3
                                <1>
7617
                                <1>
7618
                                <1> uflmdt 4:
7619
                                <1>
                                         ; Directory buffer sector read&write
7620
                                <1>
                                          ; 23/10/2016
7621
                                <1>
                                               al, [esi+LD_FATType]
7622 0000EF2C 8A4603
                                <1>
                                         mov
7623
                                <1> uflmdt 5:
                                               ebx, rw_buffer ; Common r/w sector buffer addr
7624 0000EF2F BB[9C090300]
                                <1>
                                         mov
7625
                                <1>
7626 0000EF34 20C0
                                               al, al ; 0 = Singlix FS
                                <1>
                                          and
7627 0000EF36 0F8492000000
                               <1>
                                                uflmdt_11
                                         jz
                                <1>
7629 0000EF3C 21D2
                                <1>
                                         and
                                                edx, edx
7630 0000EF3E 7521
                               <1>
                                         jnz
                                                short uflmdt_9
                               <1>
7632 0000EF40 3C02
                               <1>
                                               al, 2 ; 3 = FAT32
                                         cmp
                               -inp
-inp
-inp
-inp
-inp
7633 0000EF42 771A
                                                short uflmdt_8
7634
                               <1> mov
7635 0000EF44 89F8
                                               eax, edi ; directory entry index number
7636 0000EF46 66C1E804
                               <1>
                                         shr
                                               ax, 4 ; 16 entries per sector
7637 0000EF4A 034664
                                         add eax, [esi+LD ROOTBegin]
                               <1>
7638
                               <1>
                                         ; eax = root directory sector
7639
                               <1> uflmdt_6:
                               - push eax; *; disk sector address
7640 0000EF4D 50
7641 0000EF4E 51
                                         push ecx; first cluster
                               <1>
7642 0000EF4F B901000000
                                         mov ecx, 1
                               <1>
7643
                               <1>
                                         ; ecx = sector count
7644 0000EF54 E803090000
                               <1>
                                         call disk_read
7645 0000EF59 59
                               <1>
                                         pop ecx
7646 0000EF5A 731A
                                                short uflmdt 10
                               <1>
                                          jnc
7647 0000EF5C 58
                               <1>
                                         pop
                                                eax ; *
7648
                               <1> uflmdt 7:
7649 0000EF5D C3
                                <1>
                                         retn
7650
                                <1>
                                <1> uflmdt 8:
7652 0000EF5E 8B5632
                                                edx, [esi+LD_BPB+FAT32_RootFClust]
                               <1>
                                         mov
7653
                                <1> uflmdt_9:
                               <1> cmp
                                                edx, 2
7654 0000EF61 83FA02
7655 0000EF64 72F7
                               <1>
                                                short uflmdt_7; invalid, nothing to do
                               <1>
                               7657 0000EF66 83EA02
7658 0000EF69 89D0
7659 0000EF6B 0FB65613
7660 0000EF6F F7E2
7661 0000EF71 034668
7662
7663 0000EF74 EBD7
                                <1>
                                         jmp short uflmdt_6
7664
                                <1>
                                <1> uflmdt_10:
7665
                                      ; Directory sector buffer is ready here!
7666
                                <1>
7667
                                <1>
                                         ; OF FCLUSTER must be compared/verified
                                <1>
                                         ; edi = dir entry index number (<= 2047)
                                         and di, 0Fh; 16 entries per sector shl di, 5; dir entry index * 32
7669 0000EF76 6683E70F
                                <1>
                                               di, 5 ; dir entry index * 32
7670 0000EF7A 66C1E705
                                <1>
7671 0000EF7E 81C7[9C090300]
                                         add edi, rw_buffer
                                <1>
7672
                                <1>
7673 0000EF84 F6470B18
                                <1>
                                         test byte [edi+DirEntry_Attr], 18h; Vol & Dir
7674 0000EF88 0F856EFFFFF
                               <1>
                                         jnz
                                               uflmdt 2 ; not a valid file !
                                                dx, [edi+DirEntry FstClusHI]
7675 0000EF8E 668B5714
                               <1>
                                          mov
7676 0000EF92 C1E210
                                                edx, 16
                                <1>
                                          shl
7677 0000EF95 668B571A
                                               dx, [edi+DirEntry_FstClusLO]
                               <1>
                                         mov
                                                edx, ecx; same \overline{\text{first}} cluster?
7678 0000EF99 39CA
                               <1>
7679 0000EF9B 0F855BFFFFFF
                                         jne
                                <1>
                                               uflmdt_2 ; no !?
7680
                                <1>
                                <1>
                                         ; Update directory entry
7682 0000EFA1 E8D8B8FFFF
                                <1>
                                         call convert_current_date_time
7683
                                <1>
                                         ; OUTPUT -> DX = Date in dos dir entry format
                                                   AX = Time in dos dir entry format
7684
                                <1>
                                          ;
7685 0000EFA6 66894716
                                <1>
                                         mov
                                                [edi+DirEntry_WrtTime], ax
7686 0000EFAA 66895718
                                <1>
                                          mov
                                                [edi+DirEntry_WrtDate], dx
7687 0000EFAE 66895712
                                <1>
                                                [edi+DirEntry_LastAccDate], dx
                                          mov
                                <1>
7689 0000EFB2 58
                                <1>
                                                eax ; *
                                         pop
7690
                                <1>
                                                ebx, rw buffer ; Common r/w sector buffer addr
7691 0000EFB3 BB[9C090300]
                                <1>
                                         mov
7692 0000EFB8 B901000000
                                <1>
                                               ecx, 1
                                         mov
7693
                                <1>
                                         ; esi = logical dos description table address
                                         ; eax = disk sector number/address (LBA)
7694
                                <1>
7695
                                <1>
                                         ; ecx = sector count
7696
                                <1>
                                         ; ebx = buffer address
7697 0000EFBD E88B080000
                                <1>
                                          call disk write
7698 0000EFC2 0F8234FFFFFF
                                <1>
                                                uflmdt 2
7699
                                <1>
7700
                                          ; save directory buffer if has modified/changed sign
                                <1>
7701 0000EFC8 E84FB9FFFF
                                <1>
                                          call save_directory_buffer
7702 0000EFCD C3
                                <1>
7703
                                <1>
7704
                                <1> uflmdt 11:
                                         ; 24/10/2016
7705
                                <1>
7706
                                <1>
                                          ; Update last modification date & time of a file
7707
                                <1>
                                          ; on a disk with Singlix File System.
7708
                                <1>
7709
                                <1>
                                          ; (Method: Read the FDT -File Description Table-
```

```
7710
                                  <1>
                                            ; sector of the file and update the lmdt data fields,
7711
                                  <1>
                                           ; then write FDT sector to the disk.
7712
                                           ; /// It is easy but there is compatibility buffer
                                  <1>
7713
                                  <1>
                                           ; method also for changing directory entry data and
7714
                                  <1>
                                           ; also there are some programming issues for Singlix
7715
                                  <1>
                                            ; file system (TRFS), which are not completed yet!)
7716
                                  <1>
                                           ; Not ready yet ! (24/10/2016)
7717
                                  <1>
7718
                                  <1>
                                            ; /// Temporary code for error return ! ///
7719 0000EFCE 31C0
                                  <1>
                                            xor
                                                 eax, eax
7720 0000EFD0 F9
                                  <1>
                                            stc
7721 0000EFD1 C3
                                  <1>
                                            retn
7722
                                  <1>
7723
                                  <1> sysalloc:
                                         ; 14/10/2017
7724
                                  <1>
7725
                                           ; 20/08/2017, 01/09/2017
                                  <1>
                                           ; 20/02/2017, 04/03/2017, 15/05/2017
7726
                                  <1>
                                          ; 19/02/2017 - TRDOS 386 (TRDOS v2.0)
7727
                                  <1>
7728
                                  <1>
                                           ; (TRDOS 386 feature only!)
7729
                                  <1>
7730
                                  <1>
                                           ; Allocate Contiguous Memory Block/Pages (for user)
7731
                                  <1>
                                           ; (System call for DMA Buffer allocation etc.)
7732
                                  <1>
7733
                                  <1>
                                           ; INPUT ->
7734
                                  <1>
                                                  EBX = Virtual address (for user)
7735
                                  <1>
                                                       (Physical memory block/aperture
7736
                                  <1>
                                                       will be mapped to this virtual address)
                                                  ECX = Byte Count
7737
                                  <1>
7738
                                  <1>
                                                       (will be rounded up to page border)
7739
                                                  If ECX = 0
                                  <1>
                                            ;
7740
                                  <1>
                                                      System call will return with an error (cf=1)
7741
                                  <1>
                                                      but ECX will contain maximum size of
7742
                                                      available memory aperture and physical
                                  <1>
7743
                                  <1>
                                                      (beginning) address of that aperture
7744
                                  <1>
                                                      (which have maximum size) will be in EAX.
7745
                                  <1>
                                                  EDX = Upper limit of the requested physical memory
7746
                                  <1>
                                                        block/pages.
7747
                                  <1>
                                                       (The last byte address of the memory aperture
                                                        must not be equal to or above this limit.)
7748
                                  <1>
7749
                                  <1>
                                                  If EDX = 0
7750
                                                     there is NOLIMIT !
                                  <1>
                                                  If EDX = OFFFFFFFF (-1)
7751
                                  <1>
                                                     ESI = Lower Limit !
7752
                                  <1>
                                                         (Beginning of the block must not be 'less'
7753
                                  <1>
7754
                                  <1>
                                                         than this.) (Must be equal to or above...)
7755
                                  <1>
                                                     EDI = Upper Limit !
7756
                                  <1>
                                                        (End of the block must be !less! than this)
                                                         (The last byte addr of the memory aperture
7757
                                  <1>
7758
                                  <1>
                                                         must not be equal to or above this limit.)
7759
                                  <1>
                                           ; OUTPUT ->
7760
                                  <1>
                                                  If CF = 0
7761
                                  <1>
                                                  {\tt EAX} = Physical address of the allocated memory block
7762
                                  <1>
7763
                                  <1>
                                                  ECX = Allocated bytes (as rounded up to page borders)
7764
                                  <1>
                                                  EBX = Virtual address (as rounded up)
7765
                                  <1>
                                                  IF CF = 1
7766
                                  <1>
                                                      Requested (size of) Memory block could not be
                                                      allocated to the user!
7767
                                  <1>
7768
                                  <1>
                                                  IF CF = 1 & EAX = 0 (Insufficient memory error!)
                                                  ECX = Total number of free bytes
7769
                                  <1>
7770
                                  <1>
                                                           (not size of available contiguous bytes!)
                                                  If CF = 1 \& EAX > 0
7771
                                  <1>
                                            ;
7772
                                  <1>
                                                     there is not a memory aperture with requested size
7773
                                  <1>
                                                     but total free mem is not less than requested size.
7774
                                  <1>
                                                     EAX = Physical addr of available memory aperture
7775
                                  <1>
                                                         with max size
7776
                                                          (but it doesn't fit to the conditions!)
                                  <1>
                                                     ECX = Size of available memory aperture in bytes.
7777
                                  <1>
7778
                                  <1>
                                                  If CF = 1 \rightarrow EAX = OFFFFFFFF
7779
                                  <1>
                                                     Conditions/Parameters are wrong !
                                           ;
7780
                                  <1>
                                                     ECX is same with input value.
7781
                                  <1>
7782
                                                         Previously allocated pages will be deallocated if
                                  <1>
                                           ; Note:
                                                    new allocation conditions are met.
7783
                                  <1>
7784
                                  <1>
7785
                                  <1>
                                            ; Note: u.break control may be included in future versions
7786
                                  <1>
7787
                                  <1>
7788 0000EFD2 31C0
                                  <1>
                                            xor eax, eax; 0
7789
                                            ; 14/10/2017
                                  <1>
7790 0000EFD4 4A
                                            dec edx; is there a limit?
                                  <1>
7791 0000EFD5 7810
                                  <1>
                                            js
                                                  short sysalloc_1 ; 0 -> 0FFFFFFFFh -> NO LIMIT
7792 0000EFD7 42
                                  <1>
                                           inc edx ; > 0
7793
                                  <1>
                                            ; Check upper address limit
                                           ; (round up to page borders)
7794
                                  <1>
7795 0000EFD8 81C1FF0F0000
                                                 ecx, PAGE_SIZE-1 ; 4095
                                 <1>
                                            add
7796 0000EFDE 6681E100F0
                                 <1>
                                                 cx, ~PAGE OFF; not 4095
                                            and
7797 0000EFE3 39CA
                                 <1>
                                            cmp edx, ecx; upper limit - block size
7798 0000EFE5 7224
                                 <1>
                                                  short sysalloc_err
                                            jb
7799
                                 <1> sysalloc 1:
7800
                                 <1>
                                            ; EAX = Beginning address (physical)
7801
                                            ; EAX = 0 \rightarrow Allocate mem block from the 1st proper aperture
                                  <1>
7802
                                 <1>
                                            ; ECX = Number of bytes to be allocated
7803 0000EFE7 E89564FFFF
                                 <1>
                                            call allocate_memory_block
                                                  short sysalloc_err
7804 0000EFEC 721D
                                           jс
                                 <1>
                                            ; 01/09/2017
7805
                                 <1>
                                            sub
7806 0000EFEE 29C2
                                 <1>
                                                 edx, eax; upper limit address - beginning address
7807 0000EFF0 760F
                                 <1>
                                            jna
                                                  short sysalloc_3; begin addr not less than the limit
7808 0000EFF2 39CA
                                 <1>
                                            cmp
                                                  edx, ecx
7809 0000EFF4 720B
                                 <1>
                                                  short sysalloc_3 ; end address overs the limit
                                            jb
7810
                                 <1> sysalloc_2:
7811
                                  <1>
                                           ; EAX = Beginning (physical) addr of the allocated mem block
7812
                                 <1>
                                            ; ECX = Num of allocated bytes (rounded up to page borders)
7813 0000EFF6 50
                                  <1>
                                            push eax ; * ; 04/03/2017
                                  <1>
7814
                                            ; Here, requested contiquous memory pages have been allocated
```

```
7815
                                 <1>
                                          ; on Memory Allocation Table but user's page directory
7816
                                 <1>
                                          ; and page tables have not been updated yet!
7817 0000EFF7 51
                                          push ecx; **
                                 <1>
7818
                                 <1>
                                          ; ebx = virtual address (will be rounded up to page border)
7819
                                 <1>
                                         ; ecx = number of bytes to be deallocated
7820
                                 <1>
                                           ; will be adjusted to ebx+ecx round down - ebx round up
7821 0000EFF8 E8DF67FFFF
                                          call deallocate user pages
                                <1>
                                          jnc short sysalloc_{4}; EAX = Deallocated memory bytes
7822 0000EFFD 731F
                                 <1>
7823 0000EFFF 59
                                 <1>
                                                 ecx ; **
                                          pop
                                                eax ; *
7824 0000F000 58
                                <1>
                                          pop
                                 <1> sysalloc_3:
7825
                                        ; error !
7826
                                 <1>
                                          ; restore Memory Allocation Table Content
7827
                                <1>
                                <1> call deallocate_memory_block
7828 0000F001 E88866FFFF
7829 0000F006 31C0
                                <1>
                                        xor eax, eax; 0
7830 0000F008 48
                                                 eax ; 0FFFFFFFFh ; 15/05/2017
                                <1>
                                          dec
                                <1> dec eax; Ufffffffff ; 1<
<1> jmp short sysalloc_wrong
7831 0000F009 EB09
7832
                                <1> sysalloc err:
                                      mov ebp, [u.usp] ; ebp points to user's registers
7833 0000F00B 8B2D[60030300]
                                <1>
                                                [ebp+24], ecx; return to user with ecx value
7834 0000F011 894D18
                                <1>
                                          mov
7835
                                <1> sysalloc_wrong:
                                      ; eax = OFFFFFFFFh
                                 <1>
7837 0000F014 A3[64030300]
                                           mov [u.r0], eax
                                <1>
7838 0000F019 E924D7FFFF
                                <1>
                                         jmp
                                                error
7839
                                 <1> sysalloc 4:
7840 0000F01E 8B2D[60030300]
                                 <1>
                                          mov
                                                ebp, [u.usp] ; ebp points to user's registers
7841 0000F024 894518
                                <1>
                                                 [ebp+24], eax; return to user with ecx value
                                          mov
7842 0000F027 895D10
                                <1>
                                          mov
                                                [ebp+16], ebx; new value of ebx (rounded up)
7843 0000F02A 89C1
                                <1>
                                                 ecx, eax ; byte count (from 'deallocate_user_pages')
                                          mov
                                <1>
<1>
<1>
7844 0000F02C 5A
                                                edx ; ** ; discard (another) byte count
                                          pop
                                                 eax ; *
7845 0000F02D 58
                                          pop
7846 0000F02E A3[64030300]
                                 <1>
                                                 [u.r0], eax ; physical address
                                          mov
7847
                                 <1>
7848 0000F033 51
                                 <1>
                                          push ecx ; 20/08/2017
7849
                                 <1>
7850
                                 <1>
                                          ; Write newly allocated contiguous (physical) pages
7851
                                 <1>
                                           ; on page dir and page tables of current user/process
                                           ; as PRESENT, USER, WRITABLE
7852
                                 <1>
7853
                                 <1>
                                           ; (then clear allocated pages)
                                           call allocate user_pages
7854 0000F034 E89868FFFF
                                 <1>
7855
                                 <1>
                                           ;jnc sysret; OK! return to process with success...
7856
                                 <1>
                                           ; 20/08/2017 ('sysdma' modification)
7857
                                 <1>
7858 0000F039 59
                                 <1>
                                           pop
7859 0000F03A A1[64030300]
                                 <1>
                                                 eax, [u.r0] ; physical address (of the block)
                                           mov
7860
                                 <1>
7861 0000F03F 721D
                                 <1>
                                           jс
                                                 short sysalloc 6
7862
                                 <1>
7863 0000F041 833D[846F0100]FF
                                 <1>
                                           cmp
                                                 dword [dma_addr], OFFFFFFFF ; -1
7864 0000F048 0F8214D7FFFF
                                 <1>
                                           jb
                                                 sysret
7865
                                 <1>
7866 0000F04E A3[846F0100]
                                 <1>
                                                  [dma addr], eax; save dma address for sysdma
                                           mov
7867 0000F053 890D[886F0100]
                                 <1>
                                           mov
                                                  [dma_size], ecx; save dma buff size for sysdma
7868
                                 <1>
7869 0000F059 E904D7FFFF
                                 <1>
                                                 sysret
                                           jmp
7870
                                 <1>
7871
                                 <1> sysalloc 6:
7872
                                 <1>
7873
                                 <1>
                                           ; unexpected error ! insufficient memory !? conflict !?
                                           ; (!!?there is not a free page for a new page table?!!)
7874
                                 <1>
7875
                                 <1>
                                           ; We need to terminate process with error message !!!
                                 <1>
7877 0000F05E 8B2D[60030300]
                                                  ebp, [u.usp] ; ebp points to user's registers
                                 <1>
                                           mov
7878 0000F064 8B4D18
                                 <1>
                                                 ecx, [ebp+24] ; byte count
7879
                                 <1>
7880
                                 <1>
                                           ; 20/08/2017
7881
                                 <1>
                                          ;mov eax, [u.r0] ; physical address (of the block)
7882
                                 <1>
7883
                                 <1>
7884
                                 <1>
                                           ; restore Memory Allocation Table Content
7885 0000F067 E82266FFFF
                                 <1>
                                           call deallocate_memory_block
7886
                                 <1>
                                               byte [CRT_MODE], 3 ; 80x25 text mode?
short sysalloc_7 ; yes
7887 0000F06C 803D[225F0000]03
                                 <1>
                                           cmp
7888 0000F073 7407
                                 <1>
7889
                                 <1>
                                          ; Current mode is VGA (or CGA graphics) mode,
7890
                                 <1>
                                           ; We need to return to text mode for displaying
                                 <1>
                                           ; error message just before 'sysexit'.
7892 0000F075 B003
                                 <1>
                                           mov al, 3
                                 call _set_mode <1> sysalloc_7: <1>
7893 0000F077 E8F224FFFF
7895 0000F07C BE[C3100100]
                                           mov esi, beep_Insufficient_Memory ; error message
7896 0000F081 E84773FFFF
                                                print_msg ; print/display the message
                                 <1>
                                           call
                                                 eax, \overline{1}; ax=1 is needed for 'sysexit' procedure
7897 0000F086 B801000000
                                 <1>
                                           mov
7898 0000F08B E959D8FFFF
                                 <1>
                                                 sysexit; and terminate the process!
                                           jmp
7899
                                 <1>
7900
                                 <1> sysdalloc:
                                          ; 19/02/2017 - TRDOS 386 (TRDOS v2.0)
7901
                                 <1>
7902
                                 <1>
                                           ; (TRDOS 386 feature only!)
7903
                                 <1>
7904
                                 <1>
                                           ; Deallocate Memory Block/Pages (for user)
7905
                                 <1>
                                           ; (Complementary call for sysalloc.)
7906
                                 <1>
                                           ; INPUT ->
7907
                                 <1>
                                                 EBX = Virtual address (for user)
7908
                                 <1>
7909
                                 <1>
                                                       (will be rounded up to page border)
7910
                                 <1>
                                                 ECX = Byte Count
                                                      (will be adjusted to page borders)
7911
                                 <1>
7912
                                 <1>
                                                 If ICX = 0
7913
                                 <1>
                                                    nothing to do
7914
                                                  If EBX + ECX > User's ESP
                                 <1>
                                                    nothing to do
7915
                                 <1>
7916
                                 <1>
7917
                                 <1>
                                           ; Note: u.break control may be included in future versions
7918
                                 <1>
7919
                                 <1>
                                           ; OUTPUT ->
```

```
7920
                                  <1>
                                                  If CF = 0
7921
                                  <1>
                                                     EAX = Deallocated memory bytes
                                           ;
                                                     EBX = Virtual address (as rounded up)
7922
                                  <1>
                                           ;
7923
                                  <1>
                                                  IF CF = 1
7924
                                  <1>
                                                     EAX = 0
7925
                                  <1>
                                                        Main purpose of this call is to deallocate/release
7926
                                  <1>
7927
                                  <1>
                                                 previously allocated (physically) contiguous memory
7928
                                  <1>
                                                  pages but beginning (virtual) address may not be
7929
                                                  followed by physically contiguous pages. So, this
                                  <1>
7930
                                  <1>
                                                  system call will deallocate user's virtually
7931
                                  <1>
                                                  contiguous memory pages. Also, there is not any
                                                  objections to use this system call without sysalloc
7932
                                  <1>
7933
                                  <1>
                                                  system call; only possible objection is to lost data
                                                  within user's memory space, if the beginning address
7934
                                  <1>
7935
                                  <1>
                                                  and size is not proper.
7936
                                  <1>
                                           ; Note: Empty page tables will not be deallocated!!!
7937
                                  <1>
7938
                                  <1>
                                                    (they will be deallocated at process termination)
7939
                                  <1>
7940
                                  <1>
                                           ; Note: When the program terminates itself or when it is
7941
                                                  terminated by operating system kernel, all allocated
                                  <1>
7942
                                  <1>
                                                  memory pages will be deallocated during termination
7943
                                  <1>
                                                  stage. So, 'sysdalloc' is not necessary except
7944
                                  <1>
                                                  forgiving memory block to other programs/processes.
                                           ;
7945
                                  <1>
7946 0000F090 8B15[5C030300]
                                 <1>
                                           mov
                                                  edx, [u.sp]
                                                  eax, [edx+12] ; user's stack pointer
7947 0000F096 8B420C
                                 <1>
                                           mov
7948 0000F099 29C8
                                 <1>
                                                  eax, ecx; esp - byte count
7949 0000F09B 24FC
                                 <1>
                                           and
                                                 al, OFCh ; dword alignment
                                 <1>
7950 0000F09D 39D8
                                           cmp
                                                  eax, ebx
7951 0000F09F 7220
                                 <1>
                                           jb
                                                  short sysdalloc err; deallocation overlaps with stack
7952
                                 <1>
7953 0000F0A1 31C0
                                 <1>
                                           xor
                                                  eax, eax
7954 0000F0A3 21C9
                                 <1>
                                           and
                                                  ecx, ecx
7955 0000F0A5 7407
                                 <1>
                                           jz
                                                  short sysdalloc_2
                                 <1>
7957 0000F0A7 E83067FFFF
                                 <1>
                                           call deallocate_user_pages
7958 0000F0AC 7213
                                 <1>
                                                  short sysdalloc_err
                                           jс
                                 <1>
                                 <1> sysdalloc_2:
7960
7961 0000F0AE A3[64030300]
                                                 [u.r0], eax
                                 <1> mov
7962 0000F0B3 8B2D[60030300]
                                 <1>
                                           mov
                                                  ebp, [u.usp]
7963 0000F0B9 895D10
                                 <1>
                                                 [ebp+16], ebx; new value of ebx
7964 0000F0BC E9A1D6FFFF
                                 <1>
                                           jmp
                                                 sysret
7965
                                 <1>
7966
                                 <1> sysdalloc err:
7967 0000F0C1 A3[64030300]
                                       mov [u.r0], eax; 0
                                 <1>
7968 0000F0C6 E977D6FFFF
                                 <1>
                                           jmp
                                                 error
                                 <1>
7970
                                  <1> syscalbac:
7971
                                  <1> ; SYS CALLBACK
                                           ; 03/08/2020
7972
                                 <1>
                                         ; 16/04/2017
7973
                                  <1>
                                         ; 14/04/2017
7974
                                  <1>
                                         ; 13/04/2017
; 28/02/2017
7975
                                  <1>
7976
                                  <1>
                                         ; 26/02/2017
7977
                                  <1>
7978
                                  <1>
                                           ; 24/02/2017
7979
                                  <1>
                                           ; 21/02/2017 - TRDOS 386 (TRDOS v2.0)
7980
                                  <1>
                                          ; (TRDOS 386 feature only!)
7981
                                  <1>
7982
                                  <1>
                                           ; Link or unlink IRQ callback service to/from user (ring 3)
7983
                                  <1>
7984
                                  <1>
                                           ; INPUT ->
7985
                                  <1>
                                                  BL = IRQ number (Hardware interrupt request number)
7986
                                  <1>
                                                       (0 t0 15 but IRQ 0,1,2,6,8,14,15 are prohibited)
7987
                                  <1>
                                                       IRQ numbers 3,4,5,7,9,10,11,12,13 are valid
7988
                                  <1>
                                                       (numbers >15 are invalid)
7989
                                  <1>
7990
                                  <1>
                                                  BH = 0 = Unlink IRQ (in BL) from user (ring 3) service
7991
                                  <1>
                                                      1 = Link IRQ by using Signal Response Byte method
7992
                                                      2 = Link IRQ by using Callback service method
                                  <1>
7993
                                  <1>
                                                      3 = Link IRQ by using Auto Increment S.R.B. method
7994
                                  <1>
                                                      >3 = invalid
7995
                                  <1>
7996
                                  <1>
                                                  CL = Signal Return/Response Byte value
7997
                                  <1>
                                                  If BH = 3, kernel will put a counter value ; 03/08/2020
7998
                                  <1>
7999
                                                           (into the S.R.B. addr)
                                  <1>
                                                           between 0 to 255. (start value = CL+1)
8000
                                  <1>
8001
                                  <1>
8002
                                  <1>
                                                  NOTE: counter value, for example: even and odd numbers
8003
                                  <1>
                                                        may be used for -audio- DMA buffer switch
8004
                                  <1>
                                                        within double buffer method, etc.
8005
                                  <1>
8006
                                  <1>
                                                  EDX = Signal return (Response) byte address
8007
                                  <1>
                                                                        - or -
                                                        Interrupt/Callback service/routine address
8008
                                  <1>
                                  <1>
8009
                                                         (virtual address in user's memory space)
8010
                                  <1>
8011
                                  <1>
                                            ; OUTPUT ->
8012
                                  <1>
                                                  CF = 0 \& EAX = 0 \rightarrow Successful setting
8013
                                  <1>
                                                  CF = 1 \& EAX > 0 \rightarrow IRQ is prohibited or locked
8014
                                  <1>
8015
                                  <1>
                                                               by another process
8016
                                  <1>
                                                          eax = ERR PERM DENIED -> prohibited or locked
                                                          eax = ERR_INV_PARAMETER ->
8017
                                  <1>
8018
                                  <1>
                                                                invalid parameter/option or bad address
8019
                                  <1>
8020
                                  <1>
                                                  NOTE: Timer callbacks are set by using 'systimer'
8021
                                  <1>
                                                        system call (IRQ 0, PIT and IRQ 8, RTC)
8022
                                  <1>
8023
                                  <1>
                                                        Direct keyboard access is performed by using
8024
                                  <1>
                                                        Keyboard Interrupt (INT 32h)
```

```
8026
                                   <1>
                                                          It is prohibited here because:
8027
                                   <1>
                                                          1) Signal Response Byte method has not advantage
8028
                                   <1>
                                                             against INT 32h, function AH = 1. Also,
8029
                                   <1>
                                                             keyboard service interrupt will return with
8030
                                   <1>
                                                             ascii and scan codes (AL, AH) while
                                                             SRB method has only 1 byte space for ascii code
8031
                                   <1>
8032
                                  <1>
                                                             or scan code. One byte signal response is used
8033
                                   <1>
                                                             for ensuring very simple and very fast
8034
                                   <1>
                                                             virtual to physical memory address conversion
8035
                                   <1>
                                                             without any memory page crossover risk.
8036
                                   <1>
                                                             (Otherwise double page conversion or word
8037
                                  <1>
                                                             alignment would be needed.)
8038
                                   <1>
                                                          2) Badly written user code (callback code)
8039
                                   <1>
                                                             can prevent keyboard and timesharing functions
                                                             of the operating system via continuous and long
8040
                                   <1>
                                                             keyboard event handling by callback service.
8041
                                   <1>
8042
                                                              (It can cause to lose immediate keystroke
                                   <1>
                                                             response from hardware to user.)
8043
                                   <1>
8044
                                  <1>
                                                          3) If user will check any keyboard events, 'getkey'
8045
                                   <1>
                                                              (or 'getchar') must have more priority than other
8046
                                                              (video etc.) events because only control ability
                                   <1>
8047
                                  <1>
                                                             on a procedural infinite loop is a keyboard or
8048
                                   <1>
                                                             mouse event. So user can use keyboard function
8049
                                   <1>
                                                             at the end or at the beginning of a loop.
8050
                                   <1>
                                                             In this case, INT 32h is used for that purpose
8051
                                   <1>
                                                             and timer interrupt etc. callbacks can be used
8052
                                  <1>
                                                             for dynamic and synchronized data refresh/transfer
8053
                                   <1>
                                                             while cpu is in a static loop (without polling).
8054
                                                             Keyboard Int callback is not more useful because
                                   <1>
8055
                                   <1>
                                                             already a manual check (a key is pressed or not)
8056
                                                             can be performed (via INT 32h, AH = 1) efficiently
                                   <1>
8057
                                  <1>
                                                             in a loop to prevent a locked infinitive loop.
8058
                                   <1>
8059
                                   <1>
                                                       Disk IRQs (6,14,15) have been phohibited from ring 3
8060
                                   <1>
                                                       callback because, disk operations (file system services
                                                       etc.) are independent from user program, for fast disk r/w.
8061
                                   <1>
8062
                                  <1>
                                                       They are not more useful at ring 3 while they are in use
8063
                                   <1>
                                                       by standard diskio functions which are mandatory part of
8064
                                   <1>
                                                        (monolithic) OS kernel and mainprog command interpreter.
8065
                                   <1>
                                                       INT 33h diskio functions are enough for user level disk
8066
                                   <1>
                                                       r/w.
8067
                                  <1>
                                             ; TRDOS 386 - IRQ CALLBACK structures (parameters):
8068
                                   <1>
8069
                                   <1>
8070
                                   <1>
                                                       [u.irqlock] = 1 word, IRQ flags (0-15) that indicates
                                                                 which IRQs are locked by (that) user.
8071
                                   <1>
                                                                  Lock and unlock (by user) will change
8072
                                  <1>
8073
                                   <1>
                                                                 these flags or 'terminate process' (sysexit)
8074
                                   <1>
                                                                 will clear these flags and unlock those IRQs.
8075
                                   <1>
8076
                                   <1>
                                                                 Bit 0 is for IRQ 0 and Bit 15 is for IRQ 15
8077
                                  <1>
8078
                                   <1>
                                                      IRQ(x).owner
                                                                        : 1 byte, user, [u.uno], 0 = free (unlocked)
8079
                                   <1>
8080
                                   <1>
                                                      IRQ(x).method : 1 byte for callback method & status
8081
                                   <1>
                                                                    0 = Signal Response Byte method
8082
                                  <1>
                                                                    1 = Callback service method
8083
                                   <1>
                                                                    >1 = invalid for current 'syscalback'.
                                                                 or(+) 80h = IRQ is in use by system (ring 0)
8084
                                   <1>
8085
                                   <1>
                                                                             function (audio etc.) or
8086
                                   <1>
                                                                            a device driver.
8087
                                  <1>
                                                                 (system function will ignore the lock/owner)
8088
                                   <1>
8089
                                   <1>
                                                      IRQ(x).srb: 1 byte, Signal Return/Response byte value
8090
                                   <1>
                                                                   (a fixed value by user or a counter value
8091
                                   <1>
                                                                  from 0 to 255, which is increased by every
8092
                                  <1>
                                                                  interrupt just before putting it into
8093
                                   <1>
                                                                  the Signal Response byte address
8094
                                   <1>
                                                                  (This is not used in callback serv method)
8095
                                   <1>
8096
                                   <1>
                                                      IRQ(x).addr
                                                                       : 1 dword
8097
                                  <1>
                                                                   Signal Response Byte address (physical)
8098
                                   <1>
                                                                              -or-
8099
                                   <1>
                                                                   Callback service address (virtual)
8100
                                   <1>
8101
                                   <1>
                                                      IRQ(x).dev: 1 byte
8102
                                   <1>
                                                                   0 = Default device or kernel function
8103
                                   <1>
8104
                                                                   1-255 = Assigned device driver number
                                   <1>
8105
                                   <1>
8106
                                   <1>
                                                       (x) = 3, 4, 5, 7, 9, 10, 11, 12, 13
8107
                                   <1>
8108
                                   <1>
8109
                                   <1>
                                                   NOTE: If user's process/program calls the kernel (INT 40h)
                                                         while it is already running in a (ring 3) callback
8110
                                   <1>
8111
                                   <1>
                                                          service, kernel will force (convert) system call to
8112
                                  <1>
                                                          'sysrele' (sys release). So, this feature provides
8113
                                  <1>
                                                          easy and simple usage of callback services without
8114
                                   <1>
                                                         falling into deepless <please 'callback me' then
                                                         let me 'callback you'> cycles! (User must return
8115
                                   <1>
8116
                                   <1>
                                                          from callback service by using 'sysrele' system
                                                         call, without a significant delay. Otherwise user
8117
                                  <1>
                                                         process/program may be late to catch the next event
8118
                                  <1>
8119
                                   <1>
                                                         within same callback purpose.
8120
                                  <1>
                                   <1>
8122 0000F0CB 30C0
                                                   al, al; the caller is 'syscalbac' sign/flag
                                  <1>
                                            xor
8123 0000F0CD E865180000
                                  <1>
                                            call
                                                   set_irq_callback_service
8124
                                            ; 16/04/2017
                                  <1>
8125 0000F0D2 A3[64030300]
                                  <1>
                                            mov
                                                   [u.r0], eax
8126 0000F0D7 0F8385D6FFFF
                                  <1>
                                            jnc
                                                   sysret
8127 0000F0DD A3[C8030300]
                                  <1>
                                            mov
                                                   dword [u.error], eax
8128 0000F0E2 E95BD6FFFF
                                  <1>
                                   <1>
```

8025

```
8130
                                 <1> sysfpstat:
                                      ; 28/02/2017 - TRDOS 386 (TRDOS v2.0)
8131
8132
                                 <1>
                                          ; (TRDOS 386 feature only!)
8133
                                 <1>
8134
                                 <1>
                                          ; Set or reset FPU registers save/restore option (for user)
8135
                                 <1>
                                                        (during software task switching, wswap-rswap)
8136
                                 <1>
                                         ; INPUT ->
                                 <1>
8137
8138
                                 <1>
                                          ; BL = 0 \rightarrow reset
                                                 BL = 1 -> set (FPU register will be saved and restored)
8139
                                 <1>
                                          ;
8140
                                 <1>
                                          ; OUTPUT ->
8141
                                 <1>
8142
                                 <1>
                                           ; cf = 0 \rightarrow no error, FPU is ready...
8143
                                 <1>
                                                      (EAX = 0)
8144
                                 <1>
                                                Cf = 1 \rightarrow error, 80387 FPU is not ready!
                                           ;
8145
                                 <1>
                                                         (EAX = OFFFFFFFFh)
8146
                                 <1>
8147 0000F0E7 31C0
                                 <1>
                                           xor
                                                eax, eax
8148 0000F0E9 803D[7C650100]00
                                 <1>
                                                 byte [fpready], 0
                                           cmp
8149 0000F0F0 7613
                                 <1>
                                                short sysfpstat_err
                                          jna
8150
                                 <1>
8151 0000F0F2 80E301
                                 <1>
                                          and
                                                bl, 1; use BIT 0 only!
8152 0000F0F5 881D[DA030300]
                                <1>
                                                 [u.fpsave], bl
                                          mov
8153 0000F0FB A3[64030300]
                                <1>
                                          mov
                                                [u.r0], eax ; 0
8154 0000F100 E95DD6FFFF
                                 <1>
                                         jmp sysret
8155
                                 <1>
8156
                                 <1> sysfpstat_err:
8157 0000F105 48
                                          dec eax ; OFFFFFFFh
                                 <1>
8158 0000F106 A3[64030300]
                                 <1>
                                           mov
                                                 [u.r0], eax ; -1
8159 0000F10B E932D6FFFF
                                 <1>
                                           jmp
                                                error
8160
                                 <1>
8161
                                 <1> sysdelete: ; Delete (Remove, Unlink) File
                                         ; 29/12/2017 (TRDOS 386 = TRDOS v2.0)
8162
                                 <1>
8163
                                 <1>
                                           ; INPUT ->
8164
                                 <1>
8165
                                 <1>
                                                       EBX = File name (ASCIIZ string) address
                                           ; OUTPUT ->
8166
                                 <1>
                                                     cf = 0 -> eax = 0
8167
                                 <1>
                                          ;
                                                     cf = 1 -> Error code in AL
8168
                                 <1>
8169
                                 <1>
                                           ;
8170
                                 <1>
                                          ; Modified Registers: EAX (at the return of system call)
8171
                                 <1>
8172
                                 <1>
8173 0000F110 89DE
                                 <1>
                                          mov esi, ebx
8174
                                 <1>
                                          ; file name is forced, change directory as temporary
                                           ;mov ax, 1
8175
                                 <1>
                                          ;mov [FFF Valid], ah ; 0 ; reset
8176
                                 <1>
8177
                                 <1>
                                          ;call set_working_path
8178 0000F112 E8680B0000
                                 <1>
                                           call set_working_path_x
                                           jnc short sysdelete_1
8179 0000F117 731D
                                 <1>
8180
                                 <1>
                                          and eax, eax ; 0 -> Bad Path!
jnz short sysdelete_err
8181 0000F119 21C0
                                 <1>
8182 0000F11B 7505
                                 <1>
8183
                                 <1>
                                           ; eax = 0
8184
                                 <1> sysdelete_path_err:
8185 0000F11D B813000000
                                 <1>
                                        mov eax, ERR_INV_PATH_NAME ; 'bad path name !'
                                 <1> sysdelete err:
8187 0000F122 A3[64030300]
                                          mov [u.r0], eax
                                <1>
8188 0000F127 A3[C8030300]
                                 <1>
                                                 [u.error], eax
                                          call reset_working_path
8189 0000F12C E8230C0000
                                <1>
8190 0000F131 E90CD6FFFF
                                <1>
                                         jmp error
8191
                                 <1> sysdelete 1:
8192
                                <1> ;mov esi, FindFile_Name
8193 0000F136 66B80018
                                 <1>
                                           mov ax, 1800h; Only files
8194 0000F13A E8A891FFFF
                                 <1>
                                           call find_first_file
8195 0000F13F 72E1
                                 <1>
                                           jс
                                                 short sysdelete_err
                                 <1> sysdelete 2:
8196
                                          ; check file attributes
8197
                                 <1>
8198
                                 <1>
8199
                                 <1>
                                          ;test bl, 17 ; system, hidden, readonly, directory
                                            test bl, 7; system, hidden, readonly
8200 0000F141 F6C307
                                 <1>
8201 0000F144 7407
                                             jz short sysdelete 3
                                 <1>
                                 <1>
8202
8203 0000F146 B80B000000
                                <1>
                                             mov eax, ERR_FILE_ACCESS ; 11 = 'permission denied !'
                                             jmp short sysdelete_err
8204 0000F14B EBD5
                                 <1>
8205
                                 <1> sysdelete_3:
8206 0000F14D 6621D2
                                 <1>
                                           and dx, dx; Ambiguous filename chars used sign (DX>0)
8207 0000F150 7407
                                                 short sysdelete 4
                                 <1>
                                           jz
8208 0000F152 B81A000000
                                 <1>
                                                eax, ERR_INV_FILE_NAME ; 26 = 'invalid file name !'
8209 0000F157 EBC9
                                 <1>
                                           jmp short sysdelete_err
8210
                                 <1> sysdelete_4:
8211
                                 <1>
                                           ;mov bh, [LongName_EntryLength]
                                                [DelFile_LNEL], bh ; Long name entry length (if > 0)
8212 0000F159 883D[EE620100]
                                <1>
                                          mov
                                           ; edi = Directory Entry Offset (DirBuff)
8213
                                 <1>
                                           ; esi = Directory Entry (FFF Structure)
8214
                                 <1>
8215 0000F15F E800BBFFFF
                                 <1>
                                           call remove_file
8216 0000F164 72BC
                                 <1>
                                           jc short sysdelete_err
8217
                                <1> sysrmdir 5:
8218 0000F166 31C0
                                 <1>
                                           xor
                                                 eax, eax ; 0
8219 0000F168 A3[64030300]
                                <1>
                                                [u.r0], eax
                                           mov
8220
                                 <1>
                                           ;mov [u.error], eax
                                           call reset_working_path
8221 0000F16D E8E20B0000
                                 <1>
8222 0000F172 E9EBD5FFFF
                                 <1>
                                           jmp
                                                sysret
8223
                                 <1>
8224
                                 <1>
8225
                                 <1> sysrmdir: ; Remove (Unlink) Directory
8226
                                 <1>
                                          ; 29/12/2017 (TRDOS 386 = TRDOS v2.0)
                                 <1>
8227
                                           ; INPUT ->
8228
                                 <1>
8229
                                 <1>
                                                      EBX = Pointer to directory name
                                           ; OUTPUT ->
8230
                                 <1>
                                           ; cf = 0 \rightarrow eax = 0
8231
                                 <1>
                                                     cf = 1 -> Error code in AL
8232
                                 <1>
                                           ;
8233
                                 <1>
8234
                                 <1>
                                          ; Modified Registers: EAX (at the return of system call)
```

```
8235
8237 0000F177 803D[B3030300]00
                                <1>
                                          cmp
                                                byte [u.uno], 0 ; root (super user) ?
8238 0000F17E 7614
                                <1>
                                                short sysrmdir 0
8239
                                <1>
8240
                                <1>
                                          ;mov
                                               dword [u.r0], ERR_PERM_DENIED
8241 0000F180 B80B000000
                                                eax, ERR PERM DENIED ; ERR NOT SUPERUSER
                                <1>
                                         mov
                                          mov
                                                [u.r0], eax
8242 0000F185 A3[64030300]
                                <1>
8243 0000F18A A3[C8030300]
                                <1>
                                                [u.error], eax
                                          mov
                                <1>
8244 0000F18F E9AED5FFFF
                                          jmp
                                               error
8245
                                <1>
8246
                                <1> sysrmdir 0:
8247 0000F194 89DE
                                <1>
                                       mov esi, ebx
8248
                                <1>
                                          ; file name is forced, change directory as temporary
8249
                                <1>
                                         ;mov ax, 1
                                         ;mov [FFF_Valid], ah; 0; reset
8250
                                <1>
                                         ;call set working path
                                <1>
8252 0000F196 E8E40A0000
                                         call set_working_path_x
                                <1>
8253 0000F19B 731D
                                <1>
                                         jnc short sysrmdir_1
8254
                                <1>
                                          and eax, eax ; 0 -> Bad Path!
8255 0000F19D 21C0
                                <1>
8256 0000F19F 7505
                                <1>
                                          jnz short sysrmdir_err
                                <1>
                                          ; eax = 0
8257
                                <1> sysrmdir_not_found:
                                      mov eax, ERR_DIR_NOT_FOUND ; Directory not found !
8259 0000F1A1 B80C000000
                                <1>
8260
                                <1> sysrmdir_err:
8261 0000F1A6 A3[64030300]
                                <1> mov [u.r0], eax
                                <1>
8262 0000F1AB A3[C8030300]
                                          mov
                                               [u.error], eax
8263 0000F1B0 E89F0B0000
                                <1>
                                          call reset_working_path
                                     jmp error
8264 0000F1B5 E988D5FFFF
                               <1>
8265
                                <1> sysrmdir_1:
                                <1> ;mov esi, FindFile_Name
<1> mov ax, 0810h ; Only directories
8266
                               <1>
8267 0000F1BA 66B81008
8268 0000F1BE E82491FFFF
                               <1>
                                      call find_first_file
                                        jnc short sysrmdir 2
8269 0000F1C3 7306
                                <1>
8270
                                <1>
8271
                                <1>
                                         ; eax = 2 (File not found !)
8272 0000F1C5 3C02
                                        cmp al, 2 ; ERR_NOT_FOUND
                                <1>
8273 0000F1C7 74D8
                                <1>
                                                short sysrmdir_not_found
                                          jе
8274 0000F1C9 EBDB
                                <1>
                                          jmp
                                               short sysrmdir_err
8275
                                <1> sysrmdir_2:
                                        ; \stackrel{-}{\text{check}} directory attributes
8276
                                <1>
8277
                                <1>
                                        test bl, 7; system, hidden, readonly
8278 0000F1CB F6C307
                                <1>
                                          jz short sysrmdir 3
8279 0000F1CE 7407
                                <1>
8280
                                <1>
8281 0000F1D0 B80B000000
                                            mov eax, ERR DIR ACCESS ; 11 = 'permission denied !'
                                <1>
8282 0000F1D5 EBCF
                                <1>
                                           jmp short sysrmdir_err
8283
                                <1> sysrmdir_3:
8284 0000F1D7 6621D2
                                <1> and dx, dx; Ambiguous filename chars used sign (DX>0)
8285 0000F1DA 7407
                                <1>
                                          jz short sysrmdir_4
                                          ;mov eax, ERR_NOT_DIR ; 'not a valid directory !'
mov eax, ERR_INV_PATH_NAME ; 'bad path name !'
                                <1>
8287 0000F1DC B813000000
                                <1>
                                <1>
8288 0000F1E1 EBC3
                                         jmp short sysrmdir_err
8289
                                <1> sysrmdir_4:
8290
                                <1>
                                         ;mov bh, [LongName_EntryLength]
                                         mov [DelFile LNEL], bh ; Long name entry length (if > 0)
8291 0000F1E3 883D[EE620100]
                                <1>
                                      ; edi = Directory Entry Offset (DirBuff)
8292
                                <1>
8293
                                <1>
                                          ; esi = Directory Entry (FFF Structure)
8294 0000F1E9 E8CE97FFFF
                                         call delete_sub_directory
                                <1>
8295 0000F1EE 0F8372FFFFFF
                                <1>
                                          jnc sysrmdir_5
                                 <1>;
                                         jс
                                                short sysrmdir 6
                                <1>;
8297
8298
                                 <1>;
                                                eax, eax ; 0
8299
                                 <1> ;sysrmdir_5:
8300
                                 <1> ;
                                       mov
                                                 [u.r0], eax
8301
                                <1>;
                                          ; mov
                                               [u.error], eax
                                <1>;
                                          call reset_working_path
8302
8303
                                <1>;
                                          jmp
                                                sysret
8304
                                <1> sysrmdir_6:
8305 0000F1F4 A3[64030300]
                                <1>
                                                 [u.r0], eax
8306 0000F1F9 A3[C8030300]
                                <1>
                                          mov
                                                [u.error], eax
                                <1>
8307
8308 0000F1FE 09C0
                                <1>
                                                eax, eax ; EAX = 0 -> Directory not empty!
8309 0000F200 741C
                                <1>
                                                short sysrmdir 9
                                        jz
8310
                                <1>
8311
                                 <1>
                                          ; EAX > 0 -> Error code in AL (or AX or EAX)
8312
                                <1>
                                                dword [FAT ClusterCounter], 1
8313 0000F202 833D[A2600100]01
                                 <1>
8314 0000F209 7209
                                <1>
                                          jЬ
                                                short sysrmdir_8
                                 <1> sysrmdir_7:
8315
8316
                                 <1>
                                          ; ESI = Logical DOS Drive Description Table address
8317 0000F20B 66BB00FF
                                                 bx, OFFOOh ; BH = FFh -> use ESI for Drive parameters
                                <1>
                                          mov
                                <1>
                                                    ; BL = 0 -> Recalculate free cluster count
8319 0000F20F E834D0FFFF
                                <1>
                                          call calculate_fat_freespace
8320
                                 <1> sysrmdir 8:
8321 0000F214 E83B0B0000
                                <1>
                                          call reset_working_path
8322 0000F219 E924D5FFFF
                                <1>
                                          jmp
                                                error
8323
                                <1>
8324
                                <1> sysrmdir 9:
8325 0000F21E A1[A2600100]
                                <1>
                                          mov eax, [FAT_ClusterCounter]
8326 0000F223 09C0
                                <1>
                                          or
                                                eax, eax ; 0 ?
8327 0000F225 0F847BFFFFFF
                                <1>
                                          jz
                                                sysrmdir_err
                                <1>
                                          ; ESI = Logical DOS Drive Description Table address
                                          mov bx, 0FF01h; BH = FFh -> use ESI for Drive parameters
8329 0000F22B 66BB01FF
                                <1>
8330
                                <1>
                                                    ; BL = 1 -> add free clusters
8331 0000F22F E814D0FFFF
                                <1>
                                          call calculate_fat_freespace
8332 0000F234 09C9
                                          or ecx, ecx
                                <1>
8333 0000F236 74DC
                                <1>
                                          jz short sysrmdir_8 ; ecx = 0 -> OK
                                <1>
                                          ; ecx > 0 -> Error (Recalculation is needed)
8334
8335 0000F238 EBD1
                                <1>
                                          jmp short sysrmdir_7
8336
                                 <1>
8337
                                <1>
8338
                                 <1> syschdir: ; Change Current (Working) Drive & Directory (for user)
8339
                                          ; 30/12/2017 (TRDOS 386 = TRDOS v2.0)
                                 <1>
```

```
8340
                                  <1>
8341
                                  <1>
                                             ; INPUT ->
8342
                                                        EBX = Directory name (ASCIIZ string) address
                                  <1>
                                            ; OUTPUT ->
8343
                                  <1>
8344
                                  <1>
                                                      cf = 0 -> eax = 0
                                           ;
8345
                                  <1>
                                                      cf = 1 -> Error code in AL
8346
                                  <1>
                                  <1>
                                           ; Modified Registers: EAX (at the return of system call)
8347
8348
                                  <1>
8349
                                  <1>
                                           ; NOTE: If drive name is not included, only the working
8350
                                  <1>
                                           ; directory (for user, not for drive/OS) will be chanded.
8351
                                  <1>
                                           ; If there is a drive name (as A:, B:, C:, D: etc.)
                                           ; at the beginning of the ASCIIZ (directory) string,
8352
                                  <1>
8353
                                  <1>
                                           ; working drive and working directory (for user)
8354
                                  <1>
                                           ; will be changed together.
                                           ; (When the program is terminated, MainProg -internal
8355
                                  <1>
8356
                                  <1>
                                           ; shell- will reset working directory to the previous
                                           ; -current- logical drive's current directory again.)
8357
                                  <1>
8358
                                  <1>
8359 0000F23A 89DE
                                  <1>
                                                 esi, ebx
                                           mov
8360
                                  <1>
                                           ; file name is not forced, change directory as temporary
8361 0000F23C 31C0
                                  <1>
                                           xor eax, eax
                                           ;mov [FFF_Valid], ah; 0; reset
8362
                                  <1>
8363
                                  <1>
                                           ;call set_working_path
8364 0000F23E E8400A0000
                                 <1>
                                           call set_working_path_xx
8365 0000F243 731D
                                 <1>
                                           jnc
                                                 short syschdir_ok
                                           and eax, eax ; 0 -> Bad Path!
8366 0000F245 21C0
                                 <1>
8367 0000F247 7505
                                 <1>
                                           jnz short syschdir_err
8368
                                  <1>
                                           ; eax = 0
8369
                                 <1> syschdir_not_found:
8370 0000F249 B80C000000
                                  <1>
                                           mov eax, ERR_DIR_NOT_FOUND ; Directory not found !
8371
                                  <1> syschdir err:
8372 0000F24E A3[64030300]
                                 <1>
                                         mov [u.r0], eax
8373 0000F253 A3[C8030300]
                                 <1>
                                                 [u.error], eax
8374 0000F258 E8F70A0000
                                  <1>
                                           call reset_working_path
8375 0000F25D E9E0D4FFFF
                                 <1>
                                           jmp
                                                 error
                                  <1> syschdir ok:
8377 0000F262 31C0
                                 <1>
                                           xor eax, eax; 0
8378 0000F264 A3[64030300]
                                  <1>
                                                  [u.r0], eax
                                           ;mov [u.error], eax
8379
                                  <1>
8380 0000F269 E9F4D4FFFF
                                  <1>
                                                 sysret
                                           jmp
8381
                                  <1>
8382
                                  <1>
8383
                                  <1> syschmod: ; Get & Change File (or Directory) Attributes
8384
                                  <1>
                                           ; 30/12/2017 (TRDOS 386 = TRDOS v2.0)
8385
                                  <1>
8386
                                  <1>
                                             ; INPUT ->
                                                       EBX = File/Directory (ASCIIZ) name address
8387
                                  <1>
8388
                                  <1>
                                                      CL = New attributes (if <math>CL < 40h)
8389
                                  <1>
                                                      CL >= 40h -> Get File Attributes
                                            ; OUTPUT ->
8390
                                  <1>
8391
                                  <1>
                                                      cf = 0 -> EAX = File attributes (in AL)
                                                      cf = 1 -> Error code in AL
8392
                                  <1>
8393
                                  <1>
8394
                                  <1>
                                           ; Modified Registers: EAX (at the return of system call)
8395
                                  <1>
8396
                                  <1>
                                           ; MSDOS File Attributes: (bit value of attrib byte)
8397
                                  <1>
                                                 ATTR_READ_ONLY
                                                                     = 01h (bit 0, 'R')
8398
                                  <1>
                                                  ATTR_HIDDEN =
                                                                     02h (bit 1, 'H')
                                                                     04h (bit 2, 'S')
8399
                                                  ATTR SYSTEM =
                                  <1>
                                                                     = 08h (bit 3)
= 10h (bit 4)
8400
                                  <1>
                                                  ATTR VOLUME ID
8401
                                  <1>
                                                  ATTR DIRECTORY
                                                                            10h (bit 4)
                                                                      20h (bit 5, 'A')
8402
                                  <1>
                                                  ATTR ARCHIVE =
8403
                                  <1>
                                                  ATTR_LONG_NAME
                                                                      =
                                                                            ATTR_READONLY |
8404
                                  <1>
                                                                      ATTR HIDDEN
8405
                                  <1>
                                                                      ATTR SYSTEM
8406
                                  <1>
                                                                      ATTR VOLUME ID
8407
                                  <1>
                                                  The upper two bits of attributes must be 0.
8408
                                  <1>
8409
                                  <1>
                                                        * If ATTR_DIRECTORY is set, only directory names
                                           ; Note:
8410
                                  <1>
                                                    will be searched (and S,H,R,A attributeds of
                                                    the directory will be changed.)
8411
                                  <1>
                                                  * If ATTR_VOLUME_ID is set, 'syschmod' system call
8412
                                  <1>
8413
                                  <1>
                                                    will return with 'permission denied' error.
8414
                                  <1>
                                                  * If ATTR DIRECTORY is not set, only file names
                                                    will be searched (and S, H, R, A attributes of the
8415
                                  <1>
8416
                                  <1>
                                                    file will be changed.)
                                           ;
8417
                                  <1>
8418
                                  <1>
                                           ; (Ony Super User can change S,H,R attributes.)
8419
                                  <1>
8420 0000F26E 80F940
                                                 cl, 40h
                                  <1>
                                            cmp
8421 0000F271 7327
                                  <1>
                                                  short syschmod 0
                                            jnb
                                  <1>
8423 0000F273 F6C108
                                  <1>
                                            test cl, 08h; ATTR_VOLUME_ID
8424 0000F276 750E
                                  <1>
                                                 short syschmod_perm_err
8425
                                  <1>
8426 0000F278 803D[B3030300]00
                                 <1>
                                                  byte [u.uno], 0 ; root (super user) ?
8427 0000F27F 7619
                                 <1>
                                           jna
                                                 short syschmod_0
8428
                                  <1>
                                           ; Not super user..
test cl, 07h ; S,H,R attributes
8429
                                 <1>
8430 0000F281 F6C107
                                 <1>
8431 0000F284 7414
                                  <1>
                                           jz
                                                 short syschmod 0
                                 <1>
8432
                                 <1> syschmod perm err:
8433
                                           ;mov dword [u.r0], ERR PERM DENIED
8434
                                 <1>
8435 0000F286 B80B000000
                                 <1>
                                           mov
                                                  eax, ERR_PERM_DENIED ; 'permission denied !'
8436 0000F28B A3[64030300]
                                 <1>
                                                  [u.r0], eax
                                           mov
8437 0000F290 A3[C8030300]
                                 <1>
                                           mov
                                                 [u.error], eax
8438 0000F295 E9A8D4FFFF
                                  <1>
                                           jmp
                                                  error
8439
                                  <1>
8440
                                  <1> syschmod_0:
8441 0000F29A 880D[3C630100]
                                 <1>
                                           mov
                                                 [Attributes], cl
8442 0000F2A0 89DE
                                  <1>
                                           mov
                                                 esi, ebx
8443
                                  <1>
                                           ; file name is forced, change directory as temporary
8444
                                  <1>
                                           ;mov ax, 1
```

```
;mov [FFF_Valid], ah; 0; reset
;call set_working_path
call set_working_path_x
8445
                                 <1>
8446
                                 <1>
8447 0000F2A2 E8D8090000
                                <1>
8448 0000F2A7 731D
                                <1>
                                          jnc short syschmod_1
                                          and eax, eax ; 0 -> Bad Path! jnz short syschmod_err
8449 0000F2A9 21C0
                                <1>
8450 0000F2AB 7505
                                <1>
                                          ; eax = 0
                                <1>
                                <1> syschmod_path_not_found:
8452
8453 0000F2AD B813000000
                                <1>
                                          mov eax, ERR_INV_PATH_NAME ; 'Bad path name !'
8454
                                <1> syschmod err:
8455 0000F2B2 A3[64030300]
                                <1>
                                          mov [u.r0], eax
8456 0000F2B7 A3[C8030300]
                                <1>
                                          mov
                                                 [u.error], eax
                                          call reset_working_path
8457 0000F2BC E8930A0000
                                <1>
8458 0000F2C1 E97CD4FFFF
                                <1>
                                          jmp error
                                <1> syschmod 1:
8459
8460 0000F2C6 B008
                                <1>
                                          mov al, 08h; Except volume labels (& long names)
8461 0000F2C8 A0[3C630100]
                                <1>
                                          mov al, [Attributes]
                                          and al, 10h ;
8462 0000F2CD 2410
                                <1>
8463
                                <1>
                                          ;mov esi, FindFile_Name
                                          ;mov ax, 1800h; Only files
8464
                                <1>
                                      ;mov ax, 0810h ; Only directories
8465
                                <1>
8466 0000F2CF E81390FFFF
                                <1>
                                          call find_first_file
                                          ;jnc short syschmod_2
8467
                                 <1>
8468 0000F2D4 72DC
                                 <1>
                                          jс
                                                 short syschmod_err
8469
                                 <1>
8470
                                 <1>
                                          ;; eax = 2 (File not found !)
8471
                                 <1>
                                          ;cmp al, 2 ; ERR NOT FOUND
8472
                                 <1>
                                          ;jne short syschmod_err
8473
                                 <1>
8474
                                 <1>
                                          ;and byte [Attributes], 10h
8475
                                 <1>
                                          ;jz
                                                short syschmod_err
8476
                                 <1>
8477
                                 <1>
                                          ;; Directory not found !
8478
                                 <1>
                                          ;mov al, 3 ; ERR PATH NOT FOUND
8479
                                 <1>
                                           ;jmp short syschmod err
8480
                                 <1>
8481
                                 <1> syschmod 2:
8482 0000F2D6 6621D2
                                           and dx, dx; Ambiguous filename chars used sign (DX>0)
                                <1>
                                                 short syschmod 3
8483 0000F2D9 7407
                                 <1>
                                           jΖ
                                           mov eax, ERR INV FILE NAME ; 'invalid file name !'
8484 0000F2DB B81A000000
                                <1>
8485 0000F2E0 EBD0
                                <1>
                                          jmp short syschmod_err
8486
                                 <1> syschmod 3:
                                         ; \overline{\text{EDI}} = Directory buffer entry offset/address
8487
                                 <1>
8488
                                 <1>
                                          ; BL = File (or Directory) Attributes
                                          ; mov bl, [EDI+0Bh]
8489
                                 <1>
8490
                                 <1>
8491
                                 <1>
                                          ; check directory attributes
8492 0000F2E2 8A3D[3C630100]
                                          mov bh, [Attributes] ; new attributes
                                 <1>
                                                 bh, 40h ;>=40 -> get file/directory attributes
8493 0000F2E8 80FF40
                                 <1>
8494 0000F2EB 732D
                                 <1>
                                               short syschmod_6
                                          jnb
8495
                                 <1>
8496
                                 <1>
                                          ; set file/directory attributes
8497 0000F2ED F6C307
                                          test bl, 7; system, hidden, readonly
                                <1>
8498 0000F2F0 7409
                                <1>
                                          jz short syschmod_4
8499
                                 <1>
8500 0000F2F2 803D[B3030300]00 <1>
                                           cmp
                                                 byte [u.uno], 0 ; root (super user) ?
8501 0000F2F9 778B
                                 <1>
                                                 short syschmod_perm_err
                                           jа
                                 <1> syschmod_4:
8502
8503 0000F2FB 66817F0CA101
                                 <1>
                                                 word [edi+DirEntry_NTRes], 01A1h ; Singlix FS
                                           cmp
8504 0000F301 7424
                                <1>
                                                 short syschmod 7
                                           jе
8505
                                 <1>
8506 0000F303 887F0B
                                 <1>
                                                [edi+0Bh], bh ; Attributes (New!)
                                          mov
8507
                                 <1>
8508 0000F306 C605[AC600100]02
                                 <1>
                                                 byte [DirBuff_ValidData], 2 ; modified sign
                                 <1>
                                                                        ; to force write
8509
8510 0000F30D E80AB6FFFF
                                 <1>
                                           call save_directory_buffer
8511 0000F312 729E
                                 <1>
                                           jс
                                                 short syschmod_err
8512
                                 <1>
8513
                                 <1> syschmod_5:
8514 0000F314 8A1D[3C630100]
                                <1> mov bl, [Attributes]
                                 <1> syschmod 6:
8515
8516 0000F31A 0FB6C3
                                 <1>
                                       movzx eax, bl
8517 0000F31D A3[64030300]
                                 <1>
                                          mov [u.r0], eax
8518
                                 <1>
                                          ;mov dword [u.error], 0
                                         jmp sysret
8519 0000F322 E93BD4FFFF
                                 <1>
8520
                                 <1>
8521
                                <1> syschmod_7:
                                <1> sub eax, eax
8522 0000F327 29C0
                                                 ah, [DirBuff DRV]
8523 0000F329 8A25[AA600100]
                                 <1>
                                           mov
8524 0000F32F BE00010900
                                           mov esi, Logical DOSDisks
                                <1>
                                        add
8525 0000F334 01C6
                                 <1>
                                                    esi, eax
8526 0000F336 807E04A1
                                 <1>
                                             cmp
                                                    byte [esi+LD_FSType], 0A1h
8527 0000F33A 7307
                                                short syschmod_8
                                <1>
                                           jnc
8528 0000F33C B01D
                                 <1>
                                                al, ERR_INV_DATA ; 29 = Invalid Data
8529 0000F33E E96FFFFFF
                                 <1>
                                           jmp
                                                syschmod err
8530
                                 <1>
                                 <1> syschmod 8:
                                          ; BH = New MS-DOS File Attributes
8532
                                <1>
8533 0000F343 88F8
                                <1>
                                                al, bh ; File/Directory Attributes
                                           mov
8534 0000F345 30E4
                                <1>
                                                ah, ah ; Attributes in MS-DOS format sign
                                          xor
8535 0000F347 E8F9A0FFFF
                                <1>
                                           call change_fs_file_attributes
8536 0000F34C 0F8260FFFFFF
                                 <1>
                                           jс
                                                 syschmod err
8537 0000F352 EBC0
                                 <1>
                                           jmp
                                                short syschmod_5
8538
                                 <1>
8539
                                 <1>
8540
                                 <1> sysdrive: ; Get/Set Current (Working) Drive (for user)
8541
                                 <1>
                                          ; 30/12/2017 (TRDOS 386 = TRDOS v2.0)
8542
                                 <1>
8543
                                 <1>
                                            ; INPUT ->
                                                      BL = Logical DOS Drive number (0=A: ... 2=C:)
8544
                                 <1>
                                                   If BL = OFFh -> Get Current Drive
8545
                                 <1>
8546
                                 <1>
                                          ; OUTPUT ->
                                                    cf = 0 ->
8547
                                 <1>
                                           ;
                                                         AL = Current Drive number
8548
                                 <1>
8549
                                 <1>
                                                          AH = The Last Logical DOS Drive no.
```

```
8550
                                 <1>
                                                      cf = 1 -> Error code in AL
8551
                                  <1>
                                           ; Modified Registers: EAX (at the return of system call)
8552
                                 <1>
8553
                                 <1>
8554
                                 <1>
                                           ; NOTE: If the requested logical dos drive is ready,
8555
                                 <1>
                                                 it's current current directory will be the user's
                                                  (program's) current directory.
8556
                                 <1>
                                                 (When the program is terminated, MainProg -internal
                                 <1>
8557
                                           ;
8558
                                 <1>
                                                 shell- will reset the previous -current- logical drive
                                           ;
8559
                                 <1>
                                                  as current drive again).
                                           ;
8560
                                 <1>
8561 0000F354 80FBFF
                                 <1>
                                           cmp
                                                 bl, OFFh
8562 0000F357 7435
                                 <1>
                                                  short sysdrive_ok
                                           jе
8563 0000F359 3A1D[610D0100]
                                                 bl, [Last DOS DiskNo]
                                 <1>
                                                 short sysdrive_err
8564 0000F35F 771E
                                 <1>
                                           jа
8565
                                 <1>
                                           ; Save current drive and reset mode
8566
                                 <1>
                                           ; for 'reset_working_path' procedure (for MainProg)
8567
                                 <1>
8568 0000F361 30C0
                                 <1>
                                           xor al, al
                                           mov [SWP Mode], ax; ah = 0
8569 0000F363 66A3[78650100]
                                 <1>
8570 0000F369 A0[86590100]
                                 <1>
                                           mov al, [Current_Drv]
8571 0000F36E FEC4
                                 <1>
                                           inc
                                                 ah ; mov ah, 1
8572 0000F370 66A3[7A650100]
                                                 [SWP_DRV], ax
                                 <1>
                                           mov
8573
                                 <1>
8574 0000F376 88DA
                                 <1>
                                                 dl, bl
                                           mov
8575 0000F378 E8C77BFFFF
                                 <1>
                                           call
                                                change_current_drive
8576 0000F37D 730F
                                 <1>
                                           jnc short sysdrive_ok
8577
                                 <1> sysdrive_err:
8578 0000F37F C705[64030300]0F00- <1>
                                           mov dword [u.r0], ERR_DRV_NOT_RDY; 'drive not ready!'
8578 0000F387 0000
                                <1>
8579 0000F389 E9B4D3FFFF
                                 <1>
                                           jmp
                                                  error
8580
                                 <1> sysdrive ok:
8581 0000F38E A0[86590100]
                                 <1>
                                           mov al, [Current_Drv]
8582 0000F393 8A25[610D0100]
                                <1>
                                                 ah, [Last_DOS_DiskNo]
                                           mov
                                                 [u.r0], eax
8583 0000F399 A3[64030300]
                                 <1>
                                           mov
8584 0000F39E E9BFD3FFFF
                                 <1>
                                           jmp
                                                 sysret
                                 <1>
8586
                                 <1>
                                 <1> sysdir: ; Get Current (Working) Drive & Directory (for user)
8587
8588
                                         ; 30/12/2017 (TRDOS 386 = TRDOS v2.0)
                                 <1>
8589
                                 <1>
8590
                                  <1>
                                             ; INPUT ->
                                                       EBX = Current directory name buffer address
8591
                                 <1>
8592
                                 <1>
                                                        (Buffer length = 92 bytes)
8593
                                 <1>
                                           ; OUTPUT ->
                                                     AL = Current drive (0=A: ... 2=C:)
8594
                                 <1>
                                                    If CF = 1 -> AL = error code
8595
                                 <1>
8596
                                 <1>
8597
                                 <1>
                                           ; Modified Registers: EAX (at the return of system call)
8598
                                 <1>
8599
                                 <1>
                                           ; Note: Required directory name buffer length may be
8600
                                 <1>
                                                 <= 92 bytes for current TRDOS 386 version.
                                           ;
                                                  (7*12 \text{ name chars} + 7 \text{ slash} + 0)
8601
                                 <1>
8602
                                 <1>
8603 0000F3A3 89E5
                                 <1>
                                                ebp, esp
                                           mov
8604 0000F3A5 83EC60
                                 <1>
                                           sub
                                                 esp, 96
                                           push ebx; User's buffer address
8605 0000F3A8 53
                                 <1>
8606 0000F3A9 30D2
                                           xor dl, dl; 0 = current drive
                                 <1>
                                          call get_current_directory
jc short sysdrive_err; 'drive not ready!' error
8607 0000F3AB E890AAFFFF
                                 <1>
8608 0000F3B0 72CD
                                 <1>
8609 0000F3B2 89E6
                                         mov esi, esp ; System's buffer address
                                 <1>
                                        pop edi ; User's buffer address
8610 0000F3B4 5F
                                 <1>
8611
                                 <1>
                                           ; ecx = transfer (byte) count (<=92)
8612 0000F3B5 E843F4FFFF
                                 <1>
                                           call transfer_to_user_buffer
                                           mov esp, ebp
8613 0000F3BA 89EC
                                 <1>
8614 0000F3BC 730F
                                 <1>
                                           jnc
                                                 short sysdir_ok
                                 <1> sysdir_err:
8616 0000F3BE C705[64030300]2E00- <1>
                                                 dword [u.r0], ERR_BUFFER ; 'buffer error !'
                                           mov
8616 0000F3C6 0000
                                 <1>
8617 0000F3C8 E975D3FFFF
                                 <1>
                                           jmp
                                                  error
                                 <1> sysdir_ok:
8619 0000F3CD 8A0D[86590100]
                                 <1>
                                           mov
                                                  cl, [Current Drv]
8620 0000F3D3 890D[64030300]
                                 <1>
                                                  [u.r0], ecx
                                           mov
8621 0000F3D9 E984D3FFFF
                                 <1>
                                           jmp
                                                 sysret
8622
                                 <1>
8623
                                 <1>
8624
                                 <1> sysldrvt: ; Get copy of Logical DOS Drive Description Table
                                           ; 30/12/2017 (TRDOS 386 = TRDOS v2.0)
8625
                                 <1>
8626
                                 <1>
8627
                                             ; INPUT ->
                                 <1>
                                                      BL = Logical DOS drive number (zero based)
8628
                                  <1>
                                                        ECX = Logical DOS drv desc table buffer addr
8629
                                  <1>
8630
                                  <1>
                                                         (Buffer length = 256 bytes)
8631
                                  <1>
                                           ; OUTPUT ->
8632
                                  <1>
                                                      cf = 0 \rightarrow
                                                           AL = Current Drive number
8633
                                  <1>
8634
                                  <1>
                                                          AH = The Last Logical DOS Drive no.
8635
                                 <1>
                                                      cf = 1 -> Error code in AL
8636
                                 <1>
                                                           AH = The Last Logical DOS Drive no.
8637
                                 <1>
8638
                                 <1>
                                           ; Modified Registers: EAX (at the return of system call)
8639
                                  <1>
                                 <1>
                                           ; Note: Required description table buffer length is
8640
8641
                                 <1>
                                                  256 bytes for current TRDOS 386 version.
8642
                                 <1>
8643 0000F3DE 89CF
                                 <1>
                                           mov
                                                  edi, ecx ; Destination address (user space)
8644 0000F3E0 88DC
                                 <1>
                                                 ah, bl
                                           mov
8645 0000F3E2 30C0
                                 <1>
                                           xor
                                                 al, al
                                                  esi, Logical DOSDisks
8646 0000F3E4 BE00010900
                                 <1>
                                           mov
8647 0000F3E9 01C6
                                           add
                                 <1>
                                                 esi, eax ; Source address (system space)
8648 0000F3EB B900010000
                                 <1>
                                                  ecx, 256; Byte count
                                           mov
8649
                                 <1>
                                                         ; Logical Dos Drv Desc Table size
8650 0000F3F0 E808F4FFFF
                                 <1>
                                           call transfer to user buffer
                                                  short sysdir err
8651 0000F3F5 72C7
                                 <1>
                                           jс
8652 0000F3F7 8A2D[610D0100]
                                 <1>
                                                 ch, [Last DOS DiskNo]
                                           mov
```

```
8653 0000F3FD EBCE
                                 <1>
                                           jmp short sysdir ok
8654
                                 <1>
8655
                                 <1>
8656
                                 <1> systime: ; Get System Date&Time
8657
                                           ; 30/12/2017 (TRDOS 386 = TRDOS v2.0)
                                 <1>
8658
                                 <1>
                                             ; INPUT -> BL =
8659
                                 <1>
                                                     0 = Get Date&Time in Unix/Epoch format
8660
                                 <1>
8661
                                 <1>
                                                     1 = Get Time in MSDOS format
                                                     2 = Get Date in MSDOS format
8662
                                 <1>
8663
                                 <1>
                                                     3 = Get Date&Time in MSDOS format
8664
                                 <1>
                                                     4 & other values =
                                                       System timer ticks will be returned
8665
                                 <1>
                                                        in EAX and Carry Flag will be set.
8666
                                 <1>
8667
                                 <1>
                                                        (CF will not be set if BL = 4)
                                           ; OUTPUT ->
8668
                                 <1>
8669
                                 <1>
                                                For BL input = 3
8670
                                                   EAX = Current Time (RTC)
                                 <1>
8671
                                 <1>
                                                       AL = Second (DL in MSDOS)
8672
                                 <1>
                                                      AH = Minute (CL in MSDOS)
                                                      HW of EAX = Hour (CH in MSDOS)
8673
                                 <1>
                                                    EDX = Current System Date (RTC)
8674
                                 <1>
8675
                                                      DL = Day (DL in MSDOS)
                                 <1>
8676
                                 <1>
                                                        DH = Month (DH in MSDOS)
8677
                                 <1>
                                                        HW of EDX = Year (CX in MSDOS)
8678
                                 <1>
8679
                                 <1>
                                                 For BL input = 2
8680
                                                  EAX = Current System Date (RTC)
                                 <1>
8681
                                 <1>
                                                        DL = Day (DL in MSDOS)
                                                        DH = Month (DH in MSDOS)
8682
                                 <1>
8683
                                 <1>
                                                        HW 	ext{ of } EDX = Year (CX in MSDOS)
8684
                                 <1>
                                                 For BL input = 1
8685
                                 <1>
8686
                                 <1>
                                                   EAX = Current Time (RTC)
8687
                                 <1>
                                                       AL = Second (DL in MSDOS)
8688
                                 <1>
                                                        AH = Minute (CL in MSDOS)
                                                       HW of EAX = Hour (CH in MSDOS)
8689
                                 <1>
8690
                                 <1>
                                                 For BL input = 0
8691
                                 <1>
                                                    EAX = Unix (Epoch) Time Ticks/Seconds
8692
                                 <1>
8693
                                 <1>
8694
                                 <1>
                                                 For BL input = 4
8695
                                 <1>
                                                    EAX = System timer ticks
8696
                                 <1>
8697
                                 <1>
                                                 If CF = 1 (for other values of BL input)
8698
                                 <1>
                                                    EAX = System timer ticks (no error code!)
8699
                                 <1>
                                           ; Modified Registers: EAX, (EDX)
8700
                                 <1>
8701
                                 <1>
                                                         (at the return of system call)
8702
                                 <1>
8703
                                 <1>
8704 0000F3FF 20DB
                                 <1>
                                           and
                                                 bl, bl
8705 0000F401 750F
                                 <1>
                                           jnz
                                                 short systime_1
                                           call epoch
8706 0000F403 E82C71FFFF
                                 <1>
                                 <1> systime_0:
8708 0000F408 A3[64030300]
                                 <1>
                                                  [u.r0], eax
                                           mov
8709 0000F40D E950D3FFFF
                                                 sysret
                                 <1>
                                           gmp
                                 <1> systime_1:
8710
8711 0000F412 80FB04
                                 <1>
                                           cmp
8712 0000F415 7211
                                                 short systime 2
                                 <1>
                                           jb
8713 0000F417 A1[40590100]
                                 <1>
                                                  eax, [TIMER_LH] ; 18.2 Hz timer ticks
8714
                                 <1>
                                                              ; Note: [TIMER LH] may be set
8715
                                 <1>
                                                              ; to wrong timer value due to
                                                              ; program functions.
8716
                                 <1>
8717
                                 <1>
                                                              ; (This value must not be
8718
                                 <1>
                                                              ; accepted as [TIMER_LH]/18.2
                                 <1>
                                                              ; seconds since the midnight.)
8720 0000F41C 76EA
                                                 short systime_0
                                 <1>
                                           jna
8721 0000F41E A3[64030300]
                                 <1>
                                                  [u.r0], eax
8722 0000F423 E91AD3FFFF
                                 <1>
                                                 error; cf = 1 & [u.r0] = eax = timer ticks
                                           jmp
8723
                                 <1>
8724
                                 <1> systime 2:
8725
                                 <1>
                                          ;push ebx
8726 0000F428 E86970FFFF
                                 <1>
                                           call get_rtc_date_time
8727
                                 <1>
                                          ;pop ebx
8728 0000F42D F6C301
                                 <1>
                                           test bl, 1
8729 0000F430 7429
                                 <1>
                                           jz
                                                 short systime 4
8730 0000F432 30E4
                                 <1>
                                           xor
                                                 ah, ah
8731 0000F434 A0[B4550100]
                                 <1>
                                           mov
                                                 al, [hour]
8732 0000F439 88C2
                                 <1>
                                                 dl, al
                                          mov
8733 0000F43B C1E010
                                 <1>
                                           shl
                                                 eax, 16
8734 0000F43E A0[B8550100]
                                 <1>
                                           mov
                                                 al, [second]
                                          mov ah, [minute]
8735 0000F443 8A25[B6550100]
                                <1>
8736 0000F449 F6C302
                                 <1>
                                           test bl, 2
                                        jz short systime_0
8737 0000F44C 74BA
                                 <1>
8738
                                        ; Check time & date match risk
; (23:59:59 may cause to wrong
                                 <1>
8739
                                <1>
                                         ; date -new day with previous date-...)
8740
                                <1>
                                          cmp dl, 23
8741 0000F44E 80FA17
                                <1>
8742 0000F451 7206
                                <1>
                                                 short systime 3
                                          jb
8743 0000F453 663D3B3B
                                           cmp ax, (59*256)+59; if hour is 23:59:59
                                <1>
8744 0000F457 73CF
                                 <1>
                                                 short systime_2 ; wait for 1 second
                                           jnb
                                 <1> systime_3:
8745
                                        ; eax = time
8746
                                 <1>
8747 0000F459 89C6
                                 <1>
                                           mov esi, eax
8748
                                 <1> systime_4:
8749 0000F45B 66A1[AE550100]
                                <1>
                                                 ax, [year]
                                          mov
8750 0000F461 C1E010
                                           shl eax, 16
                                <1>
8751 0000F464 A0[B2550100]
                                 <1>
                                           mov
                                                 al, [day]
                                         mov ah, [month]
8752 0000F469 8A25[B0550100]
                                <1>
8753
                                <1>
                                       ; eax = date
                                        and bl, 1
jz short systime_0
8754 0000F46F 80E301
                                 <1>
8755 0000F472 7494
                                <1>
8756 0000F474 96
                                 <1> xchg esi, eax
                                 <1>
                                      ; eax = time, esi = date
```

```
8758 0000F475 8B2D[60030300]
                                          mov ebp, [u.usp] ; EBP points to user's registers
                                <1>
8759
                                 <1>
                                          ; (user) edx <-- (system) esi
8760 0000F47B 897514
                                          mov [ebp+20], esi; return to user with EDX value
                                <1>
8761 0000F47E EB88
                                <1>
                                              short systime 0
8762
                                 <1>
8763
                                 <1>
8764
                                 <1> sysstime: ; Set System Date&Time
8765
                                        ; 31/12/2017
                                 <1>
8766
                                 <1>
                                          ; 30/12/2017 (TRDOS 386 = TRDOS v2.0)
8767
                                 <1>
                                          ; INPUT -> BL =
8768
                                 <1>
                                               0 = Set Date&Time in Unix/Epoch format
8769
                                 <1>
8770
                                 <1>
                                                    1 = Set Time in MSDOS format
8771
                                 <1>
                                                   2 = Set Date in MSDOS format
8772
                                 <1>
                                                    3 = Set Date&Time in MSDOS format
8773
                                 <1>
                                                    4 = Set System Timer (Ticks)
8774
                                                  5 = Convert/Save current time to/as
                                 <1>
8775
                                 <1>
                                                      18.2 Hz system timer ticks
                                                  6 = Convert MSDOS Date&Time to UNIX format
8776
                                 <1>
8777
                                 <1>
                                                      without setting system date&time; (test)
                                                  7 = Convert UNIX Date&Time to MSDOS format
8778
                                 <1>
8779
                                 <1>
                                                      without setting system date&time; (test)
                                                  8-0FFh = invalid !
8780
                                 <1>
                                          ;
8781
                                 <1>
                                                  ECX = Time (or Timer) value in selected format
8782
                                 <1>
                                                  EDX = Date value in MSDOS format if BL=2,3,6
                                          ;
8783
                                 <1>
                                          ; OUTPUT ->
8784
                                 <1>
                                        ; If CF = 0 \rightarrow
8785
                                 <1>
                                                   EAX = Set value
8786
                                 <1>
                                          ;
8787
                                                If CF = 1 -> (invalid BL input)
                                 <1>
                                          ;
8788
                                 <1>
                                                 EAX = Ticks count [TIMER_LH]
8789
                                 <1>
8790
                                <1>
8791 0000F480 20DB
                                <1>
                                        and bl, bl; 0
                                          jnz
8792 0000F482 7511
                                <1>
                                                short sysstime_0
8793 0000F484 89C8
                                <1>
                                          mov
                                                eax, ecx
8794 0000F486 E83371FFFF
                                          call convert from epoch
                                <1>
                                          call set_rtc_date_time
8795 0000F48B E8DF71FFFF
                                <1>
8796 0000F490 E9CDD2FFFF
                                <1>
                                          jmp
8797
                                <1> sysstime_0:
8798 0000F495 80FB08
                                <1>
                                          cmp bl, 8
8799 0000F498 722D
                                <1>
                                          jb
                                                short sysstime 1
8800
                                <1>
                                          ; invalid input (>7)
8801 0000F49A A1[40590100]
                                <1>
                                          mov eax, [TIMER_LH]; 18.2 Hz timer ticks
8802
                                 <1>
                                                             ; Note: [TIMER_LH] may be set
8803
                                 <1>
                                                             ; to wrong timer value due to
8804
                                 <1>
                                                             ; program functions.
8805
                                 <1>
                                                             ; (This value must not be
                                                             ; accepted as [TIMER LH]/18.2
8806
                                 <1>
                                 <1>
                                                             ; seconds since the midnight.)
8808 0000F49F A3[64030300]
                                                [u.r0], eax
                                <1>
                                          mov
8809 0000F4A4 E999D2FFFF
                                 <1>
                                          jmp
                                                error; cf = 1 & [u.r0] = eax = timer ticks
                                <1>
8810
8811
                                <1> sysstime_8:
8812
                                <1> ; BL = 7
8813 0000F4A9 89C8
                                <1>
                                          mov eax, ecx; seconds since 1/1/1970 00:00:00
8814 0000F4AB E80E71FFFF
                                          call convert from epoch
                               <1>
                               <1> xor ah, ah
<1> mov al, [ho
<1> shl eax, 16
8815 0000F4B0 30E4
8816 0000F4B2 A0[B4550100]
                                                al, [hour]
8817 0000F4B7 C1E010
                                          shl eax, 16
8818 0000F4BA A0[B8550100]
                                <1>
                                          mov al, [second]
8819 0000F4BF 8A25[B6550100]
                                <1>
                                         mov
                                                ah, [minute]
8820 0000F4C5 EB92
                                <1>
                                          jmp
                                                short systime_3
8821
                                <1>
                                <1> sysstime_1:
8822
8823 0000F4C7 80FB04
                                <1> cmp bl, 4
                                          je
8824 0000F4CA 743F
                                <1>
                                                short sysstime_2 ; set system timer ticks
8825 0000F4CC 80FB05
                                      cmp bl, 5
jne short sysstime_4
                                <1>
                                     jne short sysstime_4
; convert current time to
call get_rtc_date_time
8826 0000F4CF 754B
                                <1>
8827
                                <1>
                                          ; convert current time to system timer ticks (18.2Hz)
8828 0000F4D1 E8C06FFFFF
                                <1>
8829 0000F4D6 0FB60D[B4550100]
                                <1>
                                          movzx ecx, byte [hour]
8830 0000F4DD B8100E0000
                                <1>
                                          mov eax, 60*60; 1 hour = 3600 seconds
8831 0000F4E2 F7E1
                                          mul ecx
                                <1>
                                          mov
8832 0000F4E4 89C3
                                <1>
                                                ebx, eax
8833 0000F4E6 B13C
                                                cl, 60 ; 1 minute = 60 seconds
                                <1>
                                          mov
8834 0000F4E8 0FB605[B6550100] <1>
                                          movzx eax, byte [minute]
8835 0000F4EF F7E1
                                <1>
                                          mul ecx
8836 0000F4F1 01D8
                                 <1>
                                          add
                                                eax, ebx
                                          mov cl, [second]
8837 0000F4F3 8A0D[B8550100]
                                <1>
8838 0000F4F9 01C8
                                <1>
                                          add
                                                eax, ecx
8839 0000F4FB B1B6
                                 <1>
                                          mov
                                                 cl, 182
8840 0000F4FD F7E1
                                 <1>
                                          mul
                                                 ecx
8841 0000F4FF 83C009
                                 <1>
                                                 eax, 9
                                                edx, 0
8842 0000F502 83D200
                                 <1>
                                          adc
8843 0000F505 B10A
                                 <1>
                                          mov
                                                 cl, 10
8844 0000F507 F7F1
                                 <1>
                                          div
                                                ecx
8845
                                 <1>
                                          ; eax = ((182*seconds)+9)/10
8846 0000F509 89C1
                                 <1>
                                                ecx, eax
                                          mov
8847
                                 <1> sysstime 2:
8848 0000F50B 890D[40590100]
                                                 [TIMER LH], ecx; 18.2 * seconds
                                 <1>
                                          mov
                                 <1> sysstime 3:
8850 0000F511 890D[64030300]
                                                 [u.r0], ecx
                                 <1>
                                          mov
8851 0000F517 E946D2FFFF
                                 <1>
                                                sysret
                                          jmp
                                 <1> sysstime 4:
8853 0000F51C 80FB06
                                 <1>
                                          cmp
                                                bl, 6
8854 0000F51F 7788
                                 <1>
                                                 short sysstime_8
                                          jа
8855
                                 <1>
8856 0000F521 890D[64030300]
                                 <1>
                                                 [u.r0], ecx
                                 <1>
8857
8858 0000F527 880D[B8550100]
                                 <1>
                                          mov
                                                 [second], cl
8859 0000F52D 882D[B6550100]
                                 <1>
                                          mov
                                                 [minute], ch
8860 0000F533 C1E910
                                 <1>
                                          shr
                                                 ecx, 16
8861 0000F536 880D[B4550100]
                                 <1>
                                                [hour], cl
8862
                                 <1>
                                          ; BL = 1,2,3,6
```

```
8863 0000F53C 80FB01
                                 <1>
                                          cmp bl, 1
                                       jna short sysstime_5
; BL = 2,3,6
mov [day], dl
8864 0000F53F 762A
                                 <1>
8865
                                 <1>
8866 0000F541 8815[B2550100] <1>
                                           mov [month], dh
                                        mov
8867 0000F547 8835[B0550100]
                                 <1>
8868 0000F54D C1EA10
                                 <1>
8869 0000F550 668915[AE550100] <1>
                                       mov [year], dx
                                      mov [year], dx
and bl, 3
jz short sysstime_7; 6
; BL = 2,3
test bl, 1
jz short sysstime_6; 2
; BL = 3
8870 0000F557 80E303 <1>
8871 0000F55A 742D
                                 <1>
8872
                                 <1>
8873 0000F55C F6C301 <1>8874 0000F55F 7419 <1
                                 <1>
8874 0000F55F 7419
                                           ; BL = 3
8875
                                 <1>
                                       call set_rtc_date_time
jmp sysret
8876 0000F561 E80971FFFF
                                 <1>
8877 0000F566 E9F7D1FFFF
                                 <1>
8878
                                 <1> sysstime_5:
                                 <1> ; BL = 1
8880 0000F56B E84071FFFF
                                            call set_time_bcd
                                 <1>
8881 0000F570 E80565FFFF
                                 <1>
                                            call set_rtc_time
                                           jmp sysret
8882 0000F575 E9E8D1FFFF
                                 <1>
8883
                                 <1> sysstime_6:
8884
                                 <1> ; BL = 2
                                            call set_date_bcd
8885 0000F57A E80471FFFF
                                 <1>
8886 0000F57F E86565FFFF
                                 <1>
                                            call set_rtc_date
8887 0000F584 E9D9D1FFFF
                                 <1>
                                           jmp sysret
                                 <1> sysstime_7:
8888
                                 <1> ; \overline{BL} = 6
8889
8890
                                 <1>
                                           ; [year], [month], [day],
8891
                                  <1>
                                         ; [hour], [minute], [second]
8892 0000F589 E8AB6FFFFF
                                 <1>
                                         mov ecx, eax; seconds since 1/1/1970 00:00:00
8893 0000F58E 89C1
                                 <1>
8894 0000F590 E97CFFFFFF
                                  <1>
                                           jmp sysstime_3
                                  <1>
8895
8896
                                  <1> sysrename: ; Rename File (or Directory)
                                           ; 31/12/2017 (TRDOS 386 = TRDOS v2.0)
8897
                                  <1>
8898
                                  <1>
                                           ; INPUT ->
;
8899
                                  <1>
8900
                                  <1>
                                                        EBX = File/Directory (ASCIIZ) name address
                                                     ECX = New name (in same dir, no path name)
8901
                                  <1>
                                           ; OUTPUT ->
8902
                                  <1>
8903
                                  <1>
                                                     cf = 0 \rightarrow EAX = 0
8904
                                  <1>
                                           ;
                                                      cf = 1 -> Error code in AL
8905
                                  <1>
8906 0000F595 803D[B3030300]00
                                  <1>
                                            cmp byte [u.uno], 0 ; root (super user) ?
8907 0000F59C 7614
                                            jna short sysrename 0
                                  <1>
8908
                                  <1>
8909
                                  <1> sysrename perm err:
8910
                                 <1> ;mov dword [u.r0], ERR_PERM_DENIED
                                                   eax, ERR PERM DENIED; 'permission denied!'
8911 0000F59E B80B000000
8912 0000F5A3 A3[64030300]
8913 0000F5A8 A3[C8030300]
8914 0000F5AD E990D1FFFF
8911 0000F59E B80B000000
                                 <1>
                                            mov
                                 <1>
                                                  [u.r0], eax
                                           mov
                                           mov [u.error], eax
                                 <1>
8914 0000F5AD E990D1FFFF
                                 <1>
                                          jmp error
                                 <1>
8915
8916
                                 <1> sysrename_0:
8917 0000F5B2 51
                                 <1> push ecx ; new file name address (in user space)
8918 0000F5B3 89DE
                                 <1>
                                           mov esi, ebx
                                        ; file name; mov ax, 1
8919
                                 <1>
                                           ; file name is forced, change directory as temporary
8920
                                  <1>
                                       ;mov [FFF_Valid], ah ; 0 ; reset
;call set_working_path
8921
                                  <1>
8922
                                 <1>
8923 0000F5B5 E8C5060000
                                 <1>
                                       call set_working_path_x
                                           jnc short sysrename 1
and eax, eax ; 0 -> Bad Path!
8924 0000F5BA 731E
                                 <1>
8925 0000F5BC 21C0
                                 <1>
8926 0000F5BE 7505
                                 <1>
                                            jnz short sysrename_err
8927
                                 <1>
                                           ; eax = 0
8928
                                  <1> sysrename_path_not_found:
8929 0000F5C0 B813000000
                                 <1>
                                       mov eax, ERR_INV_PATH_NAME ; 'Bad path name !'
                                 <1> sysrename_err:
8930
8931 0000F5C5 59
                                 <1>
                                         pop ecx; new file name address (in user space)
8932
                                 <1> sysrename_error:
8933 0000F5C6 A3[64030300]
                                 <1> mov [u.r0], eax
                                                   [u.error], eax
8934 0000F5CB A3[C8030300]
                                 <1>
                                           mov
                                            call reset_working_path
8935 0000F5D0 E87F070000
                                 <1>
                                          jmp error
8936 0000F5D5 E968D1FFFF
                                 <1>
                                 <1> sysrename_1:
8937
                                 <1> mov al, 08h; Except volume labels (& long names)
<1> mov al, [Attributes]
8938 0000F5DA B008
8939 0000F5DC A0[3C630100]
8940 0000F5E1 2410
                                           and al, 10h ;
                                  <1>
                                           ;mov esi, FindFile_Name
;mov ax, 1800h; Only files
8941
                                  <1>
8942
                                  <1>
                                            ;mov ax, 0810h ; Only directories
8943
                                  <1>
                                            mov ax, 0800h; Find call find_first_file
8944 0000F5E3 66B80008
                                                   ax, 0800h; Find File or Directory
                                  <1>
8945 0000F5E7 E8FB8CFFFF
                                  <1>
                                  <1>
                                            ;jnc short sysrename_2
8947 0000F5EC 72D7
                                  <1>
                                            iс
                                                  short sysrename_err
                                  <1> sysrename 2:
8948
8949
                                  <1>
                                           ; ESI = Directory Entry (FindFile DirEntry) Location
8950
                                  <1>
                                            ; EDI = Directory Buffer Directory Entry Location
8951
                                  <1>
                                            ; EAX = File Size
8952
                                            ; BL = Attributes of The File/Directory
                                  <1>
8953
                                  <1>
                                            ; BH = Long Name Yes/No Status (>0 is YES)
8954
                                  <1>
                                            ; DX > 0 : Ambiguous filename chars are used
8955
                                  <1>
                                                   dx, dx; Ambiguous filename chars used sign (DX>0)
8956 0000F5EE 6621D2
                                  <1>
8957 0000F5F1 7407
                                  <1>
                                                   short sysrename 3
                                            iΖ
                                                   eax, ERR_INV_FILE_NAME ; 'invalid file name !'
8958 0000F5F3 B81A000000
                                  <1>
                                            mov
8959 0000F5F8 EBCB
                                  <1>
                                             jmp short sysrename_err
8960
                                  <1> sysrename_3:
8961
                                  <1>
                                            ; EDI = Directory buffer entry offset/address
                                            ; BL = File (or Directory) Attributes
8962
                                  <1>
                                            ; mov bl, [EDI+0Bh]
8963
                                  <1>
8964
                                  <1>
8965 0000F5FA 5A
                                  <1>
                                            pop edx; new file name address (in user space)
                                  <1>
8966
                                  <1>
8967
                                            ; check file/directory attributes
```

```
8968 0000F5FB F6C307
                                <1>
                                           test bl, 7; system, hidden, readonly
8969 0000F5FE 759E
                                <1>
                                           jnz short sysrename perm err
8970
                                <1> sysrename_4:
8971 0000F600 66817F0CA101
                                <1>
                                          cmp word [edi+DirEntry_NTRes], 01A1h ; Singlix FS
8972 0000F606 7496
                                <1>
                                                 short sysrename_perm_err ; -temporary!-
                                          jе
8973
                                <1>
8974
                                <1>
                                          ; save old file name & file info (FFF structure)
8975 0000F608 BE[26620100]
                                <1>
                                          mov esi, FindFile_Drv
8976 0000F60D BF[6C630100]
                                <1>
                                                 edi, SourceFile Drv
                                          mov
8977 0000F612 B920000000
                                                ecx, 128/4
                                <1>
                                          mov
8978 0000F617 F3A5
                                <1>
                                                movsd
8979
                                 <1>
8980 0000F619 89D6
                                <1>
                                                esi, edx; new file name address (in user space)
                                          mov
8981 0000F61B BF[EC630100]
                                <1>
                                          mov edi, DestinationFile_Drv
8982 0000F620 E893AEFFFF
                                 <1>
                                          call parse path name
8983 0000F625 729F
                                 <1>
                                                short sysrename_error ; eax = 1 (Bad file name)
                                          jс
                                 <1>
                                          ; same drive ?
8985
                                 <1>
8986 0000F627 A0[26620100]
                                 <1>
                                          mov al, [FindFile_Drv]
8987 0000F62C 3A05[EC630100]
                                 <1>
                                                al, [DestinationFile Drv]
                                          cmp
8988
                                 <1>
                                          ;jne short sysrename_perm_err ; Permission denied
8989 0000F632 7509
                                 <1>
                                                short sysrename_5 ; Bad file name
8990
                                 <1>
8991
                                 <1>
                                          ; no path name !? (rename file in same directory)
8992 0000F634 803D[ED630100]20
                                 <1>
                                          cmp byte [DestinationFile_Directory], 20h
8993 0000F63B 7607
                                 <1>
                                           jna
                                                short sysrename_6
                                 <1> sysrename 5:
8995 0000F63D B801000000
                                          mov eax, ERR_BAD_CMD_ARG ; 1 = Bad file name
                                 <1>
8996
                                                                  ; (Bad argument)
                                 <1>
8997 0000F642 EB82
                                                 short sysrename_error
                                 <1>
                                          jmp
8998
                                 <1> sysrename_6:
8999 0000F644 803D[2E640100]20
                                 <1>
                                          cmp byte [DestinationFile_Name], 20h
9000 0000F64B 76F0
                                 <1>
                                           jna
                                                short sysrename_5
9001
                                 <1>
9002 0000F64D BE[2E640100]
                                <1>
                                                esi, DestinationFile_Name
                                          mov
                                          call check_filename; is it a valid msdos file name?
9003 0000F652 E84E90FFFF
                                <1>
9004 0000F657 0F8269FFFFFF
                                <1>
                                                sysrename_error ; 26 = ERR_INV_FILE_NAME
                                          jс
9005
                                 <1>
9006
                                 <1>
                                          ;mov esi, DestinationFile_Name
                                                ax, 0800h; Find File or Directory
9007 0000F65D 66B80008
                                <1>
                                          mov
                                          call find first file
9008 0000F661 E8818CFFFF
                                <1>
9009 0000F666 720A
                                                short sysrename 7
                                 <1>
                                          jс
9010
                                <1>
9011 0000F668 B80E000000
                                 <1>
                                          mov eax, ERR_FILE_EXISTS ; file already exists !
                                                sysrename_error
9012 0000F66D E954FFFFFF
                                 <1>
                                          jmp
                                 <1> sysrename 7:
9013
                                          ; eax = 2 (File not found!)
9014
                                 <1>
                                          cmp al, 2 ; ERR_NOT_FOUND
9015 0000F672 3C02
                                 <1>
9016 0000F674 0F854CFFFFFF
                                 <1>
                                          jne sysrename_error
9017
                                 <1>
                                          ; 31/12/2017
9018
                                 <1>
9019
                                 <1>
                                          ; Following code is also part of 'rename file' in
                                          ; 'trdosk3.s' (MainProg's 'rename' command) ; 13/11/2017
9020
                                 <1>
9021 0000F67A BE[2E640100]
                                 <1>
                                          mov esi, DestinationFile_Name ; (Rename_NewName)
                                          mov
                                                cx, [SourceFile_DirEntryNumber]
9022 0000F67F 668B0D[E6630100]
                                 <1>
9023 0000F686 66A1[D2630100]
                                 <1>
                                          mov
                                                ax, [SourceFile_DirEntry+20] ; First Cluster, HW
9024 0000F68C C1E010
                                          shl eax, 16
                                 <1>
9025 0000F68F 66A1[D8630100]
                                          mov ax, [SourceFile_DirEntry+26] ; First Cluster, LW
                                 <1>
9026 0000F695 0FB61D[BB630100]
                                 <1>
                                          movzx ebx, byte [SourceFile_LongNameEntryLength]
9027 0000F69C E85FB6FFFF
                                 <1>
                                          call rename_directory_entry
9028 0000F6A1 0F821FFFFFF
                                 <1>
                                          jc sysrename_error
9029
                                 <1>
                                          ;xor eax, eax
9030 0000F6A7 A3[64030300]
                                 <1>
                                                [u.r0], eax ; 0
                                          mov
                                          ;mov [u.error], eax
9031
                                 <1>
9032 0000F6AC E8A3060000
                                 <1>
                                          call reset_working_path
9033 0000F6B1 E9ACD0FFFF
                                 <1>
                                          jmp
                                                sysret
9034
                                 <1>
9035
                                 <1> sysmem: ; Get Total&Free Memory amount
9036
                                 <1>
                                          ; 31/12/2017 (TRDOS 386 = TRDOS v2.0)
9037
                                 <1>
9038
                                 <1>
                                            ; INPUT ->
9039
                                 <1>
                                                none
                                          ; OUTPUT ->
9040
                                 <1>
9041
                                 <1>
                                          ; EAX = Total memory count (in bytes)
                                                EBX = Virtually available memory amount (in bytes)
9042
                                 <1>
                                          ;
9043
                                 <1>
                                                      = 4GB - CORE (4MB)
9044
                                 <1>
                                                 ECX = Free memory count (in bytes)
                                          ;
                                                EDX = Calculated free memory count (in bytes)
9045
                                 <1>
9046
                                 <1>
9047 0000F6B6 A1[C4580100]
                                 <1>
                                                eax, [memory_size] ; in pages
                                          mov
9048 0000F6BB C1E00C
                                 <1>
                                          shl
                                                 eax, 12
                                                                      ; in bytes
                                                 [u.r0], eax
9049 0000F6BE A3[64030300]
                                 <1>
                                           mov
                                          call calc_free_mem
9050 0000F6C3 E8893CFFFF
                                 <1>
9051
                                 <1>
                                          ; edx = calculated free pages
9052
                                 <1>
                                          ; ecx = 0
9053 0000F6C8 8B2D[60030300]
                                 <1>
                                          mov ebp, [u.usp] ; EBP points to user's registers
9054 0000F6CE C745100000C0FF
                                                 dword [ebp+16], ECORE ; EBX (for user)
                                 <1>
                                          mov
                                                             ; 0FFC00000h ; 4GB - 4MB
9055
                                 <1>
9056 0000F6D5 C1E20C
                                                 edx, 12
                                 <1>
                                          shl
9057 0000F6D8 895514
                                                [ebp+20], edx ; EDX (for user)
                                 <1>
                                          mov
9058 0000F6DB 8B0D[C8580100]
                                <1>
                                          mov
                                                 ecx, [free_pages]
9059 0000F6E1 C1E10C
                                 <1>
                                          shl
                                                 ecx, 12 ; free bytes
                                                 [ebp+24], ecx; ECX (for user)
9060 0000F6E4 894D18
                                 <1>
                                          mov
9061
                                 <1>
                                          ;mov [free_pages], edx
9062 0000F6E7 E976D0FFFF
                                 <1>
                                          jmp
                                                sysret
9063
                                 <1>
9064
                                 <1> sysprompt:
                                          ; Set TRDOS 386 Command Interpreter (MainProg) prompt
9065
                                 <1>
                                          ; 31/12/2017 (TRDOS 386 = TRDOS v2.0)
9066
                                 <1>
9067
                                 <1>
                                            ; INPUT ->
9068
                                 <1>
9069
                                 <1>
                                                EBX = 0 -> use default prompt
                                                 EBX > 0 -> prompt string (ASCIIZ) address
9070
                                 <1>
                                          ;
9071
                                 <1>
                                                         (Max. 11 characters except ZERO tail)
9072
                                 <1>
                                          ; OUTPUT ->
```

```
9073
                                                      (EAX = 0)
9074
                                     <1>
                                                      CF = 0 \rightarrow Successful
                                               ;
9075
                                                      CF = 1 \rightarrow Failed
                                     <1>
                                               ;
9076
                                    <1>
9077 0000F6EC 21DB
                                    <1>
                                               and
                                                     ebx, ebx
9078 0000F6EE 750A
                                    <1>
                                               jnz
                                                      short sysprompt_0
                                    <1>
9080 0000F6F0 E8F685FFFF
                                               call default_command_prompt ; '['+'TRDOS'+']'
                                    <1>
9081 0000F6F5 E968D0FFFF
                                    <1>
                                                      sysret
                                               jmp
                                    <1>
                           <1>
<1>
<1>

<1>

<1>

<1>

<1>

<1>

<1>

<1>

9083
                                    <1> sysprompt_0:
9084 0000F6FA 31C0
9085 0000F6FC A3[64030300]
9086 0000F701 89DE
9087 0000F703 B90C000000
9088 0000F708 89E5
9089 0000F70A 29CC
9090 0000F70C 49
9091 0000F70D 89E7
9092 0000F70F E833F1FFFF
                                              call transfer from user buffer
9093 0000F714 7211
9094 0000F716 803E20
9095 0000F719 760C
9096 0000F71B E8DD85FFFF
9097 0000F720 89EC
9098 0000F722 E93BD0FFFF
9099
9100
9101 0000F727 89EC
9102 0000F729 E914D0FFFF
                                    <1>
                                               jmp
9103
                                    <1>
9104
                                     <1> syspath:
                                    <1> ; Get/Set Run Path
9105
9106
                                    <1>
                                               ; 31/12/2017 (TRDOS 386 = TRDOS v2.0)
9107
                                     <1>
                                              ; INPUT ->
9108
                                    <1>
                                                   EBX = 0 -> get path (to buffer address in ECX)
9109
                                     <1>
9110
                                    <1>
                                           ;
                                                      EBX > 0 -> set path
                                                      EBX = Path string buffer address (ASCIIZ)
9111
                                    <1>
9112
                                    <1>
                                                           (Path description except 'PATH=')
                                               ;
                                                      ECX = Buffer address (if EBX = 0)
9113
                                    <1>
9114
                                                           (ECX will not be used if EBX > 0)
                                     <1>
                                                      DL = Buffer size (0 = 256 byte)
9115
                                    <1>
                                              ;
9116
                                     <1>
9117
                                     <1>
                                              ; OUTPUT ->
                                               ; CF = 0 -> Successful (EAX = String length)
9118
                                    <1>
                                                      CF = 1 \rightarrow Failed (EAX = 0)
9119
                                     <1>
9120
                                    <1>
9121
                                    <1>
                                               ; NOTE: 'PATH=' or 'PATH' must be excluded
                                               ; (It must not be at the beginning of the string.)
9122
                                    <1>
9123
                                    <1>
9124 0000F72E 89E5
                                    <1>
                                               mov
                                                      ebp, esp
9125 0000F730 81EC00010000
                                    <1>
                                               sub
                                                      esp, 256
9126 0000F736 89E7
                                    <1>
                                                      edi, esp
                                    <1>
9128 0000F738 31C0
                                    <1>
                                               xor
                                                      eax, eax
9129 0000F73A A3[64030300]
                                    <1>
                                                      [u.r0], eax
                                               mov
9130
                                    <1>
9131 0000F73F 21DB
                                    <1>
                                               and
                                                     ebx, ebx
9132 0000F741 752E
                                   <1>
                                               jnz short syspath_0
9133
                                   <1>
                                               ; EBX = 0 -> get run path
9134
                                    <1>
9135 0000F743 89CB
                                   <1>
                                               mov ebx, ecx; buffer addr (in user's mem space)
9136 0000F745 BE[2E0E0100]
9137 0000F74A 0FB6CA
                                   <1>
                                               mov esi, Cmd_Path ; 'PATH' address
9137 0000F74A 0FB6CA
                                    <1>
                                               movzx ecx, dl
                                             sub cl, 1; 0 -> 255, 1 -> 0
adc cx, 1; 255 -> 256, 0 -> 1
9138 0000F74D 80E901
                                   <1>
9139 0000F750 6683D101
                                   <1>
                                            ; EDI = Output buffer
9140
                                    <1>
9141
                                    <1>
                                              ; CX = Buffer length
                                              ; AL = 0 -> use ASCIIZ word in [ESI]
9142
                                    <1>
                                              ; ESI = 'PATH' address (with zero tail)
9143
                                    <1>
                                              call get_environment_string
jc short syspath_err
9144 0000F754 E8D89DFFFF
                                    <1>
9145 0000F759 72CC
                                    <1>
                                             mov edi, ebx ; User's buffer address
9146 0000F75B 89DF
                                   <1>
                                             mov esi, esp
9147 0000F75D 89E6
                                    <1>
9148
                                               ; EDI = User's buffer address
                                    <1>
9149
                                    <1>
                                              ; ECX = transfer (byte) count
9150 0000F75F E899F0FFFF
                                    <1>
                                              call transfer_to_user_buffer
                                              jc short syspat mov [u.r0], ecx
9151 0000F764 72C1
                                    <1>
                                                      short syspath_err
9152 0000F766 890D[64030300]
                                    <1>
9153 0000F76C E9F1CFFFFF
                                    <1>
                                              jmp sysret
9154
                                     <1>
                                    <1> syspath_0:
9156 0000F771 89DE
                                    <1>
                                                      esi, ebx
9157 0000F773 0FB6CA
                                    <1>
                                               movzx ecx, dl
                                               sub cl, 1; 0 -> 255, 1 -> 0
9158 0000F776 80E901
                                   <1>
9159 0000F779 6683D101
                                              adc cx, 1; 255 -> 256, 0 -> 1
                                   <1>
9160 0000F77D E8C5F0FFFF
                                   <1>
                                             call transfer_from_user_buffer
                                            jc
                                                      short syspath err
9161 0000F782 72A3
                                    <1>
                                              ;(*) 'PATH=' will be added to
9162
                                    <1>
9163
                                    <1>
                                                      the head of the string
9164 0000F784 83EC08
                                    <1>
                                              sub
                                                      esp, 8 ;(*)
9165 0000F787 89FE
                                    <1>
                                                     esi, edi ;(*)
                                              mov
                                               call set_path_x ; (*)
9166 0000F789 E8879DFFFF
                                    <1>
9167 0000F78E 7297
                                    <1>
                                               iс
                                                      short syspath_err
9168 0000F790 8915[64030300]
                                    <1>
                                               mov
                                                      [u.r0], edx ; run path string length
9169 0000F796 E9C7CFFFFF
                                    <1>
                                                     svsret
                                               qmj
9170
                                    <1>
                                     <1> sysenv:
9171
                                             ; Get/Set Environment Variables
9172
                                    <1>
9173
                                    <1>
                                               ; 31/12/2017 (TRDOS 386 = TRDOS v2.0)
9174
                                     <1>
                                               ; INPUT ->
9175
                                    <1>
9176
                                     <1>
                                               ; EBX = 0 \rightarrow get (all) environment variables
9177
                                     <1>
                                                      (Required Buffer length = 512 bytes)
```

```
EBX > 0 -> set (one) environment variable
9178
9179
                                                   <1>
                                                                                   (If there is not a '=' after
                                                                 ;
9180
                                                  <1>
                                                                                   the environment variable name, it will
9181
                                                  <1>
                                                                                 accepted as 'get environment variable'.)
9182
                                                  <1>
                                                                                    EBX = Buffer address
                                                                          ECX = Buffer address (if EBX = 0)
9183
                                                  <1>
                                                                          (ECX will not be used if EBX > 0)
9184
                                                  <1>
                                                                                  (Note: Buffer size is 512 bytes.)
9185
                                                  <1>
9186
                                                  <1>
                                                                          DL = Buffer size (0 = 256 byte)
9187
                                                  <1>
                                                                           (For one envrionment variable)
                                                                ;
9188
                                                  <1>
                                                                ; OUTPUT ->
9189
                                                   <1>
9190
                                                  <1>
                                                                        (EAX = 0)
9191
                                                  <1>
                                                                          CF = 0 -> Successful (EAX = String length)
9192
                                                  <1>
                                                                          CF = 1 \rightarrow Failed (EAX = 0)
9193
                                                  <1>
9194
                                                  <1>
                                                                ; Note: Environment variable name, for example,
9195
                                                  <1>
                                                                         'PATH=' must be included at the beginning
9196
                                                  <1>
                                                                           of the environment string. If the variable
9197
                                                  <1>
                                                                          name is as 'PATH' but it is not as 'PATH='
                                                                          the variable string (row) will be returned.
9198
                                                  <1>
                                                                          If variable name is as 'PATH=' but there is
9199
                                                  <1>
                                                                 ;
9200
                                                  <1>
                                                                          not a following text after the variable name,
                                                                 ;
9201
                                                  <1>
                                                                          the environment variable will be reset/deleted.
9202
                                                  <1>
9203 0000F79B 89E5
                                                  <1>
                                                                 mov
                                                                          ebp, esp
9204 0000F79D 81EC00020000
                                                  <1>
                                                                 sub
                                                                          esp, 512
9205 0000F7A3 89E7
                                                  <1>
                                                                 mov
                                                                          edi, esp
9206
                                                  <1>
9207 0000F7A5 31C0
                                                  <1>
                                                                           eax, eax
                                                                 xor
9208 0000F7A7 A3[64030300]
                                                  <1>
                                                                           [u.r0], eax
                                                                 mov
                                                  <1>
9210 0000F7AC 21DB
                                                                 and
                                                  <1>
                                                                           ebx, ebx
9211 0000F7AE 7524
                                                 <1>
                                                                          short sysenv_0
9212
                                                  <1>
9213
                                                 <1>
                                                                ; EBX = 0 -> get (all) environment variables
9214 0000F7B0 89EC
                                                 <1>
                                                                mov esp, ebp
                                                                          esi, Env_Page ; Environment page
9215 0000F7B2 BE00300900
                                                 <1>
                                                                mov
9216 0000F7B7 89CF
                                                 <1>
                                                                          edi, ecx; buffer addr (in user's mem space)
                                                                mov
                                                                mov
9217 0000F7B9 B900020000
                                                 <1>
                                                                          ecx, 512
9218 0000F7BE E83AF0FFFF
                                                          call transfer_to_user_buffer
                                                 <1>
9219 0000F7C3 0F8279CFFFFF
                                                 <1>
                                                                jс
                                                                          error
9220 0000F7C9 890D[64030300]
                                                 <1>
                                                                          [u.r0], ecx
                                                                mov
9221 0000F7CF E98ECFFFFF
                                                 <1>
                                                                jmp
                                                                         sysret
9222
                                                  <1>
9223
                                                 <1> sysenv_0:
9223

9224 0000F7D4 89DE

9225 0000F7D6 0FB6CA <1> mov2...

9226 0000F7D9 80E901 <1> sub

9227 0000F7DC 6683D101 <1> adc

9228 0000F7E0 E862F0FFFF <1> call

9229 0000F7E5 723F <1> jc

<1> mov
                                                <1> mov esi, ebx ; * ; user's buffer address
                                                                movzx ecx, dl
                                                               sub cl, 1; 0 -> 255, 1 -> 0
                                                                adc cx, 1; 255 -> 256, 0 -> 1
                                                                call transfer_from_user_buffer
                                                                          short sysenv err
                                                                          esi, edi
9231 0000F7E9 8A06
                                                <1>
                                                          mov al, [esi]
                                                <1> cmp <1> cm
9232 0000F7EB 3C20
                                                                          al, 20h
9233 0000F7ED 7637
                                                                          short sysenv_err
                                                                          al, '='
9234 0000F7EF 3C3D
9235 0000F7F1 7433
                                                 <1>
                                                                          short sysenv_err
                                                                jе
9236 0000F7F3 56
                                                 <1>
                                                                 push
                                                 <1> sysenv_1:
9237
9238 0000F7F4 46
                                                <1> inc
                                                                          esi
                                                <1>
<1>
9239 0000F7F5 803E3D
                                                                 cmp
                                                                          byte [esi], '='
                                                          je
cmp
jnb
mov
                                                                          short sysenv 3
9240 0000F7F8 7433
9241 0000F7FA 803E20
                                              <1>
                                                                          byte [esi], 20h
9242 0000F7FD 73F5
                                                 <1>
                                                                          short sysenv_1
9243 0000F7FF C60600
                                                 <1>
                                                                mov
                                                                          byte [esi], 0
                                                              pop esi
9244 0000F802 5E
                                                 <1>
                                                              ; EDI = Output buffer
9245
                                                 <1>
9246
                                                 <1>
                                                                ; CX = Buffer length
                                                                xor al, al
9247 0000F803 30C0
                                                 <1>
9248
                                                 <1>
                                                                ; AL = 0 -> use ASCIIZ word in [ESI]
9249
                                                  <1>
                                                                ; ESI = Environment variable name address
9250 0000F805 E8279DFFFF
                                                                call get_environment_string
                                                 <1>
9251 0000F80A 721A
                                                 <1>
                                                             jc short sysenv_err
                                                             mov edi, ebx; *; user's bu
mov ecx, eax; String length
9252 0000F80C 89DF
                                                 <1>
                                                                         edi, ebx ; *; user's buffer address
9253 0000F80E 89C1
                                                 <1>
                                                             mov esi, esp
9254 0000F810 89E6
                                                 <1>
                                                             ; ESI = system buffer address
9255
                                                 <1>
                                                                ; EDI = User's buffer address
9256
                                                  <1>
9257
                                                  <1>
                                                                ; ECX = transfer (byte) count
9258 0000F812 E8E6EFFFFF
                                                                 call transfer to user buffer
                                                  <1>
9259 0000F817 720D
                                                  <1>
                                                                 jc
                                                                           short sysenv_err
                                                 <1>
9260 0000F819 890D[64030300]
                                                                mov
                                                                           [u.r0], ecx; transfer (byte) count
                                                  <1> sysenv_2:
9262 0000F81F 89EC
                                                  <1>
                                                                 mov
                                                                          esp, ebp
9263 0000F821 E93CCFFFFF
                                                                          sysret
                                                 <1>
                                                                 jmp
                                                 <1> sysenv err:
                                                 <1>
9265 0000F826 89EC
                                                                 mov
                                                                          esp, ebp
9266 0000F828 E915CFFFFF
                                                 <1>
                                                                 qmŗ
                                                                           error
                                               <1> sysenv_3:
9268 0000F82D 46
                                                 <1>
                                                                 inc
                                                                          esi
9269 0000F82E 803E20
                                                                          byte [esi], 20h
                                                 <1>
                                                                 cmp
                                            <1> cmp
<1> jnb
<1> mov
<1> pop
<1> call
<1> jc
9270 0000F831 73FA
                                                                          short sysenv 3
9271 0000F833 C60600
                                                                          byte [esi], 0
9272 0000F836 5E
                                                                          esi
9273 0000F837 E8B89DFFFF
                                                                         set environment string
9274 0000F83C 72E8
                                                                          short sysenv err
9275 0000F83E 8915[64030300]
                                                <1>
                                                                          [u.r0], edx
                                                               mov
9276 0000F844 EBD9
                                                 <1>
                                                                jmp
                                                                          short sysenv_2
                                                  <1>
9278
                                                  <1>
9279
                                                  <1>; temporary - 24/01/2016
9280
                                                  <1>
9281
                                                  <1> iget:
9282 0000F846 C3
                                                  <1> retn
```

```
9283
                              <1> isintr:
9284 0000F847 C3
                              <1>
                                     retn
                              <1> iopen:
9285
9286 0000F848 C3
                              <1>
9287
                              <1> iclose:
9288 0000F849 C3
                              <1>
                                      retn
9289
                              <1> sndc:
9290 0000F84A C3
                              <1>
                                   retn
9291
                              <1> access:
9292 0000F84B C3
                              <1> retn
                              <1> sleep:
9293
9294 0000F84C C3
                              <1>
                                      retn
2311
                                 %include 'trdosk7.s' ; 24/01/2016
                              1
                              <1> ; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.0) - DISK READ&WRITE : trdosk7.s
  2
  3
                              <1> ; Last Update: 25/02/2016
  5
                              <1> ; ------
  6
                              <1> ; Beginning: 24/01/2016
  7
                              <1> ; ------
  8
                              <1>; Assembler: NASM version 2.11 (trdos386.s)
  9
                              <1>; Derived from TRDOS Operating System v1.0 (8086) source code by Erdogan Tan
 10
 11
                              <1>; DISK_IO.ASM (20/07/2011)
                              12
                              <1> ; DISK_IO.ASM (c) 2009-2011 Erdogan TAN [ 04/07/2009 ] Last Update: 20/07/2011
 13
 15
                              <1> disk_write:
                                    ; 25/02/2016
                              <1>
 16
                                      ; 24/02/2016
 17
                              <1>
                                     ; 23/02/2016
 18
                              <1>
                                     cmp byte [esi+LD_LBAYes], 0
 19 0000F84D 807E0500
                              <1>
 20 0000F851 777B
                              <1>
                                               short lba_write
                                      jа
 21
                              <1>
 22
                              <1> chs_write:
                                   ; 25/02/2016
 23
                              <1>
                              <1>
                                      ; 23/02/2016
 25 0000F853 C605[75610100]03
                              <1>
                                     mov byte [disk_rw_op], 3 ; CHS write
 26 0000F85A EB0D
                                            short chs rw
                              <1>
                                     jmp
                              <1>
                              <1> disk_read:
 28
                                  ; 25/02/2016
 29
                              <1>
                                      ; 24/02/2016
 30
                              <1>
 31
                              <1>
                                     ; 23/02/2016
                                     ; 17/02/2016
; 14/02/2016
; 31/01/2016 (TRDOS 386 = TRDOS v2.0)
 32
                              <1>
 33
                              <1>
                              <1>
                                     ; 17/10/2010
; 18/04/2010
 35
                              <1>
 36
                              <1>
 37
                              <1>
                                     ; INPUT -> EAX = Logical Block Address
 38
                              <1>
 39
                              <1>
                                               ESI = Logical Dos Disk Table Offset (DRV)
                                               ECX = Sector Count
 40
                              <1>
                                      ;
 41
                              <1>
                                               EBX = Destination Buffer
                                     ; OUTPUT ->
 42
                              <1>
 43
                              <1>
                                               cf = 0 \text{ or } cf = 1
                                       ; (Modified registers: EAX, EBX, ECX, EDX)
                              <1>
 45
                              <1>
 46 0000F85C 807E0500
                              <1>
                                       cmp byte [esi+LD_LBAYes], 0
                                       ja
 47 0000F860 7775
                              <1>
                                              short lba_read
 48
                              <1>
 49
                              <1> chs read:
                              <1> ; 25/02/2016 
<1> ; 24/02/2016
 50
                                      ; 24/02/2016
 51
                              <1>
                                     ; 23/02/2016
 52
                              <1>
                                     ; 31/01/2016 (TRDOS 386 = TRDOS v2.0)
; 20/07/2011
 53
                              <1>
                              <1>
                                     ; 04/07/2009
 55
                              <1>
 56
                              <1>
                                      ; INPUT -> EAX = Logical Block Address
 57
                              <1>
 58
                              <1>
                                               ECX = Number of sectors to read
                                      ;
                                               ESI = Logical Dos Disk Table Offset (DRV)
 59
                              <1>
                              <1>
                                               EBX = Destination Buffer
 60
                                       ;
 61
                              <1>
                                               cf = 0 \text{ or } cf = 1
                              <1>
                                     ;
 62
 63
                              <1>
                                      ; (Modified registers: EAX; EBX, ECX, EDX)
 64
                              <1>
 65
                              <1>
                                       ; 23/02/2016
                                       mov byte [disk rw op], 2; CHS read
  66 0000F862 C605[75610100]02
                              <1>
 67
                              <1>
 68
                              <1> chs_rw:
                                     ;;movzx
                                                   edx, word [esi+LD_BPB+SecPerTrack]
  69
                              <1>
  70
                                       ;movzx edx, byte [esi+LD_BPB+SecPerTrack] ; <= 63</pre>
 71
                              <1>
                                       ;mov [disk rw spt], dl
 72
                              <1>
                              <1> chs_read_next_sector:
 73
 74 0000F869 C605[76610100]04
                              <1>
                                      mov byte [retry_count], 4
 75
                              <1>
 76
                              <1> chs_read_retry:
                                      ;mov [sector count], ecx; 23/02/2016
 77
                              <1>
 78
                              <1>
 79 0000F870 50
                                                              ; Linear sector #
                              <1>
                                       push eax
 80 0000F871 51
                              <1>
                                                              ; # of FAT/FILE/DIR sectors
                                       push ecx
                             <1>
                                      movzx ecx, word [esi+LD_BPB+SecPerTrack]
 82 0000F872 0FB74E1E
                             <1>
 83
                             <1>
                                      ;movzx ecx, byte [disk_rw_spt] ; 23/02/2016
                                     sub edx, edx
 84 0000F876 29D2
                             <1>
                                     div ecx
 85 0000F878 F7F1
                             <1>
 86
                             <1>
                                      ; eax = track, dx (dl ) = sector (on track)
 87
                             <1>
                                      ;sub cl, dl; 24/02/2016 (spt - sec)
                             <1>
                                     ;push ecx ; *
 88
                          <1>
<1>
 89 0000F87A 6689D1
                                      mov cx, dx inc cx
                                                              ; Sector (zero based)
                                                              ; To make it 1 based
 90 0000F87D 6641
 91 0000F87F 6651
                                      push cx
                             <1>
 92 0000F881 668B4E20
                             <1>
                                      mov cx, [esi+LD_BPB+Heads]
```

```
93 0000F885 6629D2
                                <1>
                                          sub
                                                dx, dx
 94 0000F888 F7F1
                                <1>
                                          div
                                                ecx
                                                                    ; Convert track to head & cyl
                                          ; eax (ax) = cylinder, dx (dl) = head (max. FFh)
 95
                                <1>
 96 0000F88A 88D6
                                <1>
                                          mov dh, dl
 97 0000F88C 6659
                                <1>
                                          pop
                                                 CX
                                                                    ; AX=Cyl, DH=Head, CX=Sector
 98 0000F88E 8A5602
                                <1>
                                          mov
                                                 dl, [esi+LD_PhyDrvNo]
                                <1>
100 0000F891 88C5
                                          mov
                                                ch, al
                                                                     ; NOTE: max. 1023 cylinders !
                                <1>
101 0000F893 C0CC02
                                <1>
                                          ror
                                                ah, 2
                                                                    ; Rotate 2 bits right
                                          or
102 0000F896 08E1
                                <1>
                                                cl, ah
103
                                 <1>
                                          ; 24/02/2016
104
                                 <1>
                                          ;pop eax; * (spt - sec) (example: 63 - 0 = 63)
105
                                 <1>
106
                                 <1>
                                          ;cmp eax, [sector_count]
                                          ;jb short chs_write_sectors
;je short chs_read_sectors
107
                                 <1>
108
                                 <1>
                                          ;; (# of sectors to read is more than remaining sectors on the track)
109
110
                                 <1>
                                          ;mov al, [sector_count]
111
                                 <1> ;chs_read_sectors: ; read or write !
                                          mov al, 1 ; 25/02/2016
112 0000F898 B001
                                 <1>
                                                 ah, [disk_rw_op] ; 02h = chs read, 03h = chs write
113 0000F89A 8A25[75610100]
                                <1>
                                           mov
                                 <1>
115 0000F8A0 E88849FFFF
                                           call int13h
                                                                     ; BIOS Service func ( ah ) = 2
                                 <1>
116
                                 <1>
                                                                            ; Read disk sectors
                                 <1>
117
                                                                             ; AL-sec num CH-track CL-sec
118
                                 <1>
                                                                             ; DH-head DL-drive ES:BX-buffer
119
                                 <1>
                                                                             ; CF-flag AH-stat AL-sec read
120
                                 <1>
                                                                           ; If CF = 1 then (If AH > 0)
121 0000F8A5 8825[77610100]
                                 <1>
                                          mov
                                                 [disk_rw_err], ah
122
                                 <1>
123 0000F8AB 59
                                 <1>
                                          pop
                                                 ecx
124 0000F8AC 58
                                 <1>
                                           pop
                                                 eax
125 0000F8AD 7314
                                 <1>
                                           jnc
                                                 short chs_read_ok
                                 <1>
127 0000F8AF 803D[77610100]09
                                 <1>
                                                 byte [disk_rw_err], 09h; DMA crossed 64K segment boundary
                                          cmp
128 0000F8B6 7408
                                 <1>
                                                 short chs_read_error_retn
                                 <1>
130 0000F8B8 FE0D[76610100]
                                 <1>
                                          dec
                                                 byte [retry_count]
131 0000F8BE 75B0
                                 <1>
                                                 short chs_read_retry
132
                                 <1>
                                 <1> chs_read_error_retn:
133
134 0000F8C0 F9
                                 <1>
                                          stc
135
                                 <1>
                                           ;retn
136 0000F8C1 EB69
                                 <1>
                                           jmp short update_drv_error_byte
137
                                 <1>
138
                                 <1> ;chs_write_sectors: ; read or write
                                         \overline{\phantom{a}};; (\overline{\sharp} of sectors to read is less than remaining sectors on the track)
139
                                 <1>
140
                                 <1>
                                           ;mov [sector_count], al
141
                                 <1>
                                          ;jmp short chs_read_sectors
                                 <1>
142
143
                                 <1> chs_read_ok:
                                       ;; 23/02/2016
144
                                 <1>
145
                                 <1>
                                          ;movzx edx, byte [sector_count] ; sector count (<= spt)</pre>
146
                                 <1>
                                                   ecx, edx ; remaining sector count
                                          ;jna short update_drv_error_byte
147
                                 <1>
                                          ;add eax, edx ; next disk sector
148
                                 <1>
                                          ;shl edx, 9; 512 * sector count
149
                                 <1>
150
                                 <1>
                                          ;add ebx, edx; next buffer byte address
151
                                 <1>
                                           ;jmp
                                                     chs_read_next_sector
                                          ; 25/02/2016
152
                                 <1>
153 0000F8C3 40
                                 <1>
                                          inc eax ; next sector
154 0000F8C4 81C300020000
                                 <1>
                                          add
                                                ebx, 512
                                          loop chs_read_next_sector
155 0000F8CA E29D
                                 <1>
156 0000F8CC EB5E
                                 <1>
                                                short update_drv_error_byte
                                          jmp
157
                                 <1>
158
                                 <1> lba_write:
                                      ; 23/02/2016
                                 <1>
160 0000F8CE C605[75610100]1C
                                          mov byte [disk_rw_op], 1Ch; LBA write
                                 <1>
161 0000F8D5 EB07
                                 <1>
                                                short lba_rw
                                          jmp
162
                                 <1>
163
                                 <1> lba_read:
                                      ; 23/02/2016
164
                                 <1>
                                          ; 17/02/2016
165
                                 <1>
166
                                 <1>
                                         ; 14/02/2016
                                         ; 13/02/2016
167
                                 <1>
                                          ; 31/01/2016 (TRDOS 386 = TRDOS v2.0)
168
                                 <1>
169
                                 <1>
                                          ; 10/07/2015 (Retro UNIX 386 v1)
170
                                 <1>
171
                                 <1>
                                          ; INPUT -> EAX = Logical Block Address
172
                                                    ESI = Logical Dos Disk Table Offset (DRV)
                                 <1>
173
                                 <1>
                                                    ECX = Sector Count
174
                                 <1>
                                                    EBX = Destination Buffer
                                           ; OUTPUT ->
                                 <1>
176
                                 <1>
                                                    cf = 0 \text{ or } cf = 1
177
                                           ; (Modified registers: EAX, EBX, ECX, EDX)
                                 <1>
178
                                 <1>
                                          ; LBA read/write (with private LBA function)
179
                                 <1>
                                 <1>
                                           ;((Retro UNIX 386 v1 - DISK I/O code by Erdogan Tan))
180
181
                                 <1>
182
                                 <1>
                                           ; 23/02/2016
183
                                 <1>
184 0000F8D7 C605[75610100]1B
                                          mov byte [disk rw op], 1Bh; LBA read
                                 <1>
185
                                 <1>
                                 <1> lba_rw:
186
187
                                 <1>
                                          ; 17/02/2016
188 0000F8DE 57
                                 <1>
                                          push edi
                                 <1>
190 0000F8DF 890D[78610100]
                                 <1>
                                                 [sector_count], ecx; total sector (read) count
191
                                 <1>
                                          mov dl, [esi+LD PhyDrvNo]
192 0000F8E5 8A5602
                                 <1>
193
                                 <1>
                                          ; dl = physical drive number (0,1, 80h, 81h, 82h, 83h)
194
                                 <1>
                                 <1> lba_read_next:
195
                                           cmp ecx, 256
196 0000F8E8 81F900010000
                                 <1>
197 0000F8EE 7605
                                 <1>
                                                short lba_read_rsc
                                           jna
```

```
198 0000F8F0 B900010000
                          <1>
                                 mov ecx, 256; 17/02/2016
                           <1> lba read rsc:
200 0000F8F5 290D[78610100]
                          <1> sub [sector_count], ecx ; remain sectors
201
                           <1>
202 0000F8FB 89CF
                           <1>
                                       edi, ecx
                                  mov
203 0000F8FD 89C1
                           <1>
                                  mov
                                        ecx, eax; sector number/address
                           <1>
                                mov byte [retry_count], 4
205 0000F8FF C605[76610100]04 <1>
206
                           <1> lba_read_retry:
207 0000F906 89F8
                           <1> mov eax, edi
208
                           <1>
                               ; ecx = sector number
; al = sector count (0 - 255) /// (0 = 256)
; dl = drive number
209
                           <1>
210
                           <1>
211
                           <1>
                                 ; ebx = buffer offset
                           <1>
212
213
                           <1>
                               ; Function 1Bh = LBA read, 1Ch = LBA write
; 23/02/2016
mov ah, [disk_rw_op] ; 1Bh = LBA read, 1Ch = LBA write
call int13h
                           <1>
215
                           <1>
216 0000F908 8A25[75610100]
                           <1>
217 0000F90E E81A49FFFF
                          <1>
228
                           <1>
                           <1> lba_read_stc_retn:
229
230 0000F92A F9
                          <1>
                                  stc
231
                          <1> lba_read_retn:
232 0000F92B 5F
                           <1> pop edi
233
                          <1>
                          <1> update_drv_error_byte:
235 0000F92C 9C
                          <1> pushf
ebx, drv.error; 13/02/2016
256
                          <1>
                  257
258 0000F94F 89C8
                          <1> add eax, edi ; sector number (nex
<1> shl edi, 9 ; sector count * 512
<1> add ebx, edi ; next buffer offset
259 0000F951 01F8
                                        eax, edi ; sector number (next)
260 0000F953 C1E709
261 0000F956 01FB
262
                          <1>
ecx, [sector_count] ; remaining sectors
                                        ecx, ecx
                                        short lba_read_next
266 0000F962 EBC7
                          <1>
                                       short lba_read_retn
                                  jmp
267
                           <1>
268
                          <1> lba_read_reset:
                          269 0000F964 B40D
270 0000F966 E8C248FFFF
                          <1>
                                   call int13h
                          ; al = ? (changed)

1>; ah = error code

1> jnc short lba_read_retry

1> jmp short lba_read_retr
271
272
273 0000F96B 7399
274 0000F96D EBBC
                             %include 'trdosk8.s' ; 24/01/2016
2312
  2
                           <1> ; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.2) - MAIN PROGRAM : trdosk8.s
  3
                           <1> ; ------
                           <1>; Last Update: 03/08/2020
  5
                           <1> ; Beginning: 24/01/2016
                           <1> ; ------
  7
  8
                           <1>; Assembler: NASM version 2.11 (trdos386.s)
  9
 10
                           <1>; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
 11
                           <1>; u0.s (20/11/2015), u4.s (14/10/2015)
                           12
                           <1>; Derived from TRDOS Operating System v1.0 (8086) source code by Erdogan Tan
 1.3
 14
                           <1>; TRDOS2.ASM (09/11/2011)
                           15
 16
                           <1>; DIR.ASM (c) 2004-2011 Erdogan TAN [07/01/2004] Last Update: 09/10/2011
 17
                           <1>
                           <1> set run sequence:
 18
 19
                                  ; 23/12/2016
 20
                           <1>
                                   ; 10/06/2016
                                   ; 22/05/2016
 21
                           <1>
                                   ; 20/05/2016
 22
                           <1>
                                  ; 19/05/2016 - TRDOS 386 (TRDOS v2.0)
 2.3
                           <1>
                                  ; TRDOS 386 feature only !
 24
                           <1>
 25
                           <1>
 26
                           <1>
                                   ; INPUT ->
 27
                           <1>
                                   ; AL = process number (next process)
```

```
29
                                  <1>
                                                  this process must be added to run sequence
 30
                                  <1>
 31
                                  <1>
                                                  [u.pri] = priority of present process
 32
                                  <1>
 33
                                  <1>
                                                  DL = priority (queue)
                                                      0 = background (low) ; run on background
 34
                                  <1>
                                                       1 = regular (normal) ; run as regular
 35
                                  <1>
                                                                          ; run for event
 36
                                  <1>
                                                       2 = event (high)
 37
                                  <1>
 38
                                  <1>
                                                  1) If the requested process is already running:
 39
                                  <1>
                                                     a) If present priority is high ([u.pri]=2)
 40
                                  <1>
                                                        and requested priority is also high,
                                                         there is nothing to do! Because it has been
 41
                                  <1>
                                  <1>
 42
                                                        done already (before this attempt).
 43
                                  <1>
                                                     b) If present priority is high ([u.pri]=2)
 44
                                  <1>
                                                        and requested priority is not high, there is
 45
                                  <1>
                                                        nothing to do! Because, it's current
 46
                                  <1>
                                                        run queue is unspecified, here. (It may be in
 47
                                  <1>
                                                        a waiting list or in a run queue; if the new
 48
                                  <1>
                                                       priority would be used to add it to relavant
                                                            run queue, this would be wrong, unnecessary
 49
                                  <1>
                                             ;
                                                        and destabilizing duplication!)
 50
                                  <1>
 51
                                  <1>
                                                     c) If present priority is not high ([u.pri]<2)</pre>
 52
                                  <1>
                                             ;
                                                            and requested priority is high (event),
 53
                                  <1>
                                                        process will be added to present priority's
                                  <1>
                                                        run queue and then, priority will be changed
 55
                                  <1>
                                                        to high ([u.pri]=2).
 56
                                  <1>
                                                     d) If present priority is not high ([u.pri]<2)</pre>
 57
                                  <1>
                                                        and requested priority is not high, [u.pri]
 58
                                  <1>
                                                        value will be changed. There is nothing to do
 59
                                  <1>
                                                         in addition. (The new priority value will be
 60
                                  <1>
                                                        used by 'tswap/tswitch' procedure at 'sysret'
 61
                                  <1>
                                                         or 'sysrele' stage.)
                                  <1>
 62
 63
                                  <1>
                                                  2) If the requested process is not running:
 64
                                  <1>
                                                     a) If requested priority of the requested
 65
                                  <1>
                                                         (next) process is high (event) and priority
 66
                                  <1>
                                                         of present process is not high, the requested
                                                        process will be added to ('runq_event') high
 67
                                  <1>
 68
                                  <1>
                                                        priority run queue and then present (running)
 69
                                  <1>
                                                        process will be stopped (swapped/switched out)
                                                        immediately if it is in user mode, or it's
 70
                                  <1>
                                                        [u.quant] value will be reset to 0 and (then)
 71
                                  <1>
                                                         it will be stopped at 'sysret' stage.
 72
                                  <1>
 73
                                  <1>
                                                     b) If requested priority of the requested
 74
                                  <1>
                                                        (next) process is high (event) and priority
 75
                                  <1>
                                                        of present process is also high, the requested
 76
                                  <1>
                                                        process will be added to ('runq_event') high
 77
                                  <1>
                                                        priority run queue and present (running)
                                                        process will be allowed to run until it's
 78
                                  <1>
 79
                                  <1>
                                                        time quantum will be elapsed ([u.quant]=0).
 80
                                  <1>
                                                     c) If requested priority of the requested
 81
                                  <1>
                                                        (next) process is not high ('run for event'),
 82
                                  <1>
                                                        there is nothing to do. Because, it's current
 83
                                  <1>
                                                        run queue is unspecified, here. (It may be in
 84
                                  <1>
                                                        a waiting list or in a run queue; if the new
 85
                                  <1>
                                                         priority would be used to add it to relavant
 86
                                  <1>
                                                            run queue, this would be wrong, unnecessary
 87
                                  <1>
                                                        and destabilizing duplication!)
 88
                                  <1>
                                           ; OUTPUT ->
 89
                                  <1>
 90
                                  <1>
                                                  none
 91
                                  <1>
 92
                                  <1>
                                                  [u.pri] = priority of present process
 93
                                  <1>
 94
                                  <1>
                                                  cf = 1, if the request could not be fulfilled.
 95
                                  <1>
 96
                                  <1>
                                                  NOTE:
 97
                                  <1>
                                                         * Processes in 'run as regular' queue can run
 98
                                  <1>
                                                       if there is no process in 'run for event' queue
 99
                                  <1>
                                                        ('run for event' processes have higher priority)
                                                     ^{\star} When [u.quant] time quantum of a process is
100
                                  <1>
101
                                  <1>
                                                      elapsed, it's high priority ('run for event')
                                  <1>
102
                                                       status will be disabled, it can be run in sequence
103
                                  <1>
                                                       of it's actual run queue.
                                                     * A 'run on background' process will always be
104
                                  <1>
105
                                  <1>
                                                       sequenced in 'run on background' (low priority)
106
                                  <1>
                                                       queue, it can run only when other priority queues
107
                                                       are empty. (idle time processes, e.g. printing)
                                  <1>
108
                                  <1>
                                            ; Modified registers: eax, ebx, edx
109
                                  <1>
110
                                  <1>
111
                                  <1>
112
                                  <1> srunseq_0:
113 0000F96F 3A05[B3030300]
                                 <1>
                                           cmp
                                                    al, [u.uno] ; same process ?
114 0000F975 750C
                                            jne short srunseq_2 ; no
                                 <1>
115
                                 <1>
116 0000F977 8A25[A9030300]
                                 <1>
                                                  ah, [u.pri] ; present/current priority
117 0000F97D 80FC02
                                                  ah, 2 ; 'run for event' priority level
                                 <1>
                                           cmp
                                                  short srunseq_6 ; no
118 0000F980 7221
                                 <1>
119
                                  <1>
                                 <1> srunseq_1:
120
121
                                 <1>
                                          ; there is nothing to do!
122 0000F982 C3
                                 <1>
123
                                  <1>
                                  <1> srunseq_2:
124
                                           ;; this not necessary ! 23/12/2016
125
                                  <1>
                                             ;;cmp al, nproc ; number of processes = 16
;inb short srunseg 5 ; error ! invalid process r
126
                                  <1>
127
                                  <1>
                                                                      ; error ! invalid process number
                                            ;;jnb short srunseq_5
128
                                  <1>
129
                                  <1>
                                           ; dl = priority
                                           cmp dl, 2
130 0000F983 80FA02
                                 <1>
                                                              ; event queue
131 0000F986 72FA
                                  <1>
                                                  short srunseq_1; requested process is not present
                                  <1>
132
                                                            ; process and priority of requested
```

```
; process is not high (event),
133
                                  <1>
134
                                  <1>
                                                                 ; there is nothing to do!
135
                                  <1>
136
                                  <1>
                                            ; requested process is not present process
                                  <1>
                                            ; & priority of requested process is high
137
138 0000F988 3A15[A9030300]
                                  <1>
                                                  dl, [u.pri] ; priority of present process
                                             cmp
                                                   short srunseq 3 ; is high, also
139 0000F98E 7606
                                  <1>
                                            jna
                                  <1>
140
141
                                  <1>
                                            ; present process will be swapped/switched out
142 0000F990 FE05[51650100]
                                  <1>
                                                  byte [p_change] ; 1
                                            inc
143
                                  <1>
                                  <1> srunseq_3:
144
145
                                            ; add process to 'runq_event' queue for new event
                                  <1>
146 0000F996 BB[52030300]
                                  <1>
                                                  ebx, runq_event ; high priority run queue
                                  <1>
147
148
                                  <1> srunseq_4:
                                  <1>
                                            ; al = process number
                                            ; ebx = run queue
                                  <1>
150
151 0000F99B E881EDFFFF
                                  <1>
                                            call putlu
152 0000F9A0 C3
                                  <1>
                                            retn
153
                                  <1>
                                  <1> srunseq_5:
155 0000F9A1 F5
                                  <1>
                                            cmc
156 0000F9A2 C3
                                  <1>
                                            retn
157
                                  <1>
158
                                  <1> srunseq_6:
159
                                  <1>
                                            ; present priority of the process is not high
160
                                  <1>
161 0000F9A3 8815[A9030300]
                                  <1>
                                                   [u.pri], dl ; new priority
                                                              ; (will be used by 'tswap')
162
                                  <1>
163
                                  <1>
164 0000F9A9 80FA02
                                  <1>
                                                   dl, 2
                                                                ; high priority ?
                                            cmp
165 0000F9AC 72F3
                                  <1>
                                            jb
                                                   short srunseq_5; no, there is nothing to do
                                  <1>
                                                                ; in addition
166
167
                                  <1>
                                            ; process must be added to relevant run queue, here!
168
                                  <1>
169
                                  <1>
                                            ; (new priority is high/event priority and process
170
                                  <1>
                                            ; will not be added to a run queue by 'tswap')
171
                                  <1>
172 0000F9AE BB[54030300]
                                  <1>
                                                   ebx, runq_normal; 'run as regular' queue
                                            mov
173
                                  <1>
174 0000F9B3 20E4
                                  <1>
                                            and
                                                   ah, ah ; previous value of [u.pri]
175 0000F9B5 75E4
                                  <1>
                                            jnz
                                                   short srunseq_4
176
                                  <1>
177 0000F9B7 43
                                  <1>
                                            inc
                                                   ebx
178 0000F9B8 43
                                  <1>
                                            inc
                                                   ebx
179
                                  <1>
                                            ; ebx = runq_background ; 'run on backgroud' queue
180
                                  <1>
181 0000F9B9 EBE0
                                  <1>
                                            jmp
                                                  short srunseq 4
182
                                  <1> clock:
183
                                  <1>
                                            ; 23/05/2016
184
                                  <1>
                                            ; 22/05/2016
185
                                  <1>
                                            ; 20/05/2016
                                  <1>
                                            ; 19/05/2016 - TRDOS 386 (TRDOS v2.0)
186
187
                                  <1>
                                            ; 14/05/2015 - 14/10/2015 (Retro UNIX 386 v1)
188
                                  <1>
                                            ; 07/12/2013 - 10/04/2014 (Retro UNIX 8086 v1)
                                  <1>
190 0000F9BB 803D[A8030300]00
                                  <1>
                                            cmp
                                                   byte [u.quant], 0
191 0000F9C2 772C
                                  <1>
                                            jа
                                                   short clk_1
192
                                  <1>
193 0000F9C4 803D[B3030300]01
                                  <1>
                                                     byte [u.uno], 1 ; /etc/init ? (for Retro UNIX 8086 & 386 v1)
                                            cmp
194
                                  <1>
                                                                 ; MainProg (Kernel's Command Interpreter)
                                  <1>
195
                                                                 ; for TRDOS 386.
196 0000F9CB 7623
                                  <1>
                                                   short clk_1; yes, do not swap out
                                            jna
                                  <1>
197
198 0000F9CD 803D[5B030300]FF
                                  <1>
                                                    byte [sysflg], OFFh; user or system space?
                                             cmp
199 0000F9D4 7520
                                  <1>
                                                   short clk 2
                                                                    ; system space (sysflg <> OFFh)
                                            jne
200
                                  <1>
201 0000F9D6 66833D[AA030300]00 <1>
                                                   word [u.intr], 0
                                            cmp
202 0000F9DE 7616
                                  <1>
                                                   short clk_2
                                            jna
203
                                  <1>
204
                                  <1>
                                            ; 23/05/2016
205 0000F9E0 803D[52650100]00
                                                   byte [multi_tasking], 0
                                  <1>
                                            cmp
206 0000F9E7 760D
                                  <1>
                                            jna
                                                   short clk_2
                                  <1>
207
208 0000F9E9 FE05[51650100]
                                  <1>
                                            inc
                                                   byte [p_change] ; it is time to change running process
209 0000F9EF C3
                                  <1>
                                            retn
                                  <1> clk_1:
210
211 0000F9F0 FE0D[A8030300]
                                  <1>
                                            dec
                                                   byte [u.quant]
                                  <1> clk_2:
212
213 0000F9F6 C3
                                  <1>
                                            retn
                                                  ; return to (hardware) timer interrupt routine
                                  <1>
                                  <1>; 12/10/2017
215
                                  <1> ; 15/01/2017
216
217
                                  <1>; 14/01/2017
                                  <1> ; 07/01/2017
218
219
                                  <1>; 02/01/2017
220
                                  <1> ; 17/08/2016
221
                                  <1>; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
                                  <1> int34h: ; \#IOCTL\# (I/O port access support for ring 3)
222
                                  <1>; 23/05/2016
223
224
                                          ; 20/06/2016
                                            ; 29/04/2016 - TRDOS 386 (TRDOS v2.0)
225
                                  <1>
226
                                  <1>
                                           ; INPUT ->
227
                                  <1>
                                                   AH = 0 -> read port (physical IO port) -byte-
228
                                  <1>
                                                   AH = 1 -> write port (physical IO port) -byte-
229
                                  <1>
                                                        AL = data byte
230
                                  <1>
231
                                  <1>
                                                   AH = 2 -> read port (physical IO port) -word-
                                                   AH = 3 -> write port (physical IO port) -word-
232
                                  <1>
                                                         BX = data word
233
                                  <1>
                                                  AH = 4 \rightarrow read port (physical IO port) -dword-

AH = 5 \rightarrow write port (physical IO port) -dword-
234
                                  <1>
235
                                  <1>
                                                          EBX = data dword
236
                                  <1>
237
                                  <1>
                                                   ; 12/10/2017
```

```
238
                               <1>
                                              AH = 6 -> read port (physical IO port) twice -byte-
                                              AH = 7 -> write port (physical IO port) twice -byte-
239
                               <1>
240
                               <1>
                                        ;
                                                    BX = data word
241
                               <1>
242
                               <1>
                                              DX = Port number (<= 0FFFFh)
                                        ;
243
                               <1>
                                        ; OUTPUT ->
                               <1>
245
                               <1>
                                              AL = data byte (in al, dx)
246
                               <1>
                                              AX = data word (in ax, dx)
                                              EAX = data dword (in eax, dx)
247
                               <1>
248
                               <1>
                               <1>
                                              (ECX = actual TRANSFER COUNT for string functions)
249
250
                               <1>
251
                               <1>
252
                               <1>
                                        ; Modified registers: EAX
253
                               <1>
                               <1>
255
                                        ;cmp ah, 5
                               <1>
                                        ;ja short int34h 5 ; invalid function !
256
                              <1>
257
                              <1>
                                        ; 12/10/2017
258
                              <1>
259 0000F9F7 80FC07
                                        cmp ah, 7
                              <1>
260 0000F9FA 7743
                                              short int34h_5; invalid function!
                              <1>
                                        jа
                              <1>
                                       ;; 15/01/2017
                              <1>
262
                                        ; 14/01/2017
263
                              <1>
264
                              <1>
                                        ; 02/01/2017
265
                              <1>
                                        ;;mov byte [ss:intflg], 34h ; IOCTL interrupt
266 0000F9FC FB
                              <1>
267
                              <1>
268
                              <1>
                                        ;sti ; enable interrupts
                                             byte [esp+8], 111111110b ; clear carry bit of eflags register
269 0000F9FD 80642408FE
                              <1>
                                        and
270
                              <1>
271 0000FA02 80FC01
                              <1>
                                        cmp
                                              ah, 1
272 0000FA05 7205
                              <1>
                                              short int34h_0
                                        jb
273 0000FA07 7705
                              <1>
                                        jа
                                              short int34h_1
274
                              <1>
275 0000FA09 EE
                              <1>
                                        out dx, al
276
                              <1>
                                        ;iretd
                                        jmp short int34h_iret
277 0000FA0A EB01
                              <1>
278
                              <1>
279
                              <1> int34h 0:
280 0000FA0C EC
                              <1>
                                              al, dx
                                        in
281
                              <1>
                                        ;iretd
                              <1> int34h_iret:
282
                                     ;cli ; 07/01/2017
283
                              <1>
                                        ;; 15/01/2017
                              <1>
285
                              <1>
                                       ;;mov byte [ss:intflg], 0 ; reset
286 0000FA0D CF
                              <1>
                                        iretd
287
                              <1>
                              <1> int34h 1:
288
289 0000FA0E F6C401
                              <1>
                                   test ah, 1
290 0000FA11 7516
                              <1>
                                        jnz
                                              short int34h_3; out
291
                              <1>
292
                              <1>
                                        ; in
293 0000FA13 80FC02
                              <1>
                                        cmp
                                              ah, 2
294 0000FA16 7707
                              <1>
                                              short int34h 2
                                        jа
295
                              <1>
296 0000FA18 6689D8
                              <1>
                                        mov
                                              ax, bx
297 0000FA1B 66ED
                              <1>
                                       in
                                              ax, dx
298
                              <1>
                                        ;iretd
299 0000FA1D EBEE
                              <1>
                                             short int34h_iret
                                        jmp
300
                              <1>
301
                              <1> int34h 2:
302 0000FA1F 80FC04
                              <1>
                                   cmp
                                              ah, 4
                                              short int34h_7
303 0000FA22 772C
                              <1>
                                        jа
                                                                ; 12/10/2017
                              <1>
                                       ; ah = 4
305 0000FA24 89D8
                              <1>
                                       mov
                                             eax, ebx
306 0000FA26 ED
                              <1>
                                        in
                                              eax, dx
307
                              <1>
                                        ;iretd
308 0000FA27 EBE4
                              <1>
                                        jmp
                                             short int34h_iret
309
                              <1>
                              <1> int34h_3:
310
                                              ah, 3
311 0000FA29 80FC03
                              <1>
                                        cmp
                                              short int34h 4
312 0000FA2C 7707
                              <1>
                                        jа
313
                              <1>
314 0000FA2E 6689D8
                              <1>
                                       mov
                                              ax, bx
                                        out
315 0000FA31 66EF
                              <1>
                                             dx, ax
                              <1>
                                        ;iretd
317 0000FA33 EBD8
                              <1>
                                             short int34h_iret
                                        jmp
318
                               <1>
                               <1> int34h 4:
319
320 0000FA35 80FC05
                                             ah, 5
                              <1>
                                        cmp
321 0000FA38 770B
                              <1>
                                        ja short int34h_6 ; 12/10/2017
                                       ; ah = 5
                              <1>
322
323 0000FA3A 89D8
                              <1>
                                       mov eax, ebx
324 0000FA3C EF
                             <1>
                                      out dx, eax
                             <1>
<1>
                                       ;iretd
325
326 0000FA3D EBCE
                                       jmp short int34h_iret
327
                              <1>
                              <1> int34h_5:
328
329 0000FA3F 804C240801
                                    or byte [esp+8], 1; set carry bit of eflags register
                              <1>
330 0000FA44 CF
                              <1>
                                        iretd
331
                              <1>
                                        ; 12/10/2017
332
                              <1>
                              <1> int34h_6:
333
334 0000FA45 6689D8
                             <1> mov
                                              ax, bx
335 0000FA48 EE
                             <1>
                                        out dx, al
336 0000FA49 EB00
                              <1>
                                       jmp
                                             short $+2
                                      xchg ah, al
337 0000FA4B 86E0
                             <1>
338 0000FA4D EE
                             <1>
                                      out dx, al
                             <1> ;xchg al, ah <1> ;iretd
339
340
                             <1> ;iretd
<1> jmp short int34h_8
341 0000FA4E EB06
342
                              <1> int34h_7:
```

```
343 0000FA50 EC
                               <1>
                                        in al, dx
344 0000FA51 EB00
                               <1>
                                          jmp
                                               short $+2
345 0000FA53 88C4
                               <1>
                                               ah, al
                                         mov
346 0000FA55 EC
                               <1>
                                         in
                                                al, dx
347
                               <1> int34h 8:
348 0000FA56 86C4
                                       xchg al, ah
                               <1>
349 0000FA58 CF
                               <1>
                                          iretd
350
                                <1>
351
                                <1>
352
                                <1> INT4Ah:
                                      ; 24/01/2016
353
                                <1>
354
                                         ; this procedure will be called by 'RTC INT' (in 'timer.s')
                                <1>
355 0000FA59 C3
                                <1>
                                         retn
356
                                <1>
                                <1> ; u0.s
357
                                <1>; Retro UNIX 386 v1 Kernel (v0.2) - SYSO.INC
358
                                <1> ; Last Modification: 20/11/2015
360
                                <1>
361
                                <1> com2_int:
                                <1> ; 07/11/2015
362
                                        ; 24/10/2015
363
                                <1>
                                       ; 23/10/2015
; 14/03/2015 (Retro UNIX 386 v1 - Beginning)
; 28/07/2014 (Retro UNIX 8086 v1)
364
                                <1>
365
                                <1>
                                <1>
                                        ; < serial port 2 interrupt handler >
                                <1>
367
368
                                <1>
                                        mov
369 0000FA5A 890424
                               <1>
                                               [esp], eax; overwrite call return address
370
                                <1>
                                       ;push eax
371 0000FA5D 66B80900
                               <1>
                                         mov ax, 9
372 0000FA61 EB07
                               <1>
                                               short comm_int
                                         jmp
373
                               <1> com1_int:
                                <1> 7/11/2015
374
                                         ; 24/10/2015
375
                               <1>
376 0000FA63 890424
                               <1>
                                       mov [esp], eax; overwrite call return address
                                       ; 23/10/2015
377
                                <1>
378
                               <1>
                                         ;push eax
                                       mov ax, 8
379 0000FA66 66B80800
                               <1>
380
                                <1> comm_int:
                                     ; 20/11/2015
381
                                <1>
382
                                <1>
                                         ; 18/11/2015
                                       ; 17/11/2015
; 16/11/2015
; 09/11/2015
; 08/11/2015
383
                                <1>
384
                                <1>
                                <1>
385
386
                                <1>
                                       ; 07/11/2015
; 06/11/2015 (serial4.asm, 'serial')
; 01/11/2015
387
                                <1>
388
                                <1>
                                <1>
                                        ; 26/10/2015
; 23/10/2015
                                <1>
390
391
                                <1>
392 0000FA6A 53
                               <1>
                                        push ebx
393 0000FA6B 56
                               <1>
                                         push esi
                                         push edi
394 0000FA6C 57
                                <1>
395 0000FA6D 1E
                               <1>
                                         push ds
396 0000FA6E 06
                               <1>
                                         push es
                                         ; 18/11/2015
                                <1>
398 0000FA6F 0F20DB
                               <1>
                                         mov
                                               ebx, cr3
                                         push ebx; ****
399 0000FA72 53
                               <1>
400
                               <1>
                                         push ecx; ***
401 0000FA73 51
                                <1>
                                         push edx; **
402 0000FA74 52
                               <1>
403
                                <1>
404 0000FA75 BB10000000
                                <1>
                                         mov
                                                ebx, KDATA
405 0000FA7A 8EDB
                                <1>
                                               ds, bx
                                         mov
406 0000FA7C 8EC3
                                <1>
                                               es, bx
                                         mov
                                <1>
                                         ;
408 0000FA7E 8B0D[C0580100]
                                        mov ecx, [k_page_dir]
mov cr3, ecx
                                <1>
409 0000FA84 0F22D9
                               <1>
                                       ; 20/11/2015
410
                                <1>
411
                                <1>
                                         ; Interrupt identification register
412 0000FA87 66BAFA02
                               <1>
                                         mov dx, 2FAh; COM2
413
                                <1>
414 0000FA8B 3C08
                                               al, 8
                                <1>
                                         cmp
415 0000FA8D 7702
                                <1>
                                         jа
                                                short com_i0
416
                                <1>
                                        ; 20/11/2015
417
                                <1>
                                         ; 17/11/2015
418
                                <1>
419
                                <1>
                                         ; 16/11/2015
                                        ; 15/11/2015
420
                                <1>
                                         ; 24/10/2015
421
                                <1>
422
                                <1>
                                         ; 14/03/2015 (Retro UNIX 386 v1 - Beginning)
                                         ; 28/07/2014 (Retro UNIX 8086 v1)
423
                                <1>
                                          ; < serial port 1 interrupt handler >
424
                                <1>
                                <1>
426 0000FA8F FEC6
                                <1>
                                               dh ; 3FAh ; COM1 Interrupt id. register
                                <1> com i0:
427
                                          ;push eax ; *
428
                                <1>
                                          ; 07/11/2015
                                <1>
430 0000FA91 A2[2A590100]
                                          mov byte [ccomport], al
                               <1>
431
                                <1>
                                         ; 09/11/2015
432 0000FA96 0FB7D8
                               <1>
                                         movzx ebx, ax; 8 or 9
433
                                <1>
                                         ; 17/11/2015
434
                                <1>
                                         ; reset request for response status
435 0000FA99 88A3[20590100]
                                <1>
                                          mov [ebx+req_resp-8], ah; 0
436
                                <1>
                                          ; 20/11/2015
437
                                <1>
438 0000FA9F EC
                                <1>
                                          in
                                               al, dx
                                                           ; read interrupt id. register
                                                       ; I/O DELAY
; received data available?
439 0000FAA0 EB00
                                <1>
                                          JMP
                                               $+2
440 0000FAA2 2404
                                <1>
                                          and al, 4
441 0000FAA4 7470
                                               short com eoi; (transmit. holding reg. empty)
                                <1>
                                          jz
442
                               <1>
                                          ;
                                          ; 20/11/2015
443
                                <1>
444 0000FAA6 80EA02
                                <1>
                                          sub dl, 3FAh-3F8h; data register (3F8h, 2F8h)
                                          in
445 0000FAA9 EC
                                <1>
                                               al, dx ; read character
                                          ;JMP $+2
                                                            ; I/O DELAY
446
                                <1>
447
                                <1>
                                          ; 08/11/2015
```

```
; 07/11/2015
448
                                 <1>
449 0000FAAA 89DE
                                  <1>
                                           mov esi, ebx
450 0000FAAC 89DF
                                 <1>
                                            mov
                                                  edi, ebx
451 0000FAAE 81C6[24590100] <1>
                                            add esi, rchar - 8; points to last received char
452 0000FAB4 81C7[26590100]
                                 <1>
                                            add edi, schar - 8; points to last sent char
453 0000FABA 8806
                                 <1>
                                            mov
                                                  [esi], al ; received char (current char)
                                 <1>
                                           ; query
455 0000FABC 20C0
                                 <1>
                                          and al, al
456 0000FABE 7527
                                 <1>
                                           jnz
                                                  short com i2
457
                                 <1>
                                            ; response
458
                                 <1>
                                            ; 17/11/2015
                                            ; set request for response status
459
                                  <1>
460 0000FAC0 FE83[20590100]
                                 <1>
                                            inc byte [ebx+req_resp-8] ; 1
461
                                 <1>
                                          add dx, 3FDh-3F8h; (3FDh, 2FDh)
462 0000FAC6 6683C205
                                 <1>
                                            in al, dx ; read line status register
JMP $+2 ; I/O DELAY
and al, 20h ; transmitter holding reg. empty?
jz short com_eoi ; no
463 0000FACA EC
                                 <1>
464 0000FACB EB00
                                 <1>
465 0000FACD 2420
                                 <1>
                                           jz short com_eoi ; no
mov al, 0FFh ; response
                                mov al, 0FFh ; response

<1> sub dx, 3FDh-3F8h ; data port (3F8h, 2F8h)

<1> out dx, al ; send on serial port

<1> ; 17/11/2015

<1> cmp byte [edi], 0 ; query ? (schar)

<1> jne short com_il ; no

<1> mov [edi], al . OFFL (schar)
466 0000FACF 7445
                                 <1>
467 0000FAD1 B0FF
                             <1>
468 0000FAD3 6683EA05
469 0000FAD7 EE
470
471 0000FAD8 803F00
472 0000FADB 7502
473 0000FADD 8807
                                 <1> com i1:
                                        ; 17/11/2015
475
                                 <1>
476
                                  <1>
                                            ; reset request for response status (again)
477 0000FADF FE8B[20590100]
                                            dec byte [ebx+req_resp-8]; 0
                                 <1>
                                 <1>
478 0000FAE5 EB2F
                                            jmp short com_eoi
479
                                  <1> com i2:
                                 <1>
                                           ; 08/11/2015
480
                                            cmp al, OFFh ; (response ?)
je short com_i3 ; (check for response signal)
481 0000FAE7 3CFF
                                 <1>
                                       je short com_i3
; 07/11/2015
cmp al, 04h
jne short com_i4
482 0000FAE9 7417
                                 <1>
483
                                 <1>
484 0000FAEB 3C04
                                 <1>
485 0000FAED 751C
                                 <1>
486
                                  <1>
                                            ; EOT = 04h (End of Transmit) - 'CTRL + D'
487
                                  <1>
                                            ; (an EOT char is supposed as a ctrl+brk from the terminal)
                                          ; 08/11/2015
488
                                 <1>
                                                  ; ptty -> tty 0 to 7 (pseudo screens)
                                  <1>
490 0000FAEF 861D[EE580100]
                                            xchg bl, [ptty] ; tty number (8 or 9)
                                 <1>
491 0000FAF5 E84C69FFFF
                                 <1>
                                        call ctrlbrk
                                           xchg [ptty], bl ; (restore ptty value and BL value)
;mov al, 04h ; EOT
492 0000FAFA 861D[EE580100]
                                  <1>
493
                                  <1>
                                            ; 08/11/2015
                                  <1>
495 0000FB00 EB09
                                  <1>
                                            jmp short com_i4
496
                                  <1> com_i3:
497
                                  <1> ; 08/11/2015
498
                                  <1>
                                          ; If OFFh has been received just after a query
499
                                  <1>
                                           ; (schar, ZERO), it is a response signal.
                                         ; (schar, ZER; 17/11/2015
500
                                 <1>
501 0000FB02 803F00
                                 <1>
                                            cmp byte [edi], 0; query ? (schar)
                                         ja short com_i4; no
; reset query status (sc
mov [edi], al; 0FFh
502 0000FB05 7704
                                 <1>
503
                                 <1>
                                            ; reset query status (schar)
504 0000FB07 8807
                                 <1>
                                            mov [edi], al; OFFh
505 0000FB09 FEC0
                                            inc al; 0
                                 <1>
506
                                  <1> com_i4:
                                 <1> - ; 27/07/2014
507
508
                                 <1>
                                            ; 09/07/2014
                                         shl bl, 1 add ebx, ttychr
509 0000FB0B D0E3
                                  <1>
510 0000FB0D 81C3[F0580100]
                                  <1>
                                          ; 23/07/2014 (always overwrite)
511
                                  <1>
512
                                  <1>
                                            ;;cmp word [ebx], 0
513
                                  <1>
                                            ;;ja short com_eoi
                                  <1>
515 0000FB13 668903
                                  <1>
                                            mov [ebx], ax ; Save ascii code
516
                                  <1>
                                                    ; scan code = 0
517
                                  <1> com_eoi:
518
                                  <1>
                                          ;mov al, 20h
                                            ;out 20h, al ; end of interrupt
519
                                  <1>
                                  <1>
520
                                            ; 07/11/2015
521
                                  <1>
                                            ;pop eax ; *
mov al, byte [ccomport] ; current COM port
                                  <1>
522
523 0000FB16 A0[2A590100]
                                  <1>
                                            ; al = tty number (8 or 9)
                                  <1>
525 0000FB1B E85E010000
                                  <1>
                                            call wakeup
526
                                  <1> com_iret:
527
                                  <1> ; 23/10/2015
528 0000FB20 5A
                                            pop edx; **
                                  <1>
                                            pop
                                                   ecx ; ***
529 0000FB21 59
                                  <1>
                                            ; 18/11/2015
530
                                  <1>
531
                                  <1>
                                            ;pop eax ; ****
532
                                  <1>
                                            ;mov cr3, eax
533
                                  <1>
                                            ;jmp iiret
534 0000FB22 E9C20FFFFF
                                            jmp iiretp
                                  <1>
535
                                  <1>
536
                                  <1> ;iiretp: ; 01/09/2015
                                  <1>; 28/08/2015
537
538
                                  <1>;
                                            pop eax; (*) page directory
539
                                  <1>;
                                            mov
                                                  cr3, eax
                                  <1> ;iiret:
540
                                         ; 22/08/2014
                                  <1>;
541
                                            mov al, 20h; END OF INTERRUPT COMMAND TO 8259
542
                                  <1>;
543
                                  <1>;
                                            out
                                                  20h, al ; 8259 PORT
544
                                  <1>;
545
                                  <1>;
                                            pop
                                                  es
546
                                  <1>;
                                            pop
                                                   ds
                                  <1>;
547
                                            pop
                                                  edi
548
                                  <1>;
                                                  esi
549
                                  <1>;
                                            pop
                                                  ebx ; 29/08/2014
                                  <1>;
550
                                            pop
                                                  eax
                                  <1>;
551
552
                                  <1>
```

```
<1> sp_init:
                               <1> ; 07/11/2015
554
555
                               <1>
                                        ; 29/10/2015
                                       ; 26/10/2015
556
                               <1>
                                       ; 23/10/2015
557
                               <1>
558
                               <1>
                                        ; 29/06/2015
                                        ; 14/03/2015 (Retro UNIX 386 v1 - 115200 baud)
559
                                       ; 28/07/2014 (Retro UNIX 8086 v1 - 9600 baud)
                               <1>
560
561
                               <1>
                                        ; Initialization of Serial Port Communication Parameters
                                        ; (COM1 base port address = 3F8h, COM1 Interrupt = IRQ 4)
562
                               <1>
563
                               <1>
                                        ; (COM2 base port address = 2F8h, COM1 Interrupt = IRQ 3)
564
                                <1>
565
                               <1>
                                        ; ((Modified registers: EAX, ECX, EDX, EBX))
566
                               <1>
                                       ; INPUT: (29/06/2015)
                               <1>
567
                                         ; AL = 0 \text{ for COM1}
568
                               <1>
                                                 1 for COM2
569
                                              AH = Communication parameters
570
                               <1>
571
                               <1>
572
                               <1>
                                         ; (*) Communication parameters (except BAUD RATE):
                                            Bit 4 3 2 1 0
-PARITY-- STOP BIT -WORD LENGTH-
573
                               <1>
574
                                <1>
575
                                            this one --> 00 = none 0 = 1 bit 11 = 8 bits
                               <1>
                                         ;
576
                               <1>
                                                     01 = odd   1 = 2 bits   10 = 7 bits
577
                               <1>
                                                     11 = even
578
                               <1>
                                         ; Baud rate setting bits: (29/06/2015)
579
                               <1>
                                                    Retro UNIX 386 v1 feature only !
                                               Bit 7 6 5 | Baud rate
                               <1>
580
581
                               <1>
                                               value 0 0 0 | Default (Divisor = 1)
                               <1>
582
583
                               <1>
                                                     0 0 1 | 9600 (12)
                                                              0 | 19200 (6)
584
                                <1>
                                                     0
                                                          1
                                                        1 1 | 38400 (3)
585
                               <1>
                                                     0
586
                               <1>
                                                     1 0 0 | 14400 (8)
                                                         0 1 | 28800 (4)
1 0 | 57600 (2)
587
                               <1>
                                                     1
                                                        1
588
                               <1>
                                                     1
                                                             1 | 115200 (1)
                                <1>
                                                          1
590
                               <1>
591
                               <1>
                                         ; References:
                                         ; (1) IBM PC-XT Model 286 BIOS Source Code
592
                               <1>
593
                               <1>
                                              RS232.ASM --- 10/06/1985 COMMUNICATIONS BIOS (RS232)
                                        ; (2) Award BIOS 1999 - ATORGS.ASM
594
                                <1>
595
                               <1>
                                        ; (3) http://wiki.osdev.org/Serial_Ports
596
                               <1>
597
                               <1>
                                        ; Set communication parameters for COM1 (= 03h)
598
                               <1>
                                              ebx, com1p
599 0000FB27 BB[26590100]
                                                                 ; COM1 parameters
                               <1>
                                        mov
600 0000FB2C 66BAF803
                               <1>
                                        mov dx, 3F8h
                                                                 ; COM1
601
                               <1>
                                         ; 29/10/2015
602 0000FB30 66B90103
                               <1>
                                        mov cx, 301h ; divisor = 1 (115200 baud)
603 0000FB34 E86F000000
                              <1>
                                         call sp_i3 ; call A4
604 0000FB39 A880
                               <1>
                                         test al, 80h
                                        jz short sp_i0 ; OK..
; Error !
605 0000FB3B 7410
                              <1>
                                    ; Error ; mov dx, 3F8h sub dl, 5 · f
606
                              <1>
                                              ; Error !
                              <1>
608 0000FB3D 80EA05
                              <1>
                                              dl, 5 ; 3FDh -> 3F8h
                                        mov cx, 30Eh ; divisor = 12 (9600 baud)
609 0000FB40 66B90E03
                              <1>
610 0000FB44 E85F000000
                              <1>
                                         call sp_i3 ; call A4
611 0000FB49 A880
                               <1>
                                         test al, 80h
                                         jnz short sp_i1
612 0000FB4B 7508
                               <1>
613
                               <1> sp_i0:
614
                               <1>
                                          ; (Note: Serial port interrupts will be disabled here...)
                               <1>
615
                                           ; (INT 14h initialization code disables interrupts.)
                               <1>
617 0000FB4D C603E3
                               <1>
                                              byte [ebx], 0E3h; 11100011b
                                         mov
                                         call sp_i5; 29/06/2015
618 0000FB50 E8DC000000
                               <1>
                               <1> sp_i1:
                                     inc
620 0000FB55 43
                               <1>
                                              ebx
621 0000FB56 66BAF802
                               <1>
                                              dx, 2F8h
                                                                ; COM2
                                         mov
622
                              <1>
                                        ; 29/10/2015
623 0000FB5A 66B90103
                              <1>
                                         mov cx, 301h; divisor = 1 (115200 baud)
                                        call sp_i3; call A4 test al, 80h
624 0000FB5E E845000000
                               <1>
625 0000FB63 A880
                              <1>
                                       jz short sp_i2 ; OK..
626 0000FB65 7410
                              <1>
627
                               <1>
                                               ; Error !
628
                               <1>
                                        ;mov dx, 2F8h
                                    sub dl, 5 ; 2FDh -> 2F8h
629 0000FB67 80EA05
                              <1>
630 0000FB6A 66B90E03
                               <1>
                                        mov
                                              cx, 30Eh ; divisor = 12 (9600 baud)
                                         call sp i3 ; call A4
631 0000FB6E E835000000
                               <1>
632 0000FB73 A880
                               <1>
                                         test al, 80h
633 0000FB75 7530
                               <1>
                                         jnz short sp_i7
                                <1> sp_i2:
635 0000FB77 C603E3
                                        mov byte [ebx], 0E3h; 11100011b
                               <1>
                               <1> sp_i6:
                                       ;; COM2 - enabling IRQ 3
637
                               <1>
                                        ; 07/11/2015
638
                               <1>
                                        ; 26/10/2015
                               <1>
640 0000FB7A 9C
                              <1>
                                       pushf
641 0000FB7B FA
                              <1>
                                        cli
                              <1>
                           <1>
                                       mov
643 0000FB7C 66BAFC02
                                                               ; modem control register
                                              dx, 2FCh
644 0000FB80 EC
                                                                  ; read register
                               <1>
                                        in
                                               al, dx
                                                               ; I/O DELAY
645 0000FB81 EB00
                              <1>
                                        JMP
                                              $+2
646 0000FB83 0C08
                             <1>
                                              al, 8
                                                                ; enable bit 3 (OUT2)
                                              , wille back to register; I/O DELAY ; I/O DELAY ; interrupt enable register al, dx ; read register ; I/O DETAY ; I/O DETAY
                                      out
647 0000FB85 EE
                              <1>
                              <1>
648 0000FB86 EB00
                                        JMP
                        <1>
649 0000FB88 66BAF902
                                        mov
                              <1>
                                    in
650 0000FB8C EC
                                                                ; I/O DELAY ; receiver data interrupt enable and
651 0000FB8D EB00
                              <1>
                                        JMP
652
                              <1>
                                        or al, 1
                          <1>
653 0000FB8F 0C03
                                      or
                                              al, 3
                                                                ; transmitter empty interrupt enable
                                        out dx, al
JMP $+2
                                                                ; wri
; I/O DELAY
                              <1>
<1>
654 0000FB91 EE
                                        out
                                                                        ; write back to register
                                              al, 21h
$+2
655 0000FB92 EB00
                                                             ; read interrupt mask register
656 0000FB94 E421
                              <1>
657 0000FB96 EB00
                               <1>
                                        JMP
                                                                ; I/O DELAY
```

```
and al, 0F7h out 21h, al
                              <1>
658 0000FB98 24F7
                                                               ; enable IRQ 3 (COM2)
659 0000FB9A E621
                               <1>
                                                               ; write back to register
                               <1>
660
                                        ; 23/10/2015
                               <1>
662 0000FB9C B8[5AFA0000]
                               <1>
                                        mov eax, com2 int
663 0000FBA1 A3[79FC0000]
                               <1>
                                        mov
                                              [com2_irq3], eax
                                        ; 26/10/2015
                               <1>
665 0000FBA6 9D
                               <1>
                                        popf
666
                               <1> sp_i7:
667 0000FBA7 C3
                               <1>
                                        retn
668
                               <1>
669
                               <1> sp i3:
670
                               <1> ; A4:
                                               ;---- INITIALIZE THE COMMUNICATIONS PORT
671
                               <1>
                                        ; 28/10/2015
                                        inc dl ; 3F9h (2F9h); 3F9h, COM1 Interrupt enable register mov al, 0
672 0000FBA8 FEC2
                               <1>
673 0000FBAA B000
                              <1>
674 0000FBAC EE
                              <1>
                                              dx, al
                                                                ; disable serial port interrupt
                              <1>
675 0000FBAD EB00
                                                                ; I/O DELAY
                                        JMP
                                              $+2
676 0000FBAF 80C202
                                              dl, 2 ; 3FBh (2FBh); COM1 Line control register (3FBh)
                              <1>
                                        add
                                        mov al, 80h
677 0000FBB2 B080
                              <1>
678 0000FBB4 EE
                                                                 ; SET DLAB=1 ; divisor latch access bit
                              <1>
                                        out dx, al
                                        ;---- SET BAUD RATE DIVISOR
679
                               <1>
680
                              <1>
                                        ; 26/10/2015
681 0000FBB5 80EA03
                              <1>
                                        sub dl, 3 ; 3F8h (2F8h) ; register for least significant byte
                                                                ; of the divisor value
                               <1>
682
683 0000FBB8 88C8
                               <1>
                                              al, cl; 1
684 0000FBBA EE
                               <1>
                                        out dx, al
                                                                 ; 1 = 115200 baud (Retro UNIX 386 v1)
                                                                 ; 2 = 57600 baud
                               <1>
685
686
                               <1>
                                                                 ; 3 = 38400 baud
                                                                 ; 6 = 19200 baud
687
                               <1>
688
                               <1>
                                                                 ; 12 = 9600 baud (Retro UNIX 8086 v1)
689 0000FBBB EB00
                               <1>
                                        JMP
                                              $+2
                                                                 ; I/O DELAY
690 0000FBBD 28C0
                              <1>
                                        sub
                                              al, al
691 0000FBBF FEC2
                              <1>
                                              dl ; 3F9h (2F9h)
                                                                      ; register for most significant byte
                                                            ; of the divisor value
692
                               <1>
693 0000FBC1 EE
                               <1>
                                        out
                                               dx, al ; 0
694 0000FBC2 EB00
                               <1>
                                        JMP
                                              $+2
                                                                 ; I/O DELAY
695
                               <1>
                                        ;
                                              al, ch ; 3
696 0000FBC4 88E8
                               <1>
                                                                 ; 8 data bits, 1 stop bit, no parity
                                        ; and al, 1Fh; Bits 0,1,2,3,4
697
                              <1>
698 0000FBC6 80C202
                              <1>
                                        add
                                              dl, 2 ; 3FBh (2FBh); Line control register
699 0000FBC9 EE
                               <1>
                                        out
                                              dx, al
                                        JMP $+2
                                                                 ; I/O DELAY
700 0000FBCA EB00
                              <1>
701
                              <1>
                                        ; 29/10/2015
702 0000FBCC FECA
                                        dec dl ; 3FAh (2FAh); FIFO Control register (16550/16750)
                               <1>
703 0000FBCE 30C0
                               <1>
                                        xor
                                              al, al ; 0
                                                                 ; Disable FIFOs (reset to 8250 mode)
704 0000FBD0 EE
                              <1>
                                              dx, al
                                        out
705 0000FBD1 EB00
                                              $+2
                              <1>
                                        JMP
706
                               <1> sp_i4:
707
                               <1>; A18: ;---- COMM PORT STATUS ROUTINE
708
                               <1>
                                        ; 29/06/2015 (line status after modem status)
709 0000FBD3 80C204
                               <1>
                                        add dl, 4 ; 3FEh (2FEh); Modem status register
                              <1> sp_i4s:
710
711 0000FBD6 EC
                              <1>
                                        in
                                              al, dx
                                                                ; GET MODEM CONTROL STATUS
                                              $+2 ; I/O DELAY ah, al ; PUT IN (AH) FOR RETURN
712 0000FBD7 EB00
                              <1>
                                        JMP
                                              $+2
713 0000FBD9 88C4
                              <1>
                                        mov
                                        dec dl ; 3FDh (2FDh); POINT TO LINE STATUS REGISTER
714 0000FBDB FECA
                              <1>
                                                              ; dx = 3FDh for COM1, 2FDh for COM2
715
                              <1>
                                        in al, dx
716 0000FBDD EC
                               <1>
                                                                 ; GET LINE CONTROL STATUS
717
                               <1>
                                        ; AL = Line status, AH = Modem status
718 0000FBDE C3
                               <1>
719
                               <1>
720
                               <1> sp_status:
                                      ; 29/06/2015
721
                               <1>
722
                               <1>
                                        ; 27/06/2015 (Retro UNIX 386 v1)
723
                               <1>
                                        ; Get serial port status
724 0000FBDF 66BAFE03
                               <1>
                                        mov dx, 3FEh
                                                                 ; Modem status register (COM1)
                                                                 ; dh = 2 \text{ for COM2}(al = 1)
725 0000FBE3 28C6
                               <1>
                                        sub dh, al
726
                               <1>
                                                                 ; dx = 2FEh for COM2
727 0000FBE5 EBEF
                               <1>
                                        jmp short sp_i4s
728
                               <1>
729
                               <1> sp_setp: ; Set serial port communication parameters
                                       ; 07/11/2015
730
                               <1>
731
                               <1>
                                        ; 29/10/2015
                                        ; 29/06/2015
732
                               <1>
                                        ; Retro UNIX 386 v1 feature only !
733
                               <1>
734
                               <1>
                                      ; INPUT:
735
                               <1>
                                        ; AL = 0 for COM1
736
                               <1>
737
                                                1 for COM2
                               <1>
                                        ; AH = Communication parameters (*)
738
                               <1>
                                        ; OUTPUT:
739
                               <1>
                                               CL = Line status
740
                               <1>
741
                               <1>
                                              CH = Modem status
742
                               <1>
                                            If cf = 1 -> Error code in [u.error]
743
                               <1>
                                                     'invalid parameter !'
744
                               <1>
                                                           or
                                                      'device not ready !' error
745
                               <1>
746
                               <1>
747
                               <1>
                                           (*) Communication parameters (except BAUD RATE):
                                              Bit 4 3 2 1 0
-PARITY-- STOP BIT -WORD LENGTH-
748
                               <1>
749
                               <1>
                                           this one --> 00 = none 0 = 1 bit 11 = 8 bits
750
                               <1>
751
                               <1>
                                                    01 = odd   1 = 2 bits   10 = 7 bits
752
                               <1>
                                                    11 = even
753
                               <1>
                                           Baud rate setting bits: (29/06/2015)
                                                 Retro UNIX 386 v1 feature only !
754
                               <1>
                                               Bit 7 6 5 | Baud rate
755
                               <1>
756
                               <1>
757
                               <1>
                                              value 0 0 0 | Default (Divisor = 1)
758
                               <1>
                                                    0 0 1 | 9600 (12)
759
                               <1>
                                                     0
                                                         1
                                                              0 | 19200 (6)
                                                     0 1 1 | 38400 (3)
760
                               <1>
                                                     1 0 0 | 14400 (8)
761
                               <1>
762
                               <1>
                                                     1 0 1 | 28800 (4)
```

```
1 0 | 57600 (2)
763
                                <1>
764
                                <1>
                                                           1
                                                               1 | 115200 (1)
765
                                <1>
766
                                <1>
                                         ; (COM1 base port address = 3F8h, COM1 Interrupt = IRQ 4)
767
                                <1>
                                         ; (COM2 base port address = 2F8h, COM1 Interrupt = IRQ 3)
768
                                <1>
                                         ; ((Modified registers: EAX, ECX, EDX, EBX))
769
                                <1>
770
                                <1>
771 0000FBE7 66BAF803
                                <1>
                                               dx, 3F8h
772 0000FBEB BB[26590100]
                               <1>
                                               ebx, com1p ; COM1 control byte offset
                                         mov
773 0000FBF0 3C01
                               <1>
                                         cmp al, 1
                                         ja
774 0000FBF2 776B
                                               short sp_invp_err
short sp_setp1 ; COM1 (AL = 0)
                                <1>
775 0000FBF4 7203
                                         jb
                               <1>
                               <1> dec <1> inc
776 0000FBF6 FECE
                                               dh ; 2F8h
777 0000FBF8 43
                                               ebx ; COM2 control byte offset
778
                               <1> sp_setp1:
                               <1> ; 29/10/2015
<1> mov [ebx],
779
780 0000FBF9 8823
                               <1>
                                         mov [ebx], ah
781 0000FBFB 0FB6CC
                               <1>
                                         movzx ecx, ah
782 0000FBFE C0E905
                               <1>
                                     and ah, 1Fh; communication parameters except baud rate mov al, [ecx+b_div_tbl]
                                         shr cl, 5; -> baud rate index
783 0000FC01 80E41F
                              <1>
                                        mov al, [ecx+b_div_tbl]
mov cx, ax
784 0000FC04 8A81[6EFC0000]
                               <1>
785 0000FC0A 6689C1
                               <1>
786 0000FC0D E896FFFFF
                               <1>
                                     call sp_i3
787 0000FC12 6689C1
                               <1>
                                        mov cx, ax; CL = Line status, CH = Modem status
788 0000FC15 A880
                               <1>
                                         test al, 80h
                                     jz short sp_setp2
789 0000FC17 740F
                               <1>
790 0000FC19 C603E3
                                                 byte [ebx], 0E3h; Reset to initial value (11100011b)
                               <1>
                                         mov
                               <1> stp_dnr_err:
                                       mov dword [u.error], ERR_DEV_NOT_RDY ; 'device not ready !'
792 0000FC1C C705[C8030300]0F00- <1>
792 0000FC24 0000
                               <1>
793
                               <1>
                                         ; CL = Line status, CH = Modem status
794 0000FC26 F9
                               <1>
                                         stc
795 0000FC27 C3
                               <1>
                                         retn
                               <1> sp_setp2:
796
797 0000FC28 80FE02
                               <1>
                                               dh, 2 ; COM2 (2F?h)
                                          cmp
                                          jna sp_i6
798 0000FC2B 0F8649FFFFF
                               <1>
                                                   ; COM1 (3F?h)
799
                                <1>
800
                                <1> sp_i5:
                                     ; 07/11/2015
801
                                <1>
802
                                <1>
                                        ; 26/10/2015
                                        ; 29/06/2015
803
                                <1>
804
                                <1>
805
                               <1>
                                        ;; COM1 - enabling IRQ 4
806 0000FC31 9C
                               <1>
                                       pushf
807 0000FC32 FA
                               <1>
                                         cli
                                               dx, 3FCh
                                                                  ; modem control register
808 0000FC33 66BAFC03
                               <1>
                                        mov
809 0000FC37 EC
                                       in
                               <1>
                                               al, dx
                                                                          ; read register
810 0000FC38 EB00
                               <1>
                                         JMP
                                               $+2
                                                                   ; I/O DELAY
                                        or
                                                                  ; enable bit 3 (OUT2)
811 0000FC3A 0C08
                               <1>
                                               al, 8
                                     out dx, al

JMP $+2

mov dx, 3F9h
                                                                  ; write back to register
812 0000FC3C EE
                               <1>
                                                                  ; I/O DELAY
813 0000FC3D EB00
                               <1>
                               <1>
                                               dx, 3F9h
                                                                   ; interrupt enable register
814 0000FC3F 66BAF903
                                         mov
                           a1, dx
<1> JMP $+2
<1> ;or al, 1
<1> or al, 3
<1> out
815 0000FC43 EC
                               <1> in
                                                                  ; read register
                                               al, dx
816 0000FC44 EB00
                                                                 ; I/O DELAY
                                                                 ; receiver data interrupt enable and
; transmitter empty interrupt enable
817
818 0000FC46 0C03
                                     out dx, al
819 0000FC48 EE
                                                                   ; write back to register
820 0000FC49 EB00
                               <1>
                                         JMP
                                               $+2
                                                                  ; I/O DELAY
                              <1> in al, 2 <1> JMP $+2
821 0000FC4B E421
                                               al, 21h
                                                                   ; read interrupt mask register
822 0000FC4D EB00
                                                                  ; I/O DELAY
                                                                  ; enable IRQ 4 (COM1)
                                        and
                                               al, OEFh
823 0000FC4F 24EF
                               <1>
                                              21h, al
824 0000FC51 E621
                               <1>
                                         out
                                                                   ; write back to register
825
                               <1>
                                <1>
                                         ; 23/10/2015
826
827 0000FC53 B8[63FA0000]
                                <1>
                                               eax, com1_int
                                         mov
                                                [com1\_irq4], eax
828 0000FC58 A3[75FC0000]
                                <1>
                                         mov
                                <1>
                                         ; 26/10/2015
829
830 0000FC5D 9D
                                <1>
                                         popf
831 0000FC5E C3
                                <1>
                                         retn
832
                                <1>
                                <1> sp_invp_err:
833
834 0000FC5F C705[C8030300]1700- <1>
                                               dword [u.error], ERR_INV_PARAMETER ; 'invalid parameter !'
                                        mov
834 0000FC67 0000
                               <1>
835 0000FC69 31C9
                                <1>
                                         xor
                                                ecx, ecx
836 0000FC6B 49
                                                ecx; OFFFFh
                               <1>
                                          dec
837 0000FC6C F9
                                <1>
                                          stc
838 0000FC6D C3
                                <1>
                                         retn
839
                                <1>
840
                                <1> ; 29/10/2015
                                <1> b_div_tbl: ; Baud rate divisor table (115200/divisor)
842 0000FC6E 010C0603080401
                                         db 1, 12, 6, 3, 8, 4, 1
843
                                <1>
844
                                <1>
                                <1>; 23/10/2015
845
846
                                <1> com1_irq4:
                                         dd dummy_retn
847 0000FC75 [7DFC0000]
                                <1>
                                <1> com2_irq3:
848
849 0000FC79 [7DFC0000]
                                <1>
                                         dd dummy retn
850
                                <1>
                                <1> dummy_retn:
851
852 0000FC7D C3
                                <1>
                                         retn
                                <1>
853
                                <1> wakeup:
854
                                       ; 24/01/2016
855
                                <1>
856 0000FC7E C3
                                <1>
                                          retn
857
                                <1>
                                <1> set_working_path_x:
858
859
                                <1>
                                                ; 17/10/2016 (TRDOS 386 - FFF & FNF)
860 0000FC7F 66B80100
                                <1>
                                                mov ax, 1
861
                                <1>
                                                      ; File name is needed/forced (AL=1)
862
                                <1>
                                                      ; Change directory as temporary (AH=0)
863
                                <1>
                                <1> set working path xx: ; 30/12/2017 (syschdir)
864
865
                                                ; This is needed for preventing wrong Find Next File
                                <1>
```

```
866
                                <1>
                                                 ; system call after sysopen, syscreate, sysmkdir etc.
867
                                 <1>
                                                ; Find Next File must immediate follow Find First File)
868
                                <1>
869 0000FC83 8825[74650100]
                                <1>
                                                      [FFF_Valid], ah ; 0 ; reset ; 17/10/2016
870
                                <1>
871
                                 <1> set_working_path:
                                        ; 16/10/2016
872
                                <1>
                                                ; 12/10/2016
873
                                <1>
874
                                <1>
                                                ; 10/10/2016
                                                ; 05/10/2016 - TRDOS 386 (TRDOS v2.0)
875
                                <1>
876
                                <1>
                                                ; TRDOS v1.0 (DIR.ASM, "proc_set_working_path")
877
                                 <1>
                                                    ; 27/01/2011 - 08/02/2011
878
                                <1>
879
                                <1>
                                                ; Set/Changes current drive, directory and file
880
                                <1>
                                                ; depending on command tail
881
                                <1>
                                                ; (procedure is derivated from CMD_INTR.ASM
                                                ; file or dir locating code of internal commands)
882
                                 <1>
                                                ; (This procedure is prepared for INT 21H file/dir
883
                                <1>
                                                ; functions and also to get compact code for
884
                                 <1>
885
                                <1>
                                                ; internal mainprog -command interpreter- commands)
886
                                <1>
887
                                 <1>
                                                ; INPUT: DS:SI -> Command tail (ASCIIZ string)
                                                ; AL = 0 \rightarrow any, AL > 0 \rightarrow file name is forced
888
                                <1>
889
                                 <1>
                                                ; AH = CD -> Change directory permanently
890
                                <1>
                                                ; AH <> CD -> Change directory as temporary
891
                                 <1>
892
                                 <1>
                                                ; OUTPUT: ES=DS, FindFile structure has been set
893
                                <1>
                                                ;
                                                         RUN_CDRV points previous current drive
894
                                                          DS:SI = FindFile structure address
                                 <1>
                                                ;
                                                         (DS=CS)
895
                                <1>
                                                ;
896
                                 <1>
                                                        AX, BX, CX, DX, DI will be changed
897
                                 <1>
                                                   cf = 1 \rightarrow Error code in AX (AL)
                                                ;
                                                     stc & AX = 0 -> Bad command or path name
898
                                <1>
                                                ; -----
899
                                <1>
900
                                <1>
                                                ;
                                                ; TRDOS 386 (05/10/2016)
901
                                <1>
                                                ; INPUT:
902
                                 <1>
                                                       ESI = File/Directory Path (ASCIIZ string)
903
                                <1>
                                                ;
                                                              address in user's memory space
904
                                 <1>
                                                ;
905
                                                        AL = 0 \rightarrow any
                                <1>
                                                ;
                                                        AL > 0 \rightarrow file name is forced
906
                                <1>
                                                         AH = CD -> change directory as permanent
907
                                 <1>
                                                ;
908
                                <1>
                                                        AH <> CD -> change directory as temporary
                                                ;
909
                                <1>
910
                                <1>
                                                ; OUTPUT:
                                                       FindFile structure has been set
911
                                <1>
912
                                <1>
                                                        RUN CDRV points previous current drive
                                                ;
                                                         ESI = FindFile Name address ; 12/10/2016
913
                                <1>
                                                ;
914
                                <1>
                                                ;
                                                        cf = 1 -> Error code in EAX (AL)
915
                                <1>
                                                ;
916
                                <1>
                                                         stc & EAX = 0 \rightarrow Bad command or path name
                                                ;
917
                                 <1>
                                                ;
918
                                <1>
                                                ; Modified registers: EAX, EBX, ECX, EDX, ESI, EDI
919
                                <1>
920 0000FC89 66A3[78650100]
                                <1>
                                                       [SWP_Mode], ax
                                                mov
921 0000FC8F A0[86590100]
                                <1>
                                                mov
                                                       al, [Current_Drv]
922 0000FC94 30E4
                                <1>
                                                       ah, ah
                                                xor
923 0000FC96 66A3[7A650100]
                                                      [SWP_DRV], ax
                                <1>
                                                mov
924
                                <1>
925
                                                ; TRDOS 386 ring 3 (user's page directory)
                                <1>
926
                                <1>
                                                ; to ring 0 (kernel's page directory)
927
                                <1>
                                                ; transfer modifications (05/10/2016).
928
                                <1>
929 0000FC9C 55
                                <1>
                                                push ebp
930 0000FC9D 89E5
                                <1>
                                                       ebp, esp
                                                mov
931
                                <1>
932 0000FC9F B98000000
                                <1>
                                                mov
                                                       ecx, 128; maximum path length = 128 bytes
933 0000FCA4 29CC
                                <1>
                                                sub
                                                       esp, ecx; reserve 128 bytes (buffer) on stack
934 0000FCA6 89E7
                                <1>
                                                       edi, esp ; destination address (kernel space)
                                                mov
935
                                <1>
                                                ; esi = source address (virtual, in user's memory space)
936 0000FCA8 E89AEBFFFF
                                <1>
                                                 call transfer_from_user_buffer
                                                      short loc_swp_xor_retn
937 0000FCAD 720A
                                <1>
                                                jс
938
                                <1>
939 0000FCAF 89E6
                                                       esi, esp; temporary buffer (the path) on stack
                                <1>
                                <1> loc_swp_fchar:
940
941 0000FCB1 8A06
                                <1>
                                                mov
                                                       al, [esi]
942 0000FCB3 3C20
                                <1>
                                                      al, 20h
                                                 cmp
943 0000FCB5 7711
                                <1>
                                                 jа
                                                       short loc_swp_parse_path_name
944 0000FCB7 740C
                                <1>
                                                       short loc_swp_fchar_next
                                                jе
945
                                <1>
946
                                <1> loc_swp_xor_retn:
                                       xor
947 0000FCB9 31C0
                                 <1>
                                                       eax, eax
948 0000FCBB F9
                                <1>
                                                stc
                                <1> loc_swp_retn:
949
950 0000FCBC 89EC
                                <1>
                                                mov
                                                       esp, ebp
951 0000FCBE 5D
                                <1>
                                                       ebp
                                                pop
                                <1>
953
                                <1>
                                                       esi, FindFile_Drv
                                                ;mov
954 0000FCBF BE[68620100]
                                <1>
                                                       esi, FindFile_Name ; 12/10/2016
                                                mov
955 0000FCC4 C3
                                <1>
                                                retn
956
                                <1>
957
                                <1> loc_swp_fchar_next:
958 0000FCC5 46
                                        inc esi
                                <1>
959 0000FCC6 EBE9
                                <1>
                                                       short loc_swp_fchar
                                                 jmp
960
                                <1>
                                <1> loc_swp_parse_path_name:
961
962 0000FCC8 BF[26620100]
                                         mov edi, FindFile Drv
                                <1>
963 0000FCCD E8E6A7FFFF
                                <1>
                                                 call parse path name
964 0000FCD2 72E8
                                <1>
                                                 jc short loc_swp_retn
965
                                <1>
                                <1> loc_swp_checkfile_name:
966
                                                cmp byte [SWP_Mode], 0 jna short loc_swp_drv
967 0000FCD4 803D[78650100]00
                                <1>
968 0000FCDB 761E
                                <1>
969
                                <1>
970
                                <1>
                                                ; 10/10/2016 (valid file name checking)
```

```
971 0000FCDD BE[68620100]
                                <1>
                                                       esi, FindFile_Name
                                                mov
 972 0000FCE2 803E20
                                 <1>
                                                cmp
                                                       byte [esi], 20h
                                                jna short loc_swp_xor_retn
 973 0000FCE5 76D2
                                <1>
974
                                <1>
 975
                                 <1>
                                                ; 16/10/2016
 976 0000FCE7 C605[77650100]00
                                <1>
                                                mov byte [SWP_inv_fname], 0 ; reset
                                                ; esi = file name address (ASCIIZ)
                                <1>
 978 0000FCEE E8B289FFFF
                                <1>
                                                call check_filename
979 0000FCF3 7306
                                 <1>
                                                jnc short loc_swp_drv
                                <1>
 980
 981 0000FCF5 FE05[77650100]
                                 <1>
                                                inc byte [SWP_inv_fname] ; set
                                 <1> loc_swp_drv:
 983 0000FCFB 8A35[86590100]
                                 <1>
                                                mov dh, [Current_Drv]
                                                 ;mov[RUN CDRV], dh
 984
                                <1>
 985
                                 <1>
 986 0000FD01 8A15[26620100]
                                 <1>
                                                mov dl, [FindFile_Drv]
                                 <1>
                                                ;cmpdl, dh
 988 0000FD07 3A15[86590100]
                                                cmp dl, [Current_Drv]
                                <1>
                                                       short loc swp change directory
 989 0000FD0D 740D
                                <1>
                                                jе
                                <1>
 991 0000FD0F FE05[7B650100]
                                <1>
                                                inc byte [SWP_DRV_chg]
 992 0000FD15 E82A72FFFF
                                 <1>
                                                call change current drive
 993 0000FD1A 72A0
                                                jc short loc_swp_retn ; eax = error code
                                 <1>
                                 <1>
                                                ; eax = 0
 995
                                 <1>
996
                                 <1> loc_swp_change_directory:
                                          cmp byte [FindFile_Directory], 21h
 997 0000FD1C 803D[27620100]21
                                <1>
998 0000FD23 F5
                                <1>
                                                cmc
999 0000FD24 7396
                                <1>
                                                jnc
                                                      short loc_swp_retn
1000
                                <1>
1001 0000FD26 FE05[7B650100]
                                <1>
                                                inc byte [SWP_DRV_chg]
                                                inc byte [Restore_CDIR]
mov esi, FindFile_Directory
1002 0000FD2C FE05[620D0100]
                                <1>
1003 0000FD32 BE[27620100]
                                <1>
1004 0000FD37 8A25[79650100]
                                <1>
                                                mov ah, [SWP Mode+1]
                                                call change_current_directory
1005 0000FD3D E860A1FFFF
                                 <1>
1006 0000FD42 0F8274FFFFF
                                <1>
                                                jс
                                                       loc_swp_retn ; eax = error code
                                 <1>
1008
                                 <1> loc_swp_change_prompt_dir_string:
                                             ; esi = PATH Array
1009
                                 <1>
1010
                                <1>
                                                ; eax = Current Directory First Cluster
1011
                                <1>
                                                ; edi = Logical DOS Drive Description Table
1012 0000FD48 E87AA0FFFF
                                <1>
                                                call change_prompt_dir_str
1013 0000FD4D 29C0
                                <1>
                                                sub eax, eax; 0
1014 0000FD4F E968FFFFFF
                                <1>
                                                jmp loc_swp_retn
1015
                                 <1>
1016
                                 <1> reset_working_path:
                                           ; 06/10/2016 - TRDOS 386 (TRDOS v2.0)
1017
                                 <1>
1018
                                 <1>
                                                ; TRDOS v1.0 (DIR.ASM, "proc reset working path")
1019
                                 <1>
1020
                                 <1>
                                                ; 05/02/2011 - 08/02/2011
1021
                                 <1>
1022
                                 <1>
                                                ; Restores current drive and directory
1023
                                 <1>
                                                ; INPUT: none
1024
                                 <1>
1025
                                 <1>
                                                ; OUTPUT: DL = SWP_DRV, EAX = 0 -> OK
1026
                                 <1>
1027
                                 <1>
                                                     AX = 0 -> ESI = Logical Dos Drv Desc. Table
                                                ;
1028
                                 <1>
                                                ;
1029
                                 <1>
                                                     EAX, EBX, ECX, EDX, ESI, EDI will be changed
                                                ;
1030
                                 <1>
                                                ;
1031
                                 <1>
1032
                                 <1>
1033 0000FD54 31C0
                                <1>
                                                xor
                                                       eax, eax
1034 0000FD56 48
                                 <1>
                                                dec
1035
                                 <1>
1036 0000FD57 668B15[7A650100]
                                <1>
                                                mov
                                                       dx, [SWP_DRV]
1037 0000FD5E 08F6
                                <1>
                                                or
                                                       dh, dh
1038 0000FD60 742E
                                 <1>
                                                       short loc_rwp_return
                                                jz
1039
                                 <1>
1040 0000FD62 3A15[86590100]
                                <1>
                                                      dl, [Current Drv]
                                                cmp
1041 0000FD68 7407
                                <1>
                                               jе
                                                       short loc_rwp_restore_cdir
1042
                                 <1> loc_rwp_restore_cdrv:
                                      call change_current_drive
1043 0000FD6A E8D571FFFF
                                <1>
1044 0000FD6F EB10
                                <1>
                                                jmp
                                                       short loc_rwp_restore_ok
                                <1> loc_rwp_restore_cdir:
1045
1046 0000FD71 31DB
                                <1>
                                                xor ebx, ebx
1047 0000FD73 88D7
                                <1>
                                                mov bh, dl
1048 0000FD75 BE00010900
                                                       esi, Logical_DOSDisks
                                <1>
                                                mov
1049 0000FD7A 01DE
                                <1>
                                                add
                                                      esi, ebx
                                <1>
1050
1051 0000FD7C E87A72FFFF
                                 <1>
                                                call restore_current_directory
                                 <1>
                                 <1> loc_rwp_restore_ok:
1054 0000FD81 668B15[7A650100]
                                                      dx, [SWP_DRV]
                                 <1>
1055 0000FD88 31C0
                                 <1>
                                                xor
                                                       eax, eax
1056 0000FD8A 66A3[7B650100]
                                                       [SWP DRV chg], ax
                                 <1>
                                                mov
                                 <1> loc_rwp_return:
1058 0000FD90 C3
                                 <1>
                                                retn
1059
                                 <1>
1060
                                 <1> get_file_name:
                                                ; 15/10/2016 - TRDOS 386 (TRDOS v2.0)
1061
                                 <1>
1062
                                 <1>
                                                ; Convert file name
1063
                                 <1>
                                                   from directory entry format
                                                    ; to (8.3) dot file name format
1064
                                 <1>
1065
                                 <1>
                                                 ; TRDOS v1.0 (DIR.ASM, "get_file_name")
1066
                                 <1>
1067
                                 <1>
                                                    ; 2005 - 09/10/2011
                                                 ; INPUT:
1068
                                 <1>
1069
                                 <1>
                                                   DS:SI -> Directory Entry Format File Name
                                                ;
1070
                                                       ES:DI -> DOS Dot File Name Address
                                 <1>
                                                ;
1071
                                 <1>
                                                ; OUTPUT:
1072
                                 <1>
                                                    DS:SI -> DOS Dot File Name Address
1073
                                 <1>
                                                    ; ES:DI -> Directory Entry Format File Name
1074
                                 <1>
1075
                                 <1>
                                                 ; TRDOS 386 (15/10/2016)
```

```
1076
                                               ; INPUT:
                                <1>
1077
                                <1>
                                               ;
                                                      ESI = File name addr in dir entry format
1078
                                <1>
                                                      EDI = Dot file name address (destination)
                                               ;
1079
                                <1>
                                               ; OUTPUT:
1080
                                <1>
                                                      File name is converted and moved
                                               ;
1081
                                <1>
                                                      to destination (as 8.3 dot filename)
1082
                                <1>
                                               ;
                                               ; Modified registers: EAX, ECX
1083
                                <1>
1084
                                <1>
1085
                                <1>
                                                 ; 2005 (TRDOS 8086) - 2016 (TRDOS 386)
1086
                                <1>
1087 0000FD91 57
                                <1>
                                                push edi
1088 0000FD92 56
                                                push esi
                               <1>
1089 0000FD93 AC
                               <1>
                                               lodsb
1090 0000FD94 3C20
                               <1>
                                                cmp al, 20h
1091 0000FD96 762A
                               <1>
                                                jna
                                                      short pass_gfn_ext
1092 0000FD98 56
                               <1>
                                               push esi
                                               stosb
1093 0000FD99 AA
                               <1>
                                               mov ecx, 7
1094 0000FD9A B907000000
                                <1>
                               <1> loc_gfn_next_char:
1095
1096 0000FD9F AC
                               <1>
                                              lodsb
1097 0000FDA0 3C20
                               <1>
                                               cmp
                                                     al, 20h
1098 0000FDA2 7603
                               <1>
                                               jna
                                                     short pass_gfn_fn
1099 0000FDA4 AA
                               <1>
1100 0000FDA5 E2F8
                               <1>
                                               loop loc_gfn_next_char
1101
                               <1> pass_gfn_fn:
1102 0000FDA7 5E
                               <1> pop
                                                     esi
1103 0000FDA8 83C607
                               <1>
                                                add
                                                     esi, 7
1104 0000FDAB AC
                               <1>
                                               lodsb
1105 0000FDAC 3C20
                               <1>
                                               cmp al, 20h
1106 0000FDAE 7612
                               <1>
                                               jna
                                                      short pass_gfn_ext
                                                      ah, '.
1107 0000FDB0 B42E
                               <1>
                                               mov
1108 0000FDB2 86E0
                               <1>
                                               xchg ah, al
1109 0000FDB4 66AB
                               <1>
                                               stosw
1110 0000FDB6 AC
                               <1>
                                               lodsb
1111 0000FDB7 3C20
                               <1>
                                                cmp
                                                      al, 20h
1112 0000FDB9 7607
                                                      short pass gfn ext
                               <1>
                                               jna
1113 0000FDBB AA
                               <1>
                                                stosb
1114 0000FDBC AC
                               <1>
                                                lodsb
                                               cmp al, 20h
1115 0000FDBD 3C20
                               <1>
1116 0000FDBF 7601
                                                     short pass_gfn_ext
                               <1>
                                                jna
1117 0000FDC1 AA
                               <1>
                                               stosb
1118
                               <1> pass_gfn_ext:
1119 0000FDC2 30C0
                               <1>
                                                      al, al
                                               xor
1120 0000FDC4 AA
                               <1>
                                                stosb
1121 0000FDC5 5E
                               <1>
                                                pop
                                                      esi
1122 0000FDC6 5F
                                               pop
                               <1>
                                                      edi
1123 0000FDC7 C3
                                <1>
                                               retn
1124
                                <1>
                                <1> set_hardware_int_vector:
1125
                                       ; 18/03/2017
1126
                                <1>
1127
                                <1>
                                               ; 03/03/2017
                                               ; 28/02/2017 - TRDOS 386 (TRDOS v2.0)
1128
                                <1>
1129
                                <1>
                                              ; SET/RESET HARDWARE INTERRUPT GATE
1130
                                <1>
1131
                                <1>
1132
                                               ; Changes interrupt gate descriptor table
1133
                                <1>
                                               ; (without changing default interrupt list)
1134
                                <1>
                                               ; INPUT:
1135
                                <1>
1136
                                <1>
                                                  AL = IRQ number (0 to 15)
                                                      AH > 0 \rightarrow set
1137
                                <1>
                                               ;
                                                      AH = 0 \rightarrow reset
1138
                                <1>
                                               ;
1139
                                <1>
1140
                                <1>
                                               ; Modified registers: eax, ebx, edx, edi
1141
                                <1>
                                <1>
1143 0000FDC8 C0E002
                                <1>
                                                shl al, 2; IRQ number * 4
1144 0000FDCB 0FB6D8
                                <1>
                                                movzx ebx, al
1145
                                <1>
1146 0000FDCE 08E4
                                <1>
                                                or
                                                      ah, ah
1147 0000FDD0 7508
                                <1>
                                                jnz
                                                     short shintv_1 ; set (for user call service)
1148
                                <1>
1149
                                <1>
                                                ; 18/03/2017
1150 0000FDD2 81C3[60170100]
                                <1>
                                                add ebx, IRQ_list; reset to default interrupt list
1151 0000FDD8 EB06
                                <1>
                                                jmp
                                                      short shintv_2
                                <1> shintv_1:
1153 0000FDDA 81C3[01FE0000]
                                <1>
                                                add
                                                     ebx, IRQ_u_list
1154
                                <1> shintv_2:
1155 0000FDE0 8B13
                                                     edx, [ebx] ; IRQ handler address
                                <1>
                                                mov
1156
                                <1>
                                                ; 03/03/2017
1157
                                <1>
                                                      al, 1 ; IRQ number * 8
1158 0000FDE2 D0E0
                               <1>
                                               shl
1159
                                <1>
                                               ; 18/03/2017
                                               movzx edi, al
1160 0000FDE4 0FB6F8
                                <1>
1161 0000FDE7 81C7[D8560100]
                                                      edi, idt + (8*32); IRQ 0 offset = idt + 256
                               <1>
                                                add
                               <1>
1163 0000FDED 89D0
                               <1>
                                                      eax, edx ; IRQ handler address
                                               mov
1164 0000FDEF BB00000800
                               <1>
                                                      ebx, 80000h
                                                mov
1165
                               <1>
                                                ;mov edx, eax
1166
                               <1>
1167 0000FDF4 66BA008E
                                <1>
                                                mov
                                                      dx, 8E00h
1168 0000FDF8 6689C3
                               <1>
                                                      bx, ax
                                                mov
                                                      eax, ebx; /* selector = 0x0008 = cs */
1169 0000FDFB 89D8
                               <1>
                                                           ; /* interrupt gate - dpl=0, present */
1170
                               <1>
1171 0000FDFD AB
                                                stosd ; selector & offset bits 0-15
                               <1>
1172 0000FDFE 8917
                               <1>
                                               mov [edi], edx; attributes & offset bits 16-23
1173
                               <1>
1174 0000FE00 C3
                                <1>
                               <1> IRQ_u_list:
1175
                                               ; 28/02/2017
1176
                               <1>
1177 0000FE01 [94060000]
                               <1>
                                               dd timer_int
1178 0000FE05 [080E0000]
                               <1>
                                               dd
                                                      kb int
                                               dd irg2
1179 0000FE09 [76080000]
                               <1>
1180 0000FE0D [41FE0000]
                               <1>
                                               dd IRQ service3
```

```
1181 0000FE11 [4BFE0000]
                                                dd
                                                      IRQ_service4
1182 0000FE15 [55FE0000]
                                <1>
                                                dd
                                                      IRQ service5
                                                dd fdc int
1183 0000FE19 [D7410000]
                                <1>
1184 0000FE1D [5FFE0000]
                                <1>
                                                dd IRQ_service7
                                                dd
                                                      rtc_int
1185 0000FE21 [FF070000]
                                <1>
1186 0000FE25 [69FE0000]
                                <1>
                                                dd
                                                      IRQ_service9
1187 0000FE29 [73FE0000]
                                               dd IRQ service10
                               <1>
                                                      IRQ_service11
                                               dd
1188 0000FE2D [7DFE0000]
                                <1>
1189 0000FE31 [87FE0000]
                                <1>
                                                dd
                                                      IRQ_service12
1190 0000FE35 [91FE0000]
                                <1>
                                               dd
                                                     IRQ service13
1191 0000FE39 [8A4B0000]
                                <1>
                                               dd
                                                      hdc1_int
1192 0000FE3D [B14B0000]
                                <1>
                                               dd
                                                      hdc2 int
1193
                                <1>
                                              ; 03/03/2017
1194
                                <1>
                                               ; 27/02/2017
1195
                                <1>
                                <1> IRQ_service3:
1196
1197 0000FE41 36C605[3E6B0100]03 <1> mov
                                                      byte [ss:IRQnum], 3
1198 0000FE49 EB4E
                                                jmp
                                                      short IRQ_service
                                <1>
                                <1> IRQ service4:
1199
1200 0000FE4B 36C605[3E6B0100]04 <1> mov
1201 0000FE53 EB44 <1> jmp
                                                      byte [ss:IRQnum], 4
1201 0000FE53 EB44
                                                      short IRQ_service
                                <1> IRQ service5:
1203 0000FE55 36C605[3E6B0100]05 <1> mov
1204 0000FE5D EB3A <1> jmp
                                                      byte [ss:IRQnum], 5
                                                      short IRQ_service
1205
                                <1> IRQ_service7:
1206 0000FE5F 36C605[3E6B0100]07 <1> mov
                                                      byte [ss:IRQnum], 7
1207 0000FE67 EB30
                                <1>
                                                      short IRQ_service
                                                jmp
1208
                                <1> IRQ_service9:
1209 0000FE69 36C605[3E6B0100]09 <1>
1210 0000FE71 EB26 <1>
                                                      byte [ss:IRQnum], 9
                                               mov
1210 0000FE71 EB26
                                                jmp
                                                      short IRQ_service
1211
                                <1> IRQ_service10:
1212 0000FE73 36C605[3E6B0100]0A <1> mov
                                                      byte [ss:IRQnum], 10
                                <1>
1213 0000FE7B EB1C
                                                      short IRQ_service
                                                jmp
                                <1> IRQ service11:
1215 0000FE7D 36C605[3E6B0100]0B <1> mov
                                                      byte [ss:IRQnum], 11
                                                      short IRQ service
1216 0000FE85 EB12
                                <1>
                                                jmp
                                <1> IRQ_service12:
1218 0000FE87 36C605[3E6B0100]0C <1> mov
                                                     byte [ss:IRQnum], 12
1219 0000FE8F EB08
                                                jmp
                                                      short IRQ service
                                <1>
                                <1> IRQ_service13:
1221 0000FE91 36C605[3E6B0100]0D <1> mov byte [ss:IRQnum], 13
                                <1>
                                               ;jmp short IRQ_service
1223
                                <1> IRQ_service:
                                <1> ; 13/06/2017
1224
                                           ; 11/06/2017
; 10/06/2017
; 01/03/2017, 04/03/2017
1225
                                <1>
1226
                                <1>
                                <1>
                                              ; 27/02/2017, 28/02/2017
                                <1>
1228
1229 0000FE99 1E
                                <1>
                                               push ds
1230 0000FE9A 06
                                               push es
                               <1>
1231 0000FE9B 0FA0
                               <1>
                                               push fs
1232 0000FE9D 0FA8
                                <1>
                                               push gs
                                <1>
1233
1234 0000FE9F 60
                               <1>
                                               pushad; eax,ecx,edx,ebx,esp,ebp,esi,edi
1235 0000FEA0 66B91000
                               <1>
                                               mov
                                                       CX, KDATA
1236 0000FEA4 8ED9
                               <1>
                                                mov
                                                        ds, cx
1237 0000FEA6 8EC1
                               <1>
                                                mov
                                                        es, cx
1238 0000FEA8 8EE1
                               <1>
                                                mov
                                                        fs, cx
1239 0000FEAA 8EE9
                                <1>
                                                mov
                                                        gs, cx
1240
                                <1>
1241 0000FEAC 0F20D8
                                <1>
                                                mov
                                                      eax, cr3
1242 0000FEAF A3[3A6B0100]
                                                      [IRQ cr3], eax
                                <1>
                                                mov
1243
                                <1>
1244 0000FEB4 A1[C0580100]
                                <1>
                                                      eax, [k_page_dir]
                                                mov
1245 0000FEB9 0F22D8
                                <1>
                                                      cr3, eax
                                                mov
1246
                                <1>
1247 0000FEBC A0[3E6B0100]
                                <1>
                                                mov
                                                      al, [IRQnum]
1248
                                <1>
1249
                                <1>
                                                      cl, [sysflg]
                                                ; mov
1250
                                <1>
                                                      [u.r_mode], cl ; system (0) or user mode (FFh)
                                                ; mov
                                <1> IRQsrv 0:
1251
1252 0000FEC1 0FB6D8
                                <1>
                                                movzx ebx, al
                                                      bl, [ebx+IRQenum] ; IRQ (available) index number + 1
1253 0000FEC4 8A9B[96160100]
                                <1>
                                                mov
1254
                                <1>
                                               ; 01/03/2017
1255 0000FECA FECB
                                <1>
                                               dec bl ; IRQ index number, 0 to 8
1256 0000FECC 0F8807010000
                                <1>
                                                js
                                                      IRQsrv_5 ; not available to use here!?
                                <1>
                                               ;
1258 0000FED2 80BB[046B0100]80
                                <1>
                                               cmp byte [ebx+IRQ.method], 80h; using by a dev or kernel?
1259 0000FED9 7205
                                <1>
                                               jb short IRQsrv_1 ; no
1260
                                <1>
1261
                                <1>
                                                ; If the IRQ service is already owned by TRDOS 386 kernel
1262
                                 <1>
                                                       or a Device driver
                                                ; we need to call 'dev_IRQ_service
1263
                                <1>
1264
                                <1>
1265
                                <1>
                                                : IRO number in AL
1266 0000FEDB E866020000
                                                call dev IRQ service
                                                                         ; IRQ service for device drivers
                                <1>
1267
                                <1>
                                                ; IRQ number in AL
                                <1> IRQsrv_1:
1268
                                                ; check user callback service status
1269
                                <1>
1270
                                                ; AL = IRQ number
                                <1>
1271
                                <1>
                                                ; EBX = IRQ (Available) Index number
1272
                                <1>
                                                      [u.irqwait], al ; set waiting IRQ flag
1273 0000FEE0 A2[D7030300]
                                <1>
                                                mov
                                <1>
1274
1275 0000FEE5 8A83[F26A0100]
                                <1>
                                                      al, [ebx+IRQ.owner]
                                                mov
1276 0000FEEB 20C0
                                <1>
                                                and
                                                      al, al
1277 0000FEED 0F84E6000000
                                <1>
                                                      IRQsrv 5 ; it is not owned by a user/proc
                                                jz
1278
                                <1>
1279
                                <1>
                                                ; 03/03/2017
1280 0000FEF3 89DA
                                <1>
                                                mov edx, ebx
1281 0000FEF5 C0E202
                                <1>
                                                shl
                                                      dl, 2
1282 0000FEF8 8B92[166B0100]
                                <1>
                                                mov
                                                      edx, [edx+IRQ.addr]; S.R.B. or Callback service addr
                                <1>
1284 0000FEFE 8AA3[046B0100]
                                <1>
                                                      ah, [ebx+IRQ.method]
1285 0000FF04 F6C401
                                <1>
                                                test ah, 1
```

```
short IRQsrv 4 ; Callback service method
1286 0000FF07 7534
                                  <1>
                                                   jnz
1287
                                  <1>
1288
                                  <1>
                                                   ; Signal Response Byte method
1289
                                  <1>
                                                  ;mov edx, [edx+IRQ.addr] ; Signal Response Byte address
1290
                                  <1>
                                                                          ; (Physical address, non-swappable)
                                                   ;
1291 0000FF09 80E402
                                  <1>
                                                   and
                                                         ah, 2; bit 1, (S.R.B.) counter (auto increment) method
1292 0000FF0C 8AA3[0D6B0100]
                                  <1>
                                                  mov
                                                         ah, [ebx+IRQ.srb]; Signal Response Byte value
1293 0000FF12 7408
                                  <1>
                                                         short IRQsrv_2 ; fixed S.R.B. value
                                                  jΖ
1294
                                  <1>
                                                   ; counter method (auto increment)
1295 0000FF14 FEC4
                                  <1>
                                                  inc
                                                         ah
1296 0000FF16 88A3[0D6B0100]
                                  <1>
                                                         [ebx+IRQ.srb], ah ; next (count) number
                                                   mov
                                  <1> IRQsrv 2:
1298 0000FF1C 8822
                                                          [edx], ah; put S.R.B. val to the user's S.R.B. addr
                                  <1>
                                                   mov
1299 0000FF1E C605[D7030300]00
                                                         byte [u.irqwait], 0 ; clear waiting IRQ flag
                                  <1>
1300
                                  <1>
1301 0000FF25 3A05[B3030300]
                                  <1>
                                                         al, [u.uno]
                                                   cmp
1302 0000FF2B 0F84A8000000
                                  <1>
                                                   jе
                                                         IRQsrv 5; the owner is current user/process
                                  <1> IRQsrv_3:
1303
1304
                                  <1>
                                                   ; the owner is not current user/process
1305
                                  <1>
                                                   ; AL = process number
1306 0000FF31 B202
                                                  mov
                                                         dl, 2 ; priority, 2 = event (high)
                                  <1>
1307 0000FF33 E837FAFFFF
                                  <1>
                                                   call set_run_sequence
1308
                                  <1>
1309
                                  <1>
                                                   ; [u.irqwait] = waiting IRQ number for callback service
1310
                                  <1>
1311 0000FF38 E99C000000
                                  <1>
                                                   jmp
                                                         IRQsrv_5
                                  <1> IRQsrv 4:
1313 0000FF3D 3A05[B3030300]
                                  <1>
                                                   cmp
                                                         al, [u.uno] ; is the owner is current user/process?
1314 0000FF43 75EC
                                                         short IRQsrv_3 ; no !
                                  <1>
                                                   jne
1315
                                  <1>
1316
                                  <1>
                                                   ; Check if an IRQ callback service already in progress
1317 0000FF45 803D[D8030300]00
                                  <1>
                                                         byte [u.r lock], 0
                                                   cmp
1318 0000FF4C 0F8787000000
                                                         IRQsrv_5 ; nothing to do !
                                  <1>
                                                   jа
1319
                                  <1>
                                                                     ; (we need to complete prev callback)
1320 0000FF52 803D[D4030300]00
                                  <1>
                                                         byte [u.t_lock], 0
                                                   cmp
1321 0000FF59 777E
                                  <1>
                                                   jа
                                                         short IRQsrv_5; nothing to do!
1322
                                  <1>
                                                                    ; (we need to complete timer callback)
1323
                                  <1>
                                                   ; 04/03/2017
1324
                                  <1>
1325 0000FF5B C605[D7030300]00
                                                         byte [u.irqwait], 0 ; reset/clear waiting IRQ flag
                                  <1>
                                                   mov
1326
                                  <1>
1327 0000FF62 FE05[D8030300]
                                  <1>
                                                   inc
                                                         byte [u.r lock]; 'IRQ callback service in progress' flag
1328
                                  <1>
1329 0000FF68 8A0D[5B030300]
                                                         cl, [sysflg] ; (system call) mode flag (kernel/user)
                                  <1>
                                                   mov
                                                         [u.r mode], cl; system mode (0) or user mode (FFh)
1330 0000FF6E 880D[D9030300]
                                  <1>
                                                  mov
1331
                                  <1>
1332
                                  <1>
                                                   ;
1333 0000FF74 8B2D[5C580100]
                                                         ebp, [tss.esp0] ; kernel stack address (for ring 0)
                                  <1>
                                                   mov
1334 0000FF7A 83ED14
                                  <1>
                                                   sub
                                                          ebp, 20
                                                                             ; eip, cs, eflags, esp, ss
1335 0000FF7D 892D[5C030300]
                                  <1>
                                                          [u.sp], ebp
                                                   mov
1336 0000FF83 8925[60030300]
                                  <1>
                                                         [u.usp], esp
                                                   mov
1337
                                  <1>
                                                         word [ebp+8], 200h; 22/01/2017, force enabling interrupts
1338
                                  <1>
                                                   ;or
1339
                                  <1>
1340 0000FF89 8B44241C
                                  <1>
                                                         eax, [esp+28]; pushed eax
                                                   mov
1341 0000FF8D A3[64030300]
                                  <1>
                                                   mov
                                                         [u.r0], eax
                                  <1>
1343 0000FF92 E81AE7FFFF
                                  <1>
                                                   call wswap; save user's registers & status
1344
                                  <1>
1345
                                  <1>
                                                   ; software int is in ring 0 but IRQ handler must return to ring 3
1346
                                  <1>
                                                  ; so, ring 3 return address and stack registers
1347
                                  <1>
                                                  ; (eip, cs, eflags, esp, ss)
1348
                                  <1>
                                                   ; must be copied to IRQ handler return
1349
                                  <1>
                                                  ; eip will be replaced by callback service routine address
                                  <1>
1350
1351 0000FF97 C605[5B030300]FF
                                  <1>
                                                         byte [sysflg], OFFh; user mode
1352
                                  <1>
1353
                                  <1>
                                                   ; system mode (system call)
1354
                                  <1>
                                                   ;mov ebp, [u.sp]; EIP (u), CS (UCODE), EFLAGS (u),
1355
                                  <1>
                                                                    ; ESP (u), SS (UDATA)
1356
                                  <1>
1357 0000FF9E 8B4510
                                  <1>
                                                         eax, [ebp+16]; SS (UDATA)
                                                  mov
1358 0000FFA1 89E6
                                  <1>
                                                  mov
                                                         esi, esp
1359 0000FFA3 50
                                  <1>
                                                   push
                                                         eax
1360 0000FFA4 50
                                  <1>
                                                  push
                                                         eax
1361 0000FFA5 89E7
                                  <1>
                                                   mov
                                                         edi, esp
1362 0000FFA7 893D[60030300]
                                  <1>
                                                         [u.usp], edi
                                                   mov
1363 0000FFAD B908000000
                                  <1>
                                                   mov
                                                         ecx, ((ESPACE/4) - 4); except DS, ES, FS, GS
1364 0000FFB2 F3A5
                                  <1>
                                                   rep
                                                         movsd
1365 0000FFB4 B104
                                  <1>
                                                         cl, 4
                                                   mov
1366 0000FFB6 F3AB
                                  <1>
                                                   rep
                                                         stosd
1367 0000FFB8 893D[5C030300]
                                  <1>
                                                   mov
                                                          [u.sp], edi
1368 0000FFBE 89EE
                                  <1>
                                                  mov
                                                         esi, ebp
                                                         cl, 5; EIP (u), CS (UCODE), EFLAGS (u), ESP (u), SS (UDATA)
1369 0000FFC0 B105
                                  <1>
1370 0000FFC2 F3A5
                                  <1>
                                                   rep
                                                         movsd
1371
                                  <1>
                                                   ;
1372
                                  <1>
1373 0000FFC4 8B0D[B8030300]
                                  <1>
                                                   mov
                                                         ecx, [u.pgdir]
1374 0000FFCA 890D[3A6B0100]
                                  <1>
                                                   mov
                                                         [IRQ_cr3], ecx
1375
                                  <1>
                                  <1> set IRQ callback addr:
1376
1377
                                  <1>
                                                   ; This routine sets return address
1378
                                  <1>
1379
                                  <1>
                                                  ; to start of user's interrupt
                                                  ; service (callback) address
1380
                                  <1>
1381
                                  <1>
                                                  ; INPUT:
1382
                                  <1>
                                                         EDX = callback routine/service address
1383
                                  <1>
                                                  ;
1384
                                  <1>
                                                               (virtual, not physical address!)
1385
                                  <1>
                                                          [u.sp] = kernel stack, points to
                                                  ;
                                                               user's EIP, CS, EFLAGS, ESP, SS
1386
                                  <1>
1387
                                  <1>
                                                                 registers.
                                                  ;
                                                  ; OUTPUT:
1388
                                  <1>
1389
                                  <1>
                                                         EIP (user) = callback (service) address
1390
                                  <1>
                                                         CS (user) = UCODE
```

```
1391
                                  <1>
                                                         EFLAGS (user) = flags before callback
1392
                                  <1>
                                                          ESP (user) = ESP-4 (user, before callback)
                                                  ;
                                                         [ESP] (user) = EIP (user) before callback
1393
                                  <1>
                                                  ;
1394
                                  <1>
1395
                                  <1>
                                                  ; Note: If CPU was in user mode while entering
1396
                                  <1>
                                                        the timer interrupt service routine,
                                                         'IRET' will get return to callback routine
1397
                                  <1>
1398
                                  <1>
                                                         immediately. If CPU was in system/kernel mode
                                                  ;
1399
                                  <1>
                                                         'iret' will get return to system call and
                                                  ;
                                                         then, callback routine will be return address
1400
                                  <1>
                                                  ;
1401
                                  <1>
                                                         from system call. (User's callback/service code
1402
                                  <1>
                                                         will be able to return to normal return address
                                                  ;
                                                         via a 'sysrele' system call at the end.)
1403
                                  <1>
                                                  ;
1404
                                  <1>
1405
                                  <1>
                                                  ; Note: User's IRQ callback service code must be ended
                                                         with a 'sysrele' system call!
1406
                                  <1>
1407
                                  <1>
1408
                                  <1>
                                                         For example:
                                                  ;
1409
                                  <1>
                                                  ;
                                                         audio_IRQ_callback:
1410
                                  <1>
                                                  ;
1411
                                  <1>
                                                             <load DMA buffer with audio data>
1412
                                  <1>
                                                  ;
1413
                                  <1>
                                                  ;
1414
                                  <1>
                                                             mov eax, 39 ; 'sysrele'
                                                  ;
1415
                                  <1>
                                                            int 40h ; TRDOS 386 system call (interrupt)
                                                  ;
1416
                                  <1>
1417
                                  <1>
                                                  ;mov edx, [edx+IRQ.addr] ; Callback service address
1418
                                  <1>
                                                                          ; (Virtual address)
1419
                                  <1>
                                                  ;
1420
                                  <1>
1421 0000FFD0 8B2D[5C030300]
                                  <1>
                                                  mov
                                                         ebp, [u.sp]; kernel's stack, points to EIP (user)
1422 0000FFD6 895500
                                  <1>
                                                  mov
                                                         [ebp], edx
                                  <1> IRQsrv_5:
1423
1424
                                  <1>
                                                  ; EOI & return
                                                  ; 01/08/2020
1425
                                  <1>
1426
                                  <1>
                                                  ; 11/06/2017
                                                  ; 10/06/2017
1427
                                  <1>
1428
                                  <1>
                                                  ;mov al, [IRQnum]
1429 0000FFD9 B020
                                  <1>
                                                  mov
                                                        al, 20h; 01/08/2020
1430 0000FFDB FA
                                  <1>
                                                  cli
                                                  ;cmp al, 7
1431
                                  <1>
                                                  cmp byte [IRQnum], 7; 01/08/2020 jna short IRQsrv_6
1432 0000FFDC 803D[3E6B0100]07
                                  <1>
1433 0000FFE3 7602
                                  <1>
1434
                                  <1>
                                                  ;;mov al, EOI
                                                                     ; end of interrupt
1435
                                  <1>
                                                  ;mov al, 20h; 01/08/2020
1436
                                  <1>
                                                          ; disable interrupts till stack cleared
1437
                                  <1>
                                                  ;cli
                                                   ;out INTB00, al ; For control12 #2
1438
                                  <1>
1439 0000FFE5 E6A0
                                  <1>
                                                   out
                                                         0A0h, al
                                  <1> IRQsrv_6:
1440
                                                  ;mov byte [IRQnum], 0 ; reset
1441
                                  <1>
1442
                                  <1>
                                                  ;;mov al, EOI ; end of interrupt
                                                   ;mov al, 20h; 01/08/2020
1443
                                  <1>
1444
                                  <1>
                                                              ; disable interrupts till stack cleared
1445
                                  <1>
                                                  ;out INTA00, al ; end of interrupt to 8259 - 1
1446 0000FFE7 E620
                                  <1>
                                                   out
                                                         20h, al
1447
                                  <1> IRQsrv_7:
                                                   ;; 13/06/2017
1448
                                  <1>
1449
                                  <1>
                                                   ;or word [ebp+8], 200h; force enabling interrupts
1450
                                  <1>
                                                  ;
1451 0000FFE9 8B0D[3A6B0100]
                                  <1>
                                                  mov
                                                         ecx, [IRQ_cr3]
                                                                             ; previous content of cr3 register
1452 0000FFEF 0F22D9
                                  <1>
                                                  mov
                                                        cr3, ecx ; restore cr3 register content
1453
                                  <1>
1454 0000FFF2 61
                                  <1>
                                                  popad; edi, esi, ebp, (icrement esp by 4), ebx, edx, ecx, eax
1455
                                  <1>
                                                  ;
1456 0000FFF3 0FA9
                                  <1>
                                                  pop
1457 0000FFF5 0FA1
                                  <1>
                                                        fs
                                                  pop
1458 0000FFF7 07
                                  <1>
                                                   pop
                                                        es
1459 0000FFF8 1F
                                  <1>
                                                  pop
                                                         ds
1460
                                  <1>
1461 0000FFF9 CF
                                                  iretd ; return from interrupt
                                  <1>
1462
                                  <1>
                                  <1> get_device_number:
1463
                                                 ; 08/10/2016
1464
                                  <1>
1465
                                  <1>
                                                  ; 07/10/2016 - TRDOS 386 (TRDOS v2.0)
1466
                                  <1>
1467
                                  <1>
                                                  ; This procedure compares name of requested
                                                  ; device with kernel device names and
1468
                                  <1>
                                                  ; installable device names. If names match,
1469
                                  <1>
1470
                                                  ; the relevant device index (entry) number
                                  <1>
1471
                                  <1>
                                                   ; will be returned the caller (sysopen)
                                                   ; for the requested device.
1472
                                  <1>
1473
                                  <1>
1474
                                  <1>
                                                   ; NOTE: Installable device drivers must
                                                   ; be loaded before using 'sysopen'
1475
                                  <1>
1476
                                  <1>
                                                   ; (opendev) system call.
1477
                                  <1>
                                                  ; INPUT:
1478
                                  <1>
1479
                                  <1>
                                                       ESI = device name address (ASCIIZ)
1480
                                  <1>
                                                            (in kernel's memory space)
1481
                                  <1>
                                                       max name length = 8 without '/dev/')
1482
                                  <1>
                                                       Device name will be capitalized
                                                  ;
                                                       and if there is, '/\text{dev}/' will be
1483
                                  <1>
1484
                                  <1>
                                                       removed from name before comparising)
1485
                                  <1>
                                                  ;
1486
                                  <1>
                                                  ; OUTPUT:
                                                       cf = 0 \rightarrow
1487
                                  <1>
                                                  ;
1488
                                  <1>
                                                        EAX (AL) = device entry/index number
1489
                                  <1>
                                                        cf = 1 -> device not found (installed)
1490
                                  <1>
                                                               or invalid device name
1491
                                  <1>
                                                                (AL=0)
1492
                                  <1>
                                                        device name = device name address (asciiz)
                                                  ;
1493
                                  <1>
1494
                                  <1>
                                                   ; Modified registers: EAX, EBX, ESI, EDI
1495
                                  <1>
```

```
1496 0000FFFA BF[7D650100]
                                <1>
                                                        edi, device_name
                                                 mov
1497 0000FFFF E805010000
                                 <1>
                                                 call
                                                        lodsb capitalize
1498 00010004 88C4
                                 <1>
                                                 mov
                                                        ah, al
                                                        al, '/'
1499 00010006 3C2F
                                 <1>
                                                  cmp
                                                 jne
1500 00010008 750E
                                 <1>
                                                        short gdn 1
1501 0001000A BF[7D650100]
                                 <1>
                                                  mov
                                                        edi, device_name
1502 0001000F E8F5000000
                                                        lodsb_capitalize
                                 <1>
                                                  call
                                 <1> gdn_0:
1503
1504 00010014 20C0
                                 <1>
                                                  and
                                                        al, al ; 0 ?
1505 00010016 7420
                                 <1>
                                                        short gdn_err ; null name after '/'
                                                  jz
                                 <1> gdn_1:
1506
1507 00010018 3C44
                                 <1>
                                                  cmp
                                                        al, 'D'
1508 0001001A 7517
                                 <1>
                                                  jne
                                                        short gdn_2
1509 0001001C E8E8000000
                                 <1>
                                                  call lodsb_capitalize
1510 00010021 3C45
                                 <1>
                                                  cmp
                                                        al, 'E'
1511 00010023 750E
                                                        short gdn_2
                                 <1>
                                                  jne
                                                        lodsb_capitalize
1512 00010025 E8DF000000
                                <1>
                                                  call
1513 0001002A 3C56
                                                        al, \overline{V}
                                 <1>
                                                  cmp
1514 0001002C 7505
                                 <1>
                                                  jne
                                                        short gdn_2
1515 0001002E AC
                                 <1>
                                                  lodsb
                                                        al, '/'
1516 0001002F 3C2F
                                 <1>
                                                  cmp
1517 00010031 740D
                                 <1>
                                                  jе
                                                        short gdn 4
1518
                                 <1> gdn_2:
1519 00010033 80FC2F
                                 <1>
                                                  cmp
                                                        ah, '/'
1520 00010036 750F
                                 <1>
                                                  jne
                                                        short gdn_5
1521
                                 <1> gdn_err:
                                 <1>
                                                  ; invalid device name or device not found
1523 00010038 31C0
                                 <1>
                                                  xor
                                                        eax, eax ; 0
1524 0001003A F9
                                 <1>
                                                  stc
1525 0001003B C3
                                <1>
                                                  retn
1526
                                 <1> gdn_3:
1527 0001003C 3C2F
                                                        al, '/'
                                 <1>
                                                  cmp
1528 0001003E 7507
                                 <1>
                                                        short gdn_5
                                                  jne
1529
                                 <1> gdn_4:
                                                        edi, device_name
1530 00010040 BF[7D650100]
                                 <1>
                                                  mov
1531 00010045 EB04
                                 <1>
                                                  jmp
                                                        short gdn_6
                                 <1> gdn 5:
1533 00010047 3C00
                                                        al, 0
                                 <1>
                                                  cmp
1534 00010049 7419
                                 <1>
                                                        short gdn_7
                                                  jе
                                 <1> gdn_6:
1535
1536 0001004B E8B9000000
                                                  call lodsb_capitalize
                                 <1>
1537 00010050 81FF[85650100]
                                 <1>
                                                  cmp
                                                        edi, device name + 8
                                                        short gdn_3
1538 00010056 72E4
                                 <1>
                                                  jb
1539 00010058 3C00
                                 <1>
                                                  cmp
                                                        al, 0
1540 0001005A 75DC
                                 <1>
                                                  jne
                                                        short gdn_err
1541 0001005C 81FF[7E650100]
                                                        edi, device_name + 1
                                 <1>
                                                  cmp
1542 00010062 76D4
                                 <1>
                                                        short gdn_err ; null name after '/'
                                                  jna
1543
                                 <1> gdn_7:
1544 00010064 AA
                                 <1>
                                                  stosb
                                                 ; zero padding ("NAME",0,0,0,0)
                                 <1>
1546 00010065 81FF[85650100]
                                 <1>
                                                  cmp edi, device_name + 8
                                                        short gdn_7
1547 0001006B 72F7
                                 <1>
                                                  jb
1548
                                 <1> gdn_8:
1549
                                 <1>
                                                  ; search for kernel device names
1550 0001006D BE[7D650100]
                                 <1>
                                                 mov esi, device_name
1551 00010072 BF[7C140100]
                                 <1>
                                                  mov
                                                        edi, KDEV_NAME
1552 00010077 31C0
                                 <1>
                                                        eax, eax
                                                  xor
1553
                                 <1> gdn_9:
1554 00010079 A7
                                 <1>
                                                  cmpsd
1555 0001007A 7505
                                 <1>
                                                  jne short gdn_10
1556 0001007C A7
                                 <1>
                                                  cmpsd
1557 0001007D 7503
                                 <1>
                                                  jne
                                                        short gdn 11
1558 0001007F EB2B
                                 <1>
                                                        short gdn_17; match
                                                  jmp
1559
                                 <1> gdn_10:
1560 00010081 A7
                                 <1>
                                                  cmpsd ; add esi, 4 & add edi, 4
1561
                                 <1> gdn_11:
1562 00010082 BE[7D650100]
                                 <1>
                                                  mov
                                                        esi, device_name
1563 00010087 FEC0
                                 <1>
                                                  inc
                                                        al
1564 00010089 3C16
                                 <1>
                                                        al, NumOfKernelDevNames
                                                  cmp
1565 0001008B 72EC
                                 <1>
                                                        short gdn_9
                                                  jb
1566
                                 <1> gdn_12:
                                                 ; search for installable device names
1567
                                 <1>
                                                  ; esi = offset device_name
1568
                                 <1>
1569 0001008D BF[A8650100]
                                 <1>
                                                  mov edi, IDEV_NAME
1570 00010092 28C0
                                 <1>
                                                  sub
                                                       al, al ; \overline{0}
                                 <1> gdn_13:
1571
1572 00010094 A7
                                 <1>
                                                  cmpsd
1573 00010095 7505
                                 <1>
                                                  jne
                                                        short gdn_14
1574 00010097 A7
                                 <1>
                                                  cmpsd
1575 00010098 7503
                                 <1>
                                                  jne short gdn_15
1576 0001009A EB3F
                                 <1>
                                                  jmp
                                                        short gdn_19; match
                                 <1> gdn_14:
1578 0001009C A7
                                 <1>
                                                  cmpsd ; add esi, 4 & add edi, 4
1579
                                 <1> gdn_15:
1580 0001009D BE[7D650100]
                                 <1>
                                                  mov
                                                        esi, device_name
1581 000100A2 FEC0
                                 <1>
                                                  inc
                                                        al
1582 000100A4 3C08
                                 <1>
                                                       al, NumOfInstallableDevices
                                                  cmp
1583 000100A6 72EC
                                 <1>
                                                        short gdn_13
                                                 jb
1584
                                 <1>
                                 <1> gdn_16:
1585
                                                 ; error: invalid device name (not found) !
1586 000100A8 30C0
                                 <1>
                                                  xor
                                                        al, al
1587 000100AA F9
                                 <1>
                                                  stc
1588 000100AB C3
                                 <1>
                                                  retn
1589
                                 <1>
                                 <1> gdn_17:
1590
                                                        ; name match (with one of kernel device names)
1591
                                 <1>
1592
                                 <1>
                                                 ; convert KDEV NAME index to
1593
                                                 ; KDEV_NUMBER index
                                 <1>
                                                 ; (different names are used for same devices)
1594
                                 <1>
                                                 ; (example: "COM1" & "TTY8" = device number 18)
1595
                                 <1>
                                                 mov ebx, eax; < 256
1596 000100AC 89C3
                                 <1>
1597 000100AE 8A83[2C150100]
                                 <1>
                                                 mov al, [KDEV_NUMBER+ebx]
                                 <1>
1599
                                 <1>
                                                 ; check if empty dev entry in the list
1600 000100B4 80B8[2C670100]00
                                 <1>
                                                  cmp byte [DEV_OPENMODE+eax], 0
```

```
1601 000100BB 771B
                                 <1>
                                                        short gdn_18; it must be already set
                                                 jа
1602
                                 <1>
1603
                                 <1>
                                                ; (re)set device name and access flags
1604
                                 <1>
                                                ; (remain open work will be easy after that)
1605
                                 <1>
                                                 ; (NOTE: here, data will be copied to bss section)
1606 000100BD 88C3
                                 <1>
                                                 mov
                                                       bl, al
1607 000100BF 83EF08
                                <1>
                                                 sub
                                                       edi, 8 ; kernel device name address (data)
1608 000100C2 66C1E302
                                                 shl
                                <1>
                                                       bx, 2
1609 000100C6 89BB[4A670100]
                                 <1>
                                                 mov
                                                       [DEV_NAME_PTR+ebx], edi ; (all) device names
1610 000100CC 8A98[82160100]
                                                 mov bl, [KDEV_ACCESS+eax] ; kernel dev list (data)
                                <1>
1611 000100D2 8898[78660100]
                                <1>
                                                 mov [DEV_ACCESS+eax], bl ; (all) device list (bss)
1612
                                 <1> gdn 18:
                                                 inc al ; 1 to NumOfKernelDevNames (<=7Fh)</pre>
1613 000100D8 FEC0
                                 <1>
1614
                                 <1>
                                                 ; eax = device index/entry number
1615 000100DA C3
                                 <1>
                                                 retn
1616
                                 <1>
1617
                                 <1> gdn 19:
                                                       ; name match (with one of installable device names)
1618
                                 <1>
                                                 ; al = 0 to NumOfInstallableDevices - 1 (<=7Fh)
1619
                                 <1>
1620
                                 <1>
1621 000100DB 89C3
                                                       ebx, eax
                                <1>
                                                 mov
1622 000100DD 80C316
                                 <1>
                                                 add bl, NumOfKernelDevices ; < NUMOFDEVICES
1623
                                 <1>
                                 <1>
                                                 ; check if empty dev entry in the list
1625 000100E0 80BB[2C670100]00
                                                 cmp byte [DEV_OPENMODE+ebx], 0
                                <1>
1626 000100E7 771D
                                                       short gdn_20 ; it must be already set
                                 <1>
                                                 jа
1627
                                 <1>
1628
                                 <1>
                                                ; (re)set device name and access flags
1629
                                 <1>
                                                 ; (remain open work will be easy after that)
1630 000100E9 83EF08
                                                 sub edi, 8 ; installable device name address
                                <1>
1631 000100EC 66C1E302
                                <1>
                                                 shl
                                                       bx, 2 ;*4
1632 000100F0 89BB[4A670100]
                                 <1>
                                                 mov
                                                        [DEV_NAME_PTR+ebx], edi ; (all) device names
                                                       bx, \overline{2}
1633 000100F6 66C1EB02
                                 <1>
                                                 shr
1634 000100FA 8A80[F0650100]
                                 <1>
                                                 mov al, [IDEV_FLAGS+eax] ; installable dev list
                                                      [DEV ACCESS+ebx], al; (all) device list
1635 00010100 8883[78660100]
                                 <1>
                                                 mov
1636
                                 <1> gdn_20:
1637 00010106 88D8
                                 <1>
                                                 mov al, bl
1638
                                 <1>
                                                 ; eax = device index/entry number ; < NUMOFDEVICES
1639 00010108 C3
                                 <1>
                                                 retn
                                 <1>
1640
1641
                                 <1> lodsb_capitalize:
                                      -; 07/10/2016 - TRDOS 386 (TRDOS v2.0)
1642
                                 <1>
1643
                                 <1>
                                          ; INPUT -> [esi] = character
1644
                                 <1>
                                                    edi = destination
                                        ; OUTPUT -> AL contains capitalized character
1645
                                 <1>
                                              esi = esi+1
1646
                                 <1>
                                                    edi = edi+1
1647
                                 <1>
                                          ;
                                         ;
lodsb
1648
                                 <1>
1649 00010109 AC
                                 <1>
                                          cmp al, 61h
1650 0001010A 3C61
                                <1>
1651 0001010C 7206
                                <1>
                                           jb short lodsb_cap_retn
1652 0001010E 3C7A
                                <1>
                                           cmp al, 7Ah
1653 00010110 7702
                                <1>
                                           ja short lodsb_cap_retn
1654 00010112 24DF
                                <1>
                                          and al, ODFh
1655
                                 <1> lodsb_cap_retn:
1656 00010114 AA
                                <1>
                                          stosb
1657 00010115 C3
                                 <1>
                                           retn
1658
                                 <1>
1659
                                 <1> device_open:
                                      ; 08/10/2016 - TRDOS 386 (TRDOS v2.0)
1660
                                 <1>
1661
                                 <1>
                                          ; Complete device opening work for sysopen (device)
1662
                                 <1>
                                          ; INPUT ->
1663
                                 <1>
                                         ; EAX = Device Number (AL)
1664
                                 <1>
1665
                                 <1>
                                                  CL = Open mode (1 = read, 2 = write)
1666
                                 <1>
                                                 CH = Device access byte (bit 0 = 0)
                                          ; OUTPUT ->
1667
                                 <1>
                                         ; EAX = Device Number
1668
                                 <1>
1669
                                 <1>
                                                 CF = 0 -> device has been opened
                                                CF = 1 -> device could not be opened
1670
                                 <1>
                                          ;
1671
                                 <1>
1672
                                 <1>
                                           ; Modified registers: ebx, (edx, ecx, esi, edi, ebp)
1673
                                 <1>
1674
                                 <1>
1675 00010116 89C3
                                 <1>
                                                 ebx, eax
                                          mov
1676 00010118 66C1E302
                                 <1>
                                           shl
                                                 bx, 2 ; *4
                                 <1>
1678 0001011C F6C580
                                 <1>
                                           test ch, 80h; bit 7, installable device driver flag
1679 0001011F 7406
                                 <1>
                                                 short d_open_2 ; Kernel device
                                           jz
                                           ; installable device
1680
                                 <1>
1681
                                 <1> d_open_1:
                                           jmp dword [ebx+IDEV_OADDR-4]
1682 00010121 FFA3[F4650100]
                                 <1>
                                 <1> d_open_2:
1683
1684 00010127 FFA3[3E150100]
                                                 dword [ebx+KDEV_OADDR-4]
                                 <1>
                                           jmp
1685
                                 <1>
1686
                                 <1> device_close:
1687
                                 <1>
                                         ; 08/10/2016 - TRDOS 386 (TRDOS v2.0)
1688
                                 <1>
                                           ; Complete device closing work for sysclose (device)
1689
                                 <1>
                                          ; INPUT ->
1690
                                 <1>
                                                 EAX = Device Number (AL)
1691
                                 <1>
1692
                                 <1>
                                                   CL = Open mode (1 = read, 2 = write)
                                                  CH = Device access byte (bit 0 = 0)
1693
                                 <1>
1694
                                 <1>
                                          ; OUTPUT ->
1695
                                 <1>
                                                 EAX = Device Number
1696
                                 <1>
                                                 CF = 0 -> device has been closed
1697
                                 <1>
                                                 CF = 1 -> device could not be closed
1698
                                 <1>
1699
                                 <1>
                                           ; Modified registers: ebx, (edx, ecx, esi, edi, ebp)
1700
                                 <1>
1701
                                 <1>
1702 0001012D 89C3
                                 <1>
                                                 ebx, eax
                                           mov
1703 0001012F 66C1E302
                                 <1>
                                           shl
                                                 bx, 2 ; *4
1704
                                 <1>
1705 00010133 F6C580
                                 <1>
                                           test ch, 80h; bit 7, installable device driver flag
```

```
<1> jz short d_close_
<1> ; installable device
                                                 short d_close_2 ; Kernel device
1707
1708
                                  <1> d_close_1:
1709 00010138 FFA3[14660100]
                                  <1> jmp dword [ebx+IDEV_CADDR-4]
                                  <1> d close 2:
1710
1711 0001013E FFA3[8E150100]
                                  <1>
                                          jmp
                                                  dword [ebx+KDEV_CADDR-4]
1712
                                  <1>
1713
                                  <1> rnull:
                                          ; 07/10/2016 - TRDOS 386 (TRDOS v2.0)
1714
                                  <1>
1715
                                  <1>
                                            ; read null (read from null device)
1716 00010144 C3
                                  <1>
                                   <1>
1717
                                  <1> wnull:
1718
                                        ; 07/10/2016 - TRDOS 386 (TRDOS v2.0)
1719
                                  <1>
1720
                                  <1>
                                            ; write null (write to null device)
1721 00010145 C3
                                  <1>
                                            retn
1722
                                   <1>
                                  <1> dev_IRQ_service:
1723
                                         ; 12/05/2017
1724
                                  <1>
                                            ; 13/04/2017
1725
                                  <1>
                                          ; 27/02/2017 - TRDOS 386 (TRDOS v2.0)
1726
                                  <1>
                                          ; INPUT ->
; AL =
1727
                                   <1>
                                                   AL = IRQ Number (0 to 15)
1728
                                  <1>
1729
                                  <1>
                                          push ebx
1730 00010146 53
                                  <1>
1731 00010147 0FB6D8
                                  <1>
                                            movzx ebx, al
1732 0001014A C0E302
                                  <1>
                                           shl bl, 2; * 4
                                          mov ebx, [ebx+DEV_INT_HNDLR]
and ebx, ebx
jz short dIRQ_s_retn
1733 0001014D 8B9B[B26A0100]
                                 <1>
1734 00010153 21DB
                                  <1>
1735 00010155 7404
                                  <1>
1736 00010157 50
                                  <1>
                                          push eax
1737
                                  <1>
1738 00010158 FFD3
                                  <1>
                                            call ebx
1739
                                  <1>
1740 0001015A 58
                                  <1>
                                            pop
                                                  eax
                                  <1> dIRQ_s_retn:
1741
1742 0001015B 5B
                                  <1>
                                            pop ebx
1743 0001015C C3
                                  <1>
                                             retn
1744
                                  <1>
1745
                                   <1>
1746
                                   <1> set_dev_IRQ_service:
                                         ; 13/04/2017 - TRDOS 386 (TRDOS v2.0)
1747
                                   <1>
1748
                                   <1>
                                           ; Set Device Interrupt Service
1749
                                   <1>
1750
                                   <1>
1751
                                   <1>
                                            ; INPUT ->
1752
                                            ; AL = IRQ Number
                                   <1>
1753
                                   <1>
                                                   EBX = Hardware Interrupt Service Address
1754
                                   <1>
1755
                                   <1>
                                            ; Note: There is not a validation check here
                                          ; because this procedure is called by
1756
                                   <1>
1757
                                   <1>
                                                   TRDOS 386 kernel!
                                            ;
1758
                                   <1>
                                             ;
                                                    (Even if a device driver does not exist
1759
                                   <1>
                                                 this setting may be used by sysaudio
1760
                                   <1>
                                                   and other system calls for hardware
                                           ;
1761
                                   <1>
                                                   components which use IRQ method for I/O.)
1762
                                   <1>
                                           ;push esi
1763
                                  <1>
1764 0001015D 0FB6F0
                                  <1>
                                            movzx esi, al
                                            shl si, 2; * 4
1765 00010160 66C1E602
                                  <1>
1766 00010164 899E[B26A0100]
                                  <1>
                                            mov [esi+DEV_INT_HNDLR], ebx
                                   <1>
                                            ;pop esi
1768 0001016A C3
                                   <1>
                                            retn
1769
                                   <1>
1770
                                   <1>
                                   <1> sysaudio: ; AUDIO FUNCTIONS
1771
                                       ; 28/07/2020
1772
                                   <1>
                                           ; 27/07/2020
1773
                                   <1>
                                           ; 10/10/2017
; 22/06/2017
1774
                                   <1>
1775
                                   <1>
                                           ; 28/05/2017, 04/06/2017, 05/06/2017, 10/06/2017
1776
                                   <1>
                                          ; 01/05/2017, 12/05/2017, 15/05/2017, 20/05/2017; 21/04/2017, 22/04/2017, 23/04/2017, 24/04/2017; 10/04/2017, 13/04/2017, 14/04/2017, 16/04/2017
1777
                                   <1>
1778
                                   <1>
1779
                                   <1>
                                           ; 03/04/2017 (VIA VT8237R)
; 01/04/2016 (trdosk6.s -> tdosk8.s)
1780
                                   <1>
1781
                                   <1>
1782
                                   <1>
                                            ; 16/05/2016 - TRDOS 386 (TRDOS v2.0)
1783
                                   <1>
1784
                                   <1>
                                            ; Inputs:
1785
                                   <1>
1786
                                   <1>
                                                    BH = 0 \rightarrow Beep (PC Speaker)
1787
                                   <1>
                                                        BL = Duration Counter (1 for 1/64 second)
                                                         CX = Frequency Divisor (1193180/Frequency)
1788
                                   <1>
1789
                                   <1>
                                                            (1331 for 886 Hz)
1790
                                   <1>
1791
                                                   01/04/2017
                                   <1>
1792
                                   <1>
1793
                                   <1>
                                                   BH = 1 -> DETECT (& ENABLE) AUDIO DEVICE
1794
                                   <1>
                                                        BL = 0 : PC SPEAKER
                                                          1 : SOUND BLASTER 16
1795
                                   <1>
1796
                                   <1>
                                                           2 : INTEL AC'97
1797
                                   <1>
                                                            3 : VIA VT8237R (VT8233)
1798
                                                            4 : INTEL HDA
                                   <1>
1799
                                   <1>
                                                         5-FEh : unknown/invalid
1800
                                   <1>
                                                           ; 04/06/2017
                                                          FFh : Get current audio device id
1801
                                   <1>
1802
                                   <1>
                                                   BH = 2 -> ALLOCATE AUDIO BUFFER (for user)
1803
                                   <1>
1804
                                   <1>
                                                         ECX = Audio Buffer Size (must be equal to
1805
                                   <1>
                                                                the half of DMA buffer size)
1806
                                   <1>
                                                          EDX = Virtual Address of the buffer
1807
                                   <1>
                                                                (This is not DMA buffer!)
1808
                                   <1>
                                                   BH = 3 -> INITIALIZE AUDIO DEVICE
1809
                                   <1>
1810
                                   <1>
                                                        BL = 0,2 -> for Signal Response Byte
```

1706 00010136 7406

```
CL = Signal Response Byte Value (fixed)
1812
                                  <1>
                                                                      if BL = 0
1813
                                  <1>
                                                                auto increment of S.R.B. value
1814
                                  <1>
                                                                      if BL = 2
1815
                                  <1>
                                                          EDX = Signal Response (Return) Byte Address
1816
                                  <1>
                                                       BL = 1 for CallBack Method
1817
                                  <1>
1818
                                  <1>
                                                        EDX = CallBack Service Address (Virtual)
1819
                                  <1>
1820
                                  <1>
                                                       BL > 2 -> invalid function
1821
                                  <1>
1822
                                                       (Audio buffer must be allocated before
                                  <1>
1823
                                  <1>
                                                       initialization.)
1824
                                  <1>
                                                   BH = 4 -> START TO PLAY
1825
                                  <1>
1826
                                  <1>
                                                       BL = Mode
                                                        Bit 0 = mono/stereo (1 = stereo)
1827
                                  <1>
                                                           Bit 1 = 8 bit / 16 bit (1 = 16 bit)
1828
                                  <1>
1829
                                  <1>
                                                        CX = Sampling Rate (Hz)
1830
                                  <1>
                                                  BH = 5 \rightarrow PAUSE
1831
                                  <1>
1832
                                  <1>
                                                       BL = Any
1833
                                  <1>
1834
                                  <1>
                                                   BH = 6 -> CONTINUE TO PLAY
1835
                                  <1>
                                                      BL = Any
1836
                                  <1>
1837
                                  <1>
                                                   BH = 7 \rightarrow STOP
1838
                                  <1>
                                                     BL = Any
1839
                                  <1>
1840
                                                   BH = 8 \rightarrow RESET
                                  <1>
1841
                                  <1>
                                                       BL = Any
1842
                                  <1>
                                                   BH = 9 -> CANCEL (CALLBACK or S.R.B. SERVICE)
1843
                                  <1>
1844
                                  <1>
                                                      BL = Any
1845
                                  <1>
                                                   BH = 10 -> DEALLOCATE AUDIO BUFFER (for user)
1846
                                  <1>
                                                       BL = Any
1847
                                  <1>
1848
                                  <1>
1849
                                  <1>
                                                   BH = 11 -> SET VOLUME LEVEL
1850
                                  <1>
                                                     BL: (Bit 0 to 6)
1851
                                  <1>
                                                         0 = Master (Playback, Lineout) volume
                                                       CL = Left Channel Volume
1852
                                  <1>
1853
                                  <1>
                                                       CH = Right Channel Volume
1854
                                  <1>
1855
                                  <1>
                                                        Note: If BL \ge 80h (Bit 7 of BL is set),
1856
                                  <1>
                                                        volume level will be set for next playing
                                                        (actual volume level will not be changed
1857
                                  <1>
1858
                                                       immediately)
                                  <1>
1859
                                  <1>
1860
                                  <1>
                                                   BH = 12 -> DISABLE AUDIO DEVICE
                                            ;
1861
                                  <1>
                                                       (reset audio device and unlink dma buffer)
1862
                                  <1>
                                                        BL = Any
1863
                                  <1>
1864
                                  <1>
                                                  12/05/2017
1865
                                  <1>
                                                   BH = 13 -> MAP DMA BUFFER TO USER
1866
                                  <1>
                                                       (for direct access to system's dma buffer)
1867
                                  <1>
1868
                                  <1>
                                                        ECX = map size in bytes
1869
                                  <1>
                                                           (will be rounded up to page borders)
1870
                                  <1>
                                                        EDX = Virtual Address of the buffer
1871
                                  <1>
                                                           (Will be rounded up to page borders)
1872
                                  <1>
1873
                                  <1>
                                                   05/06/2017
                                            ;
1874
                                  <1>
                                                   04/06/2017
1875
                                  <1>
                                                   BH = 14 -> GET AUDIO DEVICE INFO
1876
                                  <1>
                                                      BL: 0 = Audio Controller Info
                                                         > 0 = Invalid for now!
1877
                                  <1>
1878
                                  <1>
1879
                                  <1>
                                                   22/06/2017
1880
                                  <1>
                                                   BH = 15 -> GET CURRENT SOUND DATA (for graphics)
1881
                                  <1>
                                                      BL: 0 -> PCM OUT data
                                                         > 0 -> Invalid for now!
1882
                                  <1>
                                                      ECX = 0 -> Get DMA Buffer Pointer
1883
                                  <1>
1884
                                  <1>
                                                         EDX = Not Used
1885
                                  <1>
                                                        ECX > 0 -> Byte count for buffer (EDX)
                                                           EDX = Buffer Address (Virtual)
1886
                                  <1>
1887
                                  <1>
                                                   10/10/2017
1888
                                  <1>
                                                   BH = 16 -> UPDATE DMA BUFFER DATA
1889
                                  <1>
                                                            (by using the Audio Buffer content)
1890
                                  <1>
1891
                                  <1>
                                                        BL = 0 : Update dma half buffer in sequence
1892
                                                                (automatic destination)
                                  <1>
                                                            1 : Update 1st half of the dma buffer
1893
                                  <1>
1894
                                  <1>
                                                            2 : Update 2nd half of the dma buffer
1895
                                  <1>
                                                            3-FEh: Invalid!
                                                            FFh = Get current flag value
1896
                                  <1>
1897
                                  <1>
                                                                (Half buffer number -1)
1898
                                  <1>
1899
                                  <1>
1900
                                  <1>
                                            ; Outputs:
1901
                                  <1>
1902
                                  <1>
                                                   For BH = 0 \rightarrow Beep
1903
                                  <1>
                                                      None
1904
                                  <1>
1905
                                  <1>
                                                   01/04/2017
1906
                                  <1>
                                                   For BH = 1 -> DETECT (& ENABLE) AUDIO DEVICE
1907
                                  <1>
                                            ;
1908
                                  <1>
                                                      AH = 0 : PC SPEAKER
1909
                                  <1>
                                                          1 : SOUND BLASTER 16
1910
                                  <1>
                                                          2 : INTEL AC'97
1911
                                  <1>
                                                          3 : VIA VT8237R (VT8233)
1912
                                  <1>
                                                          4 : INTEL HDA
1913
                                                         5-FFh : unknown/invalid
                                  <1>
                                                       AL = mode status
1914
                                  <1>
1915
                                  <1>
                                                         bit 0 = mono /stereo (1 = stereo)
```

```
1916
                                  <1>
                                                          bit 1 = 8 bit / 16 bit ( 1 = 16 bit)
1917
                                  <1>
                                                      04/06/2017
                                                      EBX = PCI DEVICE/VENDOR ID (if >0)
1918
                                  <1>
1919
                                  <1>
                                                          (BX = VENDOR ID)
1920
                                  <1>
                                                       (if CF = 1 -> Error code in EAX)
1921
                                  <1>
                                                  For BH = 2 -> ALLOCATE AUDIO BUFFER (for user)
1922
                                  <1>
                                                      EAX = Physical Address of the buffer
1923
                                  <1>
1924
                                  <1>
                                                       (if CF = 1 -> Error code in EAX)
1925
                                  <1>
                                                  For BH = 3 -> INITIALIZE AUDIO DEVICE
1926
                                  <1>
1927
                                                      (if CF = 1 -> Error code in EAX)
                                  <1>
1928
                                  <1>
1929
                                  <1>
                                                  For BH = 4 -> START TO PLAY
                                                      none (if CF = 1 -> Error code in EAX)
1930
                                  <1>
1931
                                  <1>
                                                  For BH = 5 \rightarrow PAUSE
1932
                                  <1>
1933
                                  <1>
                                                      none (if CF = 1 -> Error code in EAX)
1934
                                  <1>
1935
                                  <1>
                                                  For BH = 6 -> CONTINUE TO PLAY
                                                      none (if CF = 1 -> Error code in EAX)
1936
                                  <1>
1937
                                  <1>
1938
                                                  For BH = 7 \rightarrow STOP
                                  <1>
1939
                                  <1>
                                                      none (if CF = 1 -> Error code in EAX)
1940
                                  <1>
1941
                                  <1>
                                                  For BH = 8 \rightarrow RESET
1942
                                  <1>
                                                     none (if CF = 1 -> Error code in EAX)
1943
                                  <1>
1944
                                  <1>
                                                  For BH = 9 -> CANCEL (CALLBACK or S.R.B. SERVICE)
1945
                                                      none (if CF = 1 -> Error code in EAX)
                                  <1>
1946
                                  <1>
                                                  For BH = 10 -> DEALLOCATE AUDIO BUFFER (for user)
1947
                                  <1>
                                                      none (if CF = 1 -> Error code in EAX)
1948
                                  <1>
1949
                                  <1>
1950
                                  <1>
                                                  For BH = 11 -> SET VOLUME LEVEL
1951
                                  <1>
                                                      none (if CF = 1 -> Error code in EAX)
1952
                                  <1>
1953
                                  <1>
                                                  For BH = 12 -> DISABLE AUDIO DEVICE
1954
                                  <1>
                                                      none (if CF = 1 -> Error code in EAX)
1955
                                  <1>
                                                  12/05/2017
1956
                                  <1>
1957
                                  <1>
                                                  For BH = 13 -> MAP DMA BUFFER TO USER
                                                      EAX = Physical Address of the buffer
1958
                                  <1>
                                                      (if CF = 1 -> Error code in EAX)
1959
                                  <1>
1960
                                  <1>
1961
                                  <1>
                                                  04/06/2017
                                                  For BH = 14 -> GET AUDIO DEVICE INFO
1962
                                  <1>
1963
                                  <1>
                                                  (for BL = 0) ; 05/06/2017
                                                   EAX = IRQ Number in AL
1964
                                  <1>
                                                         Audio Device Number in AH
1965
                                  <1>
1966
                                  <1>
                                                    EBX = DEV/VENDOR ID
1967
                                  <1>
                                                          (DDDDDDDDDDDDDDDVVVVVVVVVVVVVVV)
                                                    ECX = BUS/DEV/FN
1968
                                  <1>
1969
                                  <1>
                                                         (00000000BBBBBBBBBDDDDDFFF00000000)
1970
                                  <1>
                                                      EDX = NABMBAR/NAMBAR (for AC97)
1971
                                  <1>
                                                         (Low word, DX = NAMBAR address)
                                                      EDX = Base IO Addr (DX) for SB16 & VT8233
1972
                                  <1>
1973
                                  <1>
                                                      (if CF = 1 -> Error code in EAX)
1974
                                  <1>
                                                                   (ERR_DEV_NOT_RDY = 15)
1975
                                  <1>
                                                  22/06/2017
1976
                                  <1>
                                                  For BH = 15 -> GET CURRENT SOUND DATA
1977
                                  <1>
1978
                                  <1>
                                                                 (for graphics)
1979
                                  <1>
                                                  (for BL = 0)
                                                   If ECX input is 0
1980
                                  <1>
1981
                                  <1>
                                                      EAX = DMA Buffer Current Position (Offset)
1982
                                  <1>
                                                   If ECX input > 0
1983
                                  <1>
                                                    EAX = Actual transfer count
1984
                                  <1>
                                                      (Sound samples will be copied from
1985
                                  <1>
                                                      Current DMA Buffer Position to EDX
1986
                                  <1>
                                                       virtual address as EAX bytes.)
1987
                                  <1>
                                                   ((If CF = 1 -> Error code in EAX))
1988
                                  <1>
1989
                                  <1>
1990
                                  <1>
                                                  10/10/2017
                                                  For BH = 16 \rightarrow UPDATE DMA BUFFER DATA
1991
                                  <1>
1992
                                  <1>
                                                    EAX = 0, if the updated (or current)
1993
                                  <1>
                                                             half buffer is DMA half buffer 1
1994
                                  <1>
                                                      EAX = 1, if the updated (or current)
1995
                                                             half buffer is DMA half buffer 2
                                  <1>
                                                      (If CF = 1 -> Error code in EAX)
1996
                                  <1>
1997
                                  <1>
                                  <1>
1999 0001016B 80FF11
                                  <1>
                                                   bh, AUDIO1L/4
2000 0001016E 0F83EEC5FFFF
                                  <1>
                                            jnb
                                                  sysret
2001
                                  <1>
2002 00010174 C0E702
                                            shl bh, 2; *4
                                  <1>
2003 00010177 OFB6F7
                                  <1>
                                            movzx esi, bh
2004
                                  <1>
2005
                                  <1>
                                            ; 22/04/2017
2006 0001017A 31C0
                                  <1>
                                            xor
                                                   eax, eax
                                  <1>
                                                   [u.r0], eax ; 0
2007 0001017C A3[64030300]
                                            mov
                                  <1>
2009 00010181 FF96[8C010100]
                                  <1>
                                            call
                                                  dword [esi+AUDIO1]
2010
                                  <1>
                                            ;ic
                                                  error
2011 00010187 E9D6C5FFFF
                                  <1>
                                            jmp
                                                  sysret
2012
                                  <1>
2013 0001018C [AA1D0000]
                                  <1> AUDIO1:
                                                        beep ; FUNCTION = 0 (bl = Duration Counter
                                                   dd
2014
                                  <1>
                                                                           cx = Frequency Divisor
2015 00010190 [D0010100]
                                  <1>
                                                   soundc detect
                                            dd
                                                   sound alloc
2016 00010194 [6C020100]
                                  <1>
                                            dd
2017 00010198 [2A030100]
                                  <1>
                                            dd
                                                   soundc_init
2018 0001019C [E2040100]
                                  <1>
                                            dd
                                                   sound_play
2019 000101A0 [7E050100]
                                  <1>
                                            dd
2020 000101A4 [A8050100]
                                  <1>
                                            dd
                                                   sound_continue
```

```
2021 000101A8 [D2050100]
                                <1>
                                          dd
                                                sound_stop
2022 000101AC [FB050100]
                                <1>
                                          dd
                                                soundc reset
2023 000101B0 [2C060100]
                                                soundc cancel
                                <1>
                                          dd
2024 000101B4 [52060100]
                               <1>
                                          dd
                                                sound dalloc
2025 000101B8 [7D060100]
                                <1>
                                          dd
                                                sound volume
2026 000101BC [CF060100]
                                <1>
                                          dd
                                                soundc_disable
2027 000101C0 [41070100]
                                                sound dma map
                               <1>
                                        dd
                                <1>
2028 000101C4 [B0070100]
                                          dd
                                                soundc_info
2029 000101C8 [0F080100]
                                <1>
                                          dd
                                                sound data
2030 000101CC [BC080100]
                                <1>
                                          dd
                                                sound_update
2031
                                <1>
2032
                                <1> AUDIO1L
                                                EQU $ - AUDIO1
2033
                                <1>
2034
                                <1> soundc_detect:
                                       ; FUNCTION = 1
2035
                                <1>
2036
                                 <1>
                                          ; bl = Audio device type number
                                          ; (0= pc speaker, 1 = sound blaster 16, 2 = intel ac97
2037
                                 <1>
                                <1>
                                        ; 3 = via vt823x, 4 = intel HDA, 0FFh = any)
2038
2039
                                <1>
                                         ; 04/06/2017
2040
                                <1>
2041 000101D0 8A25[416B0100]
                                <1>
                                        mov
                                                ah, [audio_device]
2042 000101D6 80FBFF
                                                bl, OFFh; get current audio device id
                                <1>
                                          cmp
2043 000101D9 7408
                                <1>
                                          jе
                                                short sysaudio0
2044
                                <1>
2045 000101DB 20E4
                                <1>
                                          and
                                                ah, ah
2046 000101DD 741E
                                <1>
                                          jz
                                                short soundc_get_dev
                                <1>
                                          cmp
2048 000101DF 38DC
                                <1>
                                                ah, bl
2049 000101E1 7567
                                <1>
                                                short soundc_dev_err
                                          jne
2050
                                <1>
2051
                                <1> sysaudio0:
2052 000101E3 A0[426B0100]
                                <1> mov
                                                al, [audio_mode]
                                <1> sysaudio1:
2053
2054 000101E8 A3[64030300]
                                <1> mov
                                                [u.r0], eax
                                                ebx, [audio_vendor] ; (DEVICE/VENDOR ID)
2055 000101ED 8B1D[4C6B0100]
                                <1>
                                          mov
2056 000101F3 8B2D[60030300]
                                <1>
                                          mov
                                                 ebp, [u.usp]
2057 000101F9 895D10
                                <1>
                                         mov
                                                [ebp+16], ebx ; ebx
2058 000101FC C3
                                <1>
                                        retn
2059
                                <1>
                                <1> soundc_get_dev:
2060
                                      ; 28/05/2017
2061
                                <1>
2062
                                <1>
                                          ; 03/04/2017, 24/04/2017
                                      mov byte [audio_pci], 0 cmp bl, 3; VIA VT8233 (VT8237R) Audio Controller & AC97 Codec
2063 000101FD C605[406B0100]00
                               <1>
2064 00010204 80FB03
                                <1>
                                       ;jne short soundc_get_dev_sb
2065
                                <1>
                                          ; 28/05/2017
2066
                                <1>
                                        jb short soundc_get_dev_sb
2067 00010207 7220
                                <1>
2068 00010209 773F
                                <1>
                                                short soundc_dev_err ; temporary (28/05/2017)
                                        jа
2069
                                <1>
                                       call DetectVT8233
jc short soundc_dev_err
2070 0001020B E848180000
                                <1>
2071 00010210 7238
                                <1>
                                        ; eax = 0
2072
                                <1>
2073
                                <1>
2074
                                <1>
                                        ;mov ebx, [audio_vendor]
2075
                                <1>
                                         ; ebx = DEVICE/VENDOR ID
2076
                                <1>
                                                DDDDDDDDDDDDDVVVVVVVVVVVVVVV
                                <1>
2078 00010212 B003
                                <1>
                                          mov al, 3 ; VIA VT8237R (VT3233) Audio Controller
2079 00010214 88C4
                                <1>
                                                ah, al
                                          mov
2080
                                <1>
                                <1> soundc_get_pci_dev_ok: ; 28/05/2017
<1> inc byte [audio_pci] ; = 1
2081
2082 00010216 FE05[406B0100]
2083
                                <1> soundc_get_dev_ok:
2084
                                <1>
                                <1> soundc_get_dev_sb16_ok:
2085
2086 0001021C A2[416B0100]
                                <1>
                                       mov [audio_device], al
                                              [audio mode], ah; stereo (bit0), 16 bit (bit1) capability
2087 00010221 8825[426B0100]
                                <1>
                                          mov
2088 00010227 EBBF
                                <1>
                                          jmp short sysaudio1
2089
                                <1>
2090
                                <1> soundc_get_dev_sb:
                                      ; 24/04/2017
2091
                                <1>
2092 00010229 80FB01
                                <1>
                                          cmp bl, 1; Sound Blaster 16
2093 0001022C 750E
                                                short soundc_get_dev_ich ; 28/05/2017
                                <1>
                                          jne
2094
                                <1>
                                         call DetectSB
2095 0001022E E8491D0000
                                <1>
2096 00010233 7215
                                <1>
                                          jс
                                                short soundc_dev_err
2097 00010235 B801030000
                                <1>
                                                eax, 0301h; Sound Blaster 16
                                          mov
2098 0001023A EBE0
                                <1>
                                          jmp short soundc_get_dev_sb16_ok
2099
                                <1>
2100
                                 <1> soundc get dev ich:
                                          ; 28/05/<del>2</del>017
2101
                                 <1>
                                               bl, 2 ; Intel AC'97 Audio Controller (ICH)
2102
                                 <1>
                                          ;cmp
                                          ;jne short soundc_dev_err ; Temporary (28/05/2017)
2103
                                 <1>
2104
                                 <1>
                                                                 ; (Here will be modified just after
                                                                  ; new sound card code will be ready!)
2105
                                <1>
2106 0001023C E80A180000
                                          call DetectICH
                                <1>
2107 00010241 7207
                                <1>
                                          jc short soundc_dev_err
2108
                                <1>
                                               eax, 0302h ; AC'97 (ICH)
2109 00010243 B802030000
                                <1>
                                          mov
                                          jmp short soundc_get_pci_dev_ok
2110 00010248 EBCC
                                <1>
2111
                                <1>
                                <1> soundc_dev_err:
2112
                                          mov eax, ERR_DEV_NOT_RDY; Device not ready!
2113 0001024A B80F000000
                                <1>
2114 0001024F EB0C
                                <1>
                                          jmp short sysaudio_err
2115
                                <1>
2116
                                <1> sound_buff_error:
2117 00010251 B82E000000
                               <1> mov eax, ERR BUFFER; Buffer error!
2118 00010256 EB05
                                <1>
                                          jmp short sysaudio err
2119
                                <1>
                                <1> soundc respond err:
2120
                                       ; ERR_TIME_OUT ; 'time out !' error
2121
                                <1>
2122 00010258 B819000000
                                <1>
                                          mov eax, ERR_DEV_NOT_RESP; 'device not responding!' error
                                <1> sysaudio_err:
2123
2124 0001025D A3[64030300]
                                <1> mov [u.r0], eax
2125 00010262 A3[C8030300]
                                <1>
                                          mov [u.error], eax
```

```
2126 00010267 E9D6C4FFFF
                                  <1>
                                             jmp error
                                   <1>
                                   <1> sound alloc:
2128
2129
                                   <1> ; FUNCTION = 2
                                           ; ecx = audio buffer size (in bytes)
2130
                                   <1>
                                           ; edx = audio buffer address (virtual)
; 27/07/2020
2131
                                   <1>
                                        , 21/01/2020
; 28/05/2017
; 01/05/2017, 15/05/2017
; 21/04/2017, 24/04/2017
cmp byte [audio_pci], 0
ja short snd_alloc_0
; Max. 64KB DMA buffer | | | | |
2132
                                   <1>
2133
                                   <1>
2134
                                   <1>
2135
                                   <1>
2136 0001026C 803D[406B0100]00 <1>
2137 00010273 7708
                                   <1>
                                   <1>
2139 00010275 81F900800000
                                         cmp ecx, 32768
ja short sound
                                  <1>
2140 0001027B 77D4
                                  <1>
                                                   short sound_buff_error
                                  <1> snd_alloc_0:
2141
<1> ; 15/05/2017
2143 0001027D 81F900100000
2144 00010283 72CC
2145
                                            cmp ecx, 4096; PAGE_SIZE
                                                    short sound_buff_error
                                                    eax, [audio_buffer] ; audio buffer address (current)
                                  2148 00011
2149
2150 0001028E 8A1D[B3030300]
2151 00010294 3A1D[696B0100]
2152 0001029A 0F85FC000000
                                 <1> cmp  eax, edx; same virtual buffer address?
<1> jne  short snd_alloc_1
<1> cmp  ecx, [audio_buff_size]
<1> je  short snd_alloc_3; Nothing to do!
2153 000102A0 39D0
2154 000102A2 7508
2155 000102A4 3B0D[5C6B0100]
2156 000102AA 746C
2157
                                   <1>
                                                       ; Buffer has been set already!
2158
                                  <1> snd_alloc_1:
ebx, eax; audio buffer address (current)
                                                    [audio buffer], eax ; 0
                                                     [audio p buffer], eax ; 0
2172 000102D3 89D3
                                  <1> mov
                                                   ebx, edx
                                             ; 01/05/2017
                                  <1>
2173
2174 000102D5 BA00F0FFFF
                                  <1>
                                             mov edx, ~PAGE_OFF; truncating page offsets
2175
                                  <1>
                                                                 ; for aligning to page borders
                                        ; and eax, edx and ebx, edx and ecx, edx
2176
                                  <1>
2177 000102DA 21D3
                                  <1>
                                         , EAX = Beginning address (physical)
; EAX = 0 -> Allocate mem block from the 1st proper aperture
; ECX = Number of bytes to be allocated
call allocate_memory_block
jc sound_buff_error
; EAX = ph...
2178 000102DC 21D1
                                  <1>
2179
                                  <1>
2180
                                   <1>
2181
                                   <1>
                                   <1>
2183 000102DE E89E51FFFF
                                   <1>
2184 000102E3 0F8268FFFFFF
                                   <1>
                                   <1>
                                             ; EAX = Physical address of the allocated memory block
2185
2186
                                   <1>
                                           ; ECX = Allocated bytes (as truncated to page border)
                                            ; EBX = Virtual address (as truncated to page border)
2187
                                   <1>
2188 000102E9 50
                                  <1>
                                             push eax
2189 000102EA 53
                                  <1>
2190 000102EB 51
                                   <1>
                                             push ecx
2191 000102EC E8E055FFFF
                                  <1>
                                             call allocate_user_pages
2192 000102F1 59
                                  <1>
                                             pop
                                                   ecx
2193 000102F2 5B
                                  <1>
                                           pop
                                                   ebx
                                           pop
2194 000102F3 58
                                   <1>
2195 000102F4 722A
                                             jc short snd_alloc_4 ; insufficient memory, buff error
                                  <1>
                                           ; eax = physical address of the user's audio buffer
2196
                                  <1>
                                           ; ebx = virtual address of the user's audio buffer
2197
                                   <1>
                                             ; ecx = buffer size (in bytes)
2198
                                   <1>
                                           mov [audio_p_buffer], eax
2199 000102F6 A3[586B0100]
                                  <1>
                                           mov
                                                   [audio_buffer], ebx [audio_buff_size], ecx
2200 000102FB 891D[546B0100]
                                   <1>
2201 00010301 890D[5C6B0100]
                                  <1>
                                             mov
2202 00010307 8A15[B3030300]
                                   <1>
                                                   dl, [u.uno]
                                             mov
2203 0001030D 8815[696B0100]
                                   <1>
                                             mov
                                                   [audio_user], dl
2204 00010313 A3[64030300]
                                   <1>
                                             mov
                                                    [u.r0], eax
2205
                                   <1> snd_alloc 3:
                                         ; 27/07/2020
2206
                                   <1>
2207 00010318 C605[686B0100]00
                                                  byte [audio_flag], 0 ; clear dma half buffer flag
                                    <1>
                                             mov
2208
                                   <1>
2209 0001031F C3
                                   <1>
                                              retn
                                   <1> snd alloc 4:
2210
                                             ; 15/05/2017
2211
                                   <1>
2212
                                   <1>
                                             ; EAX = Beginning address (physical)
2213
                                   <1>
                                             ; ECX = Number of bytes to be deallocated
                                             call deallocate_memory_block
2214 00010320 E86953FFFF
                                   <1>
2215 00010325 E927FFFFF
                                   <1>
                                             jmp sound buff error ; insufficient memory, buff error
2216
                                   <1>
                                   <1> soundc init:
2217
2218
                                            ; FUNCTION = 3
                                   <1>
2219
                                   <1>
                                             ; bl = method (0= s.r.b., 1= callback, 2= auto incr s.r.b.)
                                             ; cl = signal response byte (initial or fixed) value
2220
                                    <1>
2221
                                   <1>
                                             ; edx = signal response byte or callback address
                                             ; 27/07/2020
2222
                                    <1>
                                             ; 28/05/2017
                                   <1>
2223
2224
                                   <1>
                                             ; 12/05/2017, 20/05/2017
2225
                                   <1>
                                             ; 22/04/2017, 23/04/2017, 24/04/2017
2226
                                   <1>
                                             ; 13/04/2017, 14/04/2017, 16/04/2017, 21/04/2017
2227
                                    <1>
                                             ; 03/04/2017, 10/04/2017
2228
                                   <1>
2229 0001032A A0[416B0100]
                                   <1>
                                                     al, [audio_device]
2230 0001032F 20C0
                                   <1>
                                             and
                                                    al, al
```

```
2231 00010331 7549
                                 <1>
                                          jnz
                                                short sndc_init_6
                                 <1>
2233 00010333 C605[406B0100]00
                                <1>
                                          mov
                                                byte [audio_pci], 0
                                          push edx
2234 0001033A 52
                                 <1>
2235 0001033B 53
                                 <1>
                                          push ebx
2236 0001033C 51
                                 <1>
                                          push
                                                ecx
2237 0001033D E83A1C0000
                                <1>
                                          call DetectSB
2238 00010342 7213
                                          jс
                                                short sndc_init_8
                                <1>
2239 00010344 66B80103
                                <1>
                                                ax, 0301h ; Sound Blaster 16
                                                short sndc_init_7
2240 00010348 EB1E
                                <1>
                                          jmp
2241
                                <1>
2242
                                 <1> sndc init 11:
                                      -; 2<del>8</del>/05/2017
2243
                                <1>
                                          call DetectICH; Detect AC'97 (ICH) Audio Controller
2244 0001034A E8FC160000
                                <1>
                                                 short sndc_init_7
2245 0001034F 7217
                                <1>
                                          jс
                                                ax, 0302h; Intel AC'97 Audio Device
2246 00010351 66B80203
                                <1>
                                          mov
2247 00010355 EB0B
                                                short sndc init 12 ; (PCI device)
                                <1>
                                          jmp
2248
                                <1>
2249
                                <1> sndc_init_8:
2250 00010357 E8FC160000
                                <1> call DetectVT8233
2251
                                <1>
                                          ;jc short sndc_init_7
2252 0001035C 72EC
                                <1>
                                                sndc_init_11 ; 28/05/2017
                                          jс
2253
                                <1>
                                          ; eax = 0
2254 0001035E B003
                                <1>
                                          mov al, 3 ; VIA VT8237R (VT3233) Audio Controller
2255 00010360 88C4
                                <1>
                                        mov ah, al
2256
                                 <1>
                                <1> sndc_init_12:
2258 00010362 FE05[406B0100]
                                <1>
                                          inc byte [audio_pci] ; = 1
                                <1> sndc_init_7:
2260 00010368 59
                                <1>
                                          pop
                                                ecx
2261 00010369 5B
                                <1>
2262 0001036A 5A
                                 <1>
                                          pop
                                                 edx
2263 0001036B 0F82D9FEFFFF
                                <1>
                                                 soundc_dev_err
                                          jс
2264
                                <1>
2265 00010371 A2[416B0100]
                                 <1>
                                          mov
                                                 [audio_device], al
2266 00010376 8825[426B0100]
                                <1>
                                          mov
                                                 [audio_mode], ah; stereo (bit0), 16 bit (bit1) capability
                                 <1>
2268
                                 <1> sndc_init_6:
                                      cmp_
2269 0001037C 833D[546B0100]00
                                 <1>
                                                dword [audio_buffer], 0
2270 00010383 0F86C8FEFFFF
                                                sound_buff_error
                                <1>
                                          jna
2271
                                <1>
2272 00010389 A0[B3030300]
                                <1>
                                          mov
                                                al, [u.uno]
2273 0001038E 8A25[696B0100]
                                <1>
                                          mov
                                                ah, [audio_user]
2274 00010394 08E4
                                <1>
                                                 ah, ah
2275 00010396 7418
                                 <1>
                                          jz
                                                 short sndc_init0
2276 00010398 38E0
                                <1>
                                          cmp
                                                 al, ah
2277 0001039A 7419
                                                short sndc init1
                                <1>
                                          jе
2278
                                <1>
                                <1> sndc_owner_error:
<1> mov eax, ERR_NOT_OWNER; 'permission denied!' error
2279
2280 0001039C B80B000000
2281
                                <1> sndc_perm_error:
2282 000103A1 A3[64030300]
                                <1>
                                          mov [u.r0], eax
2283 000103A6 A3[C8030300]
                                <1>
                                          mov
                                                [u.error], eax
2284 000103AB E992C3FFFF
                                <1>
                                          jmp error
                                <1> sndc_init0:
2285
2286 000103B0 A2[696B0100]
                                <1>
                                                [audio_user], al
                                         mov
                                 <1> sndc init1:
2288 000103B5 8915[6C6B0100]
                                <1>
                                          mov [audio_cb_addr], edx
2289 000103BB 881D[6A6B0100]
                                 <1>
                                                 [audio_cb_mode], bl
                                          mov
2290 000103C1 880D[6B6B0100]
                                                [audio_srb], cl
                                 <1>
                                          mov
2291
                                 <1>
2292
                                 <1>
                                          ; 27/07/2020
2293
                                 <1>
                                          ;mov byte [audio_flag], 0 ; clear dma half buffer flag
2294
                                 <1>
                                          ; 24/04/2017
2295
                                 <1>
2296 000103C7 803D[416B0100]03
                                 <1>
                                          cmp byte [audio_device], 3; VT8233 (VT8237R)
2297 000103CE 7438
                                 <1>
                                                short sndc init 9
                                          jе
2298
                                                short soundc_respond_err ; temporary (28/05/2017)
                                 <1>
                                          ;ja
2299 000103D0 803D[416B0100]01
                                 <1>
                                                byte [audio_device], 1; SB 16
                                          cmp
                                                short sndc_init_13
2300 000103D7 7510
                                 <1>
                                          jne
                                                ebx, sb16 int handler
2301 000103D9 BB[A1210100]
                                 <1>
                                          mov
                                          ; Note: 'SbInit' is at 'Start to Play' stage
2302
                                 <1>
                                          ; 20/05/2017
2303
                                 <1>
2304 000103DE 66C705[766B0100]08- <1>
                                                 word [audio_master_volume], 0808h; 2/8
2304 000103E6 08
                                <1>
2305 000103E7 EB3F
                                 <1>
                                          jmp
                                                short sndc_init_10
                                 <1> sndc init 13:
2306
                                        ; 28/05/2017
2307
                                 <1>
2308 000103E9 803D[416B0100]02
                                 <1>
                                                byte [audio_device], 2 ; AC 97 (ICH)
2309 000103F0 0F8562FEFFFF
                                                soundc_respond_err ; temporary (28/05/2017)
                                 <1>
                                          jne
2310
                                 <1>
2311 000103F6 E8FB1E0000
                                 <1>
                                          call ac97_codec_config
2312 000103FB 0F8257FEFFFF
                                <1>
                                                 soundc_respond_err ; codec error !
                                 <1>
2314 00010401 BB[DD240100]
                                 <1>
                                          mov
                                                 ebx, ac97 int handler
2315 00010406 EB20
                                 <1>
                                          jmp
                                                 short sndc_init_10
                                 <1>
2317
                                 <1> sndc_init_9:
2318
                                 <1>
                                          ;call reset_codec
2319
                                 <1>
                                          ;; eax = 1
2320
                                 <1>
                                          ;call codec_io_w16 ; w32
2321 00010408 E8C2170000
                                 <1>
                                          call init codec ; 28/05/2017
2322 0001040D 0F8245FEFFFF
                                 <1>
                                                 soundc_respond_err ; codec error !
                                          jс
                                 <1>
2324 00010413 E8E9190000
                                 <1>
                                          call channel_reset
2325
                                 <1>
                                 <1>
                                          ; setup the Codec (actually mixer registers)
2327 00010418 E809190000
                                 <1>
                                           call codec_config ; unmute codec, set rates.
                                              soundc respond err ; codec error !
2328 0001041D 0F8235FEFFFF
                                 <1>
                                 <1>
2330 00010423 BB[931D0100]
                                <1>
                                          mov ebx, vt8233_int_handler
2331
                                 <1> sndc init 10:
                                          ; 13/04/2017
2332
                                <1>
2333 00010428 A0[436B0100]
                                 <1>
                                          mov al, [audio_intr] ; IRQ number
2334 0001042D E82BFDFFFF
                                 <1>
                                          call set_dev_IRQ_service
```

```
2335
                                 <1>
2336
                                 <1>
                                           ; SETUP (audio) INTERRUPT CALLBACK SERVICE
2337 00010432 8A1D[436B0100]
                                           mov bl, [audio_intr] ; IRQ number
                                 <1>
2338 00010438 8A3D[6A6B0100]
                                 <1>
                                                 bh, [audio_cb_mode]
                                                 bh ; 1 = \overline{\text{Signal}} Response Byte method (fixed value)
2339 0001043E FEC7
                                 <1>
                                           inc
2340
                                 <1>
                                                     ; 2 = Callback service method
                                                     ; 3 = Auto Increment S.R.B. method
2341
                                 <1>
2342 00010440 8A0D[6B6B0100]
                                 <1>
                                                cl, [audio_srb]
                                          mov
2343 00010446 8B15[6C6B0100]
                                 <1>
                                                 edx, [audio_cb_addr]
                                          mov
                                          mov al, [audio_user]
2344 0001044C A0[696B0100]
                                 <1>
                                          ; 14/04/2017
2345
                                 <1>
2346 00010451 E8E1040000
                                 <1>
                                          call set_irq_callback_service
                                          ; 16/04/2017
2347
                                 <1>
2348 00010456 A3[64030300]
                                 <1>
                                          mov [u.r0], eax
2349
                                 <1>
                                          ;jnc sysret
2350 0001045B 7316
                                 <1>
                                                 short sndc_init2 ; 21/04/2017
                                           jnc
2351
                                 <1>
                                           ;
2352 0001045D A3[C8030300]
                                                 dword [u.error], eax
                                 <1>
                                          mov
2353
                                 <1>
2354 00010462 A0[436B0100]
                                 <1>
                                                 al, [audio_intr] ; IRQ number
                                           mov
2355 00010467 31DB
                                 <1>
                                           xor
                                                  ebx, ebx; reset IRQ handler address
2356 00010469 E8EFFCFFFF
                                 <1>
                                           call
                                                set_dev_IRQ_service
2357
                                 <1>
2358 0001046E E9CFC2FFFF
                                 <1>
                                                 error
2359
                                 <1>
2360
                                 <1> sndc_init2:
                                 <1> ; 21/04/2017
2362 00010473 8B0D[5C6B0100]
                                <1>
                                           mov
                                                 ecx, [audio_buff_size] ; audio buffer size
2363 00010479 D1E1
                                 <1>
                                           shl
                                                 ecx, 1 ; *2
2364 0001047B A1[606B0100]
                                                 eax, [audio_dma_buff]
                                <1>
                                           mov
2365 00010480 21C0
                                 <1>
                                           and
                                                eax, eax
2366 00010482 7415
                                          jz
                                 <1>
                                                 short sndc init3
2367
                                 <1>
2368 00010484 8B15[646B0100]
                                <1>
                                          mov
                                                 edx, [audio_dmabuff_size] ; dma buffer size
2369 0001048A 39D1
                                 <1>
                                                 ecx, edx
                                          cmp
2370 0001048C 744D
                                 <1>
                                           jе
                                                 short sndc_init5
2371
                                 <1>
2372 0001048E 87CA
                                 <1>
                                          xchg ecx, edx
2373 00010490 E8F951FFFF
                                 <1>
                                           call
                                                deallocate_memory_block
2374 00010495 87D1
                                 <1>
                                           xchg edx, ecx
2375 00010497 31C0
                                 <1>
                                           xor
                                                 eax, eax
                                 <1> sndc init3:
2376
2377
                                 <1>
                                          ; 12/05/2017
2378 00010499 803D[416B0100]01
                                <1>
                                           cmp byte [audio_device], 1; SB 16
2379 000104A0 7515
                                 <1>
                                           jne
                                                 short sndc init4
2380 000104A2 C705[606B0100]-
                                 <1>
                                          mov
                                                 dword [audio_dma_buff], sb16_dma_buffer
2380 000104A8 [00000200]
                                 <1>
2381 000104AC C705[646B0100]0000- <1>
                                                 dword [audio_dmabuff_size], 65536
                                          mov
2381 000104B4 0100
                                 <1>
2382
                                                 eax, eax
                                 <1>
                                           ;xor
2383
                                 <1>
                                           ;mov
                                                 [u.r0], eax; 0 = no error, successful
2384 000104B6 C3
                                 <1>
                                           retn
2385
                                 <1>
2386
                                 <1> sndc_init4:
2387
                                 <1>
                                        ; EAX = Beginning address (physical)
2388
                                 <1>
                                           ; {\tt EAX} = 0 -> Allocate mem block from the 1st proper aperture
                                          ; ECX = Number of bytes to be allocated (>0)
                                 <1>
2390 000104B7 E8C54FFFFF
                                 <1>
                                          call allocate_memory_block
2391 000104BC 0F828FFDFFFF
                                 <1>
                                                 sound_buff_error
2392
                                 <1>
2393
                                 <1>
                                           ; set dma buffer address and size parameters
2394 000104C2 A3[606B0100]
                                 <1>
                                           mov [audio_dma_buff], eax ; dma buffer address
2395 000104C7 890D[646B0100]
                                                  [audio_dmabuff_size], ecx; dma buffer size
                                 <1>
                                           mov
2396
                                 <1>;
                                           ; EAX = Beginning (physical) addr of the allocated mem block
                                           ; ECX = Num of allocated bytes (rounded up to page borders)
2397
                                 <1>;
2398
                                 <1>;
                                           cmp byte [audio_pci], 0 ; AC97 audio controller ?
2399
                                 <1>;
                                                 short sndc_init4
                                           jа
                                 <1>;
2400
2401
                                 <1>;
                                           ; Sound Blaster 16 uses classic DMA
                                 <1>;
2402
                                           mov edx, eax
2403
                                 <1>;
                                           add
                                                 edx, ecx
                                                 edx, 1000000h; 1st 16 MB
2404
                                 <1>;
                                           cmp
                                 <1>;
2405
                                           jna
                                                 short sndc_init4
2406
                                 <1>;
2407
                                 <1>;
                                          ; error !
2408
                                 <1>;
                                           ; restore Memory Allocation Table Content
2409
                                 <1>;
                                           ; EAX = Beginning address (physical)
                                           ; ECX = Number of bytes to be deallocated
2410
                                 <1>;
2411
                                 <1>;
                                           call deallocate_memory_block
                                           ; reset dma buffer address and size parameters
                                 <1>;
2412
2413
                                 <1>;
                                           xor eax, eax; 0
                                                  [audio_dma_buff], eax ; 0
2414
                                 <1>;
                                           mov
                                                  [audio_dmabuff_size], ecx; 0
2415
                                 <1>;
                                           mov
2416
                                                  sound buff error
                                 <1>;
                                           jmp
2417
                                 <1>;
2418
                                 <1> ;sndc_init4:
2419 000104CD 803D[416B0100]03
                                        cmp byte [audio device], 3
                                 <1>
2420
                                 <1>
                                           ;jne short sndc_init5
2421 000104D4 7506
                                 <1>
                                                 short sndc_init14 ; 28/05/2017
2422 000104D6 E867190000
                                          call set_vt8233_bdl
                                 <1>
2423
                                 <1> sndc_init5:
                                          ;sub eax, eax; 0
2424
                                 <1>
2425
                                           ; mov [u.r0], eax; 0 = no error, successful
                                 <1>
2426 000104DB C3
                                 <1>
                                 <1> sndc_init14:
2427
2428 000104DC E82E1F0000
                                          call set_ac97_bdl
                                 <1>
                                 <1>
                                           ;jmp short sndc_init5
2430 000104E1 C3
                                 <1>
2431
                                 <1>
                                 <1> sound_play:
2432
2433
                                 <1>
                                         ; FUNCTION = 4
2434
                                 <1>
                                           ; bl = Mode

; bit 0 = mono/stereo (1 = stereo)
; bit 1 = 8 bit / 16 bit (1 = 16 bit)

                                                 bit 0 = mono/stereo (1 = stereo)
2435
                                 <1>
2436
                                 <1>
2437
                                 <1>
                                          ; cx = Sampling Rate (Hz)
```

```
2438
                                  <1>
2439
                                  <1>
                                           ; 13/06/2017
                                           ; Note: Even if Mode bits are not 11b,
2440
                                  <1>
2441
                                  <1>
                                                  AC'97 Audio Controller (&Codec)
                                  <1>
2442
                                                  will play audio samples as 16 bit, stereo
2443
                                  <1>
                                                  samples.
                                                  (Program must fill the audio buffer
2444
                                  <1>
2445
                                  <1>
                                                  as required; 8 bit samples must be converted
2446
                                  <1>
                                                  to 16 bit samples and mono samples must be
2447
                                  <1>
                                                  converted to stereo samples...)
                                            ;
2448
                                  <1>
                                           ; 28/07/2020
2449
                                  <1>
                                           ; 27/07/2020
2450
                                  <1>
                                           ; 28/05/2017
2451
                                  <1>
                                           ; 15/05/2017, 20/05/2017
                                  <1>
2452
2453
                                  <1>
                                           ; 21/04/2017, 24/04/2017
                                  <1>
                                           ; ... device check at first
2455 000104E2 A0[416B0100]
                                  <1>
                                            mov al, [audio_device]
                                                  al, al; 0; pc speaker or invalid
2456 000104E7 08C0
                                  <1>
                                            or
                                                  beeper_gfx ; 'video.s' ; temporary !
2457 000104E9 0F84B518FFFF
                                  <1>
                                            jΖ
                                            cmp al, 3; VIA VT 8237R (vt8233)
2458
                                  <1>;
                                                  short snd_play_1
2459
                                  <1>;
                                            jе
                                  <1>;
2460
                                                  al, 1 ; SB 16
                                            cmp
                                  <1>;
2461
                                            jne
                                                  soundc_dev_err ; temporary !
                                  <1> ;snd_play_0:
2462
2463
                                  <1>
                                           ; ... buffer & (buffer) owner check at second
2464 000104EF 833D[546B0100]00
                                  <1>
                                            cmp dword [audio buffer], 0
                                                  sound_buff_error
2465 000104F6 0F8655FDFFFF
                                  <1>
                                            jna
                                                  al, [u.uno]
al, [audio_user]
2466 000104FC A0[B3030300]
                                  <1>
                                            mov
2467 00010501 3A05[696B0100]
                                  <1>
                                            cmp
2468 00010507 0F858FFEFFFF
                                  <1>
                                                  sndc_owner_error
2469
                                  <1>
2470 0001050D 66890D[726B0100]
                                                   [audio_freq], cx ; sample frequency (Hertz)
                                  <1>
                                            mov
2471 00010514 88D8
                                  <1>
                                           mov
                                                  al, bl
2472 00010516 2401
                                  <1>
                                            and
                                                  al, 1 ; mono/stereo (1= stereo)
                                                  al ; channels
2473 00010518 FEC0
                                  <1>
                                            inc
2474 0001051A A2[716B0100]
                                  <1>
                                                 [audio_stmo], al ; sound channels (1 or 2)
                                           mov
2475 0001051F B008
                                  <1>
                                            mov
                                                  al, 8
2476 00010521 F6C302
                                  <1>
                                            test bl, 2; bits per sample (1= 16 bit)
2477 00010524 7402
                                  <1>
                                            jz
                                                  short snd_play_bps
2478 00010526 D0E0
                                  <1>
                                            shl
                                                  al, 1
                                  <1> snd_play_bps:
2480 00010528 A2[706B0100]
                                  <1>
                                                 [audio_bps], al
                                            mov
2481
                                  <1>
                                            ; Transfer ring 3 (user's) audio buffer content to dma buffer
2482
                                  <1>
2483 0001052D 8B3D[606B0100]
                                  <1>
                                                  edi, [audio_dma_buff] ; dma buffer (ring 0)
2484 00010533 09FF
                                  <1>
                                                   edi, edi
                                            or
2485 00010535 0F8416FDFFFF
                                  <1>
                                                   sound_buff_error
                                            jΖ
2486
                                  <1>
                                            ; 27/07/2020
2487
                                  <1>
2488
                                  <1>
2489 0001053B 8B35[586B0100]
                                  <1>
                                                  esi, [audio_p_buffer] ; physical address (ring 3)
                                           mov
2490
                                  <1>
                                            ; mov
                                                 ecx, [audio_buff_size] ; 15/05/2017
2491 00010541 8B0D[646B0100]
                                  <1>
                                                  ecx, [audio_dmabuff_size]; 27/07/2020
2492
                                  <1>
                                            ;or
                                                  ecx, ecx
2493
                                  <1>
                                                  sound_buff_error
                                            ;jz
                                            ; 28/07/2020
2494
                                  <1>
2495 00010547 D1E9
                                  <1>
                                            shr
                                                  ecx, 1 ; dma half buffer size
2496
                                  <1>
2497 00010549 8035[686B0100]01
                                                  byte [audio_flag], 1; 0 -> 1, 1 -> 0
                                  <1>
2498 00010550 7502
                                  <1>
                                                  short snd_play_0 ; [audio_flag] = 1
2499
                                  <1>
                                                                 ; fill dma half buffer 1
2500
                                  <1>
                                            ; [audio_flag] = 0
2501
                                  <1>
                                  <1>
2502
                                            ; fill dma half buffer 2
2503 00010552 01CF
                                  <1>
                                            add edi, ecx
2504
                                  <1>
                                  <1> snd_play_0:
2505
                                  <1>
2506
                                            ;rep movsb
2507 00010554 C1E902
                                  <1>
                                                  ecx, 2 ; convert byte count to dword count
                                            shr
2508 00010557 F3A5
                                  <1>
2509
                                  <1>
2510
                                            ; here, if [audio_flag] = 0, interrupt handler will update
                                  <1>
2511
                                  <1>
                                                               ; dma half buffer 2
2512
                                  <1>
                                                                ; (user's audio buffer data will be
                                                                ; copied into dma half buffer 2)
2513
                                  <1>
2514
                                  <1>
                                            ;; 20/05/2017
2515
                                  <1>
                                            ;mov byte [audio_flag], 1 ; next half (on next time)
2516
                                  <1>
                                            ; 24/04/2017
2517
                                  <1>
2518 00010559 A0[416B0100]
                                            mov al, [audio_device]
                                  <1>
2519 0001055E 3C03
                                  <1>
                                                  al, 3 ; VT8233 (VT8237R)
                                            cmp
2520 00010560 7410
                                                   short snd play 1
                                 <1>
                                            jе
2521 00010562 3C01
                                  <1>
                                                  al, 1 ; Sound Blaster 16
2522 00010564 7512
                                 <1>
                                            ine
                                                  short snd_play_2 ; 28/05/2017
2523 00010566 E8DF1A0000
                                                 SbInit play
                                 <1>
                                            call
2524 0001056B 0F82E7FCFFFF
                                 <1>
                                            jс
                                                  soundc_respond_err
2525 00010571 C3
                                           retn
                                 <1>
2526
                                 <1>
                                 <1> snd_play_1:
2527
2528 00010572 E802190000
                                            call vt8233_start_play
                                 <1>
2529 00010577 C3
                                  <1>
                                            retn
2530
                                  <1>
                                  <1> snd play 2:
2531
                                         ; \overline{2}8/05/2017
2532
                                  <1>
2533
                                  <1>
                                            ;cmp al, 2 ; AC'97
2534
                                  <1>
                                           ;jne short snd_play_3
2535
                                  <1>
2536 00010578 E8C61E0000
                                  <1>
                                            call ac97_start_play
2537 0001057D C3
                                  <1>
                                           retn
2538
                                  <1>
2539
                                  <1> ;snd play 3:
2540
                                  <1>;
                                           ;call hda_start_play
2541
                                  <1>;
2542
                                  <1>
```

```
<1> sound_pause:
                                               ; FUNCTION = 5
2544
                                         <1>
2545
                                         <1>
                                                     ; Pause
2546
                                         <1>
                                                  ; 28/05/2017
                                                  ; 24/04/2017
2547
                                         <1>

<1>    ; 22/04/2017
<1>    call    snd_dev_check
<1>    jc    short snd_nothing; temporary.
<1>    call    snd_buf_check
<1>    jc    short snd_nothing; temporary.
<1>    mov    al, [audio_device]
<1>    cmp    al, 3; VIA VT 8237R (vt8233)
<1>    je    short snd_pause_1
<1>    cmp    al, 1; Sound Blaster 16
<1>    ine    short snd_pause 2; 28/05/2017

2548
2549 0001057E E814030000
2550 00010583 7275
2551 00010585 E81A030000
2552 0001058A 726E
2553 0001058C A0[416B0100]
2554 00010591 3C03
2555 00010593 7409
2556 00010595 3C01
2557 00010597 750A
                                                     jne
                                        <1>
                                                            short snd_pause_2 ; 28/05/2017
                                        <1>
2558 00010599 E98A1C0000
                                                     jmp
                                                            sb16_pause
                                        <1> snd_pause_1:
2560 0001059E E98F190000
                                        <1> jmp vt8233_pause
                                         <1> snd pause 2:
2561
                                        <1> ; 28/05/2017
2562
                                                     ;cmp al, 2 ; AC'97
2563
                                         <1>
                                                    ;jne short snd_nothing; temporary.
jmp ac97_pause
                                         <1>
2564
2565 000105A3 E929200000
                                         <1>
2566
                                         <1>
                                         <1> sound_continue:
2567
                                                 ; FUNCTION = 6
2568
                                         <1>

; Continue to play
; 28/05/2017
; 22/04/2017

call snd_dev_check
; short snd_nothing; temporary.

call snd_buf_check
; call snd_buf_check
; jc short snd_nothing; temporary.

mov al, [audio_device]
cmp al, 3; VIA VT 8237R (vt8233)
; short snd_cont_1
cmp al, 1; Sound Blaster 16
; jne short snd_cont_2; 28/05/2017
; jmp sb16_continue

2569
2570
2571
2572 000105A8 E8EA020000
2573 000105AD 724B
2574 000105AF E8F0020000
2575 000105B4 7244
2576 000105B6 A0[416B0100]
2577 000105BB 3C03
2578 000105BD 7409
2579 000105BF 3C01
2580 000105C1 750A
2581 000105C3 E9831C0000
                                        <1>
                                                     jmp
                                                             sb16_continue
                                        <1> snd_cont_1:
2582
2583 000105C8 E916190000
                                                    jmp vt8233_play
                                        <1>
2584
                                         <1> snd cont 2:
                                        <1>
                                                     ; 28/05/2017
2585
2586
                                         <1>
                                                     ;cmp al, 2 ; AC'97
                                                     ;jne short snd_nothing; temporary.
2587
                                         <1>
2588 000105CD E9C71E0000
                                         <1>
                                                     jmp
                                                            ac97_play
                                         <1>
2590
                                         <1> sound_stop:
                                                ; FUNCTION = 7
2591
                                         <1>
                                                     ; Stop playing
2592
                                         <1>
                                                 ; 28/05/2017
; 24/05/2017
; 21/04/2017, 22/04/2017, 24/04/2017
2593
                                         <1>
2594
                                         <1>
2595
                                         <1>
                                                ; 21/04/2017, 22/04/2017, 24/04/2017
call snd_dev_check
jc short snd_nothing; temporary.
; call snd_buf_check
call snd_user_check; 24/05/2017
2596 000105D2 E8C0020000
                                        <1>
2597 000105D7 7221
                                         <1>
2598
                                        <1>
2599 000105D9 E8CF020000
                                        <1>
2600 000105DE 721A
                                        <1>
                                                  jc short snd_nothing; temporary.
2601
                                         <1>
2602 000105E0 A0[416B0100]
2603 000105E5 3C03
2604 000105E7 0F844C180000
                                        <1>
                                                            al, [audio_device]
                                                  mov
                                               cmp al, [audio_device]
cmp al, 3; VIA VT 823
je vt8233_stop
; 28/05/2017
; ja short snd_nothing
                                                            al, 3; VIAVT 8237R (vt8233)
                                        <1>
                                        <1>
2605
                                        <1>
2606
                                         <1>
                                                    cmp al, 1; Sound Blaster 16 je sb16_stop
2607 000105ED 3C01
                                         <1>
2608 000105EF 0F84791C0000
                                         <1>
                                         <1>
                                                     ;cmp al, 2
2610
                                         <1>
                                                     ;je short ac97_stop
2611 000105F5 E9A91F0000
                                         <1>
                                                             ac97_stop ; temporary.
                                                     jmp
2612
                                         <1>
                                                     ;jmp hda_stop
2613
                                         <1>
2614
                                         <1> snd nothing:
                                                   ; 21/04/2017
2615
                                         <1>
2616 000105FA C3
                                         <1>
2617
                                         <1>
2618
                                         <1> soundc_reset:
2619
                                         <1> ; FUNCTION = 8
                                                  ; Reset Audio Controller
2620
                                         <1>
                                                  ; 28/05/2017
; 22/04/2017
2621
                                         <1>
2622
                                         <1>
2623 000105FB E897020000
                                                  call snd_dev_check
                                         <1>
2624 00010600 72F8
                                         <1>
                                                  jс
                                                             snd_nothing ; temporary.
2625 00010602 E89D020000
                                         <1>
                                                     call snd_buf_check
2626 00010607 72F1
                                         <1>
                                                              snd nothing ; temporary.
2627
                                         <1>
2628 00010609 A0[416B0100]
                                         <1>
                                                     mov
                                                              al, [audio_device]
                                                             al, 3; VIA VT 8237R (vt8233)
2629 0001060E 3C03
                                         <1>
                                                     cmp
2630 00010610 0F8428190000
                                         <1>
                                                     jе
                                                             vt8233_reset
2631 00010616 77E2
                                                              short snd nothing; temporary.
                                         <1>
                                                     jа
2632
                                         <1>
                                                             hda reset
                                                     ;ja
2633 00010618 3C01
                                         <1>
                                                     cmp
                                                             al, 1 ; Sound Blaster 16
2634 0001061A 0F8502200000
                                         <1>
                                                     jne
                                                             ac97 reset
2635 00010620 E89B1C0000
                                         <1>
                                                     call
                                                            sb16 reset
2636 00010625 0F822DFCFFFF
                                                             soundc respond err
                                         <1>
                                                     jс
2637 0001062B C3
                                         <1>
                                                     retn
2638
                                         <1>
2639
                                         <1> soundc cancel:
                                                    \overline{;} FUNCTION = 9
2640
                                         <1>
2641
                                         <1>
                                                     ; Cancel audio callback service
2642
                                                     ; 22/04/2017
                                         <1>
                                                    mov al, [audio_user]
2643 0001062C A0[696B0100]
                                         <1>
                                                    cmp al, [u.uno]
jne short snd_nothing
2644 00010631 3A05[B3030300]
                                         <1>
2645 00010637 75C1
                                         <1>
                                         <1>
                                                     ; RESET (audio) INTERRUPT CALLBACK SERVICE
2647 00010639 8A1D[436B0100]
                                         <1>
                                                     mov bl, [audio intr] ; IRQ number
```

```
mov al, [u.uno]
2648 0001063F A0[B3030300]
                               <1>
2649 00010644 28FF
2650 00010646 E8EC020000
                                     sub
call
                                <1>
                                               bh, bh ; 0 ; unlink IRQ from user service
                                         call set_irq_callback_service
                               <1>
2651 0001064B 0F8250FDFFFF
                               <1>
                                         jс
                                               sndc_perm_error ; 'permission denied' error
2652 00010651 C3
                                <1>
                                         retn
2653
                                <1>
                                <1> sound dalloc:
2654
                                       ; FUNCTION = 10
2655
                                <1>
2656
                                <1>
                                         ; Deallocate (ring 3) audio buffer
                                         ; 22/04/2017
2657
                                <1>
2658 00010652 A0[696B0100]
                                <1>
                                       mov al, [audio_user]
                                      cmp
2659 00010657 3A05[B3030300]
                                <1>
                                               al, [u.uno]
                                         jne short snd_nothing
2660 0001065D 759B
                                <1>
                                       mov
                                               ebx, [audio buffer]
2661 0001065F 8B1D[546B0100]
                                <1>
                                    ;or
;jz
mov
call
2662
                                <1>
                                               ebx, ebx
2663
                                <1>
                                               short snd_nothing
2664 00010665 8B0D[5C6B0100]
                               <1>
                                               ecx, [audio buff size]
2665 0001066B E86C51FFFF
                                        call deallocate_user_pages
                               <1>
2666 00010670 31C0
                                <1>
                                         xor
                                               [audio_buffer], eax ; 0
2667 00010672 A3[546B0100]
                               <1>
                                         mov
2668 00010677 A2[696B0100]
                               <1>
                                         mov [audio_user], al; 0
2669 0001067C C3
                                <1>
                                         retn
2670
                                <1>
2671
                                <1> sound_volume:
                                     ; FUNCTION = 11
2672
                                <1>
2673
                                <1>
                                         ; Set sound volume level
                                       ; 28/05/2017
2674
                                <1>
                                       ; 20/05/2017
2675
                                <1>
                                       ; 22/04/2017, 24/04/2017
2676
                                <1>
2677
                                <1>
                                         ; bl = component (0 = master/playback/lineout volume)
2678
                                <1>
                                       ; cl = left channel volume level (0 to 31)
2679
                                <1>
                                        ; ch = right channel volume level (0 to 31)
2680
                                <1>
2681 0001067D 80FB80
                                <1>
                                       cmp bl, 80h
                                       jb
2682 00010680 720E
                                <1>
                                               short snd_vol_1
2683 00010682 0F8772FFFFF
                                <1>
                                         jа
                                               snd_nothing ; temporary.
                                         ; Set volume level for next play (BL>= 80h)
                                <1>
2685 00010688 66890D[766B0100]
                               <1>
                                         mov [audio_master_volume], cx
2686 0001068F C3
                                <1>
                                         retn
2687
                                <1> snd_vol_1:
                                     ; set volume level immediate (BL< 80h)
2688
                                <1>
2689 00010690 80FB00
                                <1>
                                         cmp bl, 0
2690 00010693 0F8761FFFFFF
                               <1>
                                               snd_nothing ; temporary.
                                         jа
2691
                                <1>
2692 00010699 E8F9010000
                                <1>
                                         call snd_dev_check
2693 0001069E 0F8256FFFFFF
                                <1>
                                         jс
                                               snd_nothing ; temporary.
2694 000106A4 E8FB010000
                                <1>
                                         call snd buf check
2695 000106A9 0F824BFFFFFF
                                               snd_nothing ; temporary.
                                <1>
                                       jс
2696
                                <1>
2697 000106AF A0[416B0100]
                                               al, [audio_device]
                                <1>
                                         mov
2698 000106B4 3C03
                                <1>
                                         cmp al, 3; VIA VT 8237R (vt8233)
                                               vt8233_volume
2699 000106B6 0F849B180000
                                <1>
                                         jе
                                         ; 28/05/2017
2700
                                <1>
2701 000106BC 0F8738FFFFF
                                <1>
                                       ja snd_nothing; temporary.
2702
                                <1>
                                       ;ja
                                               hda volume
2703
                                <1>
                                         ; Sound Blaster 16
2704 000106C2 3C01
                                <1>
                                         cmp al, 1; SB 16
2705 000106C4 0F84291B0000
                                <1>
                                               sb16_volume
                                         jе
2706 000106CA E9E61D0000
                                <1>
                                               ac97_volume
                                         jmp
2707
                                <1>
                                <1> soundc_disable:
2708
                                     ; FUNCTION = 12
2709
                                <1>
2710
                                <1>
                                         ; Disable audio device (and unlink DMA memory)
2711
                                <1>
                                        ; 28/05/2017
                                       ; 24/05/2017
2712
                                <1>
2713
                                <1>
                                         ; 22/04/2017
                                        call snd dev check
2714 000106CF E8C3010000
                                <1>
2715 000106D4 0F8270FBFFFF
                                <1>
                                       jc soundc_dev_err ; temporary.
2716
                                <1>
                                         ;call snd_buf_check
                                         ;jc sndc_owner_error ; temporary.
2717
                                <1>
2718
                                <1>
                                               al, [audio_device]
2719 000106DA A0[416B0100]
                                <1>
                                         mov
                                               al, 3; VIA VT 8237R (vt8233)
2720 000106DF 3C03
                               <1>
                                         cmp
2721 000106E1 7418
                               <1>
                                         jе
                                               short snd_disable_1
                                               snd nothing; temporary.
2722 000106E3 0F8711FFFFFF
                               <1>
                                         jа
2723 000106E9 3C01
                               <1>
                                         cmp
                                               al, 1 ; Sound Blaster 16
2724 000106EB 7507
                               <1>
                                         jne
                                               short snd disable 0
2725 000106ED E87C1B0000
                               <1>
                                         call sb16_stop
2726 000106F2 EB0C
                                <1>
                                               short snd_disable_2
                                         jmp
                                <1> snd_disable_0:
2727
2728 000106F4 E8AA1E0000
                                         call ac97 stop
                                <1>
2729 000106F9 EB05
                                <1>
                                          jmp
                                                short snd_disable_2
2730
                                <1> snd_disable_1:
2731 000106FB E839170000
                                         call vt8233_stop
                                <1>
                                <1> snd_disable_2:
2732
2733 00010700 A0[436B0100]
                                <1>
                                         mov al, [audio_intr]
2734 00010705 29DB
                                              ebx, ebx; 0 = reset
                                <1>
                                         sub
2735 00010707 E851FAFFFF
                               <1>
                                         call set dev IRQ service
2736
                                <1>
2737
                               <1>
                                         ;mov al, [audio_intr]
2738 0001070C 28E4
                               <1>
                                         sub
                                                ah, ah ; 0 = reset
2739 0001070E E8B5F6FFFF
                                <1>
                                         call
                                              set_hardware_int_vector
2740
                                <1>
2741 00010713 31C0
                                <1>
                                                eax, eax
2742 00010715 A2[416B0100]
                                <1>
                                               byte [audio_device], al
                                         mov
2743 0001071A A2[436B0100]
                                <1>
                                         mov
                                               byte [audio_intr], al
2744 0001071F 8705[606B0100]
                                <1>
                                         xchg
                                              eax, [audio_dma_buff]
                                         ; 24/05/2017
2745
                                <1>
2746
                                <1>
                                         ;or
                                               eax, eax
2747
                                               short snd disable 3
                                <1>
                                         ;jz
2748
                                         ;cmp eax, sb16_dma_buffer ; default DMA buffer
                                <1>
2749
                                <1>
                                         ;je
                                               short snd_disable_3
2750 00010725 803D[406B0100]00
                                               byte [audio pci], 0; AC97 audio controller?
                                <1>
                                         cmp
2751 0001072C 7612
                                <1>
                                               short snd_disable_3
                                         jna
2752 0001072E C605[406B0100]00
                                <1>
                                               byte [audio pci], 0
                                         mov
```

```
;sub ecx, ecx
;xchg ecx, [audio_dmabuff_size]
mov ecx, [audio_dmabuff_size]
call deallocate_memory_block
2753
                                <1>
2754
                                <1>
2755 00010735 8B0D[646B0100]
                               <1>
2756 0001073B E84E4FFFFF
                               <1>
2757
                                <1> snd disable 3:
2758 00010740 C3
                                <1>
                                         retn
2759
                                <1>
2760
                                <1> sound dma map:
                                      ; \overline{\text{FUNCTION}} = 13
2761
                                <1>
2762
                                <1>
                                         ; Map audio dma buff addr to user's buffer addr
                                       ; 12/05/2017
and ecx, e
2763
                                <1>
2764 00010741 21C9
                                         and ecx, ecx jz sound_bu:
                                <1>
2765 00010743 0F8408FBFFFF
                                               sound buff error
                               <1>
2766 00010749 803D[416B0100]01 <1> cmp byte [audio_device], 1 2767 00010750 7229 <1> jb short snd_dma_map_1
                                <1> snd_dma_map_0:
2768
                               2769 00010752 A1[606B0100]
2770 00010757 21C0
                                         and
                               <1>
                                               eax, eax
                                                short snd dma map 1
2771 00010759 7420
                                <1>
                                         jz
                               <1>
                                         ;
2772
2773 0001075B 8A1D[696B0100] <1> mov bl, [audio_user]
bl, bl
                                               short snd dma map 1
                                               bl, [u.uno]
                                               sndc_owner_error
                                               ebx, [audio_dmabuff_size]
                                     and
2780 00010777 21DB
                               <1>
                                               ebx, ebx
2781 00010779 750A
                                <1>
                                               short snd_dma_map_2
                                         jnz
                               <1> snd_dma_map_1:
2782
2782
2783 0001077B B8[00000200]
                               <1> mov eax, sb16_dma_buffer
2784 00010780 BB00000100
                                <1>
                                         mov
                                               ebx, 65536
2793
                                       ; eax = physical address of (audio) dma buffer
                                       ; ebx = virtual address of (audio) dma buffer (user's pgdir)
2794
                                <1>
                                         ; ecx = page count (>0)
2795
                                <1>
2796 0001079E E85B4FFFFF
                               <1>
                                         call direct_memory_access
                                       pop eax
2797 000107A3 58
                                <1>
2798 000107A4 0F82A7FAFFFF
                                <1>
                                         jс
                                                sound_buff_error
2799 000107AA A3[64030300]
                                <1>
                                         mov
                                               [u.r0], eax
2800 000107AF C3
                                <1>
                                        retn
2801
                                <1>
2802
                                <1> soundc_info:
                                <1> ; FUNCTION = 14
2803
2804
                                <1>
                                         ; Get Audio Controller Info
                                       ; 10/06/2017
2805
                                <1>
                                       and bl, bl; 0
jz short sndc_info_0; invalid paramet
                                       ; 05/06/2017
2806
                               <1>
2807 000107B0 20DB
                                <1>
2808 000107B2 740A
                                <1>
                                <1>
2810 000107B4 B817000000
                                        mov eax, ERR_INV_PARAMETER; 23
                                <1>
2811
                                <1> ;sndc_inf_error:
2812
                                <1>; mov [u.r0], eax
2813
                                <1>;
                                         mov
                                               [u.error], eax
2814
                                <1>;
                                          jmp
                                               error
2815 000107B9 E99FFAFFFF
                                <1>
                                         jmp
                                               sysaudio_err
2816
                                <1>
2817
                                <1> sndc_info_0:
2818 000107BE E8D4000000
                                <1> call snd_dev_check
2819 000107C3 0F8281FAFFFF
                                <1>
                                         jc soundc_dev_err
                               2821 000107C9 8B1D[4C6B0100]
2822 000107CF 8B0D[486B0100]
2823
2824 000107D5 3C02
2825 000107D7 7513
                                               short sndc info 1
                                         jne
                                         ; Intel AC97 (ICH) Audio Controller (=2)
2826
                                <1>
                                     mov dx, [NABMBAR]
shl edx, 16
mov dx, [NAMBAR]
jmp short sndc_info_2
2827 000107D9 668B15[7A6B0100]
                                <1>
2828 000107E0 C1E210
                                <1>
2829 000107E3 668B15[786B0100]
                                <1>
2830 000107EA EB07
                                <1>
2831
                                <1> sndc_info_1:
                                2832
                                          ; Note: Intel HDA code (here) is not ready yet!
2833
                                <1>
                                          ; !!! SB16 or VT8233 (VT8237R) !!!
2834
                                 <1>
2835 000107EC 0FB715[466B0100]
                                <1>
                                          movzx edx, word [audio_io_base]
                                <1> sndc_info_2:
2837 000107F3 88C4
                                <1>
                                         mov ah, al ; [audio_device]
2838 000107F5 A0[436B0100]
                                               al, [audio intr]
                                <1>
                                          mov
                                <1>
2840
                                <1>
                                         ; EAX = IRQ Number in AL
2841
                                <1>
                                               Audio Device Number in AH
                                         ; EBX = DEV/VENDOR ID
2842
                                <1>
                                         ; (DDDDDDDDDDDDDDDDVVVVVVVVVVVVVV)
2843
                                <1>
2844
                                 <1>
                                         ; ECX = BUS/DEV/FN
2845
                                <1>
                                          ; (0000000BBBBBBBBDDDDDFFF00000000)
2846
                                <1>
                                         ; EDX = NABMBAR/NAMBAR (for AC97)
2847
                                <1>
                                         ; (Low word, DX = NAMBAR address)
                                          ; EDX = Base IO Addr (DX) for SB16 & VT8233
2848
                                <1>
2849
                                <1>
                                          ; 10/06/2017
2850
                                <1>
2851 000107FA A3[64030300]
                                <1>
                                                [u.r0], eax
2852 000107FF 8B2D[60030300]
                                <1>
                                                ebp, [u.usp]
                                         mov
2853 00010805 895D10
                                <1>
                                          mov
                                                [ebp+16], ebx ; ebx
2854 00010808 895514
                                <1>
                                                [ebp+20], edx; edx
                                         mov
2855 0001080B 894D18
                                <1>
                                          mov
                                                [ebp+24], ecx ; ecx
2856
                                <1>
2857 0001080E C3
                                <1>
                                         retn
```

```
2858
                                 <1>
2859
                                 <1> sound data:
                                          ; FUNCTION = 15
2860
                                 <1>
2861
                                 <1>
                                          ; Get Current Sound data for graphics
2862
                                 <1>
                                          ; 22/06/2017
2863
                                 <1>
2864 0001080F E883000000
                                                snd dev check
                                 <1>
                                           call
                                                 soundc_dev_err ; Device not ready !
2865 00010814 0F8230FAFFFF
                                 <1>
                                           jс
2866
                                 <1>
2867 0001081A 80FB00
                                 <1>
                                           cmp
                                                 bl, 0
2868 0001081D 760A
                                 <1>
                                          jna short sound_data_0
2869
                                 <1>
                                           ; Only PCM OUT buffer data is valid for now!
2870
                                 <1>
2871 0001081F B817000000
                                 <1>
                                           mov eax, ERR_INV_PARAMETER ; 23
2872 00010824 E934FAFFFF
                                 <1>
                                           jmp
                                                 sysaudio_err
2873
                                 <1>
                                 <1> sound data 0:
2875 00010829 A1[606B0100]
                                 <1>
                                          mov eax, [audio_dma_buff]
2876 0001082E 09C0
                                 <1>
                                                  eax, eax
                                           or
                                                 sound buff error
2877 00010830 0F841BFAFFFF
                                 <1>
                                           jΖ
2878
                                 <1>
2879 00010836 803D[416B0100]04
                                 <1>
                                           cmp
                                                 byte [audio_device], 4 ; Intel HDA
                                                 short sound_data_4; temporary ! (22/06/2017)
2880 0001083D 744F
                                 <1>
                                           jе
                                 <1>
2882 0001083F 21C9
                                 <1>
                                           and
                                                 ecx, ecx
2883
                                 <1>
                                           ;jnz short sound_data_1 ; sample tranfer
2884
                                 <1>
2885
                                 <1>
                                          ; Return only DMA Buffer pointer/offset...
2886
                                 <1>
                                           ; (If DMA Buffer has been mapped to user's
                                 <1>
2887
                                           ; memory space; program can get graphics
2888
                                 <1>
                                           ; data by using only this pointer value.)
2889
                                 <1>
                                 <1>
                                           ;call get_dma_buffer_offset
2890
2891
                                 <1>
                                           ;; eax = DMA buffer offset
2892
                                 <1>
                                                (!not half buffer offset!)
                                           ;;
2893
                                 <1>
                                           ;mov [u.r0], eax
2894
                                 <1>
                                           ;retn
2895
                                 <1>
2896 00010841 0F84531F0000
                                 <1>
                                                 get_dma_buffer_offset
2897
                                 <1>
                                 <1> sound data 1:
2898
2899
                                 <1>
                                           ;mov eax, [audio_dmabuff_size]
                                           ;shr eax, 1 ; half buffer size
2900
                                 <1>
                                                ecx, eax
2901
                                 <1>
2902
                                 <1>
                                                 short sound_buff_error
                                           ;ja
2903
                                 <1>
2904 00010847 3B0D[646B0100]
                                 <1>
                                           cmp
                                                 ecx, [audio dmabuff size]
2905 0001084D 0F87FEF9FFF
                                 <1>
                                                 sound_buff_error
                                         ja
2906
                                 <1>
2907 00010853 89D0
                                 <1>
                                                 eax, edx
                                          mov
2908 00010855 25FF0F0000
                                 <1>
                                           and
                                                eax, PAGE_OFF ; 4095 (OFFFh)
2909 0001085A 81F900100000
                                 <1>
                                           cmp
                                                 ecx, 4096
2910 00010860 7605
                                <1>
                                           jna
                                                 short sound_data_2
2911 00010862 B900100000
                                <1>
                                           mov ecx, 4096; max. 1 page
2912
                                 <1> sound_data_2:
2913 00010867 01C8
                                <1>
                                          add eax, ecx
2914 00010869 3D00100000
                                <1>
                                                eax, 4096
                                           cmp
2915 0001086E 7606
                                <1>
                                                 short sound_data_3
                                           jna
2916 00010870 6625FF0F
                                 <1>
                                                 ax, PAGE_OFF ; 4095 (OFFFh)
2917 00010874 29C1
                                <1>
                                           sub ecx, eax
2918
                                <1>
                                          ; here, ECX has been adjusted to fit
2919
                                 <1>
                                                 in page border.. (<= 4096, >0)
2920
                                 <1> sound_data_3:
2921 00010876 51
                                 <1>
                                          push ecx
2922 00010877 52
                                 <1>
                                           push edx
2923 00010878 89D3
                                 <1>
                                           mov
                                                 ebx, edx
2924 0001087A E86D4AFFFF
                                <1>
                                          call get_physical_addr
2925 0001087F 5A
                                 <1>
                                          pop
                                                 edx
2926 00010880 59
                                 <1>
                                          pop
2927 00010881 0F82CAF9FFFF
                                 <1>
                                                 sound_buff_error
                                           jс
2928
                                 <1>
2929
                                 <1>
                                          ; eax = physical address of user's buffer
2930 00010887 89C3
                                 <1>
                                           mov ebx, eax
2931
                                 <1>
                                          ; ecx = byte (transfer) count
2932
                                 <1>
                                           ;call get_current_sound_data
2933
                                 <1>
                                           ;retn
                                           jmp get_current_sound_data
2934 00010889 E9691E0000
                                 <1>
2935
                                 <1>
2936
                                 <1> sound_data_4:
2937
                                         ; Intel HDA code is not ready yet !
                                 <1>
2938
                                 <1>
                                           ; 22/06/2017
2939 0001088E 31C0
                                           xor eax, eax
                                 <1>
2940 00010890 48
                                 <1>
                                           dec
                                                 eax
2941 00010891 A3[64030300]
                                 <1>
                                                 [u.r0], eax; OFFFFFFFh
2942 00010896 C3
                                 <1>
                                           retn
2943
                                 <1>
2944
                                 <1> snd dev check:
                                         ; 10/06/2017
2945
                                 <1>
2946
                                 <1>
                                           ; 05/06/2017
2947
                                 <1>
                                          ; 24/05/2017
2948
                                 <1>
                                          ; 22/04/2017
2949
                                 <1>
                                          ; 21/04/2017
2950
                                 <1>
                                           ; ... device check at first
2951 00010897 A0[416B0100]
                                 <1>
                                          mov al, [audio_device]
2952 0001089C 3C01
                                 <1>
                                           cmp al, 1; SB 16
2953 0001089E 7203
                                 <1>
                                           jb
                                                 short snd_dev_chk_retn ; error !
                                 <1>
                                           ;cmp al, 4 ; Intel HDA
2955
                                 <1>
                                           ;ja
                                                 short snd_dbchk_stc ; invalid !
2956
                                 <1>
                                           ; 10/06/2017
2957 000108A0 3C05
                                 <1>
                                                 al, 5
                                           cmp
2958 000108A2 F5
                                 <1>
                                           cmc
2959
                                 <1> snd dev chk retn:
2960 000108A3 C3
                                 <1>
                                           retn
2961
                                 <1>
2962
                                 <1> snd_buf_check:
```

```
; 10/06/2017
2963
                                  <1>
2964
                                  <1>
                                           ; 22/04/2017
                                           ; 21/04/2017
2965
                                  <1>
2966
                                  <1>
                                            ; ... buffer & (buffer) owner check at second
                                            cmp dword [audio_buffer], 0
jna short snd_dbchk_stc
2967 000108A4 833D[546B0100]00
                                  <1>
2968 000108AB 760D
                                  <1>
                                  <1> snd user check:
2970 000108AD A0[B3030300]
                                  <1>
                                            mov al, [u.uno]
2971 000108B2 3A05[696B0100]
                                  <1>
                                            cmp
                                                  al, [audio user]
                                  <1>
                                            ;jne short snd_dbchk_stc
2973
                                  <1>
                                            ;retn
2974 000108B8 74E9
                                  <1>
                                            jе
                                                  short snd dev chk retn
2975
                                  <1>
2976
                                  <1> snd_dbchk_stc:
2977 000108BA F9
                                  <1>
                                            stc
2978 000108BB C3
                                  <1>
                                            retn
                                  <1>
2980
                                  <1> sound_update:
                                           \overline{}; FUNCTION = 16
2981
                                  <1>
2982
                                  <1>
                                            ; bl =
                                                 0 = automatic (sequental) update (with flag switch!)
2983
                                  <1>
                                                 1 = update dma half buffer 1 (without flag switch!)
2984
                                  <1>
                                               2 = update dma half buffer 2 (without flag switch!)
2985
                                  <1>
                                            ;
2986
                                  <1>
                                            ; FFh = get current flag value
                                  <1>
                                                0 = dma half buffer 1 (will be played next)
2987
                                                   1 = dma half buffer 2 (will be played next)
2988
                                  <1>
2989
                                  <1>
                                            ; 10/10/2017
                                  <1>
2990
2991
                                  <1>
2992
                                  <1>
                                            ; ... device check at first
2993 000108BC A0[416B0100]
                                  <1>
                                            mov al, [audio_device]
2994 000108C1 08C0
                                  <1>
                                            or
                                                  al, al; 0; pc speaker or invalid
2995 000108C3 0F8481F9FFFF
                                  <1>
                                                  soundc_dev_err
                                            jΖ
2996
                                  <1>
                                            ; ... buffer & (buffer) owner check at second
2997
                                  <1>
2998 000108C9 833D[546B0100]00
                                            cmp dword [audio buffer], 0
                                  <1>
2999 000108D0 0F867BF9FFFF
                                  <1>
                                            jna
                                                  sound buff error
3000 000108D6 A0[B3030300]
                                  <1>
                                                  al, [u.uno]
                                            mov
3001 000108DB 3A05[696B0100]
                                  <1>
                                                  al, [audio_user]
                                            cmp
3002 000108E1 0F85B5FAFFFF
                                  <1>
                                            jne sndc_owner_error
3003
                                  <1>
                                            ; Transfer ring 3 (user's) audio buffer content to dma buffer
3004
                                  <1>
3005 000108E7 8B3D[606B0100]
                                  <1>
                                            mov edi, [audio_dma_buff] ; dma buffer (ring 0)
3006 000108ED 09FF
                                  <1>
                                                   edi, edi
                                            jz
3007 000108EF 0F845CF9FFFF
                                  <1>
                                                   sound_buff_error
3008 000108F5 8B35[586B0100]
                                                   esi, [audio_p_buffer] ; physical address (ring 3)
                                  <1>
                                            mov
3009 000108FB 8B0D[5C6B0100]
                                  <1>
                                                  ecx, [audio_buff_size]
                                            mov
3010
                                  <1>
                                            ;movzx eax, byte [audio flag]
3011
                                  <1>
                                                  al, [audio_flag]
3012 00010901 A0[686B0100]
                                  <1>
                                            mov
3013
                                  <1>
3014 00010906 FEC3
                                  <1>
                                            inc
3015 00010908 7427
                                  <1>
                                            jΖ
                                                   short snd_update_3 ; bl = 0FFh
3016 0001090A FECB
                                  <1>
                                                  short snd_update_0 ; bl = 0
3017 0001090C 7411
                                  <1>
                                            jz
3018
                                  <1>
3019 0001090E 80FB02
                                  <1>
                                            cmp
                                                   short snd_update_1 ; dma half buffer 2
3020 00010911 7417
                                  <1>
                                            jе
3021 00010913 7217
                                  <1>
                                                   short snd_update_2 ; dma half buffer 1
                                            jb
3022
                                  <1>
3023
                                  <1>
                                            ; invalid parameter !
3024 00010915 B817000000
                                                  eax, ERR INV PARAMETER; 23
                                  <1>
                                            mov
3025
                                  <1>;
                                                   [u.r0], eax
                                            mov
                                                   [u.error], eax
3026
                                  <1>;
                                  <1>;
3027
                                            jmp
                                                  error
                                            jmp
3028 0001091A E93EF9FFFF
                                  <1>
                                                  sysaudio_err
3029
                                  <1>
3030
                                  <1> snd_update_0:
3031 0001091F 8035[686B0100]01
                                  <1>
                                                 byte [audio_flag], 1 ; update flag !!!
                                            xor
3032 00010926 3C01
                                  <1>
                                                  al, 1
                                            cmp
3033 00010928 7202
                                  <1>
                                            jb
                                                  short snd_update_2 ; dma half buffer 1
3034
                                  <1> snd update 1:
                                           ; dma half buffer 2
3035
                                  <1>
                                            add edi, ecx
3036 0001092A 01CF
                                  <1>
                                  <1> snd_update_2:
3037
3038
                                  <1>
                                            ;rep movsb
3039 0001092C C1E902
                                  <1>
                                            shr ecx, 2
3040 0001092F F3A5
                                  <1>
                                            rep
                                                  movsd
3041
                                  <1> snd_update_3:
3042 00010931 A3[64030300]
                                                 [u.r0], eax
                                  <1>
                                           mov
3043
                                  <1>
3044 00010936 C3
                                  <1>
                                            retn
3045
                                  <1>
3046
                                  <1> set irq callback service:
                                           ; 03/08/20<del>2</del>0
3047
                                  <1>
3048
                                  <1>
                                            ; 10/06/2017
                                            ; 12/05/2017
3049
                                  <1>
                                            ; 24/04/2017
3050
                                  <1>
3051
                                  <1>
                                            ; 22/04/2017
3052
                                  <1>
                                            ; caller: 'syscalbac' or 'sysaudio' or ...
                                            ; 13/04/2017, 14/04/2017, 17/04/2017
3053
                                  <1>
                                            ; 24/02/2017, 26/02/2017, 28/02/2017
3054
                                  <1>
                                            ; 21/02/2017 - TRDOS 386 (TRDOS v2.0)
3055
                                  <1>
3056
                                  <1>
3057
                                  <1>
                                           ; Link or unlink IRQ callback service to/from user (ring 3)
3058
                                  <1>
                                            ; INPUT ->
3059
                                  <1>
                                                  If AL = 0, the caller is 'syscalbac';
3060
                                  <1>
3061
                                  <1>
                                                      otherwise, the caller is 'sysaudio' or ...
3062
                                  <1>
                                                      (AL = user number)
3063
                                  <1>
3064
                                  <1>
                                                   BL = IRQ number (Hardware interrupt request number)
3065
                                  <1>
                                                        (0 t0 15 but IRQ 0,1,2,6,8,14,15 are prohibited)
                                                        IRQ numbers 3,4,5,7,9,10,11,12,13 are valid
3066
                                  <1>
3067
                                  <1>
                                                        (numbers >15 are invalid)
```

```
3068
                                   <1>
3069
                                   <1>
                                                   BH = 0 = Unlink IRQ (in BL) from user (ring 3) service
                                             ;
                                                       1 = Link IRQ by using Signal Response Byte method
3070
                                   <1>
3071
                                   <1>
                                                       2 = Link IRQ by using Callback service method
3072
                                   <1>
                                                        3 = Link IRQ by using Auto Increment S.R.B. method
3073
                                   <1>
                                                       >3 = invalid
3074
                                   <1>
                                                           (syscallback version will return to user)
3075
                                   <1>
3076
                                   <1>
                                                   CL = Signal Return/Response Byte value
3077
                                   <1>
3078
                                   <1>
                                                   If BH = 3, kernel will put a counter value; 03/08/2020
3079
                                   <1>
                                                              (into the S.R.B. addr)
3080
                                   <1>
                                                            between 0 to 255. (start value = CL+1)
3081
                                   <1>
3082
                                   <1>
                                                   NOTE: counter value, for example: even and odd numbers
                                                          may be used for -audio- DMA buffer switch
3083
                                   <1>
                                                          within double buffer method, etc.
3084
                                   <1>
3085
                                   <1>
3086
                                   <1>
                                                   EDX = Signal return (Response) byte address
3087
                                   <1>
                                                                          - or -
                                                          Interrupt/Callback service/routine address
3088
                                   <1>
3089
                                   <1>
3090
                                                          (virtual address in user's memory space)
                                   <1>
3091
                                   <1>
3092
                                             ; OUTPUT ->
                                   <1>
3093
                                   <1>
                                                   CF = 0 \& EAX = 0 \rightarrow Successful setting
3094
                                   <1>
                                                   CF = 1 \& EAX > 0 \rightarrow IRQ is prohibited or locked
3095
                                                                by another process
                                   <1>
3096
                                   <1>
                                                           eax = ERR_PERM_DENIED -> prohibited or locked
3097
                                                           eax = ERR_INV_PARAMETER ->
                                   <1>
3098
                                   <1>
                                                                 invalid parameter/option or bad address
3099
                                   <1>
                                             ; TRDOS 386 - IRQ CALLBACK structures (parameters):
3100
                                   <1>
3101
                                   <1>
3102
                                   <1>
                                                       [u.irqlock] = 1 word, IRQ flags (0-15) that indicates
3103
                                   <1>
                                                                 which IRQs are locked by (that) user.
3104
                                   <1>
                                                                  Lock and unlock (by user) will change
                                                                 these flags or 'terminate process' (sysexit)
3105
                                   <1>
3106
                                   <1>
                                                                 will clear these flags and unlock those IRQs.
3107
                                   <1>
3108
                                   <1>
                                                                 Bit 0 is for IRQ 0 and Bit 15 is for IRQ 15
3109
                                   <1>
3110
                                   <1>
                                                      IRQ(x).owner
                                                                        : 1 byte, user, [u.uno], 0 = free (unlocked)
3111
                                   <1>
                                                      IRQ(x).method : 1 byte for callback method & status
3112
                                   <1>
3113
                                   <1>
                                                                   0 = Signal Response Byte method
                                                                    1 = Callback service method
3114
                                   <1>
                                                                    >1 = invalid for current 'syscalback'.
3115
                                   <1>
3116
                                   <1>
                                                                 or(+) 80h = IRQ is in use by system (ring 0)
3117
                                   <1>
                                                                             function (audio etc.) or
                                                                            a device driver.
3118
                                   <1>
3119
                                   <1>
                                                                 (system function will ignore the lock/owner)
3120
                                   <1>
3121
                                   <1>
                                                      IRQ(x).srb: 1 byte, Signal Return/Response byte value
3122
                                   <1>
                                                                   (a fixed value by user or a counter value
3123
                                   <1>
                                                                  from 0 to 255, which is increased by every
3124
                                   <1>
                                                                  interrupt just before putting it into
3125
                                   <1>
                                                                  the Signal Response byte address
3126
                                   <1>
                                                                  (This is not used in callback serv method)
3127
                                   <1>
3128
                                   <1>
                                                      IRQ(x).addr
                                                                       : 1 dword
3129
                                   <1>
                                                                   Signal Response Byte address (physical)
3130
                                   <1>
                                                                              -or-
3131
                                   <1>
                                                                   Callback service address (virtual)
3132
                                   <1>
3133
                                   <1>
                                                      IRQ(x).dev: 1 byte
3134
                                   <1>
                                                                   0 = Default device or kernel function
3135
                                   <1>
                                                                              -or-
3136
                                   <1>
                                                                   1-255 = Assigned device driver number
3137
                                   <1>
3138
                                   <1>
                                                       (x) = 3, 4, 5, 7, 9, 10, 11, 12, 13
3139
                                   <1>
3140
                                   <1>
                                                   bl, 15
3141 00010937 80FB0F
                                  <1>
3142 0001093A 7729
                                  <1>
                                                   short scbs_2
                                             jа
3143
                                  <1>
3144 0001093C 80FF03
                                  <1>
                                                   bh, 3
                                             cmp
                                                    short scbs 2 ; invalid parameter
3145 0001093F 7724
                                   <1>
                                   <1>
3146
3147 00010941 0FB6FB
                                             movzx edi, bl ; save IRQ number
                                   <1>
3148
                                   <1>
                                                   ; IRQ 0,1,2,6,8,14,15 are prohibited
3149
                                   <1>
                                             ; IRQenum: ; 'trdosk9.s'
3150
                                   <1>
3151
                                   <1>
                                                  db 0,0,0,1,2,3,0,4,0,5,6,7,8,9,0,0
3152
                                   <1>
3153 00010944 0FB6B7[96160100]
                                  <1>
                                             movzx esi, byte [edi+IRQenum] ; IRQ availability
                                  <1>
                                                                       ; enumeration/index
3155
                                  <1>
                                             :dec esi
3156 0001094B 664E
                                  <1>
                                             dec
                                                   si
3157 0001094D 780F
                                  <1>
                                                   short scbs_1 ; 0 -> 0FFFFh
                                             js
3158
                                  <1>
3159
                                   <1>
                                            ; ESI = IRQ callback parameters index number (0 to 8)
3160
                                  <1>
3161 0001094F 08FF
                                  <1>
                                                   bh, bh
3162 00010951 7419
                                  <1>
                                                   short scbs_4; unlink the IRQ (in BL)
                                             jΖ
3163
                                  <1>
3164 00010953 FECF
                                  <1>
                                             dec bh
3165
                                  <1>
                                            ; bh = method (0 = signal response byte, 1 = callback)
3166
                                   <1>
                                                         (2 = auto increment of signal response byte)
3167
                                  <1>
3168 00010955 80BE[F26A0100]00
                                             cmp byte [esi+IRQ.owner], 0 ; locked ?
                                  <1>
3169 0001095C 7637
                                   <1>
                                             jna short scbs_6; no... OK...
3170
                                  <1>
                                   <1> scbs 1:
3171
3172
                                   <1>
                                            ; permission denied (prohibited IRQ)
```

```
3173 0001095E B80B000000
                                                 eax, ERR_PERM_DENIED
                                <1>
                                         mov
3174 00010963 F9
                                 <1>
                                          stc
3175 00010964 C3
                                <1>
                                          retn
3176
                                <1> scbs 2:
3177 00010965 F9
                                 <1>
                                          stc
3178
                                 <1> scbs_3:
3179 00010966 B817000000
                                <1>
                                                 eax, ERR INV PARAMETER
                                          mov
3180 0001096B C3
                                <1>
                                          retn
3181
                                <1>
3182
                                 <1> scbs_4: ; unlink the requested IRQ (if it belongs to current user)
                                       ; 10/06/2017
3183
                                 <1>
                                          ; 22/04/2017
3184
                                 <1>
                                          ; 14/04/2017
3185
                                 <1>
                                          ; If AL = 0 -> The caller is 'syscalbac'
3186
                                 <1>
3187 0001096C 8AA6[F26A0100]
                                <1>
                                          mov ah, [esi+IRQ.owner]
3188 00010972 3A25[B3030300]
                                 <1>
                                          cmp
                                                ah, [u.uno]
3189 00010978 75E4
                                                short scbs 1
                                 <1>
                                          jne
3190
                                 <1>
3191 0001097A FE0D[D6030300]
                                 <1>
                                                byte [u.irqc] ; decrease IRQ count (in use)
3192
                                 <1>
3193
                                 <1>
                                          ;sub ah, ah
3194
                                 <1>
                                          ;mov
                                                 [esi+IRQ.owner], ah; 0; free !!!
3195
                                 <1>
                                          ; and byte [esi+IRQ.method], 80h
3196
                                 <1>
                                          ;mov [esi+IRQ.srb], ah; 0
                                                [esi+IRQ.dev], ah; 0
                                          ; mov
3197
                                 <1>
3198
                                 <1>
                                                 dword [esi+IRQ.addr], 0
                                          ;mov
3199
                                 <1>
                                                dword [u.r0], 0
                                          ; mov
3200
                                 <1>
3201
                                 <1>
                                          ;mov byte [esi+IRQ.owner], 0
3202
                                 <1>
3203
                                 <1>
                                          ; 22/04/2017
3204 00010980 29C0
                                 <1>
                                          sub eax, eax
                                                [esi+IRQ.owner], al ; 0
3205 00010982 8886[F26A0100]
                                 <1>
                                          mov
3206
                                 <1>
                                          ; 10/06/2017
                                          xchg al, [esi+IRQ.method]
3207 00010988 8686[046B0100]
                                 <1>
                                                al, 80h
3208 0001098E 2480
                                 <1>
                                          and
                                          jz short scbs 12
3209 00010990 745E
                                 <1>
                                          ; Audio device must be disabled -later- ! ([IRQ.medhod] = 80h)
3210
                                 <1>
3211
                                 <1>
                                                byte [esi+IRQ.method], 80h; device drv or kernel extension?
3212
                                 <1>;
                                          cmp
                                                 short scbs_12; bh = 0 reset to default IRQ handler
3213
                                 <1>;
3214
                                 <1>;
                                 <1>;
3215
                                                al, al
                                          and
3216
                                 <1>;
                                                short scbs_5 ; the caller is 'syscalbac'
                                          jz
3217
                                 <1>;
                                          ; The caller is 'sysaudio' or ...
3218 00010992 30C0
                                 <1>
                                          xor al, al
                                                [esi+IRQ.method], al ; 0 ; reset kernel extension flag
3219
                                 <1>;
                                          mov
3220
                                 <1> ;scbs_5:
3221
                                 <1>;
                                          sub
                                                ah, ah
3222
                                 <1>
                                          ;mov [u.r0], eax; 0
3223 00010994 C3
                                 <1>
3224
                                 <1>
                                 <1> scbs_6:
3225
3226
                                <1> ; 14/04/2017
3227 00010995 20C0
                                 <1>
                                          and al, al
3228 00010997 7405
                                 <1>
                                                short scbs_7 ; the caller is 'syscalbac'
                                          ; AL = user number ([u.uno] or [audio.user] or ...)
                                 <1>
                                          ; The caller is 'sysaudio' or ...
3230
                                 <1>
3231
                                 <1>
3232
                                 <1>
                                          ; bh = method (0 = signal response byte, 1 = callback)
3233
                                 <1>
                                                (2 = auto increment of signal response byte)
                                 <1>
3235 00010999 80CF80
                                <1>
                                                bh, 80h
                                                                    ; Kernel extension flag !
                                          or
                                                short scbs_8
3236 0001099C EB0A
                                 <1>
                                          jmp
3237
                                 <1> scbs 7:
3238 0001099E 8A86[046B0100]
                                 <1>
                                                 al, [esi+IRQ.method] ; >= 80h = kernel is using this IRQ
                                          mov
                                                al, 80h; use only bit 7 (kernel function flag)
3239 000109A4 2480
                                <1>
                                          and
3240 000109A6 08C7
                                 <1>
                                                 bh, al
                                                            ; method
3241
                                 <1>
                                                              ; 0 = signal response byte, 1 = callback
                                                               ; 2 = auto increment of s.r.b.
3242
                                 <1>
                                 <1> scbs 8:
3243
3244 000109A8 A0[B3030300]
                                 <1>
                                          mov
                                                 al, [u.uno]; user (process) number (1 to 16)
3245 000109AD 8886[F26A0100]
                                                 [esi+IRQ.owner], al ; lock the IRQ for user
                                 <1>
                                          mov
3246 000109B3 88BE[046B0100]
                                 <1>
                                                 [esi+IRQ.method], bh
3247
                                 <1>
                                 <1> ;
3248
                                                bh, 1
                                          test
3249
                                 <1>;
                                                short scbs 9 ; Callback method, CX will not be used
                                          jnz
3250
                                 <1>;
3251
                                                             ; use auto increment (counter) method
                                 <1>;
3252
                                 <1>;
                                                 short scbs_10; (count can be used for buffer switch)
                                          jΖ
3253
                                 <1> ;scbs_9:
                                                 ecx, ecx ; 0
3254
                                 <1> ;
                                          xor
3255
                                 <1> scbs_10:
3256
                                                [esi+IRQ.method], bh
                                 <1>
3257 000109B9 888E[0D6B0100]
                                 <1>
                                          mov
                                                 [esi+IRQ.srb], cl
3258 000109BF C686[FB6A0100]00
                                 <1>
                                                 byte [esi+IRQ.dev], 0 ; device number is always 0
                                          mov
                                                              ; for this system call
                                 <1>
3260
                                 <1>
                                          ;test bh, 1
3261 000109C6 80E701
                                 <1>
                                                bh, 1 ; 17/04/2017
3262 000109C9 7513
                                 <1>
                                                short scbs_11 ; callback method, use virtual address
                                          jnz
3263
                                 <1>
3264 000109CB 53
                                          push ebx ; IRQ number (in BL)
                                 <1>
3265 000109CC 89D3
                                <1>
                                          mov ebx, edx
                                 <1>
                                          ; ebx = virtual address
3267
                                 <1>
                                          ; [u.pgdir] = page directory's physical address
3268 000109CE FE05[926A0100]
                                                  byte [no page swap] ; 1
                                 <1>
                                          inc
3269
                                 <1>
                                                       ; Do not add this page to swap queue
3270
                                 <1>
                                                       ; and remove it from swap queue if it is
3271
                                 <1>
                                                       ; on the queue.
3272 000109D4 E81349FFFF
                                          call get_physical_addr
                                <1>
3273 000109D9 5B
                                <1>
                                                 ebx
3274 000109DA 728A
                                <1>
                                                scbs 3 ; invalid address !
                                          jс
3275
                                <1>
                                          ; eax = physical address of the virtual address in user's space
3276 000109DC 89C2
                                 <1>
3277
                                 <1> scbs 11:
```

```
3278 000109DE 66C1E602
                                                              ; byte (index) to dword (offset)
                                  <1>
                                            shl
                                                  si, 2
3279 000109E2 8996[166B0100]
                                  <1>
                                           mov
                                                  [esi+IRQ.addr], edx
3280
                                  <1>
3281 000109E8 FE05[D6030300]
                                  <1>
                                                  byte [u.irqc]; increase IRQ (in use) count
                                  <1>
3282
                                                   bh ; 17/04/2017
3283 000109EE FEC7
                                  <1>
                                            inc
                                            ; bh > 0 -> set to requested IRQ handler (IRQ u list)
3284
                                 <1>
                                 <1> scbs_12:
3285
3286 000109F0 88D8
                                 <1>
                                                  al, bl ; IRQ number
3287 000109F2 88FC
                                                  ah, bh ; 0 = reset, >0 = set
                                 <1>
                                            mov
3288 000109F4 E8CFF3FFFF
                                 <1>
                                            call set_hardware_int_vector
                                  <1>
3290 000109F9 31C0
                                  <1>
                                                   eax, eax
                                            xor
3291
                                 <1>
                                           ;mov [u.r0], eax ; 0
3292
                                  <1>
3293 000109FB C3
                                  <1>
                                            retn ; return with success (cf=0, eax=0)
3294
                                  <1>
3295
                                  <1>
3296
                                  <1> sysdma: ; DMA FUNCTIONS
                                         ; 02/09/2017
3297
                                  <1>
3298
                                  <1>
                                           ; 28/08/2017
                                           ; 20/08/2017 - TRDOS 386 (TRDOS v2.0)
3299
                                  <1>
3300
                                  <1>
3301
                                  <1>
                                           ; Inputs:
3302
                                                  BH = 0 -> Allocate DMA buffer
                                  <1>
3303
                                  <1>
                                                    BL = 0 \rightarrow Use the system's default DMA
3304
                                  <1>
                                                              (SB16) Buffer
3305
                                  <1>
                                                         Buffer Size (max.) = 65536 bytes
3306
                                  <1>
                                                     BL > 0 -> Allocate (a new) DMA buffer
                                                     ECX = DMA Buffer Size in bytes (<=128KB)
3307
                                  <1>
3308
                                  <1>
                                                     EDX = Virtual Address of DMA buffer
3309
                                  <1>
                                                  BH = 1 -> Initialize (Start) DMA service
3310
                                  <1>
3311
                                  <1>
                                                    BL, bit 0 to 3 = Channel Number (0 to 7)
3312
                                  <1>
                                                       BL, bit 7 = Auto Initialized Mode
3313
                                  <1>
                                                                (If bit 7 is set)
                                                         bit 6 = Record (read) mode (0= playback)
3314
                                  <1>
3315
                                  <1>
                                                       ECX = byte count (0 = use dma buffer size)
3316
                                  <1>
                                                       EDX = physical buffer address
3317
                                  <1>
                                                            (0 = use dma buffer -start- address)
3318
                                  <1>
3319
                                  <1>
                                                  BH = 2 -> Get Current DMA Buffer Offset
                                                       BL = DMA channel number
3320
                                  <1>
3321
                                  <1>
3322
                                  <1>
                                                  BH = 3 -> Get Current DMA count down value
                                                       BL = DMA channel number (0 to 7)
3323
                                  <1>
3324
                                  <1>
3325
                                  <1>
                                                  BH = 4 -> Get Current DMA channel (in progress)
3326
                                  <1>
3327
                                  <1>
                                                  BH = 5 -> Get System's Default DMA Buffer Address
3328
                                  <1>
3329
                                  <1>
                                                  BH = 6 -> Get Current DMA Buffer Address
3330
                                  <1>
3331
                                  <1>
                                                  BH = 7 -> Stop DMA service
3332
                                  <1>
3333
                                  <1>
3334
                                  <1>
                                           ; Outputs:
3335
                                  <1>
3336
                                  <1>
                                                   For BH = 0; Allocate DMA buffer
3337
                                                   EAX = Physical address of DMA buffer
                                  <1>
3338
                                  <1>
                                                      ECX = Allocated buffer size in bytes
                                                           - page count * 4096 -
3339
                                  <1>
3340
                                  <1>
                                                           (may be bigger than requested)
3341
                                  <1>
                                                      If BL input > 0,
3342
                                  <1>
                                                         'sysalloc:' system call will be used with
                                                         EBX (for 'sysalloc') = EDX (for 'sysdma')
3343
                                  <1>
3344
                                  <1>
                                                         ECX is same, byte count (buffer size)
3345
                                  <1>
                                                         EDX = 1024*1024*16; 16 MB upper limit
3346
                                  <1>
                                                      If BL input = 0,
3347
                                  <1>
                                                         Default DMA buffer (SB16 buffer) will be
3348
                                  <1>
                                                          checked and if it is free, it's address
3349
                                  <1>
                                                          will be returned in EAX and it's size
3350
                                  <1>
                                                         will be returned in ECX (as 65536)
3351
                                  <1>
                                                      If CF = 1, error code is in EAX
3352
                                  <1>
                                                         EAX = -1; DMA buffer allocation error!
3353
                                  <1>
3354
                                  <1>
                                                         EAX = 11 ; 'Permission Denied' error !
3355
                                  <1>
3356
                                  <1>
                                                         Note: 'sysalloc' error return method
3357
                                                               will be applied if BL input > 0 !
                                  <1>
3358
                                  <1>
3359
                                  <1>
                                                    For BH = 1; Initialize (Start) DMA
                                                       EAX = 0 (Successful)
3360
                                  <1>
3361
                                  <1>
                                                       If CF = 1, error code is in EAX
3362
                                  <1>
                                                   For BH = 2 ; Get Current DMA Buffer Offset
3363
                                  <1>
                                                       EAX = DMA Buffer Offset (in bytes)
3364
                                  <1>
3365
                                  <1>
3366
                                  <1>
                                                        AX = DMA buffer offset
                                                       EAX bits 16 to 23 = Page register value
3367
                                  <1>
3368
                                  <1>
3369
                                  <1>
                                                   For BH = 3 ; Get Current DMA count down value
3370
                                  <1>
                                                       EAX = Count down value (remain bytes)
3371
                                  <1>
                                                   For BH = 4 ; Get Current DMA channel (in progress)
3372
                                  <1>
                                                       EAX = DMA channel number (0 to 7)
3373
                                  <1>
                                                         AH = 0 if the owner is the caller process
3374
                                  <1>
                                                         AH > 0 if the dma channel is in use by
3375
                                  <1>
3376
                                  <1>
                                                                another user/process
                                                       EAX = -1 (OFFFFFFFF)
3377
                                  <1>
3378
                                  <1>
                                                             if DMA service is not in use
3379
                                  <1>
                                                             (stopped or not initialized/started)
3380
                                  <1>
3381
                                  <1>
                                                    For BH = 5; Get System's Default DMA Buff Addr
3382
                                  <1>
                                                       EAX = Default DMA Buffer Address (Physical)
```

```
= offset 'sb16 dma buffer:'
3383
                                 <1>
3384
                                  <1>
                                                      ECX = Buffer size
3385
                                 <1>
                                                        = 65536
3386
                                 <1>
3387
                                 <1>
                                                  For BH = 6 ; Get Current DMA Buffer Address
                                                  EAX = Current DMA buffer address (Physical)
ECX = Current DMA buffer size (setting value)
3388
                                 <1>
3389
                                                    Note: These values are for current dma channel
3390
                                 <1>
3391
                                 <1>
                                                          settings for the user/process
                                                    ** For now (for current TRDOS 386 version)
3392
                                 <1>
3393
                                 <1>
                                                      only one user/process can use only one
3394
                                 <1>
                                                       dma channel & one dma buffer at same time
                                                        (no multi tasking on DMA service) !!! **
3395
                                 <1>
3396
                                 <1>
                                                    (Once, current DMA user must stop it's own DMA
                                                       DMA service, than another user/program
3397
                                 <1>
3398
                                 <1>
                                                       can use DMA service with same dma channel
                                                       or with another DMA channel.)
3399
                                 <1>
                                 <1>
3400
3401
                                 <1>
                                                  For BH = 7; Stop DMA service (for current user
                                                    and current DMA channel)
3402
                                 <1>
3403
                                 <1>
                                                      EAX = 0; successful
                                                      CF = 1 \& EAX > 0 (= -1) -> Error
3404
                                 <1>
3405
                                 <1>
3406 000109FC 80FF07
                                 <1>
                                                 bh, 7
3407 000109FF 7612
                                 <1>
                                                 short sysdma_0
                                           jna
3408
                                 <1>
                                <1> sysdma err:
3410 00010A01 31C0
                                <1>
                                           xor
                                                  eax, eax
3411 00010A03 48
                                 <1>
                                                 eax ; -1
                                 <1> sysdma_perm_err:
3412
3413 00010A04 A3[64030300]
                                <1>
                                           mov
                                                 [u.r0], eax
3414 00010A09 A3[C8030300]
                                 <1>
                                           mov
                                                 [u.error], eax ; DMA service error !
3415 00010A0E E92FBDFFFF
                                 <1>
                                           jmp
                                                 error
3416
                                 <1>
3417
                                 <1> sysdma_0:
3418 00010A13 08FF
                                 <1>
                                           or
                                                 bh, bh
3419 00010A15 0F85BA000000
                                 <1>
                                                 sysdma_1
                                           jnz
                                 <1>
3421 00010A1B 20DB
                                 <1>
                                                 bl, bl
3422 00010A1D 7416
                                 <1>
                                                 short sysdma_01
                                           jz
3423
                                 <1>
3424
                                 <1>
                                           ; redirect system call to 'sysalloc'
                                           mov ebx, edx; virtual address of DMA buffer
3425 00010A1F 89D3
                                 <1>
3426
                                 <1>
                                           ;ecx = Buffer size in bytes
3427
                                 <1>
                                           ; DMA buffer address <= 16MB upper limit
                                           mov edx, 1024*1024*16; 16MB limit for DMA buff
3428 00010A21 BA00000001
                                 <1>
                                 <1>
3430 00010A26 C705[846F0100]FFFF- <1>
                                                  dword [dma_addr], OFFFFFFFF ; -1
                                           mov
3430 00010A2E FFFF
                                 <1>
                                 <1>
3432 00010A30 E99DE5FFFF
                                 <1>
                                                  sysalloc
3433
                                 <1>
3434
                                 <1> sysdma_01:
3435 00010A35 B8[00000200]
                                 <1>
                                                  eax, sb16_dma_buffer
3436
                                 <1>
3437 00010A3A 803D[416B0100]01
                                <1>
                                                  byte [audio_device], 1
3438 00010A41 722A
                                 <1>
                                           jb
                                                  short sysdma 03
3439
                                 <1>
3440 00010A43 3B05[606B0100]
                                 <1>
                                                  eax, [audio_dma_buff]
3441 00010A49 7507
                                 <1>
                                           jne
                                                 short sysdma_02
3442
                                 <1>
                                 <1> sysdma 0 err:
3444 00010A4B B80B000000
                                 <1>
                                                 eax, ERR_PERM_DENIED
                                           mov
3445 00010A50 EBB2
                                 <1>
                                                  short sysdma_perm_err
3446
                                 <1>
3447
                                 <1> sysdma_02:
3448
                                 <1>
                                        ; Only one user is permitted for audio/dma functions
3449
                                 <1>
3450 00010A52 833D[606B0100]00
                                 <1>
                                                  dword [audio_dma_buff], 0
                                           cmp
3451 00010A59 7612
                                 <1>
                                                 short sysdma_03
                                           jna
                                 <1>
3453 00010A5B 8A1D[696B0100]
                                 <1>
                                                 bl, [audio_user]
                                           mov
3454 00010A61 08DB
                                                 bl, bl
                                 <1>
                                           or
3455 00010A63 7408
                                 <1>
                                           jΖ
                                                  short sysdma_03
3456
                                 <1>
3457 00010A65 3A1D[B3030300]
                                 <1>
                                           cmp
                                                 bl, [u.uno]
3458 00010A6B 75DE
                                 <1>
                                                 short sysdma_0_err
                                           jne
3459
                                 <1>
3460
                                 <1> sysdma_03:
                                                 bl, [dma_user]
3461 00010A6D 8A1D[816F0100]
                                 <1> mov
3462 00010A73 20DB
                                 <1>
                                           and
                                                 bl, bl
3463 00010A75 750E
                                  <1>
                                           jnz
                                                  short sysdma_04
                                 <1>
3465 00010A77 8A1D[B3030300]
                                 <1>
                                                  bl, [u.uno]
3466 00010A7D 881D[816F0100]
                                 <1>
                                           mov
                                                  [dma_user], bl
3467
                                 <1>
3468 00010A83 EB15
                                 <1>
                                                  short sysdma_05
                                           qmŗ
                                 <1>
3469
3470
                                 <1> sysdma_04:
                                                  esi, [dma_addr]
3471 00010A85 8B35[846F0100]
                                           mov
                                 <1>
3472 00010A8B 21F6
                                 <1>
                                           and
                                                 esi, esi
3473 00010A8D 740B
                                 <1>
                                           jz
                                                  short sysdma_05
3474
                                 <1>
3475 00010A8F 46
                                 <1>
                                           inc
                                                  esi ; -1 -> 0
3476 00010A90 7408
                                 <1>
                                                  short sysdma 05
                                           jz
3477
                                 <1>
3478 00010A92 3A1D[B3030300]
                                 <1>
                                                 bl, [u.uno]
                                           cmp
3479 00010A98 75B1
                                 <1>
                                                  short sysdma_0_err
                                           jne
3480
                                 <1>
3481
                                 <1> sysdma 05:
                                           ; edx = virtual address (user's buffer address)
3482
                                 <1>
3483
                                 <1>
3484 00010A9A 81F900000100
                                 <1>
                                           cmp
                                                 ecx, 65536 ; byte count (buffer size)
3485 00010AA0 0F875BFFFFFF
                                 <1>
                                           jа
                                                  sysdma_err
                                 <1>
3486
```

```
3487 00010AA6 81C1FF0F0000
                                <1>
                                           add ecx, PAGE_SIZE-1; 4095
                                                 cx, ~PAGE_OFF; not 4095
3488 00010AAC 6681E100F0
                                 <1>
                                           and
                                           ;cmp ecx, 6553\overline{6}
                                 <1>
3490
                                 <1>
                                           ;ja sysdma_err;
                                           push ecx ; buffer size (allocated pages * 4096)
push eax ; offset sb16_dma_buffer
3491 00010AB1 51
                                 <1>
3492 00010AB2 50
                                 <1>
3493 00010AB3 89D3
                                <1>
                                           mov ebx, edx
3494 00010AB5 C1E90C
                                 <1>
                                           shr ecx, 12; byte count to page count
3495
                                 <1>
                                           ; eax = physical address of (audio) dma buffer
                                           ; ebx = virtual address of (audio) dma buffer (user's pgdir)
3496
                                 <1>
3497
                                 <1>
                                           ; ecx = page count (>0)
3498 00010AB8 E8414CFFFF
                                 <1>
                                           call direct_memory_access
3499 00010ABD 58
                                 <1>
                                           pop
                                                  eax
3500 00010ABE 59
                                 <1>
                                           pop
                                                 ecx
3501 00010ABF 0F823CFFFFFF
                                 <1>
                                           jс
                                                  sysdma_err
3502
                                 <1>
3503 00010AC5 A3[846F0100]
                                 <1>
                                           mov
                                                  [dma addr], eax
3504 00010ACA 890D[886F0100]
                                 <1>
                                                  [dma_size], ecx; dma buffer size (in bytes)
                                           mov
3505
                                 <1>
                                                  [u.r0], eax; DMA Buffer Address (Physical)
3506
                                 <1>
                                           ; mov
3507
                                  <1>
3508
                                  <1>
                                           ;mov
                                                  ebp, [u.usp] ; ebp points to user's registers
                                  <1>
                                           ;mov
3509
                                                  [ebp+24], ecx; return to user with ecx value
3510
                                  <1>
                                  <1>
                                           ;jmp sysret
3511
3512
                                  <1>
3513
                                 <1>
                                           ; 28/08/2017
3514 00010AD0 E9C4000000
                                 <1>
                                           jmp
                                                 sysdma_51
3515
                                 <1>
3516
                                 <1> sysdma 1:
3517 00010AD5 80FF01
                                 <1>
                                           cmp
                                                 bh, 1
3518 00010AD8 0F87A6000000
                                                  sysdma 5
                                 <1>
                                           jа
3519
                                 <1>
3520 00010ADE F6C340
                                 <1>
                                           test bl, 40h
                                                               ; record (read) mode -BL, bit 6-
3521 00010AE1 0F851AFFFFFF
                                 <1>
                                                  sysdma_err ; not ready yet!
                                           jnz
3522
                                 <1>
3523 00010AE7 A1[846F0100]
                                 <1>
                                           mov
                                                  eax, [dma addr] ; physical address of dma buffer
3524 00010AEC 21C0
                                 <1>
                                           and
                                                  eax, eax
3525 00010AEE 0F840DFFFFFF
                                 <1>
                                                  sysdma_err
                                           jΖ
3526
                                 <1>
3527 00010AF4 09D2
                                 <1>
                                           or
                                                  edx, edx
                                                 short sysdma 11
3528 00010AF6 7504
                                 <1>
                                           jnz
3529
                                 <1>
3530 00010AF8 89C2
                                 <1>
                                                  edx, eax
3531 00010AFA EB08
                                 <1>
                                           jmp
                                                  short sysdma_12
3532
                                 <1> sysdma_11:
3533 00010AFC 39C2
                                 <1>
                                           cmp
                                                  edx, eax
3534 00010AFE 0F82FDFEFFFF
                                 <1>
                                           jb
                                                  sysdma_err
3535
                                 <1> sysdma_12:
3536 00010B04 21C9
                                 <1>
                                                  ecx, ecx
                                           and
3537 00010B06 7508
                                 <1>
                                                  short sysdma_13
3538
                                 <1>
3539 00010B08 8B0D[886F0100]
                                 <1>
                                           mov
                                                  ecx, [dma_size]
3540 00010B0E EB0C
                                 <1>
                                                  short sysdma_14
                                           jmp
3541
                                 <1> sysdma_13:
3542 00010B10 3B0D[886F0100]
                                 <1>
                                           cmp
                                                  ecx, [dma_size]
3543 00010B16 0F87E5FEFFFF
                                 <1>
                                                  sysdma err
                                           jа
3544
                                 <1> sysdma_14:
3545 00010B1C 89C6
                                 <1>
                                           mov
                                                  esi, eax
3546 00010B1E 0335[886F0100]
                                 <1>
                                                  esi, [dma_size]
                                           add
3547
                                 <1>
3548 00010B24 89D0
                                 <1>
                                           mov
                                                  eax, edx
3549 00010B26 01C8
                                 <1>
                                           add
                                                  eax, ecx
3550 00010B28 0F82D3FEFFFF
                                 <1>
                                                  sysdma_err ; 02/09/2017
                                           jс
3551
                                 <1>
3552 00010B2E 39F0
                                 <1>
                                           cmp
                                                  eax, esi
3553 00010B30 0F87CBFEFFFF
                                 <1>
                                           jа
                                                  sysdma_err
3554
                                 <1>
3555 00010B36 8B3D[606B0100]
                                 <1>
                                                  edi, [audio_dma_buff]
3556 00010B3C 8B35[846F0100]
                                 <1>
                                                  esi, [dma_addr]
                                           mov
3557
                                  <1>
3558 00010B42 09FF
                                  <1>
                                                  edi, edi
                                           or
3559 00010B44 7424
                                 <1>
                                           jz
                                                  short sysdma_16
3560
                                 <1>
                                                  byte [audio_device], 1
3561 00010B46 803D[416B0100]01
                                 <1>
                                           cmp
                                                  short sysdma 15
3562 00010B4D 7208
                                 <1>
                                           jb
3563
                                 <1>
3564
                                           ; Sound Blaster 16
                                 <1>
3565 00010B4F 39FE
                                  <1>
                                                 esi, edi
3566 00010B51 0F84F4FEFFFF
                                 <1>
                                                  sysdma_0_err ; permmission denied !
                                           jе
3567
                                  <1>
3568
                                  <1> sysdma_15:
                                                  byte [dma_mode], 48h; single mode playback
3569 00010B57 C605[836F0100]48
                                  <1>
                                           mov
                                  <1>
                                           test bl, 80h; DMA mode - BL, bit 7, auto init -
3571 00010B5E F6C380
                                  <1>
3572 00010B61 7407
                                  <1>
                                           jz
                                                  short sysdma_16
                                           ; Auto initialized playback (write) mode
                                  <1>
3574 00010B63 8005[836F0100]10
                                 <1>
                                           add byte [dma_mode], 10h; = 58h
3575
                                  <1> sysdma_16:
                                                  bl, 07h
3576 00010B6A 80E307
                                 <1>
                                           and
3577 00010B6D 881D[826F0100]
                                                  [dma_channel], bl
                                 <1>
                                           mov
3578 00010B73 8915[8C6F0100]
                                  <1>
                                           mov
                                                  [dma_start], edx
3579 00010B79 890D[906F0100]
                                 <1>
                                           mov
                                                  [dma_count], ecx
                                  <1>
3581
                                  <1>
                                           ; 28/08/2017
                                           ; call dma init
3582
                                  <1>
                                  <1>
                                           ;jmp sysret
3584 00010B7F E94B010000
                                                 dma init
                                 <1>
                                           jmp
3585
                                 <1>
                                 <1> sysdma 5:
3586
3587 00010B84 80FF05
                                                 bh, 5
                                 <1>
                                           cmp
3588 00010B87 7223
                                 <1>
                                           jb
                                                  short sysdma 3
3589 00010B89 0F87CE000000
                                 <1>
                                           jа
                                                  sysdma 6
3590
                                  <1>
3591
                                  <1>
                                           ; Get the system's default dma buffer addr and size
```

```
3592 00010B8F B8[00000200]
                                 <1>
                                                 eax, sb16 dma buffer
                                           mov
3593 00010B94 B900000100
                                 <1>
                                           mov
                                                 ecx, 6553\overline{6}; Buffer size in bytes
3594
                                 <1>
3595
                                 <1> sysdma 51:
3596
                                           ; 0 = there is not a dma buffer (in use or available)
                                 <1>
3597 00010B99 A3[64030300]
                                 <1>
                                           mov
                                                 [u.r0], eax
                                 <1>
3599 00010B9E 8B2D[60030300]
                                 <1>
                                                  ebp, [u.usp] ; ebp points to user's registers
                                           mov
3600 00010BA4 894D18
                                 <1>
                                                  [ebp+24], ecx; return to user with ecx value
                                           mov
3601
                                 <1>
3602 00010BA7 E9B6BBFFFF
                                 <1>
                                                  sysret
3603
                                  <1>
3604
                                 <1> sysdma_3:
                                           cmp
3605 00010BAC 80FF03
                                 <1>
                                                  bh, 3
3606 00010BAF 7231
                                 <1>
                                           ib
                                                  short sysdma 2
3607 00010BB1 776B
                                                  short sysdma 4
                                 <1>
                                           jа
3608
                                 <1>
                                         ; Get current dma count down value (remain bytes)
3609
                                 <1>
                                           ; 28/08/2017
3610
                                  <1>
3611 00010BB3 0FB635[826F0100]
                                 <1>
                                           movzx esi, byte [dma channel]
3612 00010BBA 0FB696[CE160100]
                                 <1>
                                           movzx edx, byte [dma_flip+esi]
3613 00010BC1 EE
                                           out dx, al
                                                                     ; flip-flop clear
                                  <1>
3614 00010BC2 8A96[AE160100]
                                 <1>
                                                 dl, [dma_cnt+esi] ; dma count register addr
                                           mov
                                                 al, dx
3615 00010BC8 EC
                                 <1>
                                           in
                                           movzx ebx, al
3616 00010BC9 0FB6D8
                                 <1>
3617 00010BCC EC
                                 <1>
                                           in
                                                  al, dx
3618 00010BCD 88C7
                                 <1>
                                                   bh, al
                                           mov
3619
                                 <1>
3620 00010BCF 6683FE04
                                 <1>
                                                  si, 4 ; channel number ?
                                           cmp
3621 00010BD3 7202
                                                  short sysdma_31 ; 8 bit dma channel
                                 <1>
                                           jb
3622
                                 <1>
3623 00010BD5 D1E3
                                  <1>
                                                  ebx, 1; word count to byte count
3624
                                 <1>
3625
                                  <1> sysdma 31:
3626 00010BD7 891D[64030300]
                                 <1>
                                                  [u.r0], ebx
                                           mov
3627
                                 <1>
3628 00010BDD E980BBFFFF
                                  <1>
                                                  sysret
                                           jmp
3629
                                  <1>
3630
                                  <1> sysdma_2:
                                       ; Get current dma buffer offset (& page)
3631
                                  <1>
                                           ; 28/08/2017
3632
                                  <1>
3633 00010BE2 0FB635[826F0100]
                                  <1>
                                           movzx esi, byte [dma channel]
3634 00010BE9 0FB696[CE160100]
                                           movzx edx, byte [dma_flip+esi]
                                 <1>
3635 00010BF0 EE
                                           out dx, al
                                                                     ; flip-flop clear
                                  <1>
                                           mov dl, [dma_adr+esi]
3636 00010BF1 8A96[A6160100]
                                 <1>
3637 00010BF7 EC
                                                                     ; get dma position
                                 <1>
                                           in
                                                  al, dx
3638 00010BF8 0FB6D8
                                 <1>
                                           movzx ebx, al
3639 00010BFB EC
                                                  al, dx
                                 <1>
                                           in
3640 00010BFC 88C7
                                 <1>
                                           mov
                                                 bh, al
                                 <1>
3642 00010BFE 6683FE04
                                 <1>
                                           cmp
                                                  si, 4 ; channel number ?
3643 00010C02 7202
                                  <1>
                                           jb
                                                  short sysdma 21 ; 8 bit dma channel
3644
                                 <1>
3645 00010C04 D1E3
                                 <1>
                                                  ebx, 1; word offset to byte offset
3646
                                  <1>
3647
                                  <1> sysdma_21:
3648 00010C06 891D[64030300]
                                 <1>
                                                  [u.r0], ebx
3649
                                  <1>
3650 00010C0C 8A96[B6160100]
                                  <1>
                                                  dl, [dma_page+esi]
3651 00010C12 EC
                                 <1>
                                           in
                                                  al, dx
                                                                      ; get dma page
3652
                                 <1>
3653
                                  <1>
                                           ;add
                                                [u.ro+2], al
3654 00010C13 0805[66030300]
                                 <1>
                                                  [u.r0+2], al
                                           or
3655
                                  <1>
3656 00010C19 E944BBFFFF
                                  <1>
                                           jmp
                                                  sysret
3657
                                  <1>
3658
                                  <1> sysdma 4:
3659
                                        ; Get current DMA channel number
                                 <1>
3660
                                 <1>
                                           ; 28/08/2017
3661 00010C1E 8A25[816F0100]
                                 <1>
                                           mov ah, [dma_user]
3662 00010C24 20E4
                                  <1>
                                           and
                                                 ah, ah
3663 00010C26 750F
                                  <1>
                                           jnz
                                                 short sysdma_42
                                  <1>
3664
3665
                                  <1> sysdma_41:
3666
                                 <1>
                                           ; Not a valid dma channel (in use)
3667 00010C28 C705[64030300]FFFF- <1>
                                                 dword [u.r0], -1; OFFFFFFFh
                                           mov
3667 00010C30 FFFF
                                  <1>
3668 00010C32 E92BBBFFFF
                                 <1>
                                           jmp
                                                  sysret
3669
                                  <1>
3670
                                  <1> sysdma_42:
3671 00010C37 8B35[846F0100]
                                  <1>
                                           mov
                                                  esi, [dma_addr]
3672 00010C3D 21F6
                                  <1>
                                           and
                                                  esi, esi
3673 00010C3F 74E7
                                 <1>
                                                  short sysdma_41
                                  <1>
3675 00010C41 46
                                  <1>
                                                  esi ; -1 -> 0
                                           inc
3676 00010C42 74E4
                                 <1>
                                           jz
                                                  short sysdma_41
                                  <1>
                                                  al, [dma_channel]
3678 00010C44 A0[826F0100]
                                 <1>
                                           mov
3679
                                  <1>
3680 00010C49 3A25[B3030300]
                                  <1>
                                                  ah, [u.uno]
                                           cmp
3681 00010C4F 7502
                                  <1>
                                                  short sysdma_43
3682
                                  <1>
3683 00010C51 30E4
                                 <1>
                                                  ah, ah ; DMA channel in use by current user
                                           xor
3684
                                 <1>
3685
                                 <1> sysdma 43:
3686 00010C53 A3[64030300]
                                                  [u.r0], eax; AL = dma channel number
                                 <1>
                                           mov
                                                           ; AH > 0 if the the channel
                                  <1>
                                                             ; in use by another user/process
3688
                                 <1>
3689 00010C58 E905BBFFFF
                                 <1>
                                 <1>
3690
3691
                                 <1> sysdma_6:
3692 00010C5D 80FF06
                                 <1>
                                                  bh, 6
                                           cmp
3693 00010C60 7710
                                 <1>
                                           jа
                                                  short sysdma 7
3694
                                  <1>
3695
                                  <1>
                                           ; 28/08/2017
```

```
3696
                                 <1>
                                           ; Get current DMA buffer addr and size
                                                 eax, [dma_addr] ; dma buffer address
ecx, [dma_size] ; dma buffer size (in bytes)
3697 00010C62 A1[846F0100]
                                 <1>
3698 00010C67 8B0D[886F0100]
                                 <1>
                                           mov
3699
                                 <1>
3700 00010C6D E927FFFFF
                                  <1>
                                                  sysdma 51
                                           jmp
3701
                                  <1>
3702
                                  <1> sysdma 7:
3703
                                           ; DMA service STOP
                                 <1>
3704 00010C72 A0[B3030300]
                                 <1>
                                                 al, [u.uno]
                                           mov
3705 00010C77 3A05[816F0100]
                                 <1>
                                           cmp
                                                  al, [dma user]
3706 00010C7D 751D
                                 <1>
                                                  short sysdma_72
                                  <1>
3708 00010C7F 28C0
                                 <1>
                                           sub
                                                  al, al ; 0
3709
                                 <1>
3710 00010C81 A2[816F0100]
                                 <1>
                                           mov
                                                  [dma_user], al ; clear user
3711
                                  <1>
3712 00010C86 8605[836F0100]
                                 <1>
                                           xchg
                                                 al, [dma_mode]
3713 00010C8C 20C0
                                  <1>
                                            and
                                                  al, al
3714
                                  <1>
                                            ;jz
                                                  short sysdma_err
3715 00010C8E 7527
                                 <1>
                                                  short sysdma_73
                                           jnz
3716
                                 <1>
3717
                                 <1> sysdma 71:
3718 00010C90 31C0
                                 <1>
                                                  eax, eax
                                           xor
3719 00010C92 A3[64030300]
                                 <1>
                                                  [u.r0], eax; 0
                                                  sysret
3720 00010C97 E9C6BAFFFF
                                 <1>
                                           jmp
3721
                                 <1>
3722
                                  <1> sysdma 72:
                                           ; 28/08/2017
3723
                                 <1>
3724 00010C9C 803D[816F0100]00
                                 <1>
                                                  byte [dma_user], 0
                                           cmp
3725 00010CA3 76EB
                                                  short sysdma_71; Nothing to do!
                                  <1>
                                           jna
3726
                                  <1>
3727 00010CA5 833D[846F0100]00
                                  <1>
                                                  dword [dma addr], 0
                                           cmp
3728 00010CAC 0F8799FDFFFF
                                                  sysdma 0 err
                                 <1>
                                           jа
3729
                                  <1>
3730 00010CB2 A2[816F0100]
                                  <1>
                                                  [dma_user], al ; reset to current user
                                           mov
3731
                                  <1>
3732
                                  <1> sysdma_73:
3733
                                           ; 28/08/2017
                                 <1>
3734 00010CB7 0FB635[826F0100]
                                           movzx esi, byte [dma channel]
                                 <1>
3735 00010CBE 0FB696[BE160100]
                                           movzx edx, byte [dma mask+esi]
                                 <1>
3736 00010CC5 A0[826F0100]
                                 <1>
                                           mov al, [dma_channel]
3737 00010CCA 0C04
                                  <1>
                                           or
                                                  al, 4
3738 00010CCC EE
                                 <1>
                                                 dx, al
                                           out
3739
                                 <1>
3740 00010CCD EBC1
                                  <1>
                                           jmp
                                                  short sysdma_71
3741
                                  <1>
                                  <1> dma_init:
3742
                                        ; 28/08/2017
3743
                                 <1>
3744
                                  <1>
                                           ; 20/08/2017
                                           ; DMA initialization
3745
                                  <1>
                                          ; 14/08/2017
3746
                                  <1>
3747
                                  <1>
                                           ; 03/08/2017, 06/08/2017, 08/08/2017
3748
                                           ; 02/07/2017, 13/07/2017, 16/07/2017, 30/07/2017
                                  <1>
3749
                                  <1>
                                           ; (Derived from 'DMA_INIT' procedure in SB16MOD.ASM)
3750
                                  <1>
                                           ; Modified for TRDOS 386 DMA buffer allocation & initialization !
3751
                                  <1>
3752 00010CCF 8B1D[8C6F0100]
                                  <1>
                                           mov
                                                  ebx, [dma start]
3753 00010CD5 8B0D[906F0100]
                                  <1>
                                                  ecx, [dma_count]
                                           mov
3754
                                  <1>
3755 00010CDB 0FB635[826F0100]
                                 <1>
                                           movzx esi, byte [dma_channel]
3756
                                 <1>
3757 00010CE2 6683FE04
                                  <1>
                                                 si, 4
                                           cmp
3758 00010CE6 7205
                                 <1>
                                            jb
                                                  short gdmi1
3759
                                  <1>
                                           ; 08/08/2017
                                            shr cx, 1; word count
3760 00010CE8 66D1E9
                                 <1>
3761 00010CEB D1EB
                                  <1>
                                            shr
                                                  ebx, 1 ; convert byte offset to word offset
3762
                                  <1> gdmi1:
3763
                                                 [dma_poff], bx ; 08/08/2017
                                 <1>
                                           ; mov
3764 00010CED 6649
                                  <1>
                                                                      ; dma size = block size - 1
                                           dec
3765
                                 <1>
3766 00010CEF 0FB696[BE160100]
                                 <1>
                                           movzx edx, byte [dma_mask+esi] ; 30/07/2017
3767 00010CF6 A0[826F0100]
                                  <1>
                                           mov
                                                  al, [dma_channel]
3768 00010CFB 0C04
                                 <1>
                                           or
                                                  al, 4
3769 00010CFD EE
                                                  dx, al
                                 <1>
                                                                      ; dma channel mask
3770
                                  <1>
3771 00010CFE 30C0
                                                  al, al ; 0 ; any value ! 08/08/2017
                                 <1>
                                           xor
3772 00010D00 8A96[CE160100]
                                 <1>
                                                  dl, [dma_flip+esi]
                                           mov
                                                                      ; flip-flop clear
3773 00010D06 EE
                                 <1>
                                           out
                                                  dx, al
3774
                                  <1>
3775 00010D07 8A96[C6160100]
                                 <1>
                                                  dl, [dma mod+esi]
                                           mov
                                                  al, [dma_channel] ; 13/07/2017
3776 00010D0D A0[826F0100]
                                  <1>
                                           mov
3777 00010D12 2403
                                                  al, 3
                                  <1>
                                            and
                                           ; 08/08/2017
                                 <1>
3779 00010D14 0A05[836F0100]
                                 <1>
                                                  al, [dma_mode] ; 58h
                                                                         ; dma mode for SB16
3780 00010D1A EE
                                 <1>
                                           out
                                                  dx, al
3781
                                 <1>
3782 00010D1B 8A96[A6160100]
                                                  dl, [dma adr+esi]
                                 <1>
                                           mov
3783 00010D21 88D8
                                 <1>
                                                  al, bl
                                           mov
3784 00010D23 EE
                                 <1>
                                                  dx, al
                                                                     ; offset low
                                           out
                                 <1>
3786 00010D24 88F8
                                 <1>
                                           mov
                                                  al, bh
3787 00010D26 EE
                                 <1>
                                                                     ; offset high
                                           out
                                                  dx, al
3788
                                 <1>
3789 00010D27 8A96[AE160100]
                                 <1>
                                                  dl, [dma_cnt+esi]
3790 00010D2D 88C8
                                 <1>
                                           mov
                                                  al, cl
3791 00010D2F EE
                                 <1>
                                           out
                                                  dx, al
                                                                     ; size low
                                 <1>
3793 00010D30 88E8
                                 <1>
                                           mov
                                                  al, ch
                                                                      ; size high
3794 00010D32 EE
                                 <1>
                                           out
                                                  dx, al
                                 <1>
3796 00010D33 8A96[B6160100]
                                <1>
                                           mov
                                                  dl, [dma_page+esi]
3797
                                 <1>
                                           ; 14/08/2017
3798 00010D39 6683FE04
                                 <1>
                                           cmp
                                                 si, 4
3799 00010D3D 7305
                                 <1>
                                                  short gdmi2
                                           jnb
3800 00010D3F C1EB10
                                 <1>
                                           shr
                                                 ebx, 16
```

```
3801 00010D42 EB06
                                   jmp short gdmi3
                           <1>
3802
                           <1> gdmi2:
                           <1> ; 09/08/2017
3803
3804 00010D44 C1EB0F
                           <1>
                                    shr ebx, 15
                                                   ; complete 16 bit shift
3805 00010D47 80E3FE
                           <1>
                                         bl, OFEh; clear bit 0 (not necessary)
                                    and
3806
                           <1> gdmi3:
3807 00010D4A 88D8
                           <1> mov
                                         al, bl
3808 00010D4C EE
                                  out
                           <1>
                                         dx, al
                                                         ; page
3809
                           <1>
3810 00010D4D 8A96[BE160100] <1> mov
3811 00010D53 A0[826F0100] <1> mov
3812 00010D58 2403 <1> and
3813 00010D5A EE <1> out
                                         dl, [dma_mask+esi]
                                         al, [dma_channel] ; 13/07/2017
                                         al, 3
3813 00010D5A EE
                                   out dx, al
                           <1>
                                                        ; dma channel unmask
3814
                           <1>
                                 ;retn
; 28/0
3815
                           <1>
                                    ; 28/08/2017
3816
                           <1>
3817 00010D5B E902BAFFFF
                           <1>
                                   jmp sysret
3818
                           <1>
3819
                            <1> otty:
3820
                            <1> sret:
                            <1> ocvt:
3821
3822
                            <1> ctty:
                            <1> cret:
3823
3824
                            <1> ccvt:
3825
                            <1> rtty:
3826
                            <1> wtty:
3827
                            <1> rmem:
3828
                            <1> wmem:
3829
                            <1> rfd:
                            <1> rhd:
3830
3831
                            <1> wfd:
3832
                            <1> whd:
                            <1> rlpt:
3833
3834
                            <1> wlpt:
3835
                            <1> rcvt:
3836
                            <1> xmtt:
3837 00010D60 C3
                            <1> retn
2313
                              %include 'trdosk9.s' ; 04/01/2016
                            1
                            <1> ; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.2) - INITIALIZED DATA : trdosk9.s
  2
  3
                            <1> ; Last Update: 01/09/2020
  5
                            <1>; ------
  6
                            <1>; Beginning: 04/01/2016
  7
  8
                            <1>; Assembler: NASM version 2.14 (trdos386.s)
 10
                            11
                            <1>; Derived from TRDOS Operating System v1.0 (8086) source code by Erdogan Tan
                            <1>; TRDOS2.ASM (09/11/2011)
 12
                            13
 14
                            <1>; DRV INIT.ASM [26/09/2009] Last Update: 07/08/2011
                            <1>; MAINPROG.ASM [17/01/2004] Last Update: 09/11/2011
 15
                            <1>; CMD_INTR.ASM [29/01/2005] Last Update: 09/11/2011
 16
 17
                            <1>; FILE.ASM [29/10/2009] Last Update: 09/10/2011
 18
                            <1>
                            <1> ; 12/02/2016
 19
                            <1> Last_DOS_DiskNo:
 20
 21 00010D61 01
                            <1>
                                        db 1 ; A: = 0 \& B: = 1
 22
                           <1>
 23
                           <1> Restore_CDIR:
 24 00010D62 FF
                           <1>
                                         db OFFh; Initial value -> any number except 0
 25
                           <1>
                            <1> msg_CRLF_temp:
 26
 27 00010D63 070D0A00
                           <1> db 07h, 0Dh, 0Ah, 0
 28
                           <1>
                           <1> Magic_Bytes:
                           <1> db 4 <1> db 1
 30 00010D67 04
 31 00010D68 01
                           <1>
 32
                           <1> mainprog_Version:
 33 00010D69 07
                           <1> db 7
 34 00010D6A 5B5452444F535D204D- <1>
                                         db "[TRDOS] Main Program v2.0.010920"
 34 00010D73 61696E2050726F6772- <1>
 34 00010D7C 616D2076322E302E30- <1>
 34 00010D85 3130393230 <1>
 35 00010D8A 0D0A
                                         db ODh, OAh
                           <1>
 36 00010D8C 286329204572646F67- <1>
                                         db "(c) Erdogan Tan 2005-2020"
 36 00010D95 616E2054616E203230- <1>
 36 00010D9E 30352D32303230
                          <1>
 37 00010DA5 0D0A00
                                         db ODh, OAh, O
                           <1>
 38
                            <1>
                            <1> MainProgCfgFile: ; 14/04/2016
 39
                                         db "MAINPROG.CFG", 0
 40 00010DA8 4D41494E50524F472E- <1>
 40 00010DB1 43464700
                        <1>
                           <1>
 41
                           <1> TRDOSPromptLabel:
 42
 43 00010DB5 5452444F53
                                  db "TRDOS"
                          <1>
 44 00010DBA 00
                          <1>
<1>
                                         db 0
 45 00010DBB 00<rept>
                                         times 5 db 0
 46 00010DC0 00
                          <1>
                                         db 0
 47
                           <1>
 48
                           <1> ; INTERNAL COMMANDS
                           <1> Command_List:
 49
                         <1> Cmd_Dir: db "DIR", 0
 50 00010DC1 44495200
 51 00010DC5 434400
                          <1> Cmd_Cd: db "

<1> Cmd_Drive: db "C:", 0

                                         db "CD", 0
 52 00010DC8 433A00
                           <1> Cmd_Exit: db "EXIT", 0
 57 00010DE2 4C4F4E474E414D4500 <1> Cmd_LongName: db "LONGNAME", 0
                      <1> Cmd_Date: db "DATE", 0 <1> Cmd_Time: db "TIME", 0
 58 00010DEB 4441544500
 59 00010DF0 54494D4500
                           <1> Cmd_Run: db "RUN", 0
 60 00010DF5 52554E00
                           <1> Cmd_Set: db "SET", 0
 61 00010DF9 53455400
```

```
<1> Cmd_Cls:
                                              db "CLS", 0
 62 00010DFD 434C5300
                              <1> Cmd_Show: db "SHOW", 0
<1> Cmd_Del: db "DEL", 0
 63 00010E01 53484F5700
                                              db "SHOW", 0
 64 00010E06 44454C00
 65 00010E0A 41545452494200 <1> Cmd_Attrib: db "ATTRIB", 0
                              <1> Cmd_Rename: db "RENAME", 0
<1> Cmd_Rmdir: db "RMDIR", 0
 66 00010E11 52454E414D4500
 67 00010E18 524D44495200
 68 00010E1E 4D4B44495200
                               <1> Cmd Mkdir: db "MKDIR", 0
 69 00010E24 434F505900
                               <1> Cmd_Copy: db "COPY", 0
 70 00010E29 4D4F564500
                              <1> Cmd_Move:
                                              db "MOVE", 0
                                              db "PATH", 0
 71 00010E2E 5041544800
                              <1> Cmd Path:
                              <1> Cmd_Mem:
                                              db "MEM", 0
 72 00010E33 4D454D00
 73 00010E37 00
                               <1>
                                              db 0
                                              db "FIND", 0
                              <1> Cmd Find:
 74 00010E38 46494E4400
                              <1> Cmd Echo: db "ECHO", 0
 75 00010E3D 4543484F00
                               <1> Cmd_Remark: db "*", 0
 76 00010E42 2A00
                              <1> Cmd_Help: db "?", 0
 77 00010E44 3F00
 78 00010E46 44455649434500
                              <1> Cmd Device: db "DEVICE", 0
                              <1> Cmd_DevList: db "DEVLIST", 0
 79 00010E4D 4445564C49535400
 80 00010E55 434844495200
                               <1> Cmd_Chdir: db "CHDIR", 0
                               <1> Cmd_Beep: db "BEEP", 0
 81 00010E5B 4245455000
 82
                               <1>
 83 00010E60 00
                                               db 0
                               <1>
 84
                               <1>
 85
                               <1>; 15/02/2016 (FILE.ASM, 09/10/2011)
                               <1> invalid fname chars:
 86
                                              db 22h, 27h, 28h, 29h, 2Ah, 2Bh, 2Ch, 2Fh
 87 00010E61 222728292A2B2C2F
                               <1>
 88 00010E69 3A3B3C3D3E3F40
                               <1>
                                              db 3Ah, 3Bh, 3Ch, 3Dh, 3Eh, 3Fh, 40h
 89 00010E70 5B5C5D5E60
                               <1>
                                              db 5Bh, 5Ch, 5Dh, 5Eh, 60h
 90
                               <1> sizeInvFnChars equ ($ - invalid_fname_chars)
 91
                               <1>;
 92
                               <1>
                               <1> Msg_Enter_Date:
 94 00010E75 456E746572206E6577- <1>
                                                  db 'Enter new date (dd-mm-yy): '
 94 00010E7E 206461746520286464- <1>
 94 00010E87 2D6D6D2D7979293A20 <1>
 95 00010E90 00
                               <1>
                                                  db 0
                               <1> Msg_Show_Date:
 97 00010E91 43757272656E742064- <1>
                                                  db 'Current date is '
 97 00010E9A 61746520697320
                              <1>
 98 00010EA1 30
                               <1> Day:
                                                  db '0'
                                               db '0'
 99 00010EA2 30
                               <1>
                                              db '/'
100 00010EA3 2F
                               <1>
                                                 db '0'
101 00010EA4 30
                               <1> Month:
                                               db '0'
102 00010EA5 30
                              <1>
                                              db '/'
103 00010EA6 2F
                               <1>
104 00010EA7 30
                               <1> Century:
                                               db '0'
105 00010EA8 30
                              <1>
                                                 db '0'
106 00010EA9 30
                               <1> Year:
                                               db '0'
107 00010EAA 30
                               <1>
                                                  db 0Dh, 0Ah, 0
108 00010EAB 0D0A00
                               <1>
109
                               <1>
110
                               <1> Msg_Enter_Time:
                                      db 'Enter new time: '
111 00010EAE 456E746572206E6577- <1>
111 00010EB7 2074696D653A20
                              <1>
112 00010EBE 00
                               <1>
                                              db 0
113
                               <1> Msg_Show_Time:
                                    db 'Current time is '
114 00010EBF 43757272656E742074- <1>
114 00010EC8 696D6520697320
                            <1>
                                                 db '0'
115 00010ECF 30
                               <1> Hour:
                                               db '0'
116 00010ED0 30
                               <1>
                                               db ':'
117 00010ED1 3A
                              <1>
118 00010ED2 30
                                               db '0'
                               <1> Minute:
119 00010ED3 30
                                               db '0'
                              <1>
                                               db ':'
120 00010ED4 3A
                              <1>
121 00010ED5 30
                               <1> Second:
                                               db '0'
                                                 '0'
122 00010ED6 30
                               <1>
                                               db
123 00010ED7 0D0A00
                               <1>
                                               db
                                                  0Dh, 0Ah, 0
124
                               <1>
125
                               <1> ; VolSize_Unit1:
                                                    dd 0
126
                               <1> ; VolSize_Unit2:
                                                   dd 0
127
                               <1>
                               <1> VolSize KiloBytes:
128
                                     db " kilobytes", ODh, OAh, O
129 00010EDA 206B696C6F62797465- <1>
129 00010EE3 730D0A00
                               <1>
<1> Volume_in_drive:
                                             db 0Dh, 0Ah
133 00010EF0 0D0A
                               <1>
134
                               <1> Vol_FS_Name:
                                       db "TR FS1 "
135 00010EF2 54522046533120
                               <1>
                                               db "Volume in drive "
136 00010EF9 566F6C756D6520696E- <1>
136 00010F02 20647269766520
                               <1>
                               <1> Vol_Drv_Name: db 30h
137 00010F09 30
138 00010F0A 3A
                                              db ":"
                               <1>
139 00010F0B 20697320
                                              db " is "
                               <1>
                                              db 0Dh, 0Ah, 0
140 00010F0F 0D0A00
                               <1>
                               <1> Dir Drive Str:
                                                  db "TR-DOS Drive "
142 00010F12 54522D444F53204472- <1>
142 00010F1B 69766520
                               <1>
                               <1> Dir_Drive_Name:
                                                  db "0:"
144 00010F1F 303A
                               <1>
145 00010F21 0D0A
                               <1>
                                                  db 0Dh, 0Ah
                               <1> Vol Str Header:
146
                                                  db "Volume Name: "
147 00010F23 566F6C756D65204E61- <1>
147 00010F2C 6D653A20
                               <1>
148
                               <1> Vol_Name:
149 00010F30 00<rept>
                                               times 64 db 0
                               <1>
150 00010F70 00
                               <1>
                                              db 0
151
                               <1> Vol_Serial_Header:
                               <1> db 0Dh, 0Ah
152 00010F71 0D0A
153 00010F73 566F6C756D65205365- <1>
                                               db "Volume Serial No: "
153 00010F7C 7269616C204E6F3A20 <1>
                               <1> Vol_Serial1:
                                         db "0000"
155 00010F85 30303030
                               <1>
156 00010F89 2D
                               <1>
                                               db "-"
```

```
157
                               <1> Vol Serial2:
                                    db "0000"
158 00010F8A 30303030
                               <1>
                              <1>
                                              db 0Dh, 0Ah, 0
159 00010F8E 0D0A00
160
                               <1>
161
                               <1> ;Vol_Tot_Sec_Str_Start:
162
                               <1>;
                                         dd 0
                               <1> Vol_Total_Sector_Header:
163
                                     db ODh, OAh
164 00010F91 0D0A
                              <1>
165 00010F93 566F6C756D65205369- <1>
                                              db "Volume Size : ", 0
165 00010F9C 7A65203A2000
                           <1>
166
                               <1> ; Vol_Tot_Sec_Str:
                                            db "0000000000"
167
                               <1>;
168
                               <1> ;Vol_Tot_Sec_Str_End:
169
                               <1>; db 0
                               170
171
                               <1> Vol_Free_Sectors_Header:
                                     db "Free Space : ", 0
173 00010FA2 467265652053706163- <1>
173 00010FAB 6520203A2000
                               <1>
174
                               <1> ; Vol_Free_Sectors_Str:
                               <1> ; db "000000000"
175
                               <1> ;Vol_Free_Sectors_Str_End:
176
177
                               <1>; db 0
178
                               <1>
179
                               <1> Dir_Str_Header:
180 00010FB1 4469726563746F7279- <1>
                                                  db "Directory: "
180 00010FBA 3A20
                              <1>
181 00010FBC 2F
                              <1> Dir_Str_Root: db "/"
                                                  times 64 db 0
182 00010FBD 00<rept>
                              <1> Dir_Str:
183 00010FFD 00000000
                              <1>
                                                  dd 0
184 00011001 00
                               <1>
                                                  db 0
185
                               <1>
                               <1> Msg_Bad_Command:
186
187 00011002 42616420636F6D6D61- <1>
                                                  db "Bad command or file name!"
187 0001100B 6E64206F722066696C- <1>
187 00011014 65206E616D6521
                              <1>
188 0001101B 0D0A00
                               <1>
                                                 db 0Dh, 0Ah, 0
189
                               <1>
190
                               <1> msgl_drv_not_ready:
191 0001101E 070D0A
                               <1> db 07h, 0Dh, 0Ah
192
                               <1>
                               <1>; CMD_INTR.ASM - 09/11/2011 - Messages
193
194
                               <1>
195
                               <1> Msg_Not_Ready_Read_Err:
196 00011021 4472697665206E6F74- <1>
                                                 db "Drive not ready or read error!"
196 0001102A 207265616479206F72- <1>
196 00011033 207265616420657272- <1>
196 0001103C 6F7221
                              <1>
197 0001103F 0D0A00
                              <1>
                                                  db ODh, OAh, O
                               <1>
199
                               <1> Msg_Not_Ready_Write_Err:
200 00011042 4472697665206E6F74- <1>
                                                 db "Drive not ready or write error!"
200 0001104B 207265616479206F72- <1>
200 00011054 207772697465206572- <1>
200 0001105D 726F7221
                              <1>
201 00011061 0D0A00
                              <1>
                                                  db 0Dh, 0Ah, 0
                              <1>
                              <1> Msg_Dir_Not_Found:
203
                                                 db "Directory not found!"
204 00011064 4469726563746F7279- <1>
204 0001106D 206E6F7420666F756E- <1>
204 00011076 6421
                              <1>
205 00011078 0D0A00
                               <1>
                                                  db 0Dh, 0Ah, 0
206
                               <1>
207
                              <1> Msg_File_Not_Found:
208 0001107B 46696C65206E6F7420- <1>
                                                  db "File not found!"
                          <1>
208 00011084 666F756E6421
                                                  db ODh, OAh, O
209 0001108A 0D0A00
                               <1>
210
                               <1>
211
                               <1> Msg_File_Directory_Not_Found:
                                                 db "File or directory not found!"
212 0001108D 46696C65206F722064- <1>
212 00011096 69726563746F727920- <1>
212 0001109F 6E6F7420666F756E64- <1>
212 000110A8 21
                               <1>
                                                  db 0Dh, 0Ah, 0
213 000110A9 0D0A00
                               <1>
214
                               <1>
                               <1> Msg_LongName_Not_Found:
215
216 000110AC 4C6F6E67206E616D65- <1>
                                                  db "Long name not found!"
216 000110B5 206E6F7420666F756E- <1>
216 000110BE 6421
                               <1>
217 000110C0 0D0A00
                                                  db ODh, OAh, O
                               <1>
218
                               <1>
                               <1> beep_Insufficient_Memory: ; 20/02/2017
219
220 000110C3 0D0A
                               <1>
                                               db 0Dh, 0Ah
221 000110C5 07
                               <1>
                                              db 07h
                               <1> Msg_Insufficient_Memory:
222
223 000110C6 496E73756666696369- <1>
                                                db "Insufficient memory!"
223 000110CF 656E74206D656D6F72- <1>
223 000110D8 7921
                              <1>
224 000110DA 0D0A00
                               <1>
                                                  db ODh, OAh, O
                               <1>
                               <1> Msg_Error_Code:
226
                                            db 'Command failed! Error code : '
227 000110DD 436F6D6D616E642066- <1>
227 000110E6 61696C656421204572- <1>
227 000110EF 726F7220636F646520- <1>
227 000110F8 3A20
                              <1>
228 000110FA 303068
                              <1> error_code_hex: db '00h'
                                      db 0Ah, 0Ah, 0
229 000110FD 0A0A00
                              <1>
230
                              <1>
231
                               <1> align 2
232
                               <1>
                               <1> ; 10/02/2016
233
                               <1> ; DIR.ASM - 09/10/2011
234
235
                               <1>
236 00011100 3C4449523E20202020- <1> Type_Dir: db '<DIR> ' ; 10 bytes
236 00011109 20
                              <1>
```

```
<1>
                                <1> File Name:
239 0001110A 20<rept>
                                <1>
                                                    times 12 db 20h
240 00011116 20
                                <1>
                                                db 20h
                                <1> Dir Or FileSize:
241
242 00011117 20<rept>
                                <1>
                                                    times 10 db 20h
243 00011121 20
                               <1>
                                                 db 20h
                                <1> File_Attribute:
244
245 00011122 20202020
                               <1>
                                                dd 20202020h
246 00011126 20
                                                db 20h
                               <1>
                               <1> File_Day:
247
248 00011127 3030
                                                    db '0','0'
                                <1>
249 00011129 2F
                                                 db '/'
                               <1>
250
                               <1> File_Month:
251 0001112A 3030
                               <1>
                                                   db '0','0'
                                                db '/'
252 0001112C 2F
                               <1>
                               <1> File Year:
                                                   db '0','0','0','0'
254 0001112D 30303030
                                <1>
                                                 db 20h
255 00011131 20
                                <1>
256
                               <1> File Hour:
                                                   db '0','0'
257 00011132 3030
                               <1>
                                <1>
                                                db ':'
258 00011134 3A
                               <1> File Minute:
259
260 00011135 3030
                               <1>
                                                   db '0','0'
261 00011137 00
                                <1>
                                                db 0
262
                                <1>
                                <1> Decimal_File_Count_Header:
264 00011138 0D0A
                                <1>
                                                db 0Dh, 0Ah
265
                                <1> Decimal_File_Count:
266 0001113A 00<rept>
                                <1> times 6 db 0
267
                                <1>
268 00011140 2066696C6528732920- <1> str_files: db " file(s) & "
268 00011149 2620
                                <1>
                                <1> Decimal_Dir_Count:
270 0001114B 00<rept>
                                                times 6 db 0
                                <1>
271
                                <1> str_dirs:
272 00011151 206469726563746F72- <1>
                                                db " directory(s) "
272 0001115A 7928732920
                               <1>
273 0001115F 0D0A00
                                <1>
                                                db ODh, OAh, O
                                <1>
275 00011162 206279746528732920- <1> str_bytes: db "byte(s) in file(s)"
275 0001116B 696E2066696C652873- <1>
275 00011174 29
                                <1>
276 00011175 0D0A00
                                <1>
                                                db 0Dh, 0Ah, 0
277
                                <1>
278
                                <1>; CMD_INTR.ASM - 09/11/2011
                                <1>; 07/\overline{10}/2010
279
                                <1> Msg_invalid_name_chars:
280
281 00011178 496E76616C69642066- <1>
                                                    db "Invalid file or directory name characters!"
281 00011181 696C65206F72206469- <1>
281 0001118A 726563746F7279206E- <1>
281 00011193 616D65206368617261- <1>
281 0001119C 637465727321
                               <1>
282 000111A2 0D0A00
                                <1>
                                                db 0Dh, 0Ah, 0
                                <1> ; 21/02/2016
284 000111A5 46696C65206F722064- <1> Msg_Name_Exists: db "File or directory name exists!"
284 000111AE 69726563746F727920- <1>
284 000111B7 6E616D652065786973- <1>
284 000111C0 747321
                                <1>
285 000111C3 0D0A00
                                                    db 0Dh, 0Ah, 0
                                <1>
286
                                <1> Msg_DoYouWantMkdir:
287 000111C6 446F20796F75207761- <1>
                                                    db "Do you want to make directory ", 0
287 000111CF 6E7420746F206D616B- <1>
287 000111D8 65206469726563746F- <1>
287 000111E1 72792000
                               <1>
288 000111E5 2028592F4E29203F20- <1> Msg_YesNo:
                                                    db " (Y/N) ? ", 0
288 000111EE 00
                       <1>
                                                       db 0, 0Dh, 0Ah, 0
289 000111EF 000D0A00
                                <1> Y_N_nextline:
290 000111F3 4F4B2E0D0A00
                                <1> Msg_OK:
                                                       db "OK.", ODh, OAh, O
291
                                <1>
                                <1> ; 27/02/2016
292
293
                                <1> Msg_DoYouWantRmDir:
                                                   db "Do you want to delete directory ", 0
294 000111F9 446F20796F75207761- <1>
294 00011202 6E7420746F2064656C- <1>
294 0001120B 657465206469726563- <1>
294 00011214 746F72792000
                                <1>
                                <1> Msg_Dir_Not_Empty:
296 0001121A 4469726563746F7279- <1>
                                                    db "Directory not empty!"
296 00011223 206E6F7420656D7074- <1>
296 0001122C 7921
                                <1>
297 0001122E 0D0A00
                                <1>
                                                    db 0Dh, 0Ah, 0
298
                                <1>
                                <1> Msg_DoYouWantDelete:
300 00011231 446F20796F75207761- <1>
                                                    db "Do you want to delete file ",0
300 0001123A 6E7420746F2064656C- <1>
300 00011243 6574652066696C6520- <1>
300 0001124C 00
301
                                <1>
302 0001124D 44656C657465642E2E- <1> Msg_Deleted:
                                                    db "Deleted...", ODh, OAh, O
302 00011256 2E0D0A00
                               <1>
303
                                <1>
                                <1> Msg Permission Denied:
304
305 0001125A 07
                                <1>
                                                   db 7
306 0001125B 5065726D697373696F- <1>
                                                    db "Permission denied!", ODh, OAh, O
306 00011264 6E2064656E69656421- <1>
306 0001126D 0D0A00
                                <1>
                                <1>
                                <1> ; 04/03/2016
308
                                <1> Msg New:
                                                    db "New "
309 00011270 4E657720
310 00011274 00
                                                    db 0
                                <1>
                                <1> Str Attributes:
311
312 00011275 417474726962757465- <1>
                                                    db "Attributes : "
312 0001127E 73203A20
                               <1>
                                                    db "NORMAL"
313 00011282 4E4F524D414C
                                <1> Attr_Chars:
314 00011288 00
                                <1>
                                                    db 0
```

237

```
315
                                 <1>
316
                                 <1>; 06/03/2016
317
                                 <1>; CMD_INTR.ASM - 16/11/2010
                                <1> Msg_DoYouWantRename:
319 00011289 446F20796F75207761- <1>
                                                    db "Do you want to rename ", 0
319 00011292 6E7420746F2072656E- <1>
319 0001129B 616D652000
                                <1> Rename_File: db "file ", 0
320 000112A0 66696C652000
321 000112A6 6469726563746F7279- <1> Rename_Directory: db "directory ", 0
321 000112AF 2000
                                <1>
322 000112B1 00<rept>
                                <1> Rename_OldName: times 13 db 0
                                <1> Msg File rename as: db " as "
323 000112BE 20617320
324 000112C2 00<rept>
                                <1> Rename_NewName: times 13 db 0
325
                                <1>
326
                                <1>; 08/03/2016
                                 <1>; CMD INTR.ASM - 01/08/2010 - 23/04/2011
327
                                 <1> msg_not_same_drv:
                                                    db "Not same drive!"
329 000112CF 4E6F742073616D6520- <1>
329 000112D8 647269766521
                                <1>
330 000112DE 0D0A00
                                                    db 0Dh, 0Ah, 0
                                <1>
331
                                <1>
                                <1> Msg_DoYouWantMoveFile:
333 000112E1 446F20796F75207761- <1>
                                                    db "Do you want to move file", 0
333 000112EA 6E7420746F206D6F76- <1>
333 000112F3 652066696C6500
                                <1>
334
                                 <1>
                                <1> msg_insufficient_disk_space:
336 000112FA 496E73756666696369- <1>
                                                     db "Insufficient disk space!"
336 00011303 656E74206469736B20- <1>
336 0001130C 737061636521
                                <1>
337 00011312 0D0A00
                                <1>
                                                     db 0Dh, 0Ah, 0
338
                                 <1>
339
                                <1> ; 01/08/2010
                                <1> msg_source_file:
                                                 db ODh, OAh, "Source file name
341 00011315 0D0A536F7572636520- <1>
341 0001131E 66696C65206E616D65- <1>
341 00011327 2020202020203A2020- <1>
341 00011330 20
                                <1>
                                 <1> msg_source_file_drv:
342
343 00011331 203A00
                                <1> db ":", 0
344
                                <1> msg_destination_file:
                                                 db ODh, OAh, "Destination file name: "
345 00011334 0D0A44657374696E61- <1>
345 0001133D 74696F6E2066696C65- <1>
345 00011346 206E616D65203A2020- <1>
345 0001134F 20
                                 <1>
346
                                 <1> msg_destination_file_drv:
                                <1> db ":", 0
347 00011350 203A00
                                <1> msg_copy_nextline:
348
                                                db ODh, OAh, O
349 00011353 0D0A00
                                <1>
350
                                 <1>
351
                                 <1> ; 15/03/2016
352
                                 <1> ; CMD_INTR.ASM
353
                                 <1>
354
                                 <1> Msg_DoYouWantOverWriteFile:
355 00011356 446F20796F75207761- <1>
                                                    db "Do you want to overwrite file ",0
355 0001135F 6E7420746F206F7665- <1>
355 00011368 727772697465206669- <1>
355 00011371 6C652000
                                <1>
356
                                <1>
                                <1> Msg_DoYouWantCopyFile:
358 00011375 446F20796F75207761- <1>
                                                    db "Do you want to copy file",0
358 0001137E 6E7420746F20636F70- <1>
358 00011387 792066696C6500
                                <1>
359
                                 <1>
360
                                <1> Msg_read_file_error_before_EOF:
361 0001138E 46696C652072656164- <1>
                                                db "File reading error! (before EOF)"
361 00011397 696E67206572726F72- <1>
361 000113A0 2120286265666F7265- <1>
361 000113A9 20454F4629
                                <1>
362 000113AE 0A0A00
                                <1>
                                                 db 0Ah, 0Ah, 0
363
                                 <1>
364
                                 <1> ; 18/03/2016
                                 <1> ; TRDOS 386 (v2.0) mainprog copy procedure
365
                                 <1> msg_reading:
                                                 db "Reading... ", 0
367 000113B1 52656164696E672E2E- <1>
367 000113BA 2E2000
                                <1>
                                 <1> msg_writing:
                                                 db "Writing...", 0
369 000113BD 57726974696E672E2E- <1>
369 000113C6 2E2000
                                 <1>
                                 <1> percentagestr:
                                                 db " %", 0 ; " 0%" .. "100%"
371 000113C9 2020202500
                                <1>
                                 <1> ; 11/04/2016
                                <1> Msg_No_Set_Space:
374 000113CE 496E73756666696369- <1>
                                                    db "Insufficient environment space!"
374 000113D7 656E7420656E766972- <1>
374 000113E0 6F6E6D656E74207370- <1>
374 000113E9 61636521
375 000113ED 0D0A00
                                                    db 0Dh, 0Ah, 0
                                <1>
                                <1> ; 18/04/2016
376
                                <1> isc_msg:
377
378 000113F0 0D0A
                                                 db 0Dh, 0Ah
                                <1>
379 000113F2 494E56414C49442053- <1>
                                                 db "INVALID SYSTEM CALL", 0
379 000113FB 595354454D2043414C- <1>
379 00011404 4C00
                                <1>
                                <1> usi_msg:
380
381 00011406 0D0A
                                                 db 0Dh, 0Ah
                                <1>
382 00011408 554E444546494E4544- <1>
                                                 db "UNDEFINED SOFTWARE INTERRUPT", 0
382 00011411 20534F465457415245- <1>
382 0001141A 20494E544552525550- <1>
382 00011423 5400
                                <1>
                                <1> ifc msg:
384 00011425 0D0A
                                <1>
                                                 db 0Dh, 0Ah
                                                 db "INVALID FUNCTION CALL"
385 00011427 494E56414C49442046- <1>
385 00011430 554E4354494F4E2043- <1>
385 00011439 414C4C
```

```
<1> inv_msg_for_trdos_v2:
386
387 0001143C 20
                                 <1> db 20h <1> db "for
388 0001143D 666F72205452444F53- <1>
                                                 db "for TRDOS v2!"
388 00011446 20763221 <1>
389 0001144A 07
                                 <1>
                                                 db 07h
                                                 db 0Dh, 0Ah
390 0001144B 0D0A
                                <1>
391 0001144D 0D0A
                               <1>
                                                 db 0Dh, 0Ah
                            <1>
<1>
<1>
                                       db "INT "
392 0001144F 494E5420
393 00011453 303068
                                <1> int_num_str: db "00h"
                                <1> db 0Dh, 0Ah
<1> db "EAX: "
394 00011456 0D0A
395 00011458 454158203A20 <1>
                                                  db "EAX : "
396 0001145E 303030303030303068- <1> eax str:
                                                 db "00000000h", 0Dh, 0Ah
396 00011467 0D0A
                          <1>
<1>
                                                  db "EIP : "
397 00011469 454950203A20
398 0001146F 303030303030303068- <1> eip_str:
                                                 db "00000000h", 0Dh, 0Ah, 0
398 00011478 0D0A00
                                <1>
                                 <1>; 07/10/2016
400
                                 <1> ; Device names & parameters (for kernel devices)
401
402
                                 <1>
403 0001147B 90
                                 <1> align 2
                                 <1> KDEV NAME:
405 0001147C 5454590000000000
                                                  db 'TTY',0,0,0,0,0; 1
                                 <1>
406 00011484 4D454D0000000000
                                 <1>
                                                 db 'MEM',0,0,0,0,0; 2
407 0001148C 4644300000000000
                                                 db 'FD0',0,0,0,0,0; 3
                                 <1>
                                                 db 'FD1',0,0,0,0,0; 4
408 00011494 4644310000000000
                                 <1>
                                               db 'HDO',0,0,0,0,0; 5
db 'HD1',0,0,0,0,0; 6
409 0001149C 484430000000000
                                 <1>
410 000114A4 4844310000000000
                                 <1>
                                                db 'HD2',0,0,0,0,0; 7 db 'HD3',0,0,0,0; 8
411 000114AC 4844320000000000
                                 <1>
412 000114B4 4844330000000000
                                 <1>
413 000114BC 4C50540000000000
                                 <1>
                                               db 'LPT',0,0,0,0,0; 9
                                                db 'TTY0',0,0,0,0; 10
db 'TTY1',0,0,0,0; 11
414 000114C4 5454593000000000
                                 <1>
415 000114CC 5454593100000000
                                 <1>
                                               db 'TTY2',0,0,0,0; 12
416 000114D4 5454593200000000
                                 <1>
                                                db 'TTY3',0,0,0,0; 13
db 'TTY4',0,0,0,0; 14
417 000114DC 5454593300000000
                                 <1>
418 000114E4 5454593400000000
                                 <1>
419 000114EC 5454593500000000
                                               db 'TTY5',0,0,0,0; 15
                                               db 'TTY6',0,0,0,0; 16 db 'TTY7',0,0,0,0; 17
420 000114F4 5454593600000000
                                 <1>
421 000114FC 5454593700000000
                                 <1>
422 00011504 5454593800000000
                                                db 'TTY8',0,0,0,0; 18
                                 <1>
423 0001150C 5454593900000000
                                               db 'TTY9',0,0,0,0 ; 19
                                 <1>
                                                db 'COM1',0,0,0,0; 18
db 'COM2',0,0,0,0; 19
424 00011514 434F4D3100000000
                                 <1>
425 0001151C 434F4D3200000000
                                 <1>
                                                ;db 'CONSOLE',0
426
                                 <1>
                                                ;db 'PRINTER',0 ; 9
427
                                 <1>
                                                 ;db 'CDROM' ; 20
428
                                 <1>
                                                 ;db 'CDROM0' ; 20
429
                                 <1>
                                                ;db 'CDROM1' ; 21
430
                                 <1>
431
                                 <1>
                                                 ;db 'DVD'
                                                 ;db 'DVD0'
432
                                 <1>
                                                 ;db 'DVD1'
                                                              ; 23
433
                                 <1>
                                                 ;db 'USB'
434
                                 <1>
                                                 ;db 'USB0'
                                                                ; 24
435
                                 <1>
                                                 ;db 'USB1'
                                                              ; 25
436
                                 <1>
                                                 ;db 'USB3'
437
                                 <1>
                                                 , db 'KEYBOARD'; 27
438
                                 <1>
439
                                 <1>
                                                 ;db 'MOUSE' ; 28
;db 'SOUND' ; 29
440
                                 <1>
441
                                 <1>
                                                 ;db 'VGA',0,0,0,0; 30
442
                                 <1>
                                                 ;db 'CGA',0,0,0,0 ; 31
443
                                 <1>
                                                 ;db 'AUDIO',0,0,0 ; 29
444
                                 <1>
                                                 ;db 'VIDEO',0,0,0 ; 32
                                 <1>
445
446
                                 <1>
                                                ;db 'MUSIC',0,0,0 ; 33
                                                 ;db 'ETHERNET' ; 34
447
                                 <1>
                                                 ;db 'SD0',0,0,0,0,0; 35
448
                                 <1>
                                                 ;db 'SD1',0,0,0,0,0; 36
449
                                 <1>
                                                 ;db 'SD2',0,0,0,0,0; 37
450
                                 <1>
451
                                 <1>
                                                 ;db 'SD3',0,0,0,0,0; 38
                                                 ;db 'SATA0' ; 35
452
                                 <1>
                                                ;db 'SATA1' ; 36
;db 'SATA2' ; 37
;db 'SATA3' ; 38
453
                                 <1>
454
                                 <1>
                                 <1>
455
                                                 ;db 'PATA0',0,0,0 ; 5
456
                                 <1>
                                                 ;db 'PATA1',0,0,0 ; 6 ;db 'PATA2',0,0,0 ; 7
457
                                 <1>
458
                                 <1>
                                                 ;db 'PATA3',0,0,0 ; 8
459
                                                 ;db 'WIRELESS'
460
                                 <1>
                                  <1>
                                                  ;db 'HDMI',0,0,0,0; 40
461
462 00011524 4E554C4C00000000
                                                  db 'NULL',0,0,0,0; 0
                                 <1>
463
                                 <1>
                                  <1> NumOfKernelDevNames equ ($-KDEV_NAME) / 8 ; 20 (07/10/2016)
465
                                 <1>
                                 <1> KDEV NUMBER:
467 0001152C 010203040506070809 <1>
                                                 db 1,2,3,4,5,6,7,8,9
468 00011535 0A0B0C0D0E0F101112- <1>
                                                  db 10,11,12,13,14,15,16,17,18,19
468 0001153E 13
                                 <1>
469 0001153F 121300
                                <1>
                                                  db 18,19,0
470
                                 <1>
471
                                 <1> NumOfKernelDevices equ $ - KDEV NUMBER
472
                                 <1>
473
                                 <1> KDEV OADDR:
474 00011542 [600D0100]
                                                  dd otty ;tty ; 1
                                <1>
475 00011546 [600D0100]
                                <1>
                                                  dd sret ; mem ; 2
476 0001154A [600D0100]
                        <1>
                                                 dd sret ;fd0 ; 3
477 0001154E [600D0100]
                                                 dd sret ;fd1 ; 4
478 00011552 [600D0100]
                                                dd sret ;hd0 ; 5
                                               dd sret ;hd1 ; 6
dd sret ;hd2 ; 7
dd sret ;hd3 ; 8
479 00011556 [600D0100]
480 0001155A [600D0100]
481 0001155E [600D0100]
                                               dd sret ;lpt ; 9
482 00011562 [600D0100]
                                             dd ocvt ;tty0 ; 10
dd ocvt ;tty1 ; 11
483 00011566 [600D0100]
484 0001156A [600D0100]
                                <1>
                                               dd ocvt ;tty2 ; 12
485 0001156E [600D0100]
486 00011572 [600D0100]
                                <1>
                                                 dd ocvt ;tty3 ; 13
```

```
dd ocvt ;
dd ocvt ;tty5 ; 15
dd ocvt ;tty6 ; 16
                                                 <1>
                                                                          dd ocvt ;tty8 ; 18
                                                                           dd ocvt ;tty9 ; 19
                                                                           ;dd ocvt ;com1 ; 18
                                                  <1>
   494
                                                                           ;dd ocvt ;com2 ; 19
   495 0001158E [600D0100]
                                                  <1>
                                                                           dd sret ;null ; 20
  496

497 00011592 [600D0100]

498 00011596 [600D0100]

499 0001159A [600D0100]

500 0001159E [600D0100]

510 dd cret ;fdd ; 3

501 000115A2 [600D0100]

510 dd cret ;fdd ; 4

501 000115A2 [600D0100]

510 dd cret ;hdd ; 5

502 000115A6 [600D0100]

510 dd cret ;hdd ; 6

503 000115AA [600D0100]

510 dd cret ;hdd ; 6

503 000115AA [600D0100]

510 dd cret ;hdd ; 7

504 000115B2 [600D0100]

510 dd cret ;hdd ; 8

505 000115B2 [600D0100]

510 dd cret ;hdd ; 8

505 000115B2 [600D0100]

510 dd cret ;hdd ; 9

510 000115B4 [600D0100]

510 dd cret ;hdd ; 9

511 000115C2 [600D0100]

510 dd cret ;hdd ; 9

512 000115C4 [600D0100]

510 dd cret ;hdd ; 9

513 000115D4 [600D0100]

510 dd cret ;tty4 ; 14

511 000115C4 [600D0100]

510 dd cret ;tty4 ; 14

512 000115D4 [600D0100]

510 dd cret ;tty5 ; 15

512 000115D4 [600D0100]

510 dd cret ;tty6 ; 16

513 000115D4 [600D0100]

510 dd cret ;tty7 ; 17

514 000115D4 [600D0100]

510 dd cret ;tty9 ; 19

516

517

518 000115D5 [600D0100]

510 dd cret ;tty9 ; 19

511 ;dd cret ;com2 ; 19

512 dd cret ;nu11 ; 20
                                                  <1> KDEV CADDR:
   496
                                                                          ;dd ccvt ;com1 ; 18
                                                                           ;dd ccvt ;com2 ; 19
   518 000115DE [600D0100]
                                                   <1>
                                                                           dd cret ;null ; 20
  <1>
   519
   541
                                                   <1>
                                                                           ;dd rcvt ;com2 ; 19
542 0001162E [44010100]
                                                  <1>
                                                                           dd rnull ; null ; 20
   565 0001167E [45010100]
                                                   <1>
                                                                            dd wnull ; null ; 20
   566
                                                    <1>
                                                    <1> ; DEV_ACCESS bits:
   567
                                                            ; bit 0 = accessable by normal users ; bit 1 = read access permission
                                                    <1>
   569
                                                    <1>
   570
                                                                  ; bit 2 = write access permission
                                                    <1>
                                                                  ; bit 3 = IOCTL permission to users
   571
                                                    <1>
   572
                                                    <1>
                                                                  ; bit 4 = block device if it is set
   573
                                                    <1>
                                                                  ; bit 5 = 16 bit or 1024 byte data
                                                                 ; bit 6 = 32 bit or 2048 byte data
   574
                                                    <1>
                                                                  ; bit 7 = installable device driver
   575
                                                    <1>
   576
                                                    <1>
                                                    <1> KDEV ACCESS: ; 08/10/2016
   577
   578 00011682 07
                                                                           db 00000111b; tty, 1
                                                    <1>
                                                                            db 00000111b; mem, 2
   579 00011683 07
                                                    <1>
   580 00011684 8F
                                                    <1>
                                                                            db 10001111b; fd0, 3
   581 00011685 8F
                                                    <1>
                                                                           db 10001111b; fd1, 4
                                                                           db 10001111b; hd0, 5
   582 00011686 8F
                                                   <1>
   583 00011687 8F
                                                    <1>
                                                                           db 10001111b; hd1, 6
   584 00011688 8F
                                                                         db 10001111b; hd2, 7
                                                   <1>
   585 00011689 8F
                                                   <1>
                                                                           db 10001111b; hd3, 8
   586 0001168A 07
                                                                          db 00000111b ; lpt, 9
                                                   <1>
   587 0001168B 07
                                                                           db 00000111b; tty0, 10
                                                  <1>
   588 0001168C 07
                                                   <1>
                                                                           db 00000111b; tty1, 11
   589 0001168D 07
                                                                           db 00000111b; tty2, 12
                                                  <1>
   590 0001168E 07
                                                   <1>
                                                                           db 00000111b; tty3, 13
   591 0001168F 07
                                                    <1>
                                                                            db 00000111b; tty4, 14
```

```
db 00000111b; tty5, 15
 592 00011690 07
                                 <1>
 593 00011691 07
                                 <1>
                                                 db 00000111b; tty6, 16
                                                db 00000111b; tty7, 17
594 00011692 07
                                 <1>
 595 00011693 07
                                <1>
                                                db 00000111b; tty8, 18
 596 00011694 07
                                 <1>
                                                db 00000111b; tty9, 19
597
                                 <1>
                                                ;db 00000111b; com1, 18
                                                ;db 00000111b; com2, 19
                                 <1>
                                                 db 00000000b ; null, 0
 599 00011695 00
                                 <1>
 600
                                 <1>
 601
                                 <1> ; 07/10/2016
 602
                                 <1> NumOfInstallableDevices equ 8
 603
                                 <1> NUMIDEV
                                              equ NumOfInstallableDevices ; 8
                                 <1> NUMOFDEVICES equ NumOfKernelDevices + NumOfInstallableDevices
 604
 605
                                 <1>
                                 <1>; 26/02/2017
 606
 607
                                 <1>; IRQ Callback (& Signal Response Byte) service availibity
 608
                                 <1> ; 'syscalbac'
                                 609
 610
                                 <1>; IRQ 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
                                 <1> ; --- -- -- -- -- -- -- -- -- -- --
 611
                                 <1> ; --- 00 00 00 01 02 03 00 04 00 05 06 07 08 09 00 00
 612
                                 613
                                 <1> IRQenum:
 614
 615 00011696 000000010203000400- <1>
                                         db 0,0,0,1,2,3,0,4,0,5,6,7,8,9,0,0
 615 0001169F 05060708090000
                                 <1>
 616
                                 <1>
 617
                                 <1>; 28/08/2017
                                 <1>; 20/08/2017
 618
 619
                                 <1> ; DMA Registers (for 'sysdma')
                                 <1>; 02/07/2017 (sb16mod.s)
 620
                                                db 0,2,4,6,0C0h,0C4h,0C8h,0CCh
 621 000116A6 00020406C0C4C8CC
                                 <1> dma_adr:
 622 000116AE 01030507C2C6CACE
                                 <1> dma cnt:
                                                 db 1,3,5,7,0C2h,0C6h,0CAh,0CEh
 623 000116B6 878381828F8B898A
                                                db 87h,83h,81h,82h,8Fh,8Bh,89h,8Ah; 03/08/2017
                                 <1> dma_page:
 624 000116BE 0A0A0A0AD4D4D4D4
                                 <1> dma_mask:
                                                 db 0Ah, 0Ah, 0Ah, 0Ah, 0D4h, 0D4h, 0D4h
 625 000116C6 0B0B0B0BD6D6D6D6
                                 <1> dma mod:
                                                 db 0Bh, 0Bh, 0Bh, 0Bh, 0D6h, 0D6h, 0D6h
626 000116CE 0C0C0C0CD8D8D8D8
                                 <1> dma_flip:
                                                db 0Ch, 0Ch, 0Ch, 0Ch, 0D8h, 0D8h, 0D8h, 0D8h
2314
2315
                                     ; 27/08/2014
2316
                                     scr_row:
2317 000116D6 E0810B00
                                         dd 0B8000h + 0A0h + 0A0h + 0A0h ; Row 3
2318
                                     scr_col:
2319 000116DA 00000000
                                          dd 0
2320
2321 000116DE 90<rept>
                                     Align 4
                                          ; 15/04/2016
2322
2323
                                           ; TRDOS 386 (TRDOS v2.0)
2324
                                          ; 21/08/2014
2325
2326
                                     ilist:
2327
                                          :times
                                                       32 dd cpu_except ; INT 0 to INT 1Fh
2328
2329
                                          ; Exception list
                                          ; 25/08/2014
2330
2331 000116E0 [20090000]
                                           dd
                                                 exc0
                                                      ; Oh, Divide-by-zero Error
2332 000116E4 [27090000]
                                                 exc1
                                          dd
2333 000116E8 [2E090000]
                                          dd
                                                 exc2
2334 000116EC [35090000]
                                          dd
                                                 exc3
2335 000116F0 [39090000]
                                          dd
                                                 exc4
2336 000116F4 [3D090000]
                                          dd
                                                 exc5
2337 000116F8 [41090000]
                                                      ; 06h, Invalid Opcode
                                          dd
                                                 exc6
2338 000116FC [45090000]
                                          dd
                                                 exc7
2339 00011700 [49090000]
                                          dd
                                                 exc8
2340 00011704 [4D090000]
                                          dd
                                                 exc9
2341 00011708 [51090000]
                                          dd
                                                 exc10
2342 0001170C [55090000]
                                          dd
                                                 exc11
2343 00011710 [59090000]
                                          dd
                                                 exc12
2344 00011714 [5D090000]
                                          dd
                                                 exc13 ; ODh, General Protection Fault
2345 00011718 [61090000]
                                          dd
                                                 exc14 ; OEh, Page Fault
2346 0001171C [65090000]
                                          dd
                                                 exc15
2347 00011720 [69090000]
                                          dd
                                                 exc16
2348 00011724 [6D090000]
                                          dd
                                                 exc17
2349 00011728 [71090000]
                                          dd
                                                 exc18
2350 0001172C [75090000]
                                          dd
                                                 exc19
2351 00011730 [79090000]
                                          dd
                                                 exc20
2352 00011734 [7D090000]
                                          dd
                                                 exc21
2353 00011738 [81090000]
                                          dd
                                                 exc22
2354 0001173C [85090000]
                                          dd
                                                 exc23
2355 00011740 [89090000]
                                          dd
                                                 exc24
2356 00011744 [8D090000]
                                          dd
                                                 exc25
2357 00011748 [91090000]
                                          dd
                                                 exc26
2358 0001174C [95090000]
                                           dd
                                                 exc27
2359 00011750 [99090000]
                                           dd
                                                 exc28
2360 00011754 [9D090000]
                                           dd
                                                 exc29
2361 00011758 [A1090000]
                                           dd
                                                 exc30
2362 0001175C [A5090000]
                                          dd
                                                 exc31
                                     IRQ_list: ; 28/02/2017 ('syscalbac')
2363
                                          ; Interrupt list
2364
2365 00011760 [94060000]
                                           dd
                                                 timer_int
                                                              ; INT 20h
2366
                                                 ;dd
                                                       irq0
                                                              ; 24/01/2016
2367 00011764 [080E0000]
                                           dd
                                                 kb int
2368
                                                 ; dd
                                                       irq1
2369 00011768 [76080000]
                                           dd
                                                 irq2
                                                 ; COM2 int
2370
2371 0001176C [7A080000]
                                           dd
                                                 irq3
2372
                                                 ; COM1 int
2373 00011770 [85080000]
                                           dd
                                                 irq4
                                                 irq5
2374 00011774 [90080000]
                                           dd
                                     ;DISKETTE INT: ;06/02/2015
2375
2376 00011778 [D7410000]
                                           dd
                                                fdc_int
                                                                    ; 16/02/2015, IRQ 6 handler
2377
                                                 ;dd irq6
2378
                                     ; Default IRQ 7 handler against spurious IRQs (from master PIC)
2379
                                     ; 25/02/2015 (source: http://wiki.osdev.org/8259 PIC)
2380 0001177C [FF0B0000]
                                                 default_irq7 ; 25/02/2015
2381
                                                 ;dd irq7
2382
                                     ; Real Time Clock Interrupt
```

```
2383 00011780 [FF070000]
                                                       rtc_int ; 23/02/2015, IRQ 8 handler
                                                dd
                                                       ;dd irq8 ; INT 28h
2384
2385 00011784 [A0080000]
                                                dd
                                                       ira9
2386 00011788 [A4080000]
                                                dd
                                                      irq10
2387 0001178C [A8080000]
                                                dd
                                                       irq11
2388 00011790 [AC080000]
                                                dd
                                                       irq12
2389 00011794 [B0080000]
                                                dd irq13
                                         ;HDISK_INT1: ;06/02/2015
2390
                                         dd hdc1_int ; 21/02/2015, IRQ 14 handler
;dd irq14
2391 00011798 [8A4B0000]
2392
                                          ;HDISK_INT2: ;06/02/2015
2393
                                               dd hdc2_int ; 21/02/2015, IRQ 15 handler ; dd irq15 ; INT 2Fh
2394 0001179C [B14B0000]
2395
                                                       ; 14/08/2015
2396
                                                ;dd sysent ; INT 30h (system calls)
2397
2398
                                                ; 15/04/2016
2399
                                                ; TRDOS 386(TRDOS v2.0) Software Interrupts
2400
2401
2402 000117A0 [FD170100]
                                                dd
                                                       int30h
                                                                     ; Reserved for
                                               ; !!! Retro UNIX (RUNIX) !!!
; !!! SINGLIX !!! System Calls

dd int31h ; Video BIOS (IBM PC/AT, Int 10h)

dd int32h ; Keyboard Functions (IBM PC/AT, Int 16h)

dd int33h ; DISK I/O (IBM PC/AT, Int 13h)

dd int34h ; #IOCTL# (I/O port access support for ring 3)

dd int35h ; Time/Date Functions (IBM PC/AT, Int 1Ah)

dd ignore_int ; INT 36h : Timer Functions

dd ignore_int ; INT 37h

dd ignore_int ; INT 38h

dd ignore_int : INT 38h

dd ignore_int : INT 38h
                                                                     ; !!! Retro UNIX (RUNIX) !!!
2403
2405 000117A4 [FB140000]
2406 000117A8 [270C0000]
2407 000117AC [8E420000]
2408 000117B0 [F7F90000]
2409 000117B4 [DF590000]
2410 000117B8 [B30A0000]
2411 000117BC [B30A0000]
2412 000117C0 [B30A0000]
                                                      ignore_int ; INT 39h
2413 000117C4 [B30A0000]
                                                dd
                                                      ignore_int ; INT 3Ah
ignore_int ; INT 3Bh
2414 000117C8 [B30A0000]
                                                dd
2415 000117CC [B30A0000]
                                                dd
2416 000117D0 [B30A0000]
                                                dd
                                                       ignore_int ; INT 3Ch
                                                       ignore_int ; INT 3Dh
ignore_int ; INT 3Eh
2417 000117D4 [B30A0000]
                                                dd
2418 000117D8 [B30A0000]
                                                dd
                                                       ignore_int ; INT 3Fh
2419 000117DC [B30A0000]
                                                dd
2420 000117E0 [10C60000]
                                                       sysent ; INT 40h : !!! TRDOS 386 System Calls !!!
                                                dd
2421
                                                ;dd
                                                      ignore_int
2422 000117E4 00000000
                                                dd
2423
2424
                                         ; 20/08/2014
                                           ; /* This is the default interrupt "handler" :-) */
2425
2426
                                            ; Linux v0.12 (head.s)
                                          int_msg:
2427
                                           db "Unknown interrupt ! ", 0
2428 000117E8 556E6B6E6F776E2069-
2428 000117F1 6E7465727275707420-
2428 000117FA 212000
2429
2430
                                          ; 15/04/2016
2431
                                          ; TRDOS 386 (TRDOS v2.0)
2432
                                          ; 29/04/2016
2433
2434
                                          int30h:
2435
                                          trdos_isc_routine:
2436
                                               ; 02/05/2016
                                                ; 01/05/2016
2437
                                               ; 29/04/2016
2438
2439
                                                ; 18/04/2016
2440
                                                ; 15/04/2016 (TRDOS 386 = TRDOS v2.0)
                                                ; 17/04/2011 (TRDOS v1.0, 'IFC.ASM')
2441
                                                ; 03/02/2011 ('trdos ifc routine')
2442
2443
2444 000117FD 8B1C24
                                                mov
                                                      ebx, [esp] ; EIP (next)
2445 00011800 83EB02
                                                      ebx, 2 ; EIP (CD ##h)
                                                sub
2446
2447 00011803 89C1
                                                mov
                                                       ecx, eax
2448 00011805 8A4301
                                                       al, [ebx+1] ; CDh ##h
                                                mov
2449
2450 00011808 66BA1000
                                                       dx, KDATA
                                                mov
2451 0001180C 8EDA
                                                mov
                                                       ds, dx
2452 0001180E 8EC2
                                                mov
                                                       es, dx
2453
2454 00011810 FC
                                                cld
2455 00011811 8B15[C0580100]
                                                       edx, [k_page_dir]
                                               mov
2456 00011817 0F22DA
                                                       cr3, edx
                                               mov
2458 0001181A E8B31AFFFF
                                               call bytetohex
2459 0001181F 66A3[53140100]
                                                      [int_num_str], ax
                                               mov
2461 00011825 89D8
                                                       eax, ebx ; EIP
                                                mov
2462 00011827 E8E61AFFFF
                                                call
                                                       dwordtohex
2463 0001182C 8915[6F140100]
                                                        [eip_str], edx
                                                mov
2464 00011832 A3[73140100]
                                                mov
                                                        [eip_str+4], eax
2465
2466 00011837 89C8
                                                        eax, ecx
                                                mov
2467 00011839 E8D41AFFFF
                                               call dwordtohex
2468 0001183E 8915[5E140100]
                                                       [eax str], edx
                                               mov
2469 00011844 A3[62140100]
                                                       [eax_str+4], eax
                                               mov
2471 00011849 43
                                               inc
                                                       ebx
2472 0001184A 8A03
                                                       al, [ebx]; Interrupt number
                                               mov
2473
                                          trdos isc handler:
2474
2475 0001184C 80FE30
                                                cmp
                                                      dh, 30h; Retro UNIX, SINGLIX System calls
2476 0001184F 7507
                                                jne
                                                       short trdos_usi_handler
2477 00011851 BE[F0130100]
                                                       esi, isc msq
                                                mov
2478 00011856 EB05
                                                jmp short loc write inv system call msg
2479
                                          trdos_usi_handler:
2480
2481 00011858 BE[06140100]
                                                mov esi, usi_msg
2482
                                          loc_write_inv_system_call_msg:
2483
2484 0001185D E86B4BFFFF
                                                call print_msg
2485
                                                ; 29/04/2016
```

```
2486 00011862 BE[3C140100]
                                       mov esi, inv_msg_for_trdos v2
2487 00011867 E8614BFFFF
                                        call print msg
2489
                                    loc ifc terminate process:
2490
                                       ; u.uno = process number
2491
                                         ; 29/04/2016
2492
                                         ; 02/05/2016
2493
2494 0001186C FE05[5B030300]
                                         inc byte [sysflg] ; OFFh -> 0
                                         mov eax, 1
2496 00011872 B801000000
2497 00011877 E96DB0FFFF
                                         jmp sysexit
2498
2499
                                    ; 07/03/2015
2500
                                   ; Temporary Code
2501
                                    display_disks:
2502 0001187C 803D[565D0000]00
                                        cmp byte [fd0 type], 0
2503 00011883 7605
                                              short ddsks1
                                         jna
                                         call pdskm
2504 00011885 E87D000000
                                    ddsks1:
                                               byte [fd1 type], 0
2506 0001188A 803D[575D0000]00
                                     cmp
2507 00011891 760C
                                               short ddsks2
                                         jna
                                               byte [dskx], '1'
2508 00011893 C605[D7190100]31
                                         mov
2509 0001189A E868000000
                                     call pdskm
2510
                                    ddsks2:
                                               byte [hd0_type], 0
2511 0001189F 803D[585D0000]00
                                        cmp
2512 000118A6 7654
                                         jna
                                               short ddsk6
2513 000118A8 66C705[D5190100]68-
                                               word [dsktype], 'hd'
                                       mov
2513 000118B0 64
2514 000118B1 C605[D7190100]30
                                       mov
                                               byte [dskx], '0'
2515 000118B8 E84A000000
                                       call pdskm
2516
                                    ddsks3:
                                               byte [hd1_type], 0
2517 000118BD 803D[595D0000]00
                                    cmp
2518 000118C4 7636
                                               short ddsk6
                                         jna
2519 000118C6 C605[D7190100]31
                                               byte [dskx], '1'
                                         mov
2520 000118CD E835000000
                                        call
                                               pdskm
                                    ddsks4:
2522 000118D2 803D[5A5D0000]00
                                    cmp
                                               byte [hd2_type], 0
                                               short ddsk6
2523 000118D9 7621
                                         jna
2524 000118DB C605[D7190100]32
                                               byte [dskx], '2'
                                         mov
2525 000118E2 E820000000
                                        call pdskm
                                    ddsks5:
2526
                                               byte [hd3_type], 0
2527 000118E7 803D[5B5D0000]00
                                    cmp
2528 000118EE 760C
                                         jna
                                               short ddsk6
2529 000118F0 C605[D7190100]33
                                               byte [dskx], '3'
                                         mov
2530 000118F7 E80B000000
                                         call pdskm
                                    ddsk6:
2532 000118FC BE[FF190100]
                                               esi, nextline
                                    mov
                                               pdskml
2533 00011901 E806000000
                                         call
                                    pdskm_ok:
2535 00011906 C3
                                        retn
2536
                                    pdskm:
2537 00011907 BE[D3190100]
                                     mov
                                               esi, dsk_ready_msg
2538
                                    pdskml:
2539 0001190C AC
                                    lodsb
2540 0001190D 08C0
                                         or al, al
                                         jz short pdskm_ok
2541 0001190F 74F5
                                         push esi
2542 00011911 56
2543
                                         ; 13/05/2016
                                         mov ebx, 7 ; Black background,
2544 00011912 BB07000000
                                                  ; light gray forecolor
2545
2546
                                                     ; Video page 0 (bh=0)
                                         call _write_tty
2547 00011917 E89F03FFFF
                                         pop esi
2548 0001191C 5E
2549 0001191D EBED
                                         jmp short pdskml
2550
2551 0001191F 90
                                    Align 2
                                       ; 21/08/2014
2552
2553
                                    exc_msg:
                                    db "CPU exception ! "
2554 00011920 435055206578636570-
2554 00011929 74696F6E202120
                                    excnstr: ; 25/08/2014
2555
                                    db "??h", " EIP : "
2556 00011930 3F3F68202045495020-
2556 00011939 3A20
2557
                                    EIPstr: ; 29/08/2014
2558 0001193B 00<rept>
                                         times 12 db 0
2559
                                         ; 23/02/2015
2560
2561
                                         ; 25/08/2014
2562
                                    ;scounter:
                                       db 5
2563
2564
                                         db 19
2565
2566
                                    ; 06/11/2014
2567
                                    ; Memory Information message
2568
                                    ; 14/08/2015
2569
                                    msg_memory_info:
2570 00011947 07
                                         db
                                               07h
2571 00011948 0D0A
                                         db
                                                ODh, OAh
                                                "MEMORY ALLOCATION INFO", ODh, OAh, ODh, OAh
                                         ; db
                                                "Total memory : "
2573 0001194A 546F74616C206D656D-
                                         db
2573 00011953 6F7279203A20
2574
                                    mem_total_b_str: ; 10 digits
2575 00011959 303030303030303030-
                                         db
                                               "0000000000 bytes", 0Dh, 0Ah
2575 00011962 302062797465730D0A
                                                               ", 20h, 20h, 20h
2576 0001196B 2020202020202020-
                                         db
2576 00011974 202020202020202020
                                    mem_total_p_str: ; 7 digits
2577
2578 0001197D 303030303030302070-
                                         db
                                                "0000000 pages", 0Dh, 0Ah
2578 00011986 616765730D0A
                                               0Dh, 0Ah
2579 0001198C 0D0A
                                         db
2580 0001198E 46726565206D656D6F-
                                         db
                                                "Free memory : "
2580 00011997 727920203A20
                                    free_mem_b_str: ; 10 digits
                                             "????????? bytes", ODh, OAh
2582 0001199D 3F3F3F3F3F3F3F3F3F-
                                         db _
```

```
2583 000119AF 2020202020202020-
                                     db
                                                          ", 20h, 20h, 20h
2583 000119B8 202020202020202020
2584
                                 free_mem_p_str: ; 7 digits
2585 000119C1 3F3F3F3F3F3F3F2070-
                                      db
                                           "??????? pages", ODh, OAh
2585 000119CA 616765730D0A
2586 000119D0 0D0A00
                                            0Dh, 0Ah, 0
2587
2588
                                 dsk_ready_msg:
2589 000119D3 0D0A
                                     db
                                            ODh, OAh
2590
                                 dsktype:
2591 000119D5 6664
                                            'fd'
                                      db
2592
                                 dskx:
2593 000119D7 30
                                            '0'
                                      db
2594 000119D8 20
                                            20h
                                      db
2595 000119D9 697320524541445920-
                                      db
                                            'is READY ...'
2595 000119E2 2E2E2E
                                            0
2596 000119E5 00
                                      db
2597
2598
                                 setup_error_msg:
2599 000119E6 0D0A
                                      db 0Dh, 0Ah
2600 000119E8 4469736B2053657475-
                                      db 'Disk Setup Error !'
2600 000119F1 70204572726F722021
2601 000119FA 0D0A00
                                      db 0Dh, 0Ah,0
2602
2603
                                 next2line: ; 08/02/2016
2604 000119FD 0D0A
                                     db
                                           ODh, OAh
                                 nextline:
2605
2606 000119FF 0D0A00
                                      db
                                            0Dh, 0Ah, 0
2607
2608
                                 ; KERNEL - SYSINIT Messages
2609
                                 ; 24/08/2015
                                 ; 13/04/2015 - (Retro UNIX 386 v1 Beginning)
2610
2611
                                 ; 14/07/2013
2612
                                 ;kernel_init_err_msg:
2613
                                      db 0Dh, 0Ah
                                      db 07h
2614
                                      db 'Kernel initialization ERROR !'
2615
2616
                                      db ODh, OAh, O
2617
                                 ;welcome_msg:
2618
2619
                                 ;
                                      db 0Dh, 0Ah
                                      db 07h
2620
2621
                                      db 'Welcome to TRDOS 386 Operating System !'
                                      db ODh, OAh
2622
                                      db 'by Erdogan Tan - 31/12/2017 (v2.0.0)'
2623
                                      db ODh, OAh, O
2624
2625
2626
                                 panic_msg:
                                      db 0Dh, 0Ah, 07h
2627 00011A02 0D0A07
2628 00011A05 4552524F523A204B65-
                                      db 'ERROR: Kernel Panic !'
2628 00011A0E 726E656C2050616E69-
2628 00011A17 632021
                                      db 0Dh, 0Ah, 0
2629 00011A1A 0D0A00
2630
2631
                                 ;msgl_drv_not_ready:
                                      db 07h, 0Dh, 0Ah
2632
                                 ;
2633
                                        db 'Drive not ready or read error !'
2634
                                        db ODh, OAh, O
2635
2636
                                 starting_msg:
2637 00011A1D 5475726B6973682052-
                                      db "Turkish Rational DOS v2.0 [01/09/2020] ...", 0
2637 00011A26 6174696F6E616C2044-
2637 00011A2F 4F532076322E30205B-
2637 00011A38 30312F30392F323032-
2637 00011A41 305D202E2E2E00
2638
                                 NextLine:
                                      db 0Dh, 0Ah, 0
2639 00011A48 0D0A00
2640
                                 %include 'audio.s'; 03/04/2017
2641
                              1
  2
                              <1>; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.2 - audio.s
  3
                              <1> ; Last Update: 01/09/2020
                              <1> ; -----
  5
                              <1> ; Beginning: 03/04/2017
  6
  7
  8
                              <1>; Assembler: NASM version 2.11 (trdos386.s)
                              9
 10
                             <1>
                              <1>; AUDIO CONTROLLER & CODEC DEFINITIONS & CODE FOR TRDOS 386
 11
 12
 13
                              15
                              16
                              <1>
 17
                              <1> ; PCI EQUATES
 18
                             <1>
                             <1> BITO EQU 1
 19
                             <1> BIT1 EOU 2
 20
 21
                             <1> BIT2 EQU 4
                             <1> BIT3 EQU 8
 22
                             <1> BIT4 EQU 10h
 23
 24
                             <1> BIT5 EQU 20h
                             <1> BIT6 EQU 40h <1> BIT7 EQU 80h
 25
 26
 27
                             <1> BIT8 EQU 100h
                             <1> BIT9 EQU 200h
 28
                             <1> BIT10 EQU 400h
 29
                             <1> BIT11 EQU 800h
 30
                             <1> BIT12 EQU 1000h
 31
                             <1> BIT13 EQU 2000h
 32
                             <1> BIT14 EQU 4000h
 33
                             <1> BIT15 EQU 8000h
 34
 35
                             <1> BIT16 EQU 10000h
```

2582 000119A6 3F2062797465730D0A

```
<1> BIT17 EQU 20000h
   36
   37
                                    <1> BIT18 EQU 40000h
                                    <1> BIT19 EQU 80000h
   38
   39
                                    <1> BIT20 EQU 100000h
   40
                                    <1> BIT21 EQU 200000h
   41
                                    <1> BIT22 EQU 400000h
                                    <1> BIT23 EQU 800000h
   42
                                    <1> BIT24 EQU 1000000h
   43
   44
                                    <1> BIT25 EQU 2000000h
                                    <1> BIT26 EQU 4000000h
   45
   46
                                    <1> BIT27 EQU 8000000h
   47
                                    <1> BIT28 EQU 1000000h
   48
                                    <1> BIT29 EQU 2000000h
   49
                                    <1> BIT30 EQU 4000000h
   50
                                    <1> BIT31 EQU 80000000h
   51
                                    <1> NOT_BIT31 EQU 7FFFFFFFh
   52
   53
                                    <1> ; PCI equates
   54
                                    <1> ; PCI function address (PFA)
   55
                                    <1>; bit 31 = 1
   56
                                    <1> ; bit 23:16 = bus number
                                                                     (0-255)
   57
                                    <1>; bit 15:11 = device number (0-31)
                                    <1>; bit 10:8 = function number (0-7)
   58
   59
                                    <1>; bit 7:0 = register number (0-255)
   60
                                    <1>
   61
                                    <1> IO_ADDR_MASK
                                                       EQU
                                                                OFFFEh ; mask off bit 0 for reading BARs
   62
                                    <1> PCI INDEX PORT EQU
                                                                0CF8h
   63
                                    <1> PCI_DATA_PORT
                                                       EOU
                                                                0CFCh
    64
                                    <1> PCI32
                                                        EQU
                                                                BIT31
                                                                       ; bitflag to signal 32bit access
                                                                      ; bitflag for 16bit access
   65
                                    <1> PCI16
                                                                BIT30
                                                       EQU
   66
                                    67
                                    <1>
                                    <1> PCI FN0
                                                       EQU
                                                                0 << 8
   68
   69
                                    <1> PCI FN1
                                                       EQU
                                                                1 << 8
   70
                                    <1> PCI FN2
                                                                2 << 8
                                                       EQU
   71
                                    <1> PCI_FN3
                                                       EQU
                                                                3 << 8
                                                                4 << 8
   72
                                    <1> PCI FN4
                                                       EQU
   73
                                    <1> PCI_FN5
                                                                5 << 8
                                                       EQU
   74
                                    <1> PCI_FN6
                                                       EQU
                                                                6 << 8
   75
                                    <1> PCI_FN7
                                                                7 << 8
                                                       EQU
   76
                                    <1>
   77
                                    <1> PCI CMD REG EQU
                                                          04h
                                                                ; reg 04, command reg
                                                                BITO ; i/o decode enable
   78
                                    <1> IO_ENA
                                                          EQU
                                    <1> MEM ENA
   79
                                                          EQU
                                                                 BIT1 ; memory decode enable
                                                                  BIT2 ; bus master enable
   80
                                    <1> BM ENA
                                                          EQU
   81
                                    <1>
   82
                                    <1>; VIA VT8233 EQUATES
   83
                                    <1>
   84
                                    <1> VIA VID
                                                          equ 1106h
                                                                      ; VIA's PCI vendor ID
                                    <1> VT8233_DID
                                                       equ 3059h; VT8233 (VT8235) device ID
   85
   86
                                    <1>
   87
                                    <1> PCI IO BASE
                                                            equ 10h
                                    <1> AC97 INT_LINE
                                                            equ 3Ch
   88
                                                            equ 41h
   89
                                    <1> VIA_ACLINK_CTRL
   90
                                    <1> VIA ACLINK STAT
                                                            equ 40h
   91
                                    <1> VIA_ACLINK_C00_READY equ 01h ; primary codec ready
   92
                                    <1>
                                                         equ 80h ; dword
   93
                                    <1> VIA_REG_AC97
   94
                                    <1>
                                                                       80h; 0: disable, 1: enable
   95
                                    <1> VIA ACLINK CTRL ENABLE
                                                                 equ
   96
                                    <1> VIA_ACLINK_CTRL_RESET
                                                                       40h; 0: assert, 1: de-assert
                                                                 equ
                                    <1> VIA_ACLINK_CTRL_SYNC
<1> VIA_ACLINK_CTRL_VRAequ
   97
                                                                       20h ; 0: release SYNC, 1: force SYNC hi
                                                                 equ
   98
                                                                 08h ; 0: disable VRA, 1: enable VRA
   99
                                    <1> VIA_ACLINK_CTRL_PCMequ
                                                                 04h; 0: disable PCM, 1: enable PCM
  100
                                    <1>
                                                                       ; 3D Audio Channel slots 3/4
  104
                                    <1> VIA_ACLINK_CTRL_INIT
                                                                 equ (VIA_ACLINK_CTRL_ENABLE +
VIA_ACLINK_CTRL_RESET +
                                                     VIA_ACLINK_CTRL_PCM +
                                                                                                          VIA_ACLINK_CTRL_VRA)
                                    <1>
  105
  106
                                    <1> CODEC_AUX_VOL
                                                                 equ 04h
  107
                                    <1> VIA_REG_AC97_BUSY equ 01000000h ; (1<<24)</pre>
  108
                                    <1> VIA_REG_AC97_CMD_SHIFT
                                                                equ 10h ; 16
                                    <1> VIA REG AC97 PRIMARY VALID equ 02000000h ; (1<<25)
  109
                                    <1> VIA_REG_AC97_READ equ 00800000h; (1<<23)
  110
  111
                                    <1> VIA_REG_AC97_CODEC_ID_SHIFT        equ    1Eh ; 30
                                    <1> VIA_REG_AC97_CODEC_ID_PRIMARY equ 0
<1> VIA_REG_AC97_DATA_SHIFT equ 0
  112
  113
  114
                                    <1> VIADEV PLAYBACK
                                                                equ
                                    <1> VIA_REG_OFFSET_STATUS
  115
                                                                equ
                                                                     0
                                                                         ;; byte - channel status
                                                                     01h ;; byte - channel control
                                    <1> VIA_REG_OFFSET_CONTROL
  116
                                                                equ
  117
                                    <1> VIA REG CTRL START equ
                                                                80h ;; WO
  118
                                    <1> VIA_REG_CTRL_TERMINATE
                                                                equ
                                                                     40h ;; WO
                                                                          ;; RW
                                    <1> VIA_REG_CTRL_PAUSE
  119
                                                                      08h
                                                                equ
                                                                      01h ;; RW - probably reset? undocumented
  120
                                    <1> VIA_REG_CTRL_RESET
                                                                equ
                                    <1> VIA REG OFFSET STOP_IDX equ
                                                                     08h ;; dword - stop index, channel type, sample rate
  121
                                                                     200000h ;; RW
  122
                                    <1> VIA8233_REG_TYPE_16BIT equ
  123
                                    <1> VIA8233_REG_TYPE_STEREO equ
                                                                     100000h ;; RW
  124
                                    <1> VIA REG OFFSET CURR INDEX equ OFh ;; byte - channel current index (for via8233 only)
  125
                                    <1> VIA_REG_OFFSET_TABLE_PTR equ 04h ;; dword - channel table pointer
                                    <1> VIA REG OFFSET CURR PTR equ
                                                                     04h ;; dword - channel current pointer
  126
                                    <1> VIA REG OFS PLAYBACK_VOLUME_L equ 02h ;; byte
  127
  128
                                    <1> VIA_REG_OFS_PLAYBACK_VOLUME_R equ 03h ;; byte
                                    <1> VIA REG CTRL AUTOSTART
  129
                                                                 equ
                                                                       20h
                                    <1> VIA REG CTRL INT EOL
  130
                                                                 equ
                                                                       02h
                                                                 equ 01h
  131
                                    <1> VIA_REG_CTRL_INT_FLAG
                                    <1> VIA REG CTRL INT equ
  134
                                                                (VIA_REG_CTRL_INT_FLAG +
                                                                                                                       VIA_REG_CTRL_INT_EOL
                               VIA_REG_CTRL_AUTOSTART)
  135
                                    <1>
                                    <1> VIA REG STAT STOP IDX
                                                                     10h
                                                                            ;; RO ; 27/07/2020
  136
                                                                 equ
  137
                                    <1>
                                                                      ; current index = stop index
                                                                       04h ;; RWC
                                    <1> VIA REG STAT STOPPED
  138
                                                                 equ
                                                                       ;; RWC
  139
                                    <1> VIA_REG_STAT_EOL equ
                                                                 02h
  140
                                    <1> VIA_REG_STAT_FLAG equ
                                                                 01h
                                                                       ;; RWC
                                    <1> VIA_REG_STAT_ACTIVEequ
  141
                                                                 80h
                                                                       ;; RO
                                    <1>; 28/11/2016
  142
  143
                                    <1> VIA_REG_STAT_LAST equ
                                                                 40h
                                                                       ;; RO
```

```
145
                                 <1>
146
                                                                ; and End of Block
147
                                 <1>
                                 <1> VIA_REG_OFFSET_CURR_COUNT equ 0Ch ;; dword - channel current count, index
148
149
                                 <1>
150
                                 <1> PORTB
                                                             010h ; Refresh signal status
                                 <1> REFRESH_STATUS
                                                       EQU
151
152
                                 <1>
153
                                 <1> ; AC97 Codec registers.
154
                                 <1>
155
                                 <1>; 22/07/2020
                                 <1> ; REALTEK ALC655 and ADI SOUNDMAX AD1980 CODEC MIXER REGISTERS
156
157
158
                                 <1> ; each codec/mixer register is 16bits
159
                                 <1>
                                 <1> CODEC RESET REG
160
                                                                            00h ; reset codec
                                 <1> CODEC_MASTER_VOL_REG
                                                                            02h ; master volume
161
                                                                    equ
162
                                 <1> CODEC_HP_VOL_REG
                                                                             04h ; headphone volume ; AD1980
                                                                    equ
                                 <1> CODEC MASTER MONO VOL REG
163
                                                                            06h ; master mono volume (mono-out)
                                                                    equ
                                 <1> ; CODEC MASTER TONE REG
164
                                                                    equ
                                                                            08h ; master tone (R+L) ; (not used)
                                 <1> CODEC PCBEEP VOL REG
165
                                                                            OAh ; PC beep volume ; ALC655
                                                                    equ
                                 <1> CODEC PHONE VOL REG
                                                                            OCh ; phone volume
166
                                                                    equ
167
                                 <1> CODEC_MIC_VOL_REG
                                                                    equ
                                                                            OEh ; mic volume
                                 <1> CODEC_LINE_IN_VOL_REG <1> CODEC_CD_VOL_REG
                                                                            10h ; line in volume
168
                                                                    equ
169
                                                                    equ
                                                                            12h ; CD volume
170
                                 <1> ; CODEC VID VOL REG
                                                                            14h ; video volume ; (not used)
                                                                    equ
                                 <1> CODEC_AUX_VOL_REG
171
                                                                    equ
                                                                            16h ; aux volume
172
                                 <1> CODEC_PCM_OUT_REG
                                                                            18h ; PCM out volume
                                                                    equ
                                 <1> CODEC RECORD SELECT REG
173
                                                                            1Ah ; record select
                                                                    equ
174
                                 <1> CODEC RECORD VOL REG
                                                                            1Ch ; record volume (record gain)
                                                                    equ
                                 <1> ; CODEC RECORD MIC VOL REG
175
                                                                            1Eh ; record mic volume ; (not used)
                                                                    equ
                                 <1> CODEC_GP_REG
                                                                            20h ; general purpose
176
                                                                    equ
177
                                 <1> ; CODEC 3D CONTROL REG
                                                                            22h ; 3D control
                                                                     equ
178
                                 <1> ;; CODEC AUDIO INT PAGING REG
                                                                           24h ; audio int & paging ; (not used)
                                                                    equ
179
                                 <1> CODEC_POWER_CTRL_REG
                                                                            26h ; power down control
                                                                     equ
                                 <1> CODEC EXT AUDIO REG
180
                                                                            28h ; extended audio ID
                                                                    equ
                                 <1> CODEC_EXT_AUDIO_CTRL_REG
181
                                                                            2Ah ; extended audio status/control
                                                                    equ
                                                                             2Ch ; PCM front sample rate
182
                                 <1> CODEC_PCM_FRONT_DACRATE_REG
                                                                     equ
                                 <1> CODEC PCM SURND DACRATE REG
                                                                            2Eh ; PCM surround sample rate
183
                                                                    equ
                                                                            30h ; PCM Center/LFE sample rate
184
                                 <1> CODEC PCM LFE DACRATE REG
                                                                    equ
                                 <1> CODEC_LR_ADCRATE_REG <1> CODEC_MIC_ADCRATE_REG
185
                                                                    equ
                                                                             32h ; PCM input sample rate
                                                                            34h ; mic in sample rate ; AD1980
186
                                                                    equ
187
                                 <1> CODEC_PCM_LFE_VOL_REG
                                                                            36h ; PCM Center/LFE volume
                                                                     equ
                                 <1> CODEC_PCM_SURND_VOL_REG
188
                                                                            38h ; PCM surround volume
                                                                    equ
                                 <1> ; CODEC_SPDIF_CTRL_REG
                                                                            3Ah ; S/PDIF control
189
                                                                     equ
                                 <1>; 22/07/2020
190
                                 <1> CODEC_MISC_CRTL_BITS_REG equ
                                                                    76h
191
                                                                          ; misc control bits ; AD1980
192
                                 <1>;
                                 <1> CODEC VENDOR ID1
                                                                     7Ch
                                                                          ; REALTEK: 414Ch, ADI: 4144h
193
                                                              equ
194
                                 <1> CODEC_VENDOR_ID2
                                                                     7Eh
                                                                          ; REALTEK: 4760h, ADI: 5370h
                                                              equ
195
                                 <1>
                                 <1>; VT8233 SGD bits (21/04/2017)
196
197
                                 <1> FLAG EQU BIT30
198
                                 <1> EOL EQU BIT31
199
                                 <1>
200
                                 <1> ; INTEL ICH EQUATES
                                 <1> ; 28/05/2017
201
                                                  equ 8086h ; Intel's PCI vendor ID
equ 2415h ; ICH (82801AA) device ID
202
                                 <1> INTEL_VID equ
203
                                 <1> ICH DID
                                                             ; native audio mixer Base Address Register
                                 <1> NAMBAR REG
204
                                                     equ10h
205
                                 <1> NABMBAR REG
                                                    equ14h
                                                             ; native audio bus mastering Base Addr Reg
206
                                 <1>
                                                 equ
207
                                 <1> PI_CR_REG
                                                            OBh ; PCM in Control Register
                                                         1Bh ; PCM out Control Register
208
                                 <1> PO_CR_REG
                                                equ
209
                                 <1> MC_CR_REG
                                                         2Bh
                                                                 ; MIC in Control Register
                                                 equ
210
                                 <1>
                                 <1> PI_SR_REG
                                                         6
                                                                 ; PCM in Status register
211
                                                 equ
212
                                 <1> PO SR REG
                                                         16h
                                                                 ; PCM out Status register
                                                 equ
213
                                 <1> MC_SR_REG
                                                                 ; MIC in Status register
                                                         26h
                                                 equ
214
                                 <1>
215
                                 <1> IOCE
                                                         BIT4
                                                                 ; interrupt on complete enable.
                                                 equ
                                 <1> FEIFE
                                                         BIT3
216
                                                 equ
                                                                 ; set if you want an interrupt to fire
                                                         BIT2
                                                                ; last valid buffer interrupt enable.
217
                                 <1> LVBIE
                                                 equ
                                                                 ; reset registers. Nukes all regs
218
                                 <1> RR
                                                         BIT1
                                                 equ
                                                                     ; except bits 4:2 of this register.
219
                                 <1>
220
                                 <1>
                                                                    ; Only set this bit if BIT 0 is 0
                                                                 ; Run/Pause
221
                                 <1> RPBM
                                                 equ
                                                         BIT0
222
                                 <1>
                                                            ; set this bit to start the codec!
223
                                 <1>
                                                         0
224
                                 <1> PI BDBAR REG equ
                                                                 ; PCM in buffer descriptor BAR
225
                                 <1> PO_BDBAR_REG equ
                                                         10h
                                                                 ; PCM out buffer descriptor BAR
226
                                 <1> MC_BDBAR_REG equ
                                                         20h
                                                                 ; MIC in buffer descriptor BAR
227
228
                                 <1> PI CIV REG equ
                                                         4
                                                                 ; PCM in current Index value (RO)
                                 <1> PO_CIV_REG
                                                                 ; PCM out current Index value (RO)
229
                                                 equ
                                                         14h
                                 <1> MC_CIV_REG
230
                                                         24h
                                                                 ; MIC in current Index value (RO)
                                                 equ
231
                                 <1>
                                 <1> PI LVI REG
232
                                                         5
                                                                 ; PCM in Last Valid Index
                                                 equ
                                                                 ; PCM out Last Valid Index
233
                                 <1> PO LVI REG
                                                         15h
                                                 equ
234
                                 <1> MC_LVI_REG
                                                         25h
                                                                 ; MIC in Last Valid Index
                                                 equ
235
                                 <1>
                                 <1> IOC
                                                         BIT31; Fire an interrupt whenever this
236
                                                 equ
237
                                                              ; buffer is complete.
                                 <1>
238
                                 <1> BUP
                                                         BIT30; Buffer Underrun Policy.
                                                 equ
239
                                 <1>
240
                                 <1> GLOB CNT REG equ
                                                                 ; Global Control Register
241
                                 <1> GLOB_STS_REG equ
                                                         30h
                                                                 ; Global Status register (RO)
242
                                 <1>
243
                                 <1> CTRL_ST_CREADY
                                                             BIT8+BIT9+BIT28 ; Primary Codec Ready
244
                                 <1>
245
                                 <1> CODEC REG POWERDOWN
                                                           equ 26h
                                 <1> CODEC_REG_ST
246
                                                           equ 26h
247
                                 <1>
248
                                 <1> ; 22/06/2017
```

<1> VIA\_REG\_STAT\_TRIGGER\_QUEUED equ 08h ;; RO

144

```
<1> PO PICB REG equ 18h ; PCM Out Position In Current Buffer Register
249
250
251
                               <1> ; CODE
252
253
                               254
                               <1>
255
                               <1>; CODE for INTEL ICH AC'97 AUDIO CONTROLLER
256
                               <1>
257
                               <1> DetectICH:
                               <1> ; 10/06/2017
258
                                      ; 05/06/2017
259
                               <1>

<1> ; 29/05/2017
<1> ; 28/05/2017
<1> mov eax, (ICH_DID << 16) + INTEL_VID
</pre>
260
261
262 00011A4B B886801524
                              <1> call pciFindDevice
<1> jnc short d_ac97_1
263 00011A50 E876000000
                                                 short d_ac97_1
264 00011A55 730D
                               <1> d ac97 0:
                               <1> ; couldn't find the audio device!
266
267 00011A57 C3
                               <1>
268
                               <1>
                               <1>; CODE for VIA VT8233 AUDIO CONTROLLER
269
270
                               <1>
                               <1> DetectVT8233:
271
                                      ; 10/06/2017
272
                               <1>
                                      ; 05/06/2017
273
                               <1>
                               <1> ; 29/05/2017
<1> ; 03/04/2017
274
                              , 03/04/201/
<1> mov eax, (VT8233_DID << 16) + VIA_VID
<1> call poiFindDowice
276 00011A58 B806115930
                              <1> call pciFindDevice
<1>; jnc short d_vt8233_0
277 00011A5D E869000000
278
                               <1>; couldn't find the audio device!
279
280
                               <1> ; retn
281 00011A62 72F3
                                              short d_ac97_0 ; 28/05/2017
                               <1>
                                        jс
282
                               <1> d_vt8233_0:
                                     ; \overline{2}4/03/2017 ('player.asm')
283
                               <1>
                                        ; 12/11/2016
284
                               <1>
                                      ; Erdogan Tan - 8/11/2016
285
                               <1>
                                     ; References: Kolibrios - vt823x.asm (2016)
286
                               <1>
287
                               <1>
                                                    VIA VT8235 V-Link South Bridge (VT8235-VIA.PDF) (2002)
                                                    lowlevel.eu - AC97 (2016)
288
                               <1>
289
                               <1>
                                                    .wav player for DOS by Jeff Leyda (2002) -this file-
                                                    Linux kernel - via82xx.c (2016)
290
                               <1>
                               <1> d_ac97_1:
291
292
                               <1>
                                      ; eax = BUS/DEV/FN
                                        ; 0000000BBBBBBBBDDDDDFFF00000000
293
                               <1>
294
                               <1>
                                        ; edx = DEV/VENDOR
                                        ; DDDDDDDDDDDDDDDDVVVVVVVVVVVVVVVV
                               <1>
                               <1>
296
297 00011A64 A3[486B0100]
                               <1>
                                        mov [audio_dev_id], eax
                                        mov [audio_vendor], edx
298 00011A69 8915[4C6B0100]
                               <1>
299
                               <1>
300
                               <1>
                                        ; init controller
301 00011A6F B004
                                        mov al, PCI_CMD_REG ; command register (04h)
                               <1>
302 00011A71 E8E2000000
                               <1>
                                       call pciRegRead32
303
                               <1>
304
                               <1>
                                        ; eax = BUS/DEV/FN/REG
                                        ; edx = STATUS/COMMAND
305
                               <1>
                                              SSSSSSSSSSSSSCCCCCCCCCCCCCCC
306
                               <1>
307 00011A76 8915[506B0100]
                               <1>
                                              [audio_stats_cmd], edx
308
                               <1>
309 00011A7C B010
                               <1>
                                             al, PCI_IO_BASE ; IO base address register (10h)
                                        mov
310
                               <1>
                                              al, NAMBAR_REG ; Native Audio Mixer BAR (10h)
                                        ; mov
311 00011A7E E8D5000000
                               <1>
                                        call
                                              pciRegRead32
                               <1>
312
313 00011A83 66813D[4C6B0100]86- <1>
                                              word [audio_vendor], 8086h; AC'97 ?
                                        cmp
313 00011A8B 80
                              <1>
314 00011A8C 751F
                               <1>
                                        jne
                                              short d_vt8233_1
                              <1>
315
316 00011A8E 6683E2FE
                              <1>
                                               dx, OfffEh; Audio Codec IO_ADDR_MASK
317 00011A92 668915[786B0100] <1>
                                               [NAMBAR], dx
                                        mov
318
                               <1>
319 00011A99 B014
                               <1>
                                               al, NABMBAR REG; Native Audio Bus Mastering BAR (14h)
                                        mov
320 00011A9B E8B8000000
                               <1>
                                        call
                                              pciRegRead32
                               <1>
322 00011AA0 6683E2C0
                               <1>
                                              dx, OFFCOh ; Audio Controller IO ADDR MASK
                                        and
323 00011AA4 668915[7A6B0100]
                                              [NABMBAR], dx
                              <1>
                                        mov
                                        ;mov [audio_io_base], dx
                               <1>
325
                               <1>
                                        jmp short d ac97 2
326 00011AAB EB0B
                               <1>
327
                               <1>
                               <1> d_vt8233 1:
328
329 00011AAD 6683E2C0
                               <1>
                                         and
                                                dx, 0FFC0h ; Audio Controller IO_ADDR_MASK
330 00011AB1 668915[466B0100]
                               <1>
                                          mov
                                                  [audio_io_base], dx
                               <1>
                               <1> d_ac97_2:
332
                                        ; 10/06/2017
333
                               <1>
334 00011AB8 B03C
                                        mov al, AC97 INT LINE; Interrupt Line Register (3Ch)
                               <1>
                                        ;call pciRegRead32
335
                               <1>
336 00011ABA E886000000
                              <1>
                                        call pciRegRead8
337
                               <1>
                                        ; and edx, OFFh
338
                               <1>
                                              dx, 0FFh
339 00011ABF 6681E2FF00
                               <1>
                                        and
340
                               <1>
341 00011AC4 8815[436B0100]
                               <1>
                                               [audio_intr], dl
342
                               <1>
343 00011ACA C3
                               <1>
                                        retn
344
                               <1>
                                        ;; (Note: Interrupts are already enabled by TRDOS 386 kernel!)
345
                               <1>
346
                               <1>
                                        ;mov cx, dx
347
                               <1>
                                        ;in al, OA1h ; irq 8-15
348
                               <1>
349
                               <1>
                                        ;mov ah, al
                                        ;in al, 21h ; irq 0-7
350
                               <1>
351
                               <1>
                                        ;btr ax, dx ; unmask ; 17/03/2017
352
                               <1>
                                        ;;bts ax, dx ; MASK interrupt ; 10/06/2017
```

```
353
                                 <1>
                                           ;out 21h, al ; irq <= 7
354
                                 <1>
                                           ;mov al, ah
                                           ;out 0A1h, al; irq > 7
355
                                 <1>
356
                                 <1>
357
                                 <1>
358
                                 <1>
                                           ; 10/06/2017
359
                                 <1>
                                          ; === Intel ICH I/O Controller Hub Datasheet, Section 8.1.16 ===
360
                                 <1>
                                          ; PRQ[n]_ROUT Register (61h, PRQB) Bit 7:
361
                                 <1>
                                           ; Interrupt Routing Enable (IRQEN).
362
                                 <1>
                                          ; 0 = The corresponding PIRQ is routed to one of the ISA-compatible
363
                                 <1>
                                                interrupts specified in bits[3:0].
364
                                 <1>
                                          ; 1 = \text{The PIRQ} is not routed to the 8259.
                                           ; Note: If the PIRQ is intended to cause an interrupt to the ICH's \,
365
                                 <1>
366
                                 <1>
                                                 integrated I/O APIC, then this bit should be set to 0 and
                                                 the APIC EN bit should be set to 1.
367
                                 <1>
368
                                 <1>
                                                 The IRQEN must be set to 0 and the PIRQ routed to
369
                                 <1>
                                                 an 8259 interrupt via the IRQ Routing filed (bits[3:0).
370
                                 <1>
                                                 The corresponding 8259 interrupt must be masked via the
                                                 appropriated bit in the 8259's OCW1 (Interrupt Mask)
371
                                 <1>
372
                                 <1>
                                                 register. The IOAPIC must then be enabled by setting
373
                                 <1>
                                                 the APIC_EN bit in the GEN_CNTL register.
374
                                 <1>
375
                                 <1>
                                           ;mov eax, 0F861h ; D31:F0
376
                                 <1>
                                                 ;AL=61h : PIRQ[B] Routing Control Reg, LPC interface
377
                                 <1>
                                           ;; mov dl, [audio intr]
378
                                 <1>
                                           ;call pciRegWrite8
379
                                 <1>
                                           ;; mov al, ODOh ; General Control Register (GEN CTL)
380
                                 <1>
                                           ;;call pciRegRead32
381
                                 <1>
                                           ;;or edx, 100h; Bit 8, APIC_EN (Enable I/O APIC)
                                           ;;;call
                                                      pciRegWrite32
382
                                 <1>
383
                                 <1>
                                           ;;and edx, \sim 100h
                                           ;;call pciRegWrite32 ; ; Bit 8, APIC EN (Disable I/O APIC)
384
                                 <1>
385
                                 <1>
386
                                 <1>
                                                              ; 8259 ELCR2
387
                                 <1>
                                           ;mov dx, 4D1h
                                                al, dx
388
                                 <1>
                                           ;in
                                           ;mov ah, al
389
                                 <1>
                                                              ; 8259 ELCR1
390
                                 <1>
                                           ;; mov dx, 4D0h
391
                                 <1>
                                           ;dec dl
392
                                 <1>
                                           ;in al, dx
                                           ;bts ax, cx
393
                                 <1>
                                           ;;mov dx, 4D0h
394
                                 <1>
                                           ;out dx, al
                                                              ; set level-triggered mode
395
                                 <1>
396
                                 <1>
                                           ;mov al, ah ; 29/05/2017
397
                                 <1>
                                          ;;mov dx, 4D1h
398
                                 <1>
                                           ;inc dl
399
                                 <1>
                                           ;out dx, al
                                                             ; set level-triggered mode
400
                                 <1>
401
                                 <1>
                                           ;xor eax, eax; 0
402
                                 <1>
403
                                 <1>
                                           ;retn
404
                                 <1>
405
                                 <1>; CODE for PCI
406
                                 <1>
407
                                 <1> pciFindDevice:
408
                                 <1>
                                           ; 03/04/2017 ('pci.asm', 20/03/2017)
409
                                 <1>
                                 <1>
                                           ; scan through PCI space looking for a device+vendor ID
410
411
                                 <1>
                                 <1>
                                          ; Entry: EAX=Device+Vendor ID
412
413
                                 <1>
414
                                 <1>
                                          ; Exit: EAX=PCI address if device found
                                 <1>
                                                EDX=Device+Vendor ID
415
416
                                 <1>
                                                   CY clear if found, set if not found. EAX invalid if CY set.
417
                                 <1>
418
                                 <1>
                                           ; Destroys: ebx, esi, edi, cl
419
                                 <1>
420
                                 <1>
421
                                 <1>
                                           ;push ecx
422 00011ACB 50
                                 <1>
                                           push eax
                                           ;push esi
423
                                 <1>
                                           ;push edi
424
                                 <1>
425
                                 <1>
426 00011ACC 89C6
                                 <1>
                                             mov
                                                                             ; save off vend+device ID
427 00011ACE BF00FFFF7F
                                 <1>
                                                     edi, (80000000h - 100h); start with bus 0, dev 0 func 0
                                            mov
428
                                 <1>
                                 <1> nextPCIdevice:
430 00011AD3 81C700010000
                                                     edi, 100h
                                 <1>
                                             add
431 00011AD9 81FF00F8FF80
                                 <1>
                                                     edi, 80FFF800h
                                                                            ; scanned all devices?
                                             cmp
432 00011ADF F9
                                 <1>
                                             stc
433 00011AE0 740C
                                 <1>
                                             jе
                                                     short PCIScanExit
                                                                           ; not found
                                 <1>
435 00011AE2 89F8
                                <1>
                                             mov
                                                     eax, edi
                                                                             ; read PCI registers
436 00011AE4 E86F000000
                                 <1>
                                                     pciRegRead32
                                 <1>
                                                     edx, esi
437 00011AE9 39F2
                                             cmp
                                                                             ; found device?
438 00011AEB 75E6
                                 <1>
                                             jne
                                                     short nextPCIdevice
439 00011AED F8
                                <1>
                                             clc
                                <1>
440
441
                                <1> PCIScanExit:
442 00011AEE 9C
                                          pushf
                                <1>
                                                 eax, NOT_BIT31 ; 19/03/2017
443 00011AEF B8FFFFFF7F
                                <1>
                                           mov
444 00011AF4 21F8
                                 <1>
                                           and
                                                 eax, edi ; return only bus/dev/fn #
445 00011AF6 9D
                                 <1>
                                           popf
446
                                 <1>
447
                                 <1>
                                          ;pop edi
448
                                 <1>
                                           ;pop
                                                 esi
449 00011AF7 5A
                                 <1>
                                                 edx
                                           pop
450
                                 <1>
                                           ;pop
                                                ecx
451 00011AF8 C3
                                 <1>
                                           retn
452
                                 <1>
453
                                 <1> pciRegRead:
454
                                 <1>
                                          ; 03/04/2017 ('pci.asm', 20/03/2017)
455
                                 <1>
456
                                 <1>
                                           ; 8/16/32bit PCI reader
457
                                 <1>
```

```
; Entry: EAX=PCI Bus/Device/fn/register number
458
                               <1>
459
                                <1>
                                                    BIT30 set if 32 bit access requested
                                                    BIT29 set if 16 bit access requested
                               <1>
460
461
                               <1>
                                                    otherwise defaults to 8 bit read
                               <1>
462
463
                               <1>
                                         ; Exit: DL,DX,EDX register data depending on requested read size
464
                               <1>
465
                               <1>
                                       ; Notel: this routine is meant to be called via pciRegRead8,
466
                               <1>
                                                pciRegread16 or pciRegRead32, listed below.
467
                               <1>
468
                               <1>
                                         ; Note2: don't attempt to read 32 bits of data from a non dword
469
                               <1>
                                                aligned reg number. Likewise, don't do 16 bit reads from
470
                               <1>
                                                non word aligned reg #
471
                               <1>
472 00011AF9 53
                               <1>
                                         push ebx
473 00011AFA 51
                               <1>
                                        push ecx
                                                                 ; save eax, dh
474 00011AFB 89C3
                               <1>
                                         mov
                                                ebx, eax
475 00011AFD 88F1
                                                  cl, dh
                               <1>
                                          mov
476
                               <1>
477 00011AFF 25FFFFFF3F
                                                   eax, NOT PCI32 PCI16 ; clear out data size request
                               <1>
                                           and
                                                                  ; make a PCI access request
478 00011B04 0D00000080
                               <1>
                                           or
                                                   eax, BIT31
                                                  al, ~3 ; NOT 3
                                                                        ; force index to be dword
479 00011B09 24FC
                               <1>
                                           and
480
                               <1>
481 00011B0B 66BAF80C
                               <1>
                                                   dx, PCI_INDEX_PORT
482 00011B0F EF
                               <1>
                                                                        ; write PCI selector
                                          out dx, eax
483
                               <1>
484 00011B10 66BAFC0C
                               <1>
                                          mov
                                                   dx, PCI DATA PORT
485 00011B14 88D8
                               <1>
                                           mov
                                                  al, bl
486 00011B16 2403
                               <1>
                                                                        ; figure out which port to
                                           and
                                                  al, 3
487 00011B18 00C2
                               <1>
                                         add
                                                                        ; read to
                                                  dl, al
488
                               <1>
489 00011B1A F7C300000C0
                               <1>
                                        test
                                                ebx, PCI32+PCI16
490 00011B20 7507
                               <1>
                                          jnz
                                                short _pregr0
                                         in al, dx
491 00011B22 EC
                               <1>
                                                                  ; return 8 bits of data
492 00011B23 88C2
                               <1>
                                        mov dl, al
493 00011B25 88CE
                               <1>
                                         mov
                                               dh, cl
                                                                        ; restore dh for 8 bit read
                                               short _pregr2
494 00011B27 EB12
                               <1>
                                        jmp
495
                               <1> _pregr0:
496 00011B29 F7C300000080
                               <1>
                                                 ebx, PCI32
497 00011B2F 7507
                              <1>
                                         jnz short _pregr1
                                               ax, dx
498 00011B31 66ED
                               <1>
                                         in
                                                                        ; return 16 bits of data
499 00011B33 6689C2
                              <1>
                                         mov
                                                dx, ax
                                      jmp
500 00011B36 EB03
                              <1>
                                               short _pregr2
                              <1> _pregr1:
502 00011B38 ED
                               <1>
                                      in
                                               eax, dx
                                                                        ; return 32 bits of data
503 00011B39 89C2
                               <1>
                                         mov
                                               edx, eax
                               <1> _pregr2:
505 00011B3B 89D8
                                    mov
                               <1>
                                                eax, ebx
                                                                  ; restore eax
                                                eax, NOT PCI32 PCI16 ; clear out data size request
506 00011B3D 25FFFFFF3F
                               <1>
                                         and
507 00011B42 59
                               <1>
                                        pop ecx
508 00011B43 5B
                               <1>
                                        pop
509 00011B44 C3
                               <1>
                                        retn
                               <1>
510
                               <1> pciRegRead8:
511
512 00011B45 25FFFFFF3F
                               <1>
                                                   eax, NOT_PCI32_PCI16 ; set up 8 bit read size
                                           and
513 00011B4A EBAD
                               <1>
                                                   short pciRegRead; call generic PCI access
                                           jmp
                               <1>
                               <1> pciRegRead16:
515
516 00011B4C 25FFFFFF3F
                               <1>
                                                   eax, NOT_PCI32_PCI16 ; set up 16 bit read size
                                           and
517 00011B51 0D00000040
                                                   eax, PCI16
                               <1>
                                           or
                                                                        ; call generic PCI access
518 00011B56 EBA1
                               <1>
                                                   short pciRegRead
                                           jmp
                               <1>
520
                               <1> pciRegRead32:
521 00011B58 25FFFFFF3F
                               <1>
                                                   eax, NOT_PCI32_PCI16 ; set up 32 bit read size
                                                   eax, PCI\overline{3}2
522 00011B5D 0D00000080
                               <1>
                                           or
                                                                ; call generic PCI access
523 00011B62 EB95
                               <1>
                                                   pciRegRead
                                           jmp
524
                               <1>
                               <1> pciRegWrite:
525
526
                               <1>
                                        ; 03/04/2017 ('pci.asm', 29/11/2016)
527
                               <1>
528
                               <1>
                                         ; 8/16/32bit PCI writer
529
                                <1>
                                         ; Entry: EAX=PCI Bus/Device/fn/register number
530
                               <1>
                                                    BIT31 set if 32 bit access requested
531
                               <1>
532
                               <1>
                                                    BIT30 set if 16 bit access requested
533
                               <1>
                                                    otherwise defaults to 8bit read
534
                                <1>
                                                 DL/DX/EDX data to write depending on size
535
                               <1>
536
                                         ; Notel: this routine is meant to be called via pciRegWrite8,
                                <1>
537
                                             pciRegWrite16 or pciRegWrite32 as detailed below.
                               <1>
538
                                <1>
539
                                <1>
                                         ; Note2: don't attempt to write 32bits of data from a non dword
                                                aligned reg number. Likewise, don't do 16 bit writes from
540
                               <1>
                               <1>
541
                                                non word aligned reg #
                               <1>
542
543 00011B64 53
                               <1>
                                         push ebx
544 00011B65 51
                                        push ecx
                              <1>
                                        mov
                              <1>
545 00011B66 89C3
                                                  ebx, eax
                                                               ; save eax, edx
546 00011B68 89D1
                              <1>
                                          mov
                                                  ecx, edx
547 00011B6A 25FFFFFF3F
                              <1>
                                                eax, NOT_PCI32_PCI16
                                        and
                                                                      ; clear out data size request
                                        or
                                                                       ; make a PCI access request
                                                  eax, BIT31
548 00011B6F 0D00000080
                              <1>
                                                  al, ~3 ; NOT 3
549 00011B74 24FC
                               <1>
                                         and
                                                                       ; force index to be dword
550
                               <1>
                                                  dx, PCI INDEX PORT
551 00011B76 66BAF80C
                              <1>
                                         mov
552 00011B7A EF
                               <1>
                                         out dx, eax
                                                                        ; write PCI selector
553
                               <1>
554 00011B7B 66BAFC0C
                              <1>
                                                   dx, PCI_DATA_PORT
                                          mov
555 00011B7F 88D8
                                        mov
                              <1>
                                                   al, bl
                                                                        ; figure out which port to
556 00011B81 2403
                               <1>
                                           and
                                                   al, 3
                                         add
557 00011B83 00C2
                                                  dl, al
                              <1>
                                                                        ; write to
                              <1>
                                       test
559 00011B85 F7C300000C0
                              <1>
                                               ebx, PCI32+PCI16
560 00011B8B 7505
                               <1>
                                        jnz short _pregw0
561 00011B8D 88C8
                               <1>
                                         mov al, cl
                                                                        ; put data into al
562 00011B8F EE
                               <1>
                                         out dx, al
```

```
563 00011B90 EB12
                              <1>
                                       jmp
                                             short _pregw2
                               <1> _pregw0:
565 00011B92 F7C30000080
                              <1>
                                   test
                                              ebx, PCI32
                                        jnz short _pregw1
566 00011B98 7507
                              <1>
567 00011B9A 6689C8
                              <1>
                                        mov ax, cx
                                                                ; put data into ax
568 00011B9D 66EF
                              <1>
                                        out
                                              dx, ax
                                        jmp short _pregw2
                             <1>
569 00011B9F EB03
                              <1> _pregw1:
570
571 00011BA1 89C8
                              <1>
                                                                ; put data into eax
                                              eax, ecx
572 00011BA3 EF
                              <1>
                                        out
                                              dx, eax
573
                             <1> _pregw2:
                                    mov
                             <1>
<1>
574 00011BA4 89D8
                                                 eax, ebx
                                                                 ; restore eax
                                                 eax, NOT_PCI32_PCI16 ; clear out data size request
575 00011BA6 25FFFFFF3F
                                         and
576 00011BAB 89CA
                             <1>
                                       mov edx, ecx ; restore dx
                                    pop ecx
pop ebx
retn
                              <1>
577 00011BAD 59
578 00011BAE 5B
                              <1>
579 00011BAF C3
                              <1>
580
                              <1>
                              <1> pciRegWrite8:
581
                                                 eax, NOT PCI32_PCI16 ; set up 8 bit write size
582 00011BB0 25FFFFFF3F
                              <1>
                                    and
583 00011BB5 EBAD
                              <1>
                                          jmp short pciRegWrite ; call generic PCI access
                              <1>
585
                              <1> pciRegWrite16:
586 00011BB7 25FFFFFF3F
                              <1>
                                                 eax, NOT_PCI32_PCI16 ; set up 16 bit write size
587 00011BBC 0D00000040
                              <1>
                                                 eax, PCI16 ; call generic PCI access
                                          or
588 00011BC1 EBA1
                              <1>
                                          jmp short pciRegWrite
                              <1>
                              <1> pciRegWrite32:
590
591 00011BC3 25FFFFFF3F
                              <1>
                                                 eax, NOT_PCI32_PCI16 ; set up 32 bit write size
                                          and
592 00011BC8 0D00000080
                                                 eax, PCI32
                              <1>
                                                                      ; call generic PCI access
                                          or
593 00011BCD EB95
                              <1>
                                          jmp pciRegWrite
594
                               <1>
                              <1> init_codec:
595
                                   ; 05/06/2017
596
                              <1>
597
                               <1>
                                        ; 28/05/2017 - Erdogan Tan (Ref: KolibriOS, vt823x.asm)
598
                              <1>
                                    mov
599 00011BCF A1[486B0100]
                                             eax, [audio dev id]
                              <1>
                                    mov
600 00011BD4 B041
                              <1>
                                             al, VIA_ACLINK_CTRL
601 00011BD6 E86AFFFFF
                              <1>
                                       call pciRegRead8
602
                              <1>
                                       ; ?
                             603 00011BDB B040
604 00011BDD E863FFFFFF
605 00011BE2 F6C201
606 00011BE5 7508
                                       call reset_codec
jnc short _codec_ready_2 ; eax = 1
607 00011BE7 E80E000000
608 00011BEC 7306
609 00011BEE C3
                              <1> _codec_ready_1:
<1> mov eax
610
611 00011BEF B801000000
                                    mov eax, 1
612
                              <1> _codec_ready_2:
613 00011BF4 E886000000
                              <1> call codec_io_w16
614
                               <1> detect codec:
615 00011BF9 C3
                              <1>
                                       retn
                              <1>
616
617
                               <1> reset_codec:
618
                              <1> ; 16/04/2017
619
                               <1>
                                       ; 23/03/2017
620
                              <1>
                                      ; ('codec.asm')
                               <1>
                                        ; 12/11/2016 - Erdogan Tan (Ref: KolibriOS, vt823x.asm)
621
                                      mov eax, [audio_dev_id]
622 00011BFA A1[486B0100]
                              <1>
623 00011BFF B041
                              <1>
                                      mov al, VIA_ACLINK_CTRL
624 00011C01 B2E0
                                             mov dl, VIA ACLINK CTRL ENABLE + VIA ACLINK CTRL RESET + VIA ACLINK CTRL SYNC
                               <1>
                                       call pciRegWrite8
625 00011C03 E8A8FFFFFF
                              <1>
                              <1>
626
                                       call delay_100ms ; wait 100 ms
627 00011C08 E849000000
                              <1>
628
                              <1> _rc_cold:
                                    call
629 00011C0D E814000000
                              <1>
                                               cold reset
630 00011C12 7301
                              <1>
                                        jnc
                                               short _reset_codec_ok
631
                               <1>
                                       ; 16/04/2017
632
                               <1>
                                      ;xor eax, eax ; timeout error
633
                               <1>
634
                               <1>
                                           ;stc
635 00011C14 C3
                               <1>
                                        retn
636
                               <1>
                               <1> _reset_codec_ok:
637
638
                               <1>
                                     ; 01/09/2020
639
                               <1>
                                        ; 15/08/2020
640
                               <1>
                                       ; 27/07/2020
641
                               <1>
                                        ; also reset codec by using index control register 0 of AD1980 or ALC655
642
                               <1>
                                        ; (to fix line out -2 channels audio playing- problem on AD1980 codec)
643
                               <1>
644 00011C15 29C0
                               <1>
                                        sub eax, eax
645 00011C17 BA0000000
                                              edx, CODEC_RESET_REG ; 00h ; Reset register
                               <1>
                                        mov
                               <1>
                                        call codec write
646 00011C1C E8CA000000
                               <1>
647
                                        ; sub eax, eax
648
                               <1>
                                       ; 01/09/2020
649
                               <1>
650
                               <1>
                                        ; 15/08/2020
651
                               <1>
                                        ; AD1980 BugFix
652
                               <1>
                                        ; (set HPSEL -headphone amp to be driven from mixer- and
                                              CLDIS - center and LFE disable- bits)
653
                               <1>
                                       ;mov eax, 0C00h; HPSEL = bit 10, CLDIS = bit 11; 01/09/2020
;mov edx, CODEC_MISC_CRTL_BITS_REG; 76h; Misc Ctrl Bits; AD1980
654
                               <1>
                               <1>
655
656
                               <1>
                                        ;call codec_write
657
                               <1>
658 00011C21 31C0
                               <1>
                                         xor
                                                 eax, eax
                               <1>
                                          ;moval, VIA_ACLINK_C00_READY ; 1
660 00011C23 FEC0
                               <1>
                                          inc al
661 00011C25 C3
                               <1>
                                        retn
                               <1>
662
663
                               <1> cold_reset:
                                     ; 16/04/2017
664
                               <1>
                                        ; 23/03/2017
665
                               <1>
                               <1>
666
                               <1>
                                        ; 12/11/2016 - Erdogan Tan (Ref: KolibriOS, vt823x.asm)
667
```

```
<1>
668
                                          ;mov eax, [audio_dev_id]
669
                                <1>
                                          ;mov al, VIA ACLINK CTRL
670 00011C26 30D2
                                                dl, dl ; 0
                                <1>
                                          xor
671 00011C28 E883FFFFFF
                                <1>
                                          call pciRegWrite8
                                <1>
672
673 00011C2D E824000000
                                <1>
                                          call delay_100ms ; wait 100 ms
674
                                <1>
                                          ;; ACLink on, deassert ACLink reset, VSR, SGD data out
675
                                <1>
676
                                <1>
                                          ;; note - FM data out has trouble with non VRA codecs !!
677
                                <1>
678
                                <1>
                                          ;mov eax, [audio_dev_id]
679
                                                al, VIA ACLINK CTRL
                                <1>
                                          ;mov
                                                dl, VIA ACLINK CTRL INIT
680 00011C32 B2CC
                                <1>
                                          mov
681 00011C34 E877FFFFF
                                          call pciRegWrite8
                                <1>
                                <1>
682
683 00011C39 B910000000
                                          mov
                                                ecx, 16
                                <1>
                                                         ; total 2s
684
                                <1>
685
                                <1> _crst_wait:
                                         ;mov eax, [audio dev id]
686
                                <1>
                                          mov al, VIA_ACLINK_STAT
687 00011C3E B040
                                <1>
688 00011C40 E800FFFFF
                                <1>
                                          call pciRegRead8
                                <1>
690 00011C45 F6C201
                                           test
                                                    dl, VIA_ACLINK_C00_READY
                                <1>
691 00011C48 750B
                                <1>
                                          jnz
                                                    short _crst_ok
692
                                <1>
693 00011C4A 51
                                <1>
                                         push ecx
694 00011C4B E806000000
                                <1>
                                         call delay_100ms
695 00011C50 59
                                <1>
                                          pop
                                                ecx
                                <1>
697 00011C51 49
                                <1>
                                            dec
                                                    ecx
698 00011C52 75EA
                                <1>
                                           jnz
                                                    short _crst_wait
699
                                <1>
700
                                <1> _crst_fail:
701 00011C54 F9
                                <1>
                                <1> _crst_ok:
702
703 00011C55 C3
                                <1>
704
                                <1>
                                <1> delay_100ms:
705
                                      ; 29/05/2017
706
                                <1>
707
                                <1>
                                         ; 24/03/2017 ('codec.asm')
                                       ; wait 100 ms
mov ecx, 400 ; 400*0.25ms
708
                                <1>
709 00011C56 B990010000
                                <1>
                                <1> _delay_x_ms:
710
                                      call delay1 4ms
711 00011C5B E803000000
                                <1>
712 00011C60 E2F9
                                <1>
                                          loop_delay_x_ms
713 00011C62 C3
                                <1>
                                          retn
714
                                <1>
                                            delay1\_4ms - Delay for 1/4 millisecond.
715
                                <1>;
716
                                <1>;
                                            1mS = 1000us
717
                                <1>;
                                            Entry:
718
                                <1>;
                                             None
719
                                <1>;
                                            Exit:
                                <1>;
720
                                            None
721
                                <1>;
722
                                <1>;
                                            Modified:
723
                                <1>;
                                             None
724
                                <1>;
725
                                <1>
                                          ; 29/05/2017
726
                                <1>
                                          ; 23/04/2017
727
                                <1>
                                         ; 05/03/2017 (TRDOS 386)
728
                                <1>
729
                                <1>
                                          ; ('UTILS.ASM')
730
                                <1> delay1_4ms:
                                            push
731 00011C63 50
                                <1>
                                                    eax
732 00011C64 51
                                <1>
                                            push
                                                   ecx
733 00011C65 B110
                                <1>
                                            mov cl, 16
                                                             ; close enough.
                                <1>
735 00011C67 E461
                                                al, PORTB ; 61h
                                <1>
                                          in
736
                                <1>
737 00011C69 2410
                                <1>
                                                al, REFRESH STATUS ; 10h
                                          and
                                                            ; Start toggle state
738 00011C6B 88C5
                                <1>
                                          mov
                                                ch, al
739
                                <1> _d4ms1:
740 00011C6D E461
                                                al, PORTB
                                                             ; Read system control port
                                <1>
                                          in
741
                                <1>
                                                al, REFRESH_STATUS; Refresh toggles 15.085 microseconds
742 00011C6F 2410
                                <1>
                                          and
743 00011C71 38C5
                                <1>
                                          cmp
                                                ch, al
744 00011C73 74F8
                                <1>
                                                short _d4ms1 ; Wait for state change
                                          jе
745
                                <1>
746 00011C75 88C5
                                <1>
                                                ch, al
                                                           ; Update with new state
747 00011C77 FEC9
                                <1>
                                          dec
                                                cl
                                                short _d4ms1
748 00011C79 75F2
                                <1>
749
                                <1>
750 00011C7B F8
                                          clc ; 29/05/2017
                                <1>
751
                                <1>
752 00011C7C 59
                                <1>
                                            pop
                                                    ecx
753 00011C7D 58
                                <1>
                                            pop
                                                    eax
754 00011C7E C3
                                <1>
                                            retn
755
                                <1>
                                <1>; 10/04/2017 (TRDOS 386)
756
757
                                <1> ; 12/11/2016
758
                                <1>
                                <1> codec_io_w16: ;w32
759
                                         -, ('codec.asm')
760
                                <1>
                                          mov dx, [audio_io_base]
761 00011C7F 668B15[466B0100]
                                <1>
                                           add dx, VIA_REG_AC97
762 00011C86 6681C28000
                                <1>
                                          out dx, eax
763 00011C8B EF
                                <1>
764 00011C8C C3
                                <1>
                                          retn
765
                                <1>
                                <1> codec io r16: ;r32
766
                                       ; ('codec.asm')
767
                                <1>
768 00011C8D 668B15[466B0100]
                                           mov dx, [audio_io_base]
                               <1>
                                                  dx, VIA REG_AC97
769 00011C94 6681C28000
                                <1>
                                            add
770 00011C99 ED
                                            in eax, dx
                                <1>
771 00011C9A C3
                                <1>
                                            retn
772
                                <1>
```

```
773
                                <1> ctrl_io_w8:
                                     ; ('codec.asm')
                                <1>
775 00011C9B 660315[466B0100]
                                          add
                                <1>
                                                dx, [audio_io_base]
776 00011CA2 EE
                                <1>
                                           out dx, al
777 00011CA3 C3
                                <1>
                                           retn
778
                                <1>
                                <1> ctrl_io_r8:
779
                                     ; ('codec.asm')
780
                                <1>
781 00011CA4 660315[466B0100]
                               <1>
                                          add
                                                 dx, [audio_io_base]
                                           in al, dx
782 00011CAB EC
                                <1>
783 00011CAC C3
                                <1>
                                           retn
784
                                <1>
                                <1> ctrl_io_w32:
785
                                     ; ('codec.asm')
786
                               <1>
787 00011CAD 660315[466B0100]
                                          add
                               <1>
                                                  dx, [audio_io_base]
788 00011CB4 EF
                                <1>
                                           out dx, eax
789 00011CB5 C3
                                <1>
                                           retn
790
                                <1>
                                <1> ctrl io r32:
791
792
                                <1> ; ('codec.asm')
793 00011CB6 660315[466B0100]
                               <1>
                                          add dx, [audio_io_base]
794 00011CBD ED
                                       in eax, dx
                                <1>
795 00011CBE C3
                                <1>
                                          retn
796
                                <1>
797
                                <1> codec_read:
798
                                <1>
                                       ; 12/11/2016 - Erdogan Tan (Ref: KolibriOS, vt823x.asm)
799
                                <1>
                                           ; Use only primary codec.
800
                                <1>
                                           ; eax = register
801 00011CBF C1E010
                                <1>
                                           shl
                                                   eax, VIA_REG_AC97_CMD_SHIFT
802 00011CC2 0D00008002
                                                   eax, VIA_REG_AC97_PRIMARY_VALID + VIA_REG_AC97_READ
                               <1>
                                          or
803
                               <1>
804 00011CC7 E8B3FFFFFF
                                <1>
                                         call
                                               codec_io_w16
805
                                <1>
806
                               <1>
                                               ; codec_valid
807 00011CCC E831000000
                               <1>
                                       call codec_check_ready
                                         jnc short _cr_ok
808 00011CD1 7301
                               <1>
                               <1>
810 00011CD3 C3
                               <1>
                                         retn
811
                                <1>
                               <1> _cr_ok:
812
                                     ; wait 25 ms
813
                               <1>
814 00011CD4 B950000000
                               <1>
                                         mov ecx, 80; (100*0.25 ms)
                               <1> _cr_wloop:
815
                                        call delay1 4ms
816 00011CD9 E885FFFFF
                               <1>
817 00011CDE E2F9
                               <1>
                                         loop _cr_wloop
818
                               <1>
819 00011CE0 E8A8FFFFFF
                               <1>
                                           call codec io r16
                                                   eax, \overline{0}FF\overline{F}Fh
820 00011CE5 25FFFF0000
                                           and
                               <1>
821 00011CEA C3
                               <1>
                                           retn
822
                                <1>
                                <1> codec_write:
823
                                     ; 12/11/2016 - Erdogan Tan (Ref: KolibriOS, vt823x.asm)
824
                                <1>
825
                                <1>
                                           ; Use only primary codec.
826
                                <1>
827
                                <1>
                                         ; eax = data (volume)
828
                                <1>
                                         ; edx = register (mixer register)
                                <1>
830 00011CEB C1E210
                                <1>
                                         shl
                                                 edx, VIA_REG_AC97_CMD_SHIFT
831
                                <1>
832 00011CEE C1E000
                               <1>
                                                   eax, VIA_REG_AC97_DATA_SHIFT ; shl eax, 0
                                           shl
833 00011CF1 09C2
                               <1>
                                                   edx, eax
                                           or
                                <1>
                                                   eax, VIA_REG_AC97_CODEC_ID_PRIMARY
835 00011CF3 B800000000
                               <1>
                                           mov
836 00011CF8 C1E01E
                               <1>
                                           shl
                                                   eax, VIA_REG_AC97_CODEC_ID_SHIFT
837 00011CFB 09D0
                                <1>
                                                   eax, edx
                                           or
838
                                <1>
839 00011CFD E87DFFFFFF
                                <1>
                                           call
                                                   codec io w16
                                                   [codec.regs+esi], ax
                                <1>
840
                                           ;mov
841
                                <1>
                                           ;call
842
                                <1>
                                                     codec_check_ready
843
                                <1>
                                               ;retn
844
                                <1>
                                         ;jmp short _codec_check_ready
                                <1>
845
846
                                <1> codec_check_ready:
                                        ; 12/11/2016 - Erdogan Tan (Ref: KolibriOS, vt823x.asm)
847
                                <1>
848
                                <1>
                                <1> _codec_check_ready:
850 00011D02 B914000000
                                <1>
                                       mov ecx, 20
                                                           ; total 2s
                                <1> ccr wait:
851
852 00011D07 51
                                        push ecx
                                <1>
853
                                <1>
854 00011D08 E880FFFFFF
                                <1>
                                           call
                                                   codec io r16
                                                   eax, VIA_REG_AC97_BUSY
855 00011D0D A90000001
                               <1>
                                           test
856 00011D12 740B
                                <1>
                                                   short _ccr_ok
                                           jΖ
857
                                <1>
858 00011D14 E83DFFFFFF
                                         call delay_100ms
                                <1>
                               <1>
860 00011D19 59
                               <1>
                                         pop
                                                ecx
861
                               <1>
862 00011D1A 49
                               <1>
                                         dec
                                                 ecx
863 00011D1B 75EA
                               <1>
                                         jnz short _ccr_wait
                                <1>
865 00011D1D F9
                               <1>
                                           stc
866 00011D1E C3
                               <1>
                                           retn
867
                               <1>
                               <1> ccr ok:
868
                                        pop ecx and eax, 0FFFFh
869 00011D1F 59
                               <1>
870 00011D20 25FFFF0000
                               <1>
871 00011D25 C3
                               <1>
                                         retn
872
                               <1>
873
                               <1> codec_config:
                                <1> ; 10/06/2017
874
875
                                <1>
                                         ; 29/05/2017
                                       ; 24/04/2017
876
                                <1>
877
                                <1>
                                      ; 21/04/2017
```

```
878
                                <1>
                                        ; 16/04/2017 (TRDOS 386 Kernel)
879
                                <1>
                                         ; 15/11/2016 ('codec.asm', 'player.com')
                                         ; 14/11/2016
880
                                <1>
881
                                <1>
                                         ; 12/11/2016 - Erdogan Tan
882
                                <1>
                                                    (Ref: KolibriOS, 'setup_codec', codec.inc)
883
                                <1>
884 00011D26 B802020000
                               <1>
                                         mov
                                                eax, 0202h
885 00011D2B 66A3[766B0100]
                                               [audio_master_volume], ax
                               <1>
                                        mov
886 00011D31 66B81F1F
                               <1>
                                               ax, 1F1Fh; 31,31
                                         mov
                                               edx, CODEC_MASTER_VOL_REG ; 02h ; Line Out
887 00011D35 BA02000000
                               <1>
                                         mov
888 00011D3A E8ACFFFFF
                               <1>
                                         call codec_write
889
                               <1>
                                         ;jc
                                               short cconfig error
890
                               <1>
891
                               <1>
                                         ;mov
                                                eax, 0202h
892 00011D3F 66B80202
                                                ax, 0202h
                               <1>
                                        mov
893 00011D43 BA18000000
                               <1>
                                         mov
                                                edx, CODEC_PCM_OUT_REG ; 18h ; Wave Output (Stereo)
894 00011D48 E89EFFFFF
                               <1>
                                         call codec write
895
                               <1>
                                       ;jc short cconfig_error
896
                               <1>
897
                               <1>
                                                eax, 0202h
                                         ;mov
898 00011D4D 66B80202
                               <1>
                                         mov
                                               ax, 0202h
899 00011D51 BA04000000
                                               edx, CODEC_AUX_VOL ; 04h ; CODEC_HP VOL REG ; HeadPhone
                               <1>
                                         mov
                                         call codec_write
900 00011D56 E890FFFFF
                               <1>
                               <1>
                                         ;jc short cconfig_error
902
                                <1>
903
                                <1>
                                         ;mov
                                                eax, 08h
                               <1>
                                         ;mov ax, 08h
                                         mov ax, 8008h; Mute
905 00011D5B 66B80880
                               <1>
906 00011D5F BA0C000000
                               <1>
                                               edx, OCh ; AC97_PHONE_VOL ; TAD Input (Mono)
                                         mov
                                         call codec write
907 00011D64 E882FFFFF
                               <1>
908
                               <1>
                                         ;jc short cconfig_error
909
                                <1>
                                               eax, 0808h
910
                               <1>
                                         ;mov
                                         mov ax, 0808h
911 00011D69 66B80808
                               <1>
                                          mov edx, CODEC LINE IN VOL REG ; 10h ; Line Input (Stereo)
912 00011D6D BA10000000
                               <1>
913 00011D72 E874FFFFF
                               <1>
                                         call codec_write
                                         ;jc short cconfig_error
                               <1>
915
                               <1>
916
                               <1>
                                                eax, 0808h
                                                 ax, 0808h
917 00011D77 66B80808
                               <1>
                                         mov
918 00011D7B BA12000000
                               <1>
                                         mov edx, CODEC_CD_VOL_REG ; 12h ; CR Input (Stereo)
                                         call codec_write
919 00011D80 E866FFFFF
                               <1>
920
                               <1>
                                         ;jc short cconfig_error
921
                               <1>
                                               eax, 0808h
922
                               <1>
                                         ;mov
923 00011D85 66B80808
                               <1>
                                         mov
                                                 ax, 0808h
                                         mov edx, CODEC_AUX_VOL_REG ; 16h ; Aux Input (Stereo)
924 00011D89 BA16000000
                               <1>
                                         ;call codec_write
925
                               <1>
926
                                <1>
                                         ;;jc short cconfig_error
                                         jmp codec_write; 10/06/2017
927 00011D8E E958FFFFFF
                               <1>
928
                                <1>
929
                                <1>;
                                         ; Extended Audio Status (2Ah)
                                         mov eax, CODEC_EXT_AUDIO_CTRL_REG ; 2Ah
930
                                <1>;
931
                                <1>;
                                         call codec_read
                                         and eax, 0ffffh - 2 ; clear DRA (BIT1); or eax, 1 ; set VRA (BIT0) or eax, 5 ; VRA (BIT0) & S/PDIF (BIT2); 14/11/2016
932
                                <1>;
933
                                <1>;
934
                                <1>;
                                <1>;
                                              edx, CODEC_EXT_AUDIO_CTRL_REG
935
                                         mov
936
                                <1>;
                                         call codec_write
                                <1>;
                                         ;jc short cconfig_error
937
938
                                <1>;
939
                                <1> ;set sample rate:
940
                                         ;movzx eax, word [audio_freq]
                                <1>;
941
                                <1>;
                                         mov ax, [audio_freq]
942
                                <1>;
                                               edx, CODEC PCM FRONT DACRATE REG ; 2Ch ; PCM Front DAC Rate
                                         mov
943
                                <1>;
                                         ;call codec_write
944
                                <1>;
                                         ;retn
945
                                               codec_write
                                <1>;
                                         jmp
946
                                <1>
947
                                <1> ;cconfig_error:
                                <1> ;
948
949
                                <1>
950
                                <1> vt8233_int_handler:
                                     ; 27/07/2020
951
                                <1>
952
                                <1>
                                         ; 22/07/2020
                                         ; Interrupt Handler for VIA VT8237R Audio Controller
953
                                <1>
954
                                <1>
                                         ; Note: called by 'dev IRQ service'
955
                                <1>
                                        ; 14/10/2017
956
                                <1>
                                         ; 09/10/2017, 10/10/2017, 12/10/2017
957
                                         ; 13/06/2017
                                <1>
                                         ; 21/04/2017 (TRDOS 386 kernel, 'audio.s')
958
                                <1>
                                         ; 24/03/2017 - 'PLAYER.COM' ('player.asm')
959
                                <1>
960
                                <1>
                                <1>
                                         ; push eax ; * must be saved !
961
962
                                <1>
                                         ;push edx
963
                                <1>
                                         ;push ecx
964
                                <1>
                                         ; push ebx ; * must be saved !
965
                                <1>
                                         ;push esi
966
                                <1>
                                         ;push edi
967
                                <1>
968
                                <1>
                                         ;cmp byte [audio_busy], 1
969
                                <1>
                                         ;jnb short _ih0 ; 09/10/2017
970
                                <1>
971
                                <1>
                                         ;mov byte [audio_flag_eol], 0
972
                                <1>
973 00011D93 66BA0000
                                <1>
                                          mov
                                                   dx, VIADEV_PLAYBACK + VIA_REG_OFFSET_STATUS
974 00011D97 E808FFFFF
                                <1>
                                                  ctrl_io_r8
                                          call
975
                                <1>
976 00011D9C A880
                                <1>
                                         test
                                                 al, VIA_REG_STAT_ACTIVE
977 00011D9E 7417
                                                   short _ih0 ; 09/10/2017
                               <1>
                                          jz
978
                               <1>
979 00011DA0 2407
                                <1>
                                                   al, VIA_REG_STAT_EOL + VIA_REG_STAT_FLAG + VIA_REG_STAT_STOPPED
                                          and
980 00011DA2 A2[756B0100]
                               <1>
                                         mov [audio_flag_eol], al
981 00011DA7 740E
                                <1>
                                          jz short _ih0 ; 09/10/2017
                                <1>
982
```

```
983
                                         ; 09/10/2017
 984
                                 <1>
                                          ;mov byte [audio busy], 1
 985
                                 <1>
 986 00011DA9 803D[746B0100]01
                                <1>
                                                 byte [audio_play_cmd], 1
                                                 short _ih1 ; 10/10/2017
 987 00011DB0 7315
                                 <1>
                                           jnb
 988
                                 <1>
 989 00011DB2 E84A000000
                                <1>
                                           call channel_reset
                                 <1> _ih0:
 990
991
                                 <1>
                                          ; 09/10/2017
 992 00011DB7 A0[756B0100]
                                <1>
                                                    al, [audio_flag_eol] ;; ack ;;
                                           mov
                                             mov
                                                     dx, VIADEV_PLAYBACK + VIA_REG_OFFSET_STATUS
 993 00011DBC 66BA0000
                                <1>
 994 00011DC0 E8D6FEFFFF
                                 <1>
                                             call
                                                    ctrl io w8
 995 00011DC5 EB39
                                <1>
                                           jmp short _ih4
996
                                <1> _ih1:
                                <1> vt8233_tuneLoop:
 997
998 00011DC7 A0[756B0100]
                                <1>
                                            mov
                                                     al, [audio_flag_eol] ;; ack ;;
                                                     dx, VIADEV PLAYBACK + VIA_REG_OFFSET_STATUS
999 00011DCC 66BA0000
                                <1>
                                             mov
1000 00011DD0 E8C6FEFFFF
                                            call
                                 <1>
                                                    ctrl_io_w8
1001
                                 <1>
                                          ; 22/07/2020
1002
                                 <1>
                                          ;; 12/10/2017
1003
                                 <1>
                                          ;mov byte [audio flag], 0 ; Reset
1004
                                 <1>
1005
                                 <1>
1006
                                 <1>
                                          ; 10/10/2017
                                           ; 09/10/2017
1007
                                 <1>
1008
                                 <1>
                                           ;test byte [audio_flag_eol], VIA_REG_STAT_FLAG
1009
                                 <1>
                                           ;jz short _ih2 ; EOL
1010
                                 <1>
1011
                                 <1>
                                          ; 22/07/2020
1012
                                 <1>
                                           ; 14/10/2017
1013
                                 <1>
                                           ;test byte [audio_flag_eol], VIA_REG_STAT_EOL
1014
                                 <1>
                                           ;jnz short _ih2 ; EOL
1015
                                 <1>
                                                          ; (Half Buffer 2 has been completed
1016
                                 <1>
                                                           ; and Half Buffer 1 will be played.)
1017
                                 <1>
1018
                                 <1>
                                           ; FLAG
1019
                                 <1>
                                           ; (Half Buffer 1 has been completed
1020
                                 <1>
                                           ; and Half Buffer 2 will be played.)
1021
                                 <1>
1022
                                 <1>
                                           ; 14/10/2017
                                          ;; (Continue to play.)
1023
                                 <1>
1024
                                 <1>
                                           ; mov al, VIA REG CTRL INT
                                                 ; or al, VIA_REG_CTRL_START
1025
                                 <1>
                                                  ; mov dx, VIADEV PLAYBACK + VIA REG OFFSET CONTROL
1026
                                 <1>
1027
                                 <1>
                                            ;call
                                                       ctrl_io_w8
1028
                                 <1>
                                           ; 12/10/2017
                                           ;mov byte [audio flag], 1
1029
                                 <1>
1030
                                 <1>
1031
                                 <1>
                                           ; 22/07/2020
1032
                                 <1>
                                           ;inc byte [audio_flag] ; = 1
1033
                                 <1> _ih2:
1034
                                 <1>
                                          ; 10/10/2017
1035 00011DD5 8B3D[606B0100]
                                 <1>
                                           mov edi, [audio_dma_buff]
1036 00011DDB 8B0D[646B0100]
                                 <1>
                                           mov ecx, [audio_dmabuff_size]
1037 00011DE1 D1E9
                                 <1>
                                           shr ecx, 1; dma buff size / 2 = half buffer size
1038
                                 <1>
1039
                                 <1>
                                           ; 22/07/2020
1040
                                           ; 12/10/2017
                                 <1>
1041
                                 <1>
                                           ;cmp byte [audio_flag], 0
1042
                                                short _ih3 ; Playing Half Buffer 2 (Current: FLAG)
                                 <1>
                                           ;ja
1043
                                 <1>
1044
                                 <1>
                                           ; 27/07/2020
                                           ; 22/07/2020
1045
                                 <1>
1046 00011DE3 F605[686B0100]01
                                <1>
                                           test byte [audio_flag], 1 ; Current flag value
1047 00011DEA 7402
                                 <1>
                                                 short _ih3 ; Half Buffer 1 must be filled
1048
                                 <1>
1049
                                 <1>
                                           ; Half Buffer 2 must be filled
1050 00011DEC 01CF
                                 <1>
                                           add edi, ecx
1051
                                 <1> ih3:
1052
                                 <1>
                                           ; Update half buffer 2 while playing half buffer 1
                                           ; Update half buffer 1 while playing half buffer 2
1053
                                 <1>
1054
                                 <1>
1055 00011DEE 8B35[586B0100]
                                 <1>
                                           mov
                                                  esi, [audio_p_buffer] ; phy addr of audio buff
1056 00011DF4 C1E902
                                 <1>
                                               ecx, 2 ; half buff size / 4
1057 00011DF7 F3A5
                                 <1>
                                                 movsd
                                           rep
1058
                                 <1>
                                 <1>
                                           ; switch flag value ;
1060 00011DF9 8035[686B0100]01
                                 <1>
                                          xor byte [audio_flag], 1
                                           ; 12/10/2017
1061
                                 <1>
                                           ; [audio_flag] = 0 : Playing dma half buffer 2
1062
                                 <1>
                                                          ; Next buffer (to update) is dma half buff 1
1063
                                 <1>
                                                         = 1 : Playing dma half buffer 1
1064
                                  <1>
1065
                                 <1>
                                                           ; Next buffer (to update) is dma half buff 2
                                 <1> ih4:
1066
                                           ; 28/05/2017
1067
                                 <1>
1068
                                 <1>
                                           ;mov byte [audio_busy], 0 ; 09/10/2017
1069
                                 <1>
1070
                                 <1>
                                           ;pop edi
1071
                                 <1>
                                           ;pop
                                                 esi
1072
                                 <1>
                                                ebx ; * must be restored !
                                           ;pop
1073
                                 <1>
                                           ;pop ecx
1074
                                 <1>
                                                 edx
                                           ;pop
1075
                                                 eax ; * must be restored !
                                 <1>
                                           ;pop
1076
                                 <1>
1077 00011E00 C3
                                 <1>
                                           retn
1078
                                 <1>
1079
                                 <1> channel reset:
                                          ; 24/06/2017
1080
                                 <1>
1081
                                 <1>
                                           ; 29/05/2017
                                           ; 23/03/2017
1082
                                 <1>
1083
                                 <1>
                                          ; 14/11/2016 - Erdogan Tan
1084
                                 <1>
                                           ; 12/11/2016 - Erdogan Tan (Ref: KolibriOS, vt823x.asm)
1085 00011E01 BA01000000
                                             mov edx, VIA_REG_OFFSET_CONTROL
                                 <1>
                                             ;mov eax, VIA REG CTRL PAUSE + VIA REG CTRL TERMINATE + VIA REG CTRL RESET
                                 <1>
1087 00011E06 B848000000
                                 <1>
                                             mov eax, VIA REG CTRL PAUSE + VIA REG CTRL TERMINATE ; 24/06/2017
```

<1>

```
1089
                                <1>
                                        ;mov edx, VIA_REG_OFFSET_CONTROL
;call ctrl_io_r8
1090
                               <1>
1091
                               <1>
1092
                                <1>
1093
                               <1>
                                         ; wait for 50 ms
                                       mov ecx, 160; (200*0.25 ms); 29/05/2017
1094 00011E10 B9A000000
                               <1>
                               <1> _ch_rst_wait:
1095
                                     call delay1_4ms
1096 00011E15 E849FEFFFF
                               <1>
1097 00011E1A 49
                               <1>
                                         dec ecx
1098 00011E1B 75F8
                               <1>
                                       jnz short _ch_rst_wait
1099
                               <1>
                        1100
1101 00011E1D BA01000000
1102 00011E22 31C0
1103 00011E24 E872FEFFFF
                               <1>
                                     ; clear interrupts
mov edx, VIA_REG_OFFSET_STATUS
1105
                               <1>
1106 00011E29 BA00000000
                               <1>
                                       mov eax, 3
1107 00011E2E B803000000
                               <1>
1108 00011E33 E863FEFFFF
                                        call ctrl_io_w8
                               <1>
                                <1>
                                        ;mov edx, VIA_REG_OFFSET_CURR_PTR
1110
                                <1>
1111
                                <1>
                                       ;xor eax, eax
1112
                                <1>
                                         ;call ctrl_io_w32
1113
                                <1>
1114 00011E38 C3
                                <1>
                                          retn
1115
                                <1>
1116
                                <1> vt8233_stop: ; 22/04/2017
1117 00011E39 C605[746B0100]00
                               <1> mov byte [audio_play_cmd], 0; stop!
1118
                                <1> _tlp2:
                                       ; 24/06/2017
1119
                                <1>
                                           ; finished with song, stop everything
1120
                                <1>
1121
                                <1>
                                       ;mov al, VIA_REG_CTRL_INT
                                        ;or al, VIA_REG_CTRL_TERMINATE
1122
                                <1>
                                      ;or al, VIA_REG_CTRL_TERMINATE
;mov dx, VIADEV_PLAYBACK + VIA_REG_OFFSET_CONTROL
1123
                                <1>
                                         ;call
                                                    ctrl io w8
1124
                                <1>
1125
                                <1>
1126
                                <1>
                                          ;call
                                                     channel_reset
1127
                                <1>
                                        ;retn
1128
                                <1>
1129 00011E40 EBBF
                                <1>
                                        jmp short channel_reset
1130
                                <1>
1131
                                <1> set_vt8233_bdl: ; Set VT8237R Buffer Descriptor List
                                     ; 22/07/2020 - TRDOS 386 v2.0.2
1132
                                <1>
1133
                                <1>
                                         ; 28/05/2017
                                         ; 21/04/2017 (TRDOS 386 kernel, 'audio.s')
1134
                                <1>
                                       ; 24/03/2017 - 'PLAYER.COM' ('via_wav.asm' - 29/11/2016)
1135
                                <1>
1136
                                <1>
1137
                                <1>
                                       ; eax = dma buffer address = [audio_DMA_buff]
1138
                               <1>
                                       ; ecx = dma buffer buffer size = [audio_dmabuff_size]
1139
                                <1>
1140 00011E42 D1E9
                                        shr ecx, 1 ; dma half buffer size
                               <1>
1141 00011E44 89CE
                               <1>
                                       mov esi, ecx
1142
                               <1>
                                        mov
1143 00011E46 BF[7C6B0100]
                                                  edi, audio_bdl_buff ; get BDL address
                               <1>
                                                ecx, 32 / \frac{1}{2}
                                                                        ; make 32 entries in BDL
1144 00011E4B B91000000
                               <1>
1145
                               <1>
                                       jmp short s_vt8233_bdl1
1146 00011E50 EB05
                                <1>
1147
                               <1>
                               <1> s_vt8233_bd10:
1148
                                       ; set buffer descriptor 0 to start of data file in memory
1149
                                <1>
1150
                               <1>
1151 00011E52 A1[606B0100]
                               <1>
                                        mov eax, [audio_dma_buff] ; Physical address of DMA buffer
                                <1>
1152
1153
                                <1> s_vt8233_bdl1:
1154 00011E57 AB
                               <1>
                                       stosd
                                                                 ; store dmabuffer1 address
1155
                                <1>
1156 00011E58 89C2
                               <1>
                                         mov edx, eax
1157
                                <1>
                                <1>; VIA VT8235.PDF: (Page 110) (Erdogan Tan, 29/11/2016)
1158
1159
                                <1>
                                       ;
                                               Audio SGD Table Format
1160
                                <1>
                                         ;
1161
                                <1>
                                               63 62 61-56 55-32 31-0
                                <1>
1162
1163
                                <1>
1164
                                <1>
                                               EOL FLAG -reserved- Base Base
1165
                                <1>
                                                               Count Address
                                                                 [23:0] [31:0]
1166
                                <1>
                                <1>
                                               EOL: End Of Link.
1167
                                               1 indicates this block is the last of the link.
1168
                                <1>
                                                    If the channel "Interrupt on {\tt EOL''} bit is set, then
                                <1>
1170
                                <1>
                                                    an interrupt is generated at the end of the transfer.
1171
                                <1>
1172
                                <1>
                                                FLAG: Block Flag. If set, transfer pauses at the end of this
                                                     block. If the channel "Interrupt on FLAG" bit is set,
1173
                                <1>
                                                      then an interrupt is generated at the end of this block.
1174
                                <1>
                                         ;
1175
                                <1>
1176 00011E5A 89F0
                                <1>
                                                eax, esi ; DMA half buffer size
                                         mov
1177 00011E5C 01C2
                                <1>
                                         add
                                               edx, eax
1178 00011E5E 0D00000040
                                <1>
                                         or
                                                eax, FLAG
                                <1>
                                         ;or
                                                eax, EOL
1180 00011E63 AB
                                <1>
                                         stosd
1181
                                <1>
                                <1> ; 2nd buffer:
1182
1183
                                <1>
1184 00011E64 89D0
                                <1>
                                           mov eax, edx; Physical address of the 2nd half of DMA buffer
1185 00011E66 AB
                                <1>
                                         stosd
                                                     ; store dmabuffer2 address
1186
                                <1>
1187
                                <1>; set length to [audio dmabuff size]/2
1188
                                <1>; Set control (bits 31:16) to BUP, bits 15:0=number of samples
1189
                                <1>;
1190 00011E67 89F0
                                <1>
                                         mov
                                              eax, esi ; DMA half buffer size
1191
                                <1>
                                         ; 22/07/2020
1192
                                <1>
                                         ;or eax, EOL
```

call ctrl io w8

<1>

1088 00011E0B E88BFEFFFF

```
1193 00011E69 0D00000040
                                <1>
                                          or
                                                eax, FLAG
1194 00011E6E AB
                                <1>
                                          stosd
1195
                                <1>
1196 00011E6F E2E1
                                <1>
                                                  s_vt8233_bd10
                                          loop
1197
                                 <1>
                                          ; 22/07/2020
1198
                                 <1>
1199 00011E71 814FFC00000080
                                <1>
                                          or dword [edi-4], EOL
1200
                                <1>
1201 00011E78 C3
                                <1>
1202
                                 <1>
1203
                                 <1> vt8233_start_play:
                                        ; 01/09/2020
1204
                                 <1>
                                          ; 22/07/2020
1205
                                 <1>
                                        ; start to play audio data via VT8233 audio controller
1206
                                 <1>
                                        ; 13/06/2017
1207
                                 <1>
                                          ; 10/06/2017
1208
                                 <1>
                                          ; 24/04/2017
1209
                                 <1>
                                         ; 21/04/2017 (TRDOS 386 kernel, 'audio.s')
1210
                                 <1>
1211
                                 <1>
                                          ; 24/03/2017 - 'PLAYER.COM' ('via_wav.asm' - 29/11/2016)
1212
                                 <1>
                                          ; write buffer descriptor list address
1213
                                 <1>
                                          ; Extended Audio Status (2Ah)
1214
                                 <1>
1215 00011E79 B82A000000
                                          mov eax, CODEC_EXT_AUDIO_CTRL_REG ; 2Ah
                                <1>
1216 00011E7E E83CFEFFFF
                                <1>
                                          call codec_read
                                          and eax, 0FFFFh - 2
                                                                          ; clear DRA (BIT1)
1217 00011E83 25FDFF0000
                                <1>
1218
                                <1>
                                          ;or
                                                  eax, 1
                                                                          ; set VRA (BITO)
                                        or eax, 5; or al, 5;
                                <1>
                                                             ; VRA (BITO) & S/PDIF (BIT2) ; 14/11/2016
1220 00011E88 0C05
                                <1>
1221
                                 <1>
                                          ; 01/09/2020
                                          ; or eax, 3805h; AD1980 (PRK, PRJ, PRI = 1 .. only front DAC)
1222
                                 <1>
1223
                                 <1>
                                          ; 01/09/2020
1224
                                 <1>
                                          ;mov edx, CODEC_EXT_AUDIO_CTRL_REG
                                          ;cmp word [audio_freq], OBB80h; 48 kHz
1225
                                 <1>
1226
                                 <1>
                                          ;jne short set_extd_audio_status_1
                                          ; and al, OFEh; disable VRA bit (set sample rate to 48000 Hz)
1227
                                 <1>
1228
                                <1>
                                          ;jmp short set_extd_audio_status_2
                                <1>; set extd audio status 1:
1230 00011E8A BA2A000000
                                <1>
                                      mov edx, CODEC_EXT_AUDIO_CTRL_REG
1231 00011E8F E857FEFFFF
                                <1>
                                          call codec_write
                                          ;jc short cconfig_error
1232
                                <1>
1233
                                <1>
1234
                                <1> set_sample_rate:
1235
                                <1> ;movzx eax, word [audio_freq]
1236 00011E94 66A1[726B0100]
                                <1>
                                          mov ax, [audio_freq]
1237 00011E9A BA2C000000
                                <1>
                                         mov
                                               edx, CODEC_PCM_FRONT_DACRATE_REG ; 2Ch ; PCM Front DAC Rate
1238
                                <1> ;set_extd_audio_status_2:
1239 00011E9F E847FEFFFF
                                <1>
                                         call codec_write
1240
                                <1>
1241
                                 <1>
                                          ; 01/09/2020
                                          ; set AD1980 MCB register (Index 76h) to 0C00h
1242
                                 <1>
1243
                                 <1>
                                          ; (CLDIS, HPSEL)
1244
                                 <1>
                                          ;mov ax, 0C00h
                                          ;mov edx, CODEC_MISC_CRTL_BITS_REG ; 76h
1245
                                 <1>
1246
                                 <1>
                                                             ; Miscellaneous Control Bit Register
1247
                                 <1>
                                          ;call codec_write
1248
                                 <1>
                                 <1>
1250 00011EA4 B8[7C6B0100]
                                 <1>
                                          mov eax, audio_bdl_buff
1251
                                 <1>
                                          ; 12/11/2016 - Erdogan Tan
1252
                                <1>
1253
                                <1>
                                          ; (Ref: KolibriOS, vt823x.asm, 'create_primary_buff')
                                          mov edx, VIADEV_PLAYBACK + VIA_REG_OFFSET TABLE PTR
1254 00011EA9 BA0400000
                                 <1>
1255 00011EAE E8FAFDFFFF
                                <1>
                                           call ctrl_io_w32
1256
                                <1>
1257
                                <1>
                                          ;call codec_check_ready
1258
                                <1>
1259 00011EB3 66BA0200
                                <1>
                                          mov dx, VIADEV_PLAYBACK + VIA_REG_OFS_PLAYBACK_VOLUME_L
                                <1>
1260
                                          ;moveax, 2; 31
1261 00011EB7 B01F
                                <1>
                                          mov al, 31
1262 00011EB9 2A05[766B0100]
                                <1>
                                          sub al, [audio_master_volume_1]
1263 00011EBF E8D7FDFFFF
                                <1>
                                          call ctrl_io_w8
1264
                                 <1>
1265
                                <1>
                                          ;call codec_check_ready
1266
                                <1>
                                          mov
                                                   dx, VIADEV_PLAYBACK + VIA_REG_OFS_PLAYBACK_VOLUME R
1267 00011EC4 66BA0300
                                <1>
                                           ;movax, 2 ; 31
1268
                                <1>
1269 00011EC8 B01F
                                <1>
                                          mov al, 31
1270 00011ECA 2A05[776B0100]
                                          sub al, [audio_master_volume_r]
                                <1>
1271 00011ED0 E8C6FDFFFF
                                 <1>
                                          call ctrl_io_w8
1272
                                 <1>
1273
                                 <1>
                                          ;call codec_check_ready
1274
                                 <1>;
1275
1276
                                 <1> ; All set. Let's play some music.
1277
                                 <1>;
                                 <1>;
1278
1279
                                 <1>
                                                ;mov dx, VIADEV PLAYBACK + VIA REG OFFSET STOP IDX
1280
                                                   ax, VIA8233 REG TYPE 16BIT or VIA8233 REG TYPE STEREO or 0xffffff or 0xff000000
                                 <1>
                                            ; mov
1281
                                 <1>
                                            ;call
                                                   ctrl_io_w32
1282
                                 <1>
1283
                                 <1>
                                          ;call codec_check_ready
1284
                                 <1>
                                          ; 08/12/2016
1285
                                 <1>
                                          ; 07/10/2016
1286
                                 <1>
                                           ;;mov al, 1
1287
                                 <1>
1288
                                 <1>
                                            ;moval, 31
1289
                                 <1>
                                          ; 22/07/2020
1290 00011ED5 B0FF
                                                al, OFFh
                                 <1>
                                          mov
1291 00011ED7 E813000000
                                 <1>
                                          call
                                                 set_VT8233_LastValidIndex
1292
                                 <1>
1293 00011EDC C605[746B0100]01
                                 <1>
                                          mov
                                                byte [audio_play_cmd], 1 ; play command (do not stop) !
1294
                                 <1>
                                          ; 22/07/2020
1295
                                 <1>
1296
                                 <1>
                                          ;mov byte [audio_flag], 0 ; clear half buffer flag
1297
                                 <1>
```

```
1298
                                <1> vt8233_play: ; continue to play
                                      ; 22/04/2017
1299
                                <1>
                                         ;mov al, VIA REG CTRL INT
1300
                                <1>
1301
                                <1>
                                         ;or al, VIA_REG_CTRL_START
1302
                                <1>
                                                     al, VIA_REG_CTRL_AUTOSTART + VIA_REG_CTRL_START
                                           ;;mov
                                         ; 22/07/2020
1303
                                <1>
                                        mov al, VIA REG CTRL AUTOSTART + VIA REG CTRL START + VIA REG CTRL INT FLAG
1304 00011EE3 B0A1
                                <1>
1305
                                <1>
1306 00011EE5 66BA0100
                                <1>
                                                 dx, VIADEV_PLAYBACK + VIA_REG_OFFSET_CONTROL
1307 00011EE9 E8ADFDFFFF
                                         call ctrl io w8
                               <1>
1308
                                <1>
                                         ;call codec_check_ready
1309
                                <1>
                                         ;retn
1310
                                <1>
                                         ;jmp codec_check_ready
1311 00011EEE C3
                                <1>
1312
                                <1>
                                <1> ;input AL = index # to stop on
1313
                                <1> set_VT8233_LastValidIndex:
1314
                                       ; 23/07/2020
1315
                                <1>
1316
                                <1>
                                         ; 10/06/2017
                                        ; 21/04/2017 (TRDOS 386 kernel, 'audio.s')
1317
                                <1>
                                        ; 24/03/2017 - 'PLAYER.COM' ('via wav.asm' - 29/11/2016)
1318
                                <1>
1319
                                <1>
                                         ; 19/11/2016
                                         ; 14/11/2016 - Erdogan Tan (Ref: VIA VT8235.PDF, Page 110)
1320
                                <1>
1321
                                <1>
                                         ; 12/11/2016 - Erdogan Tan
                                         ; (Ref: KolibriOS, vt823x.asm, 'create_primary_buff')
1322
                                <1>
1323
                                <1>
                                         ;push edx
1324
                                <1>
                                         ;push ax
                                       push eax ; 23/07/2020
1325 00011EEF 50
                                <1>
                                <1>
                                         ;push ecx
1327 00011EF0 0FB705[726B0100] <1>
                                         movzx eax, word [audio_freq] ; Hertz
1328 00011EF7 BA00001000
                                <1>
                                         mov edx, 100000h; 2^20 = 1048576
1329 00011EFC F7E2
                                <1>
                                         mul
                                               edx
                                         mov ecx, 48000
1330 00011EFE B980BB0000
                               <1>
1331 00011F03 F7F1
                               <1>
                                       div ecx
                                       ;and eax, OFFFFFh
1332
                               <1>
1333
                               <1>
                                         ;pop ecx
                               <1>
                                         ;pop dx
                                     pop
                                               edx ; 23/07/2020
1335 00011F05 5A
                               <1>
                                       shl
or
1336 00011F06 C1E218
                               <1>
                                               edx, 24 ; STOP Index Setting: Bit 24 to 31
1337 00011F09 09D0
                               <1>
                                               eax, edx
                                     ; 19/11/2016
cmp byte [
ine short
1338
                               <1>
                                        cmp byte [audio_bps], 16
jne short sLVI_1
1339 00011F0B 803D[706B0100]10
                               <1>
1340 00011F12 7505
                               <1>
                        1341 00011F14 0D00002000
                                               eax, VIA8233_REG_TYPE_16BIT
1342
                               <1> sLVI 1:
1343 00011F19 803D[716B0100]02 <1>
                                         cmp
                                               byte [audio_stmo], 2
1344 00011F20 7505
                               <1>
                                         jne short sLVI 2
                               <1> or
1345 00011F22 0D00001000
                                               eax, VIA8233_REG_TYPE_STEREO
1346
                               <1> sLVI_2:
                               <1> mov
                                                edx, VIADEV PLAYBACK + VIA REG OFFSET STOP IDX
1347 00011F27 BA08000000
1348 00011F2C E87CFDFFFF
                                         call ctrl_io_w32
                               <1>
1349
                                <1>
                                         ;call codec_check_ready
1350
                               <1>
                                         ;pop edx
                                       retn
1351 00011F31 C3
                               <1>
1352
                                <1>
1353
                                <1> vt8233_pause: ; pause
                                <1> ; 10/06/2017
1354
                                         ; 22/04/2017
1355
                                <1>
                                       ;mov al, VIA_REG_CTRL_INT
1356
                                <1>
1357
                                <1>
                                                   al, VIA_REG_CTRL_PAUSE
                                         ;or
                                       ; 23/07/2020
1358
                                <1>
1359 00011F32 B029
                                <1>
                                       mov al, VIA_REG_CTRL_PAUSE+VIA_REG_CTRL_INT_FLAG+VIA_REG_CTRL_AUTOSTART
                                <1>
1360
1361 00011F34 66BA0100
                                <1>
                                                 dx, VIADEV_PLAYBACK + VIA_REG_OFFSET_CONTROL
                                         call ctrl_io_w8
1362 00011F38 E85EFDFFFF
                                <1>
1363
                                <1>
                                         ;call codec_check_ready
1364
                                <1>
                                         ;retn
1365
                                <1>
                                         ;jmp codec_check_ready
1366 00011F3D C3
                                <1>
                                         retn
1367
                                <1>
1368
                                <1> vt8233_reset:
                                      ; 22/04/2017
1369
                                <1>
                                         ; reset VT8237R (vt8233) Audio Controller
1370
                                <1>
1371
                                <1>
                                       ;cmp byte [audio_play_cmd], 1
                                       ;jna short vt8233_rst_0
mov byte [audio play cr
                                <1>
1372
1373 00011F3E C605[746B0100]00
                                               byte [audio_play_cmd], 0 ; stop !
                               <1>
                                <1> vt8233 rst 0:
1375 00011F45 E8B0FCFFFF
                                     call reset_codec
                                <1>
1376 00011F4A 720A
                                <1>
                                               short vt8233_rst_1 ; codec error !
                                         jс
1377
                                         : eax = 1
                                <1>
1378 00011F4C E82EFDFFFF
                                       call codec_io_w16; w32
                                <1>
1379 00011F51 E8ABFEFFFF
                                <1>
                                         call channel_reset
                                <1> vt8233 rst_1:
1380
1381 00011F56 C3
                                <1>
1382
                                <1>
                                <1> vt8233_volume:
1383
1384
                                <1>
                                       ; set VT8237R (vt8233) sound volume level
                                         ; 24/04/2017
1385
                                <1>
                                        ; 22/04/2017
; bl = component (0 = master/playback/lineout volume)
1386
                                <1>
1387
                                <1>
1388
                                <1>
                                        ; cl = left channel volume level (0 to 31)
                                        ; ch = right channel volume level (0 to 31)
1389
                                <1>
1390
                                <1>
1391 00011F57 08DB
                               <1>
                                               bl, bl
                                        or
                                               short vt8233_vol_1 ; temporary !
1392 00011F59 7520
                                <1>
                                       jnz
mov
1393 00011F5B 66B81F1F
                               <1>
                                               ax, 1F1Fh ; 31,31
                                       cmp
1394 00011F5F 38C1
                               <1>
                                               cl, al
                                       ja
                                                short vt8233 vol 1 ; temporary !
1395 00011F61 7718
                               <1>
1396 00011F63 38E5
                               <1>
                                        cmp
                                               ch, ah
                                        ja
                                                short vt8233 vol 1 ; temporary !
1397 00011F65 7714
                               <1>
1398 00011F67 66890D[766B0100] <1> mov
                                                [audio_master_volume], cx
                               <1> sub ax, cx
<1> mov edx, CODEC_M
<1> call codec_write
1399 00011F6E 6629C8
1400 00011F71 BA02000000
                                               edx, CODEC_MASTER_VOL_REG ; 02h ; Line Out
1401 00011F76 E870FDFFFF
                                <1> vt8233_vol_1:
```

```
1403 00011F7B C3
                                <1>
                                          retn
1404
                                <1>
1405
                                <1>; CODE for SOUND BLASTER 16
1406
                                <1>
1407
                                <1> DetectSB:
                                         ; 24/04/2017
1408
                                <1>
1409
                                <1>
                                          ;pushad
                                <1> ScanPort:
1410
1411 00011F7C 66BB1002
                                <1>
                                                 bx, 210h ; start scanning ports
                                       mov
1412
                                <1>
                                                            ; 210h, 220h, .. 260h
1413
                                <1> ResetDSP:
1414 00011F80 6689DA
                                                 dx, bx
                                <1>
                                         mov
                                                                 ; try to reset the DSP.
                                               dx, 06h
1415 00011F83 6683C206
                               <1>
                                          add
1416 00011F87 B001
                               <1>
                                          mov
                                               al, 1
1417 00011F89 EE
                               <1>
                                               dx, al
                                         out
1418
                               <1>
1419 00011F8A EC
                                                al, dx
                               <1>
                                         in
1420 00011F8B EC
                                         in
                               <1>
                                                al, dx
1421 00011F8C EC
                                <1>
                                          in
                                                al, dx
1422 00011F8D EC
                               <1>
                                                al, dx
                                         in
1423
                                <1>
1424 00011F8E 30C0
                                                 al, al
                                <1>
                                         xor
1425 00011F90 EE
                                <1>
                                                dx, al
                                         out
                                <1>
1427 00011F91 6683C208
                                <1>
                                         add
                                                dx, 08h
1428 00011F95 66B96400
                                <1>
                                          mov
                                                cx, 100
                               <1> WaitID:
1430 00011F99 EC
                               <1>
                                         in
                                                al, dx
1431 00011F9A 08C0
                                <1>
                                          or
                                                al, al
1432 00011F9C 7804
                               <1>
                                                 short GetID
                                          is
                                                WaitID
1433 00011F9E E2F9
                               <1>
                                          loop
1434 00011FA0 EB0F
                                <1>
                                          jmp
                                                 short NextPort
                                <1> GetID:
1435
1436 00011FA2 6683EA04
                               <1> sub
                                               dx, 04h
                                                al, dx
1437 00011FA6 EC
                                <1>
                                          in
1438 00011FA7 3CAA
                               <1>
                                          cmp
                                                al, OAAh
1439 00011FA9 7413
                                                 short Found
                               <1>
                                          jе
                                          add
1440 00011FAB 6683C204
                               <1>
                                                 dx, 04h
                                         loop
1441 00011FAF E2E8
                                <1>
                                                 WaitID
1442
                                <1> NextPort:
1443 00011FB1 6683C310
                               <1>
                                          add
                                                 bx, 10h
                                                                   ; if not response,
                                                           ; try the next port.
1444 00011FB5 6681FB6002
                                <1>
                                          cmp
                                                 bx, 260h
1445 00011FBA 76C4
                                <1>
                                                 short ResetDSP
                                          jbe
1446 00011FBC F9
                                <1>
                                          stc
1447 00011FBD C3
                                <1>
                                         retn
                                <1> Found:
1448
1449 00011FBE 66891D[466B0100]
                               <1>
                                     mov
                                                 [audio_io_base], bx
                                                                        ; SB Port Address Found!
                                <1> ScanIRQ:
1450
1451
                                <1> SetIrqs:
1452 00011FC5 28C0
                                <1> sub
                                               al, al ; 0
1453 00011FC7 A2[3E6B0100]
                                <1>
                                          mov
                                               [IRQnum], al ; reset
1454 00011FCC A2[436B0100]
                                <1>
                                         mov
                                                [audio_intr], al ; reset
1455
                                <1>
1456
                                <1>
                                        ; ah > 0 \rightarrow set IRQ vector
                                         ; al = IRQ number
;mov ax, 103h ; IRQ 3
1457
                                <1>
1458
                                <1>
                                         ; call set hardware int vector
1459
                                <1>
                                         ;mov ax, 104h ; IRQ 4
1460
                                <1>
1461
                                <1>
                                         ;call set_hardware_int_vector
1462 00011FD1 66B80501
                                <1>
                                         mov ax, 105h; IRQ 5
1463 00011FD5 E8EEDDFFFF
                                <1>
                                         call set_hardware_int_vector
1464 00011FDA 66B80701
                                <1>
                                         mov
                                               ax, 107h ; IRQ 7
1465 00011FDE E8E5DDFFFF
                                         call set_hardware_int_vector
                                <1>
                                <1>
1467 00011FE3 668B15[466B0100]
                                <1>
                                                 dx, [audio_io_base] ; tells to the SB to
                                         mov
1468 00011FEA 6683C20C
                                <1>
                                          add
                                                 dx, 0Ch
                                                                      ; generate a IRQ!
                                <1> WaitSb:
1470 00011FEE EC
                                <1>
                                                al, dx
                                         in
1471 00011FEF 08C0
                                <1>
                                                 al, al
                                          or
1472 00011FF1 78FB
                                <1>
                                                 short WaitSb
                                          js
                                                 al, 0F2h
1473 00011FF3 B0F2
                                <1>
                                          mov
1474 00011FF5 EE
                                <1>
                                         out
                                                dx, al
1475
                                <1>
                                                ecx, ecx ; wait until IRQ level
1476 00011FF6 31C9
                                <1>
                                <1> WaitIRO:
1477
                                <1>
1478 00011FF8 A0[3E6B0100]
                                                al, [IRQnum]
                                         mov
1479 00011FFD 3C00
                                <1>
                                                al, 0 ; is changed or timeout.
                                          cmp
1480 00011FFF 7706
                                <1>
                                          jа
                                                short IrqOk
1481 00012001 6649
                                <1>
                                          dec
1482 00012003 75F3
                                <1>
                                                short WaitIRQ
                                          jnz
1483 00012005 EB15
                                <1>
                                          jmp
                                                short RestoreIrqs
1484
                                <1> IrqOk:
1485 00012007 A2[436B0100]
                               <1>
                                          mov
                                                [audio_intr], al ; set
1486 0001200C 668B15[466B0100] <1>
                                                 dx, [audio_io_base]
1487 00012013 6683C20E
                                <1>
                                          add
                                                 dx, 0Eh
1488 00012017 EC
                                <1>
                                          in
                                                al, dx; SB acknowledge.
1489 00012018 B020
                               <1>
                                                al, 20h
                                         mov
1490 0001201A E620
                               <1>
                                       out 20h, al
                                                            ; Hardware acknowledge.
1491
                                <1>
                                <1> RestoreIrqs:
1492
                                         ; ah = 0 -> reset IRQ vector
1493
                                <1>
1494
                                <1>
                                         ; al = IRQ number
                                         ;mov ax, 3 ; IRQ 3
1495
                                <1>
1496
                                <1>
                                        ;call set_hardware_int_vector
                                       ; mov ax, \frac{1}{4}; IRQ \frac{1}{4}
1497
                                <1>
1498
                                <1>
                                         ;call set_hardware_int_vector
                                       mov ax, 5; IRQ \overline{5}
1499 0001201C 66B80500
                                <1>
1500 00012020 E8A3DDFFFF
                               <1>
                                         call set_hardware_int_vector
1501 00012025 66B80700
                                <1>
                                         mov
                                               ax, 7 ; IRQ 7
                                        call set_hardware_int_vector
1502 00012029 E89ADDFFFF
                               <1>
                               1503
1504 0001202E 31D2
                                                edx, edx
1505 00012030 8915[486B0100]
                                                [audio_dev_id], edx ; 0
1506 00012036 8915[4C6B0100]
                                                [audio_vendor], edx ; 0
1507 0001203C 8915[506B0100]
                                               [audio_stats_cmd], edx ; 0
```

```
1509
                                  <1>
                                            ;popad
1510
                                  <1>
1511 00012042 803D[436B0100]01
                                  <1>
                                                    byte [audio_intr], 1 ; IRQ level was changed?
1512
                                  <1>
1513 00012049 C3
                                  <1>
                                            retn
1514
                                  <1>
                                                  SbOut 1
1515
                                  <1> %macro
1516
                                  <1> %%Wait:
1517
                                  <1>
                                                  al, dx
                                           in
1518
                                  <1>
                                                  al, al
                                            or
1519
                                  <1>
                                                  short %%Wait
                                            js
1520
                                  <1>
                                                  al, %1
                                            mov
1521
                                  <1>
                                            out
                                                  dx, al
                                  <1> %endmacro
1522
1523
                                  <1>
                                  <1> SbInit_play:
1524
                                         ; 22/10/2017
1525
                                  <1>
                                           ; 20/10/2017
1526
                                  <1>
1527
                                  <1>
                                           ; 06/10/2017
1528
                                  <1>
                                           ; 13/07/2017, 09/08/2017
                                           ; 24/04/2017, 15/05/2017, 24/06/2017
1529
                                  <1>
1530
                                  <1>
                                           ;pushad
1531
                                  <1> SetBuffer:
                                                 byte [DmaFlag], 0
1532
                                  <1>
                                           ;mov
1533
                                  <1>
1534 0001204A 8B1D[606B0100]
                                  <1>
                                                  ebx, [audio_dma_buff] ; physical addr of DMA buff
                                           mov
1535 00012050 89DF
                                  <1>
                                           mov
                                                  edi, ebx
1536 00012052 8B0D[646B0100]
                                  <1>
                                                   ecx, [audio_dmabuff_size]
                                           mov
1537
                                  <1>
1538 00012058 803D[706B0100]10
                                  <1>
                                            cmp
                                                  byte [audio_bps], 16
1539 0001205F 7531
                                  <1>
                                                 short sbInit 0 ; set 8 bit DMA buffer
                                           jne
1540
                                  <1>
1541
                                  <1>
                                           ; 09/08/2017
1542
                                  <1>
                                           ; convert byte count to word count
                                            shr ecx, 1
1543 00012061 D1E9
                                 <1>
                                            dec ecx; word count - 1
1544 00012063 49
                                  <1>
                                  <1>
1545
                                            ; convert byte offset to word offset
1546 00012064 D1EB
                                  <1>
                                                 ebx, 1
1547
                                 <1>
                                            ; 16 bit DMA buffer setting (DMA channel 5)
1548
                                  <1>
1549 00012066 B005
                                                   al, 05h; set mask bit for channel 5 (4+1)
                                  <1>
                                            mov
1550 00012068 E6D4
                                                  OD4h, al
                                  <1>
                                            out
1551
                                  <1>
                                                   al, al ; stops all DMA processes on selected channel
1552 0001206A 30C0
                                  <1>
                                            xor
1553 0001206C E6D8
                                  <1>
                                            out
                                                  OD8h, al ; clear selected channel register
                                  <1>
1555 0001206E 88D8
                                                               ; byte 0 of DMA buffer offset in words (physical)
                                  <1>
                                            mov
                                                   al, bl
1556 00012070 E6C4
                                  <1>
                                                  OC4h, al ; DMA channel 5 port number
1557
                                  <1>
1558 00012072 88F8
                                  <1>
                                                    al, bh
                                                            ; byte 1 of DMA buffer offset in words (physical)
                                            mov
1559 00012074 E6C4
                                  <1>
                                            out
                                                  0C4h, al
1560
                                  <1>
1561
                                  <1>
                                            ; 09/08/2017
1562 00012076 C1EB0F
                                  <1>
                                                  ebx, 15
                                                                ; complete 16 bit shift
                                            shr
1563 00012079 80E3FE
                                  <1>
                                                  bl, OFEh; clear bit 0 (not necessary, it will be ignored)
                                  <1>
1565 0001207C 88D8
                                                   al, bl ; byte 2 of DMA buffer address (physical)
                                  <1>
                                            mov
1566 0001207E E68B
                                  <1>
                                                   8Bh, al ; page register port addr for channel 5 ; 13/07/2017
                                            out
                                 <1>
1567
1568 00012080 88C8
                                 <1>
                                                    al, cl ; low byte of DMA count - 1
                                            mov
1569 00012082 E6C6
                                  <1>
                                                  OC6h, al ; count register port addr for channel 1
                                            out
1570
                                  <1>
1571 00012084 88E8
                                  <1>
                                                            ; high byte of DMA count - 1
                                                    al, ch
1572 00012086 E6C6
                                  <1>
                                            out
                                                  0C6h, al
1573
                                  <1>
1574
                                  <1>
                                            ; channel 5, read, autoinitialized, single mode
1575
                                  <1>
                                            ;mov al, 49h
1576 00012088 B059
                                  <1>
                                                  al, 59h ; 06/10/2017
1577 0001208A E6D6
                                  <1>
                                                  OD6h, al ; DMA mode register port address
                                            out
1578
                                  <1>
1579 0001208C B001
                                  <1>
                                                    al, 01h ; clear mask bit for channel 1
                                            mov
                                                  OD4h, al ; DMA mask register port address
1580 0001208E E6D4
                                  <1>
                                            out
1581
                                  <1>
1582 00012090 EB28
                                  <1>
                                            gmj
                                                  short ClearBuffer
1583
                                  <1>
1584
                                  <1> sbInit 0:
1585 00012092 49
                                  <1>
                                                   ecx ; 09/08/2017
1586
                                  <1>
                                            ; 8 bit DMA buffer setting (DMA channel 1)
1587
                                  <1>
1588 00012093 B005
                                  <1>
                                            mov
                                                  al, 05h; set mask bit for channel 1 (4+1)
1589 00012095 E60A
                                  <1>
                                            out
                                                   OAh, al ; DMA mask register
                                  <1>
1591 00012097 30C0
                                  <1>
                                                    al, al ; stops all DMA processes on selected channel
1592 00012099 E60C
                                  <1>
                                                  OCh, al ; clear selected channel register
                                            out
1593
                                  <1>
1594 0001209B 88D8
                                  <1>
                                                               ; byte 0 of DMA buffer address (physical)
                                            mov
1595 0001209D E602
                                  <1>
                                                  02h, al ; DMA channel 1 port number
                                            out
1596
                                  <1>
1597 0001209F 88F8
                                  <1>
                                                    al, bh ; byte 1 of DMA buffer address (physical)
                                            mov
1598 000120A1 E602
                                  <1>
                                            out
                                                  02h, al
1599
                                  <1>
                                                  ebx, 16
1600 000120A3 C1EB10
                                  <1>
                                            shr
                                  <1>
1601
1602 000120A6 88D8
                                  <1>
                                                    al, bl ; byte 2 of DMA buffer address (physical)
                                            mov
1603 000120A8 E683
                                  <1>
                                            out
                                                  83h, al ; page register port addr for channel 1
                                  <1>
1605 000120AA 88C8
                                                    al, cl ; low byte of DMA count - 1
                                  <1>
                                            mov
1606 000120AC E603
                                  <1>
                                            out
                                                   03h, al ; count register port addr for channel 1
                                  <1>
1607
1608 000120AE 88E8
                                                    al, ch ; high byte of DMA count - 1
                                  <1>
                                            mov
1609 000120B0 E603
                                  <1>
                                                  03h, al
                                            out
1610
                                  <1>
1611
                                  <1>
                                            ; channel 1, read, autoinitialized, single mode
1612
                                  <1>
                                            ;mov al, 49h
```

1508

<1>

```
1613 000120B2 B059
                                 <1>
                                                 al, 59h; 06/10/2017
                                           mov
1614 000120B4 E60B
                                 <1>
                                           out
                                                 OBh, al ; DMA mode register port address
1615
                                 <1>
1616 000120B6 B001
                                 <1>
                                                   al, 01h ; clear mask bit for channel 1
                                           mov
1617 000120B8 E60A
                                 <1>
                                                 OAh, al ; DMA mask register port address
                                           out
1618
                                 <1>
1619
                                 <1> ClearBuffer:
1620
                                          ;;mov edi, [audio_dma_buff]
                                 <1>
1621
                                 <1>
                                           ;; mov ecx, [audio_dmabuff_size]
1622
                                 <1>
                                          ;inc ecx
1623
                                 <1>
                                           ;mov
                                                   al, 80h
                                 <1>
1624
                                          ;;cld
1625
                                 <1>
                                          ;rep
                                                   stosb
                                 <1> SetIrq:
1626
                                 <1>
                                          ;mov ebx, SbIrqhandler
1627
1628
                                 <1>
                                           ;mov al, [audio_intr] ; IRQ number
                                          ;call set dev IRQ service
1629
                                 <1>
                                 <1>
                                          ;; SETUP (audio) INTERRUPT CALLBACK SERVICE
1630
1631
                                 <1>
                                           ;mov bl, [audio_intr] ; IRQ number
1632
                                 <1>
                                          ;mov bh, [audio_cb_mode]
1633
                                 <1>
                                          ;inc bh ; 1 = Signal Response Byte method (fixed value)
                                          ;
;
                                                    ; 2 = Callback service method
1634
                                 <1>
                                                     ; 3 = Auto Increment S.R.B. method
1635
                                 <1>
1636
                                 <1>
                                           ; mov
                                                cl, [audio_srb]
                                 <1>
1637
                                           ;mov edx, [audio_cb_addr]
1638
                                 <1>
                                           ;mov
                                                 al, [audio_user]
1639
                                 <1>
                                           ;call set_irq_callback_service
                                 <1> ResetDsp:
1640
1641 000120BA 668B15[466B0100]
                                 <1>
                                          mov
                                                  dx, [audio_io_base]
1642 000120C1 6683C206
                                 <1>
                                                  dx, 06h
                                           add
1643 000120C5 B001
                                 <1>
                                           mov
                                                  al, 1
1644 000120C7 EE
                                 <1>
                                          out
                                                 dx, al
1645
                                 <1>
1646 000120C8 EC
                                <1>
                                          in
                                                 al, dx
                                                 al, dx
1647 000120C9 EC
                                 <1>
                                           in
1648 000120CA EC
                                 <1>
                                           in
                                                 al, dx
1649 000120CB EC
                                 <1>
                                                 al, dx
                                          in
1650
                                 <1>
1651 000120CC 30C0
                                 <1>
                                                  al, al
                                           xor
1652 000120CE EE
                                 <1>
                                                 dx, al
                                          out
1653
                                 <1>
1654 000120CF 66B96400
                                 <1>
                                          mov
                                                  cx, 100
1655 000120D3 28E4
                                 <1>
                                          sub
                                                 ah, ah ; 0
1656
                                 <1> WaitId:
1657 000120D5 668B15[466B0100]
                                <1>
                                       mov
                                                  dx, [audio_io_base]
1658 000120DC 6683C20E
                                 <1>
                                           add
                                                  dx, 0Eh
1659 000120E0 EC
                                 <1>
                                          in
                                                 al, dx
1660 000120E1 08C0
                                 <1>
                                           or
                                                  al, al
1661 000120E3 7807
                                           js
                                 <1>
                                                   short sb_GetId
1662 000120E5 E2EE
                                 <1>
                                           loop
                                                  WaitId
1663 000120E7 E9B4000000
                                 <1>
                                           jmp
                                                  sb_Exit
1664
                                 <1> sb GetId:
1665 000120EC 668B15[466B0100]
                                <1>
                                          mov
                                                  dx, [audio_io_base]
                                                  dx, 0Ah
1666 000120F3 6683C20A
                                <1>
                                           add
1667 000120F7 EC
                                 <1>
                                           in
                                                 al, dx
1668 000120F8 3CAA
                                 <1>
                                           cmp
                                                  al, OAAh
1669 000120FA 7407
                                 <1>
                                                  short SbOk
                                           jе
                                                 WaitId
1670 000120FC E2D7
                                 <1>
                                          loop
1671 000120FE E99D000000
                                 <1>
                                                 sb_Exit
                                           jmp
                                 <1> SbOk:
1673 00012103 668B15[466B0100]
                                <1>
                                           mov
                                                   dx, [audio_io_base]
1674 0001210A 6683C20C
                                 <1>
                                           add
                                                  dx, 0Ch
                                           SbOut
1675
                                 <1>
                                                  OD1h ; Turn on speaker
1675
                                 <2> %%Wait:
                                 <2> in al, dx
1675 0001210E EC
1675 0001210F 08C0
                                 <2> or al, al
                                 <2> js short %%Wait
1675 00012111 78FB
                                 <2> mov al, %1
1675 00012113 B0D1
                                 <2> out dx, al
1675 00012115 EE
1676
                                 <1>
                                          SbOut 41h; 8 bit or 16 bit transfer
                                 <2> %%Wait:
1676
1676 00012116 EC
                                 \langle 2 \rangle in al, dx
1676 00012117 08C0
                                 <2> or al, al
1676 00012119 78FB
                                 <2> js short %%Wait
                                 <2> mov al, %1
1676 0001211B B041
1676 0001211D EE
                                 <2> out dx, al
1677 0001211E 668B1D[726B0100]
                                 <1>
                                          mov bx, [audio freq]; sampling rate (Hz)
1678
                                 <1>
                                           SbOut bh ; sampling rate high byte
                                 <2> %%Wait:
1678
1678 00012125 EC
                                 <2> in al, dx
1678 00012126 08C0
                                 <2> or al, al
                                 <2> js short %%Wait
<2> mov al, %1
1678 00012128 78FB
1678 0001212A 88F8
1678 0001212C EE
                                 <2> out dx, al
1679
                                 <1>
                                         SbOut bl ; sampling rate low byte
                                <2> %%Wait:
1679
1679 0001212D EC
                                <2> in al, dx
1679 0001212E 08C0
                                <2> or al, al
1679 00012130 78FB
                                <2> js short %%Wait
                                <2> mov al, %1
1679 00012132 88D8
1679 00012134 EE
                                 <2> out dx, al
1680
                                 <1>
1681
                                 <1>
                                           ; 22/05/2017
1682 00012135 E8C0000000
                                 <1>
                                           call sb16_volume_initial; 15/05/2017
1683
                                 <1>
                                           ; 20/05/2017
1684
                                 <1>
                                           ;call sb16_volume
1685
                                 <1>
1686
                                 <1> StartDma:
1687
                                 <1>
                                          ; autoinitialized mode
1688 0001213A 803D[706B0100]10
                                           cmp byte [audio bps], 16; 16 bit samples
                                <1>
1689 00012141 7411
                                 <1>
                                         jе
                                               short sb_play_1
1690
                                 <1>
                                          ; 8 bit samples
1691 00012143 66BBC600
                                 <1>
                                          mov bx, 0C6h; 8 bit output (0C6h)
1692 00012147 803D[716B0100]02
                                <1>
                                           cmp byte [audio_stmo], 2; 1 = mono, 2 = stereo
1693 0001214E 7214
                                 <1>
                                           jb
                                                short sb_play_2
```

```
1695 00012152 EB10
                                <1>
                                         jmp
                                               short sb_play_2
                               <1> sb_play_1:
1696
1697
                               <1> ; 16 bit samples
1698 00012154 66BBB610
                                <1>
                                         mov bx, 10B6h; 16 bit output (0B6h)
1699 00012158 803D[716B0100]02 <1>
                                         cmp
                                               byte [audio_stmo], 2 ; 1 = mono, 2 = stereo
                                          jb short sb_play_2
1700 0001215F 7203
                               <1>
                                     jb snort sp_{p_1ay_2} add bh, 20h ; 16 bit stereo (30h)
1701 00012161 80C720
                               <1>
1702
                                <1> sb_play_2:
1703
                               <1> ; PCM output (8/16 bit mono autoinitialized transfer)
                                         SbOut bl ; bCommand
1704
                                <1>
1704
                                <2> %%Wait:
                               <2> in al, dx
1704 00012164 EC
1704 00012165 08C0
                               <2> or al, al
                               <2> js short %%Wait
<2> mov al, %1
1704 00012167 78FB
1704 00012169 88D8
                               <2> out dx, al
1704 0001216B EE
1705
                               <1>
                                        SbOut bh ; bMode
                                <2> %%Wait:
1705
1705 0001216C EC
                               <2> in al, dx
1705 0001216D 08C0
                               <2> or al, al
1705 0001216F 78FB
                               <2> js short %%Wait
                               <2> mov al, %1
1705 00012171 88F8
1705 00012173 EE
                               <2> out dx, al
<1>
                                         ; 20/10/2017
1709 0001217C 803D[706B0100]10 <1>
                                       cmp byte [audio_bps], 16; 16 bit DMA
1710 00012183 7502
                                <1>
                                     Jiie
shr
                                         jne
                                               short sb_play_3
1711 00012185 D1EB
                               <1>
                                               ebx, 1 ; byte count to word count
1712
                               <1> sb_play_3:
                               <1> dec bx ; wBlkSize is one less than the actual size
1713 00012187 664B
                                          SbOut bl
                               <1>
1714
1714
                               <2> %%Wait:
1714 00012189 EC
                               <2> in al, dx
                               <2> or al, al
1714 0001218A 08C0
1714 0001218C 78FB
                               <2> js short %%Wait
1714 0001218E 88D8
                               <2> mov al, %1
1714 00012190 EE
                               <2> out dx, al
1715
                               <1> SbOut
1715
                               <2> %%Wait:
                               <2> in al, dx
1715 00012191 EC
                               <2> or al, al
1715 00012192 08C0
1715 00012194 78FB
                               <2> js short %%Wait
                                <2> mov al, %1
1715 00012196 88F8
                                <2> out dx, al
1715 00012198 EE
                                <1>
1717 00012199 C605[746B0100]01
                                         mov byte [audio_play_cmd], 1 ; playing !
                               <1>
1718
                                <1>
                                         ;; Set Voice and master volumes
1719
                                <1>
1720
                                <1>
                                         ;mov dx, [audio_io_base]
1721
                                <1>
                                         ;add dl, 4 ; Mixer chip Register Address Port
                                         ;SbOut 30h ; select Master Volume Register (L) ;inc dl ; Mixer chip Register Data Port
1722
                                <1>
1723
                                <1>
1724
                                <1>
                                         ;SbOut 0F8h ; Max. volume value is 31 (31*8)
                                         ;dec dl
1725
                                <1>
1726
                                <1>
                                         ;SbOut 31h ; select Master Volume Register (R)
1727
                                <1>
                                         ;inc dl
1728
                                <1>
                                          ;SbOut 0F8h ; Max. volume value is 31 (31*8)
1729
                                <1>
                                         ;dec dl
1730
                                <1>
                                         ;SbOut 32h ; select Voice Volume Register (L)
1731
                                <1>
                                         ;inc dl
1732
                                <1>
                                          ;SbOut 0F8h ; Max. volume value is 31 (31*8)
                                         ;dec dl
1733
                                <1>
1734
                                <1>
                                          ;SbOut 33h ; select Voice Volume Register (R)
1735
                                <1>
                                          ;inc dl
1736
                                <1>
                                         ;SbOut 0F8h ; Max. volume value is 31 (31*8)
1737
                                <1>
                                         ;;
1738
                                <1>
                                          ;dec dl
1739
                                <1>
                                          ;SbOut 44h ; select Treble Register (L)
1740
                                <1>
                                          ;inc dl
                                          ;SbOut OFOh ; Max. Treble value is 15 (15*16)
1741
                                <1>
                                          ;dec dl
1742
                                <1>
1743
                                <1>
                                         ;SbOut 45h
                                                     ; select Treble Register (R)
                                         ;inc dl
1744
                                <1>
1745
                                <1>
                                          ;SbOut OFOh ; Max. Treble value is 15 (15*16)
1746
                                <1>
                                         ;dec dl
1747
                                <1>
                                          ;SbOut 46h ; select Bass Register (L)
1748
                                <1>
                                          ;inc dl
1749
                                          ;SbOut OFOh ; Max. Bass value is 15 (15*16)
                                <1>
1750
                                <1>
                                          ;dec dl
                                                      ; select Bass Register (R)
1751
                                 <1>
                                          ;SbOut 47h
                                          ;inc dl
1752
                                <1>
1753
                                <1>
                                          ;SbOut OFOh ; Max. Bass value is 15 (15*16)
1754
                                <1>
1755
                                <1> sb_Exit:
                                         ;popad
1756
                                <1>
1757 000121A0 C3
                                <1>
                                          retn
1758
                                <1>
1759
                                <1> sb16 int handler:
                                          ; Interrupt Handler for Sound Blaster 16 Audio Card
1760
                                <1>
                                          ; Note: called by 'dev IRQ service'
1761
                                <1>
1762
                                <1>
                                          ; 20/10/2017
1763
                                <1>
                                         ; 12/10/2017
1764
                                <1>
                                         ; 10/10/2017
                                          ; 12/05/2017, 09/10/2017
1765
                                <1>
                                         ; 24/04/2017 (TRDOS 386 kernel, 'audio.s')
1766
                                <1>
                                <1>
                                          ; 10/03/2017 - 'PLAYWAV.PRG' ('playwav.s')
1767
1768
                                <1>
1769
                                <1>
                                          ; push eax ; * must be saved !
                                          ;push ebx ; * must be saved !
1770
                                <1>
1771
                                <1>
                                          ;push ecx
1772
                                <1>
                                          ;push edx
1773
                                <1>
                                          ;push esi
1774
                                <1>
                                          ;push edi
```

mov bh, 20h ; 8 bit stereo (20h)

1694 00012150 B720

<1>

```
1775
1776 000121A1 668B15[466B0100]
                                 <1>
                                          mov
                                                  dx, [audio io base]
1777
                                          ; 20/10/2017
                                 <1>
1778 000121A8 80C20F
                                 <1>
                                                  dl, OFh; 2xFh (DSP 16 bit intr ack)
1779 000121AB 803D[706B0100]10
                                 <1>
                                           cmp
                                                 byte [audio bps], 16
1780 000121B2 7402
                                 <1>
                                           jе
                                                 short sb_irq_16bit_ack
                                 <1> sb_irq_8bit_ack:
1782 000121B4 FECA
                                 <1>
                                           dec dl ; 2xEh (DSP 8 bit intr ack)
1783
                                 <1> sb_irq_16bit_ack:
1784 000121B6 EC
                                 <1>
                                                 al, dx
                                          in
1785
                                 <1>
1786
                                 <1>
                                           ;cmp byte [audio busy], 0
1787
                                 <1>
                                                 short sb_irq_h3
                                           ;ja
1788
                                 <1>
1789
                                 <1>
                                          ;mov
                                                byte [audio_busy], 1
1790
                                 <1>
1791 000121B7 803D[746B0100]01
                                <1>
                                           cmp
                                                 byte [audio play cmd], 1
1792 000121BE 7307
                                 <1>
                                           jnb
                                                 short sb_irq_h1
                                 <1> sb_irq_h0:
1793
1794 000121C0 E8A9000000
                                 <1>
                                           call sb16 stop
1795 000121C5 EB2B
                                 <1>
                                           jmp
                                                short sb_irq_h3
1796
                                 <1> sb_irq_h1:
1797
                                          ;call sb16 tuneloop
                                 <1>
1798
                                 <1>
                                          ; 09/10/2017
                                 <1> sb16 tuneloop:
1799
1800 000121C7 8B3D[606B0100]
                                 <1>
                                          mov
                                                edi, [audio_dma_buff]
1801 000121CD 8B0D[646B0100]
                                 <1>
                                          mov
                                                ecx, [audio dmabuff size]
                                          shr ecx, 1; dma buff size / 2 = half buffer size
1802 000121D3 D1E9
                                 <1>
1803
                                 <1>
                                          ; 22/05/2017
1804
                                 <1>
1805 000121D5 F605[686B0100]01
                                 <1>
                                          test byte [audio_flag], 1 ; Current flag value
                                                 short sb tlp1 ; EOL (Half Buffer 1 must be filled)
1806 000121DC 7402
                                 <1>
                                           jz
1807
                                 <1>
                                           ; FLAG (Half Buffer 2 must be filled)
1808 000121DE 01CF
                                 <1>
                                           add edi, ecx
1809
                                 <1>
                                           ; 15/05/2017
1810
                                 <1> sb_tlp1:
1811 000121E0 8B35[586B0100]
                                 <1>
                                                 esi, [audio_p_buffer] ; phy addr of audio buff
                                          mov
1812
                                 <1>
                                           ;rep movsb
1813 000121E6 C1E902
                                 <1>
                                                 ecx, 2; half buff size / 4
                                           shr
1814 000121E9 F3A5
                                 <1>
                                           rep
                                                movsd
1815
                                 <1>
                                          ;retn
1816
                                 <1>
                                          ; 10/10/2017
1817
                                 <1>
                                           ; switch flag value
1818
                                 <1>
1819 000121EB 8035[686B0100]01
                                 <1>
                                           xor byte [audio_flag], 1
1820
                                 <1>
1821
                                 <1>
                                           ; 12/10/2017
1822
                                 <1>
                                           ; [audio_flag] = 0 : Playing dma half buffer 2 (odd intr count)
1823
                                 <1>
                                                          ; Next buffer (to update) is dma half buff 1
                                                        = 1 : Playing dma half buffer 1 (even intr count)
1824
                                 <1>
                                                           ; Next buffer (to update) is dma half buff 2
1825
                                 <1>
1826
                                 <1>
1827
                                 <1> sb_irq_h3:
1828
                                 <1>
                                                 byte [audio_busy], 0
1829
                                 <1>
1830
                                 <1>
                                                 edi
                                           ;pop
1831
                                 <1>
                                           ;pop
                                                 esi
1832
                                 <1>
                                           ;pop
                                                 edx
1833
                                 <1>
                                           ;pop
1834
                                 <1>
                                                ebx ; * must be restored !
                                           ;pop
1835
                                 <1>
                                                 eax ; * must be restored !
                                           ;pop
1836
                                 <1>
1837 000121F2 C3
                                 <1>
                                           retn
1838
                                 <1>
                                 <1> sb16_volume:
1839
1840
                                 <1>
                                          ; 22/10/2017
                                 <1>
                                           ; mov [audio master volume 1], cl
1842
                                 <1>
                                          ; mov [audio_master_volume_h], ch
1843 000121F3 66890D[766B0100]
                                 <1>
                                                [audio_master_volume], cx
1844
                                 <1> sb16_volume_initial:
1845 000121FA 6652
                                 <1>
                                          push dx ; DX (port address) must be saved
                                                 dx, [audio_io_base]
1846 000121FC 668B15[466B0100]
                                 <1>
                                           mov
1847 00012203 6683C204
                                                dx, 4 ; Mixer chip address port
                                 <1>
                                           add
                                                al, 22h; master volume
1848 00012207 B022
                                 <1>
1849 00012209 EE
                                 <1>
                                          out
                                                 dx, al
1850 0001220A 6642
                                 <1>
                                          inc
                                                 dx
1851 0001220C 8A25[766B0100]
                                 <1>
                                                 ah, [audio master volume 1]
                                          mov
                                                 ah, 2 ; 32 -> 8 level
1852 00012212 C0EC02
                                 <1>
                                           shr
1853 00012215 C0E405
                                 <1>
                                           shl
                                                 ah, 5 ; bit 5 to 7
1854 00012218 A0[776B0100]
                                 <1>
                                                 al, [audio_master_volume r]
                                           mov
1855 0001221D C0E802
                                                 al, 2 ; 32 -> 8 1evel
                                 <1>
                                           shr
                                                 al, OFh
                                 <1>
                                           ;and
1857 00012220 D0E0
                                                al, 1 ; bit 1 to 3
                                <1>
                                           shl
1858 00012222 08E0
                                <1>
                                                 al, ah
1859 00012224 EE
                                <1>
                                          out
                                                dx, al
1860 00012225 665A
                                <1>
                                          pop
                                                dx ; DX (port address) must be restored
                                         retn
1861 00012227 C3
                                <1>
1862
                                <1>
1863
                                 <1> sb16_pause:
1864 00012228 668B15[466B0100] <1> mov dx, [audio_io_base]
1865 0001222F 6683C20C
                                           add dx, OCh; Command & Data Port
                                <1>
1866 00012233 803D[706B0100]10
                                          cmp byte [audio_bps], 16; 16 bit samples
je short sb_pause_1
                                <1>
1867 0001223A 7404
                                <1>
                                <1>
                                          ; 8 bit samples
                                      mov bl, ODOh; 8 bit DMA mode
1869 0001223C B3D0
                                <1>
1870 0001223E EB02
                                <1>
                                                short sb_pause_2
                                          jmp
                                <1> sb_pause_1:
1872
                                       ; 16 bit samples
                                <1>
1873 00012240 B3D5
                                <1>
                                          mov bl, OD5h; 16 bit DMA mode
                                <1> sb_pause_2:
                                <1>
1875
                                         SbOut bl ; bCommand
                               <2> %%Wait:
<2> in al, dx
1875
1875 00012242 EC
1875 00012243 08C0
                               <2> or al, al
1875 00012245 78FB
                                <2> js short %%Wait
```

```
1875 00012247 88D8
                                 <2> mov al, %1
1875 00012249 EE
                                 <2> out dx, al
                                 <1> sb_pause_3:
1876
1877 0001224A C3
                                 <1>
1878
                                 <1>
1879
                                 <1> sb16 continue:
1880 0001224B 668B15[466B0100] <1> mov dx, [audio_io_base]
                                         add
1881 00012252 6683C20C
                                                 dx, OCh ; Command & Data Port
                                 <1>
                                      cmp byte [audio_bps], 16; 16 bit samples je short sb_cont_1; 8 bit samples mov bl, 0D4h; 8 bit DMA mode jmp short sb_cont_2
1882 00012256 803D[706B0100]10
                                <1>
1883 0001225D 7404
                                 <1>
1884
                                 <1>
1885 0001225F B3D4
                                 <1>
1886 00012261 EB02
                                 <1>
1887
                                 <1> sb_cont_1:
                                 <1> ; 16 bit samples

1888
1889 00012263 B3D6
                                 <1>
                                           mov bl, OD6h; 16 bit DMA mode
                                 <1> sb_cont_2:
1890
                                 <1> SbOut bl; bCommand
1891
1891
                                 <2> %%Wait:
1891 00012265 EC
                                 <2> in al, dx
1891 00012266 08C0
                                <2> or al, al
                                <2> js short %%Wait
<2> mov al, %1
1891 00012268 78FB
1891 0001226A 88D8
1891 0001226C EE
                                 <2> out dx, al
                                 <1> sb_cont_3:
1892
1893 0001226D C3
                                 <1>
1894
                                 <1>
1895
                                 <1> sb16_stop:
1896
                                 <1>
                                           ; 24/04/2017
1897 0001226E 803D[746B0100]00
                                 <1>
                                           cmp byte [audio_play_cmd], 0
1898 00012275 7648
                                 <1>
                                         jna short sb16_stop_4
1899
                                 <1>
                                           ; 22/05/2017
1900
                                 <1>
                                         mov dx, [audio_io_base]
1901 00012277 668B15[466B0100]
                                 <1>
                                         add
                                                 dx, 0Ch
1902 0001227E 6683C20C
                                 <1>
                                       mov bl, OD9h; exit auto-initialize 16 bit transfer; stop autoinitialized DMA transfer mode cmp byte [audio_bps], 16; 16 bit samples
1903
                                 <1>
1904 00012282 B3D9
                                 <1>
1905
                                 <1>
                                           cmp byte [audio_bps], 16; 16 bit samples
je short sb16_stop_1
1906 00012284 803D[706B0100]10
                                 <1>
1907 0001228B 7402
                                 <1>
1908
                                           ;mov bl, ODAh ; exit auto-initialize 8 bit transfer
                                 <1>
1909 0001228D FEC3
                                 <1>
                                           inc bl
1910
                                 <1> sb16_stop_1:
1911
                                 <1>
                                           SbOut bl ; exit auto-initialize transfer command
1911
                                 <2> %%Wait:
1911 0001228F EC
                                 <2> in al, dx
1911 00012290 08C0
                                <2> or al, al
1911 00012292 78FB
                                 <2> js short %%Wait
1911 00012294 88D8
                                 <2> mov al, %1
                                 <2> out dx, al
1911 00012296 EE
1912
                                 <1>
1913 00012297 30C0
                                 <1>
                                                   al, al; stops all DMA processes on selected channel
                                           xor
1914
                                 <1>
1915 00012299 803D[706B0100]10
                                <1>
                                                 byte [audio_bps], 16; 16 bit samples
                                                  short sb16_stop_2
1916 000122A0 7404
                                 <1>
                                           jе
1917 000122A2 E60C
                                 <1>
                                                  OCh, al ; clear selected channel register
                                           out
1918 000122A4 EB02
                                 <1>
                                                 short sb16_stop_3
                                           jmp
1919
                                 <1>
1920
                                  <1> sb16_stop_2:
1921 000122A6 E6D8
                                 <1>
                                       out OD8h, al ; clear selected channel register
1922
                                 <1>
                                 <1> sb16_stop_3:
1924 000122A8 C605[746B0100]00
                                 <1> mov byte [audio_play_cmd], 0; stop!
1925
                                  <1> SbDone:
                                  <1> ;mov dx, [audio_io_base]
1926
1927
                                 <1>
                                           ;add dx, 0Ch
1928
                                 <1>
                                           SbOut 0D0h
                                 <2> %%Wait:
1928
1928 000122AF EC
                                 <2> in al, dx
                                 <2> or al, al
1928 000122B0 08C0
1928 000122B2 78FB
                                 <2> js short %%Wait
                                 <2> mov al, %1
1928 000122B4 B0D0
                                 <2> out dx, al
1928 000122B6 EE
1929
                                 <1>
                                           SbOut
                                 <2> %%Wait:
1929
                                 <2> in al, dx
1929 000122B7 EC
1929 000122B8 08C0
                                 <2> or al, al
1929 000122BA 78FB
                                 <2> js short %%Wait
                                 <2> mov al, %1
1929 000122BC B0D3
1929 000122BE EE
                                 <2> out dx, al
                                 <1> sb16_stop_4:
1930
1931 000122BF C3
                                  <1>
                                           retn
1932
                                  <1>
1933
                                  <1> sb16_reset:
                                       ; 24/04/2017
1934
                                 <1>
                                           mov
1935 000122C0 668B15[466B0100]
                                                 dx, [audio io base]; try to reset the DSP.
                                 <1>
1936 000122C7 6683C206
                                                   dx, 06h
                                 <1>
                                           add
                                                 al, 1
1937 000122CB B001
                                 <1>
                                           mov
                                                 dx, al
1938 000122CD EE
                                 <1>
                                           out
                                 <1>
1940 000122CE EC
                                 <1>
                                          in
                                                  al, dx
1941 000122CF EC
                                 <1>
                                           in
                                                  al, dx
1942 000122D0 EC
                                 <1>
                                           in
                                                  al, dx
                                                  al, dx
1943 000122D1 EC
                                 <1>
                                           in
1944
                                 <1>
1945 000122D2 30C0
                                 <1>
                                           xor
                                                  al, al
1946 000122D4 EE
                                 <1>
                                                  dx, al
                                          out
1947
                                 <1>
                                                   dx, 08h
1948 000122D5 6683C208
                                 <1>
                                           add
1949 000122D9 66B96400
                                 <1>
                                                 cx, 100
                                           mov
                                 <1> sbrstWaitID:
1950
1951 000122DD EC
                                 <1>
                                           in
                                                 al, dx
1952 000122DE 08C0
                                 <1>
                                           or
                                                   al, al
1953 000122E0 7804
                                 <1>
                                                   short sbrstGetID
                                           js
                                 <1>
                                                  sbrstWaitID
1954 000122E2 E2F9
                                           loop
```

```
1955 000122E4 F9
                                           stc
1956 000122E5 C3
                                 <1>
                                           retn
1957
                                <1> sbrstGetID:
1958 000122E6 6683EA04
                                <1>
                                           sub
                                                   dx, 04h
1959 000122EA EC
                                <1>
                                           in
                                                 al, dx
1960 000122EB 3CAA
                                <1>
                                           cmp
                                                 al, OAAh
1961 000122ED 7406
                                                   short sb rst retn
                                <1>
                                           jе
1962 000122EF 6683C204
                                          add
                                <1>
                                                  dx, 04h
1963 000122F3 E2E8
                                <1>
                                          loop
                                                  sbrstWaitID
                                <1> sb_rst_retn:
1964
1965 000122F5 C3
                                 <1>
                                           retn
1966
                                 <1>
1967
                                 <1> ac97_codec_config:
                                       ; 10<del>/</del>06/2017
1968
                                 <1>
1969
                                 <1>
                                          ; 05/06/2017
1970
                                 <1>
                                          ; 29/05/2017
1971
                                          ; 28/05/2017 (TRDOS 386, 'audio.s')
                                 <1>
                                         ; 07/11/2016 (Erdogan Tan)
1972
                                 <1>
1973
                                 <1>
                                          ; Derived from 'codecConfig' procedure in 'CODEC.ASM'
1974
                                 <1>
                                          ; .wav player for DOS by Jeff Leyda (02/09/2002)
1975
                                 <1>
1976
                                 <1>
                                           ;; 'PLAYER.ASM'
1977
                                 <1>
                                           ;; get ICH base address regs for mixer and bus master
1978
                                 <1>
1979
                                 <1> init_ac97_controller: ; 10/06/2017
1980 000122F6 A1[486B0100]
                                 <1>
                                          mov eax, [audio_dev_id]
1981
                                 <1>
                                           ;moval, NAMBAR REG
                                           ;;call pciRegRead16
                                                                                ; read PCI registers 10-11
1982
                                 <1>
1983
                                 <1>
                                            ;call
                                                      pciRegRead32
1984
                                 <1>
                                          ; and dx, IO ADDR MASK
                                                                          ; mask off BITO
1985
                                 <1>
                                          ;;and edx, IO_ADDR_MASK
1986
                                 <1>
                                            ;mov [NAMBAR], dx
1987
                                 <1>
                                                                           ; save audio mixer base addr
1988
                                 <1>
1989
                                 <1>
                                                    al, NABMBAR REG
                                            ; mov
1990
                                 <1>
                                            ;;call
                                                       pciRegRead16
                                                       pciRegRead32
1991
                                 <1>
                                            ;call
                                           ; and dx, OFFCOh ; IO_ADDR_MASK
1992
                                 <1>
                                           ;;and edx, 0FFC0h
1993
                                 <1>
1994
                                 <1>
1995
                                                                                 ; save bus master base addr
                                 <1>
                                            ; mov
                                                   [NABMBAR], dx
1996
                                 <1>
1997
                                 <1>
                                           ;mov eax, [audio_dev_id]
1998 000122FB B004
                                 <1>
                                           mov al, PCI_CMD_REG
                                                     pciRegRead8
                                                                                 ; read PCI command register
1999
                                 <1>
                                            ;call
2000 000122FD E84AF8FFFF
                                 <1>
                                             call pciRegRead16
                                           or dl, IO_ENA+BM_ENA; call pciRegWrite8
2001 00012302 80CA05
                                 <1>
                                                                                  ; enable IO and bus master
2002
                                 <1>
2003 00012305 E8ADF8FFFF
                                 <1>
                                           call pciRegWrite16
2004
                                 <1>
2005
                                 <1>
                                           ; 'CODEC.ASM'
2006
                                 <1>
                                           ; enable codec, unmute stuff, set output rate
2007
                                 <1>
2008
                                 <1>;
                                           ; entry: [audio_freq] = desired sample rate
2009
                                 <1>
2010
                                 <1>;
                                           mov
                                                        dx, [NAMBAR]
                                                        dx, CODEC EXT AUDIO CTRL REG
2011
                                 <1>;
                                           add
                                                                                        ; 2Ah
                                 <1>;
2012
                                           in
                                                        ax, dx
2013
                                 <1>;
                                                 ax, 1
                                           or
                                           out dx, ax
2014
                                 <1>;
                                                                                 ; Enable variable rate audio
2015
                                 <1>
                                 <1>;
2016
                                            ;call
                                                      delay1 4ms
                                                      delay1 4ms
2017
                                 <1>;
                                             ;call
2018
                                 <1>;
                                            ;call
                                                      delay1_4ms
                                 <1>;
2019
                                            ;call
                                                      delay1_4ms
2020
                                 <1>
2021
                                 <1>;
                                          mov ax, [audio_freq]
                                                                           ; sample rate
2022
                                 <1>
                                                        dx, [NAMBAR]
2023
                                 <1>;
2024
                                 <1>;
                                                       dx, CODEC_PCM_FRONT_DACRATE_REG ; 2Ch
                                           add
                                                                                 ; out sample rate
2025
                                 <1>;
                                           out dx, ax
2026
                                 <1>
                                             ;call
2027
                                 <1>;
                                                        delay1_4ms
                                 <1>;
2028
                                            ;call
                                                        delay1_4ms
2029
                                 <1>;
                                            ;call
                                                       delay1_4ms
2030
                                 <1>;
                                             ;call
                                                       delay1_4ms
2031
                                 <1>
2032
                                 <1>
                                           ;mov dx, [NAMBAR]
                                                                           ; mixer base address
2033
                                 <1>
                                           ; add dx, CODEC_RESET_REG
                                                                             ; reset register
2034
                                            ; movax, 42
                                 <1>
2035
                                 <1>
                                          ;out dx, ax
                                                                                 ; reset
                                 <1>
2036
                                           ;mov dx, [NABMBAR]
2037
                                 <1>
                                                                                  ; bus master base address
2038
                                 <1>
                                           ; add dx, GLOB STS REG
2039
                                 <1>
                                            ;movax, 2
2040
                                 <1>
                                          ;out dx, ax
2041
                                 <1>
2042 0001230A E847F9FFFF
                                 <1>
                                            call
                                                   delay_100ms ; 29/05/2017
2043
                                 <1>
2044
                                 <1> init ac97 codec:
2045
                                 <1>
                                          ; 10/06/2017
2046
                                 <1>
                                          ; 29/05/2017
2047
                                 <1>
                                           ; 28/05/2017 - Erdogan Tan (Ref: KolibriOS, intelac97.asm)
2048
                                 <1>
2049 0001230F 66BA2C00
                                 <1>
                                                 dx, GLOB CNT REG; 2Ch
                                          mov
2050 00012313 660315[7A6B0100]
                                <1>
                                           add
                                                 dx, [NABMBAR]
2051 0001231A ED
                                 <1>
                                           in
                                                 eax, dx
2052
                                 <1>
                                          ; ?
2053 0001231B 66BA3000
                                 <1>
                                           mov
                                                 dx, GLOB_STS_REG; 30h
                                                 dx, [NABMBAR]
2054 0001231F 660315[7A6B0100] <1>
                                          add
2055 00012326 ED
                                <1>
                                          in
                                                 eax, dx
2056
                                 <1>
2057 00012327 83F8FF
                                 <1>
                                           cmp
                                                 eax, Offffffff ; -1
2058 0001232A 744B
                                 <1>
                                                 short init_ac97_codec_err1
                                           jе
2059
                                 <1>
```

```
test eax, CTRL ST CREADY
2060 0001232C A900030010
                       <1>
2061 00012331 7507
                             <1>
                                    jnz short _ac97_codec_ready
                             <1>
2063 00012333 E8EF020000
                             <1>
                                      call reset_ac97_codec
2064 00012338 723E
                             <1>
                                      jc short init_ac97_codec_err2
                             <1>
2065
                             <1> ac97 codec ready:
2067 0001233A 668B15[786B0100] <1>
                                      mov dx, [NAMBAR]
2068
                             <1>
                                      ;add
                                           dx, 0 ; ac_reg_0 ; reset register
2069 00012341 66EF
                             <1>
                                      out
                                            dx, ax
2070
                             <1>
2071 00012343 31C0
                                      xor eax, eax; 0
                              <1>
2073 0001234C 6683C226
                                            dx, CODEC_REG_POWERDOWN
                                     out dx, ax
2074 00012350 66EF
                             <1>
2075
                             <1>
                                   ; 10/06/2017
; 29/05/2017
; wait for 1 second
mov ecx, 1000; 1000*0.25ms = 1s
2076
                             <1>
2077
                             <1>
2078
                             <1>
                     2079 00012352 B9E8030000
                             <1>
2080
2081 00012357 E807F9FFFF
2082 0001235C E802F9FFFF
2083 00012361 E8FDF8FFFF
2084 00012366 E8F8F8FFFF
2085
2086
2087 0001236B 66ED
2088 0001236D 6683E00F
2089 00012371 3C0F
2090 00012373 7404
2091 00012375 E2E0
2092
2093
                             <1> init_ac97_codec_err1:
2094 00012377 F9
                             <1> stc
                             <1> init_ac97_codec_err2:
2095
2096 00012378 C3
                             <1>
                                     retn
2097
                             <1>
2098
                             <1> _ac97_codec_init_ok:
2100 0001237B 66BA2C00 <1>
                                  mov al, 2; force set 16-bit 2-channel PCM
                                      mov dx, GLOB_CNT_REG; 2Ch
2101 0001237F 660315[7A6B0100]
                                            dx, [NABMBAR]
                             <1>
                                      add
                                      out dx, eax
2102 00012386 EF
                             <1>
2103
                              <1>
2104
                              <1>
                                     ;call delay1_4ms
2105
                              <1>
                                      ; 10/06/2017
                              <1>
2107 00012387 E849020000
                              <1>
                                      call reset_ac97_controller
2108
                              <1>
                                      call setup_ac97_codec
2109
                              <1>;
2110
                              <1>;
2111
                              <1> ;detect ac97 codec:
                              <1>;
2112
                                     retn
2113
                              <1>
2114
                              <1> setup_ac97_codec:
2115
                              <1> ; 22/07/2020
                                      ; 10/06/2017
                              <1>
                                  ; 29/05/2017

mov eax, 0202h

mov [audio_master_volume], ax
                              <1>
2117
2118 0001238C B802020000
                             <1>
2119 00012391 66A3[766B0100]
                             <1>
2120 00012397 66B81F1F
                                   mov ax, 1F1Fh; 31, 31
                             <1>
                              <1>
dx, [NAMBAR]
                                      mov
                                      add
                                            dx, CODEC_MASTER_VOL_REG ;02h
2124 000123A6 6631C0
                             <1>
                                      xor
                                            ax, ax ; volume attenuation = 0 (max. volume)
2125 000123A9 66EF
                             <1>
                                      out
                                             dx, ax
                             <1>
2127 000123AB 668B15[786B0100] <1>
                                      mov
                                              dx, [NAMBAR]
2128 000123B2 6683C206
                             <1>
                                      add
                                             dx, CODEC_MASTER_MONO_VOL_REG ;06h
                                      ;xor
2129
                             <1>
                                           ax, ax
2130 000123B6 66EF
                             <1>
                                      out
                                              dx, ax
                              <1>
2132 000123B8 668B15[786B0100] <1>
                                              dx, [NAMBAR]
                                      mov
                                              dx, CODEC PCBEEP VOL REG; OAh
2133 000123BF 6683C20A
                             <1>
                                    add
                                      ;xor
                                              ax, ax
2134
                              <1>
2135 000123C3 66EF
                             <1>
                                      out
                                              dx, ax
                             <1>
2137 000123C5 668B15[786B0100]
                             <1>
                                      mov
                                              dx, [NAMBAR]
2138 000123CC 6683C218
                              <1>
                                       add
                                              dx, CODEC_PCM_OUT_REG ;18h
2139
                              <1>
                                              ax, ax
                                      ;xor
2140 000123D0 66EF
                              <1>
                                       out
                                              dx, ax
2141
                              <1>
2142 000123D2 66B80880
                                              ax, 8008h; Mute
                             <1>
                                       mov
2143 000123D6 668B15[786B0100] <1>
                                              dx, [NAMBAR]
                                       ; 22/07/2020
2144
                              <1>
2145 000123DD 6683C20C
                                       add dx, CODEC_PHONE_VOL_REG
                              <1>
                                                                          ;0Ch
                                                         ; AC97 PHONE VOL ; TAD Input (Mono)
2146
                             <1>
2147 000123E1 66EF
                             <1>
                                      out
                                             dx, ax
2148
                              <1>
                                       mov ax, 0808h
2149 000123E3 66B80808
                             <1>
2150 000123E7 668B15[786B0100] <1>
                                      mov
                                              dx, [NAMBAR]
2151 000123EE 6683C210
                              <1>
                                       add dx, CODEC LINE IN VOL REG ; 10h ; Line Input (Stereo)
2152 000123F2 66EF
                              <1>
                                       out
                                             dx, ax
                             <1>
                                      ;mov ax, 0808h
2154
                              <1>
2155 000123F4 668B15[786B0100] <1>
                                      mov
                                             dx, [NAMBAR]
2156 000123FB 6683C212
                              <1>
                                      add dx, CODEC_CD_VOL_REG ;12h ; CR Input (Stereo)
2157 000123FF 66EF
                              <1>
                                      out dx, ax
2158
                              <1>
                                      ;mov ax, 0808h
2159
                             <1>
2160 00012401 668B15[786B0100] <1>
                                      mov dx, [NAMBAR]
2161 00012408 6683C216
                              <1>
                                       add dx, CODEC_AUX_VOL_REG ;16h ; Aux Input (Stereo)
2162 0001240C 66EF
                              <1>
                                       out dx, ax
2163
                              <1>
2164
                              <1>
                                      ;call delay1_4ms
```

```
delay1 4ms
2165
                                  <1>
                                             ;call
2166
                                  <1>
                                             ;call
                                                      delay1 4ms
                                                      delay1_4ms
                                             ;call
2167
                                 <1>
2168
                                 <1>
2169
                                 <1> detect_ac97_codec:
2170 0001240E C3
                                 <1>
                                             retn
2171
                                 <1>
                                  <1> set_ac97_bdl: ; Set AC97 (ICH) Buffer Descriptor List
2172
                                        ; 17/06/2017
2173
                                 <1>
                                           ; 11/06/2017
2174
                                 <1>
2175
                                  <1>
                                           ; 28/05/2017
                                           ; eax = dma buffer address = [audio DMA buff]
2176
                                  <1>
                                           ; ecx = dma buffer buffer size = [audio_dmabuff_size]
2177
                                 <1>
2178
                                 <1>
2179 0001240F D1E9
                                  <1>
                                           shr ecx, 1 ; dma half buffer size
2180 00012411 89CE
                                                  esi, ecx
                                  <1>
                                           mov
                                  <1>
                                                      edi, audio_bdl_buff ; get BDL address
2182 00012413 BF[7C6B0100]
                                  <1>
                                             mov
2183 00012418 B910000000
                                  <1>
                                             mov
                                                      ecx, 32 / 2
                                                                            ; make 32 entries in BDL
2184
                                  <1>
2185 0001241D EB05
                                  <1>
                                           jmp
                                                short s_ac97_bdl1
2186
                                  <1>
                                  <1> s_ac97_bd10:
2187
2188
                                  <1>
                                           ; set buffer descriptor 0 to start of data file in memory
                                  <1>
2189
2190 0001241F A1[606B0100]
                                  <1>
                                                eax, [audio_dma_buff] ; Physical address of DMA buffer
2191
                                  <1>
                                  <1> s_ac97_bdl1:
2192
2193 00012424 AB
                                  <1>
                                                                      ; store dmabuffer1 address
                                           stosd
2194
                                  <1>
2195 00012425 89C2
                                  <1>
                                           mov
                                                 edx, eax
2196
                                  <1>
2197
                                  <1>;
2198
                                  <1> ; Buffer Descriptors List
                                  <1> ; As stated earlier, each buffer descriptor list is a set of (up to) 32
2199
2200
                                  <1>; descriptors, each 8 bytes in length. Bytes 0-3 of a descriptor entry point
                                  <1>; to a chunk of memory to either play from or record to. Bytes 4-7 of an
2201
2202
                                  <1> ; entry describe various control things detailed below.
2203
                                  <1>;
                                  <1>; Buffer pointers must always be aligned on a Dword boundry.
2204
2205
                                  <1>;
2206
                                  <1>;
2207
                                  <1>
2208
                                  <1> ;IOC
                                                                       BIT31; Fire an interrupt whenever this
2209
                                  <1>
                                                                             ; buffer is complete.
2210
                                  <1>
                                  <1> ;BUP
2211
                                                                       BIT30 ; Buffer Underrun Policy.
                                                               equ
2212
                                  <1>
                                                                             ; if this buffer is the last buffer
2213
                                  <1>
                                                                              ; in a playback, fill the remaining
2214
                                  <1>
                                                                              ; samples with 0 (silence) or not.
2215
                                  <1>
                                                                              ; It's a good idea to set this to 1
2216
                                  <1>
                                                                              ; for the last buffer in playback,
                                                                              ; otherwise you're likely to get a lot
2217
                                  <1>
2218
                                  <1>
                                                                              ; of noise at the end of the sound.
2219
                                  <1>
2220
                                  <1> ;
2221
                                  <1>; Bits 15:0 contain the length of the buffer, in number of samples, which
2222
                                  <1>; are 16 bits each, coupled in left and right pairs, or 32bits each.
2223
                                  <1> ; Luckily for us, that's the same format as .wav files.
2224
2225
                                  <1>; A value of FFFF is 65536 samples. Running at 44.1Khz, that's just about
                                  <1> ; 1.5 seconds of sample time. FFFF \star 32bits is 1FFFFh bytes or 128k of data.
2226
2227
                                  <1> ;
2228
                                  <1> ; A value of 0 in these bits means play no samples.
2229
                                  <1>;
2230
                                 <1>
2231 00012427 89F0
                                 <1>
                                           mov
                                                 eax, esi ; DMA half buffer size
                                           add
2232 00012429 01C2
                                 <1>
                                                  edx, eax
2233 0001242B D1E8
                                 <1>
                                                  eax, 1 ; count of 16 bit samples
                                            shr
                                                 eax, IOC+BUP
2234
                                 <1>
                                           ;or
2235 0001242D 0D00000080
                                 <1>
                                           or
                                                  eax, IOC ; 11/06/2017
2236 00012432 AB
                                  <1>
                                           stosd
                                  <1>
2237
                                  <1> ; 2nd buffer:
2238
2239
                                  <1>
2240 00012433 89D0
                                              mov eax, edx; Physical address of the 2nd half of DMA buffer
                                  <1>
2241 00012435 AB
                                  <1>
                                                         ; store dmabuffer2 address
                                           stosd
2242
                                  <1>
2243
                                  <1> ; set length to [audio_dmabuff_size]/2
2244
                                  <1>; Set control (bits 31:16) to BUP, bits 15:0=number of samples
                                  <1> ;
2245
                                                  eax, esi ; DMA half buffer size
2246 00012436 89F0
                                  <1>
                                           mov
2247 00012438 D1E8
                                                  eax, 1 ; count of 16 bit samples
                                 <1>
                                            shr
                                  <1>
                                                  eax, IOC+BUP
                                                  eax, IOC ; 11/06/2017
2249 0001243A 0D00000080
                                  <1>
                                           or
                                           stosd
2250 0001243F AB
                                 <1>
                                 <1>
2252 00012440 E2DD
                                 <1>
                                                  s ac97 bd10
                                           loop
2253
                                 <1>
2254 00012442 C3
                                 <1>
                                           retn
2255
                                  <1>
2256
                                  <1> ac97_start_play:
                                           ; 28/05/2017
2257
                                  <1>
                                           ; Derived from 'playWav' procedure in 'ICHWAV.ASM'
2258
                                  <1>
2259
                                  <1>
                                           ; .wav player for DOS by Jeff Leyda (02/09/2002)
2260
                                  <1>
2261
                                  <1>
                                           ; set output rate
                                           ; entry: [audio_freq] = desired sample rate
2262
                                  <1>
2263
                                  <1>
2264 00012443 668B15[786B0100]
                                                        dx, [NAMBAR]
                                 <1>
                                           mov
                                           add
                                                         dx, CODEC_EXT_AUDIO_CTRL_REG
2265 0001244A 6683C22A
                                 <1>
                                                                                       ; 2Ah
2266 0001244E 66ED
                                  <1>
                                           in
                                                        ax, dx
2267 00012450 6683C801
                                 <1>
                                           or
                                                  ax, 1
2268 00012454 66EF
                                  <1>
                                                                                 ; Enable variable rate audio
                                           out dx, ax
2269
                                  <1>
```

```
2270
                                 <1>
                                                    delay1_4ms
                                           ;call
2271
                                 <1>
                                           ;call
                                                    delay1 4ms
                                                    delay1 4ms
2272
                                 <1>
                                           ;call
2273
                                 <1>
                                          ;call
                                                    delay1_4ms
2274
                                 <1>
2275 00012456 66A1[726B0100]
                                          mov
                                 <1>
                                                 ax, [audio_freq]
                                                                          ; sample rate
                                 <1>
2277 0001245C 668B15[786B0100]
                                                       dx, [NAMBAR]
                                 <1>
                                          mov
2278 00012463 6683C22C
                                 <1>
                                          add
                                                      dx, CODEC_PCM_FRONT_DACRATE_REG ; 2Ch
2279 00012467 66EF
                                 <1>
                                                dx, ax
                                          out
                                                                      ; out sample rate
2280
                                 <1>
2281
                                                    delay1 4ms
                                 <1>
                                           ;call
                                                    delay1_4ms
2282
                                 <1>
                                           ;call
2283
                                 <1>
                                           ;call
                                                    delay1_4ms
2284
                                 <1>
                                           ;call
                                                    delay1_4ms
2285
                                 <1>
                                 <1> ; register reset the DMA engine. This may cause a pop noise on the output
2287
2288
                                 <1>; lines when the device is reset. Prolly a better idea to mute output, then
2289
                                 <1> ; reset.
2290
                                 <1> ;
2291 00012469 668B15[7A6B0100]
                                                    dx, [NABMBAR]
                                 <1>
                                            mov
2292 00012470 6683C21B
                                            add
                                                    dx, PO_CR_REG
                                                                                 ; set pointer to Cntl reg
                                 <1>
2293 00012474 B002
                                 <1>
                                            mov
                                                    al, RR
                                                                                 ; set reset
2294 00012476 EE
                                 <1>
                                           out
                                                    dx, al
                                                                                  ; self clearing bit
2295
                                 <1>;
2296
                                 <1>;
                                          mov edi, audio bdl buff
2297
                                          mov edx, [audio_dmabuff_size]
                                 <1>;
2298
                                 <1>;
                                                edx, 1
                                          shr
                                          mov
                                                ecx, 32/2
2299
                                 <1>;
2300
                                 <1> ;ac97_set_bdl_buffer:
2301
                                 <1> ; 1st half of DMA buffer
                                 <1>;
                                          mov eax, [audio_dma_buff]
2302
2303
                                 <1>;
                                        push eax
                                          stosd
2304
                                 <1>;
                                          mov eax, edx; dma buffer size / 2
2305
                                 <1>;
2306
                                 <1>;
                                          or
                                                eax, IOC+BUP
2307
                                 <1>;
                                          stosd
2308
                                 <1>;
                                          pop eax
                                          ; 2nd half of DMA buffer
2309
                                 <1>;
2310
                                 <1>;
                                          add eax, edx
                                          stosd
2311
                                 <1>;
                                 <1>;
                                          mov eax, edx; dma buffer size / 2
2312
2313
                                 <1>;
                                          or eax, IOC+BUP
2314
                                 <1>;
                                          stosd
2315
                                 <1>;
                                          loop ac97_set_bdl_buffer
2316
                                 <1> ; tell the DMA engine where to find our list of Buffer Descriptors.
2317
2318
                                 <1>; this 32bit value is a flat mode memory offset (ie no segment:offset)
2319
                                 <1>; write NABMBAR+10h with offset of buffer descriptor list
2320
2321
                                 <1>;
                                           mov eax, audio_bdl_buff
2322 00012477 B8[7C6B0100]
                                <1>
2323 0001247C 668B15[7A6B0100]
                                <1>
                                          mov dx, [NABMBAR]
2324 00012483 6683C210
                                 <1>
                                          add
                                                dx, PO BDBAR REG
2325 00012487 EF
                                 <1>
                                          out
                                                dx, eax
                                 <1>;
2327
                                 <1> ; All set. Let's play some music.
2328
                                 <1>;
                                 <1>;
2330 00012488 B81F000000
                                 <1>
                                          mov
                                                eax, 31
2331 0001248D E816000000
                                                 set ac97 LastValidIndex
                                 <1>
                                          call
2332
                                 <1>
2333 00012492 C605[746B0100]01
                                 <1>
                                          mov byte [audio_play_cmd], 1 ; play command (do not stop) !
2334
                                 <1>
2335
                                 <1> ac97_play: ; continue to play (after pause)
                                      ; 11/06/2017
2336
                                 <1>
                                          ; 29/05/2017
2337
                                 <1>
                                        ; 28/05/2017

mov dx, [NABMBAR]

add dx, PO_CR_REG
2338
                                 <1>
2339 00012499 668B15[7A6B0100]
                                 <1>
                                                  dx, PO CR REG
2340 000124A0 6683C21B
                                 <1>
                                                                          ; PCM out control register
                                          mov al, IOCE+RPBM; 29/05/2017
2341 000124A4 B011
                                 <1>
2342
                                 <1>
                                            ;mov al, 1Dh ; (Ref: KolibriOS, intelac97.asm, 'play:')
                                          out dx, al
2343 000124A6 EE
                                 <1>
2344
                                 <1>
                                          ;mov byte [audio play cmd], 1 ; play command (do not stop) !
2345
                                 <1>
                                 <1>
2347 000124A7 C3
                                 <1>
                                          retn
2348
                                 <1>
2349
                                 <1> ;input AL = index # to stop on
                                 <1> set_ac97_LastValidIndex:
2350
2351
                                 <1>
                                          ; 28/05/2017
                                          ; Derived from 'setLastValidIndex' procedure in 'ICHWAV.ASM'
2352
                                 <1>
2353
                                 <1>
                                          ; .wav player for DOS by Jeff Leyda (02/09/2002)
2354 000124A8 668B15[7A6B0100]
                                 <1>
                                          mov dx, [NABMBAR]
2355 000124AF 6683C215
                                <1>
                                          add
                                                dx, PO_LVI_REG
2356 000124B3 EE
                                 <1>
                                          out dx, al
2357
                                <1>
                                          ;mov [audio_lvi], al ; for ac97_int_handler
2358 000124B4 C3
                                <1>
2359
                                <1>
                                 <1> ac97_volume:
2360
2361
                                 <1>
                                          ; 28/05/2017
2362
                                 <1>
                                          ; bl = component (0 = master/playback/lineout volume)
2363
                                 <1>
                                          ; cl = left channel volume level (0 to 31)
2364
                                 <1>
                                          ; ch = right channel volume level (0 to 31)
2365
                                 <1>
2366 000124B5 08DB
                                <1>
                                          or
                                                short ac97_vol_1 ; temporary !
2367 000124B7 7523
                                <1>
                                          jnz
2368 000124B9 66B81F1F
                                <1>
                                          mov
                                                 ax, 1F1Fh ; 31,31
2369 000124BD 38C1
                                <1>
                                          cmp
                                                cl, al
2370 000124BF 771B
                                                 short ac97_vol_1 ; temporary !
                                <1>
                                          jа
2371 000124C1 38E5
                                <1>
                                                 ch, ah
                                          cmp
2372 000124C3 7717
                                <1>
                                          jа
                                                 short ac97_vol_1 ; temporary !
2373 000124C5 66890D[766B0100] <1>
                                                 [audio master volume], cx
2374 000124CC 6629C8
                                <1>
                                          sub
                                                ax, cx
```

```
2375 000124CF 668B15[786B0100] <1>
                                         mov
                                                  dx, [NAMBAR]
2376 000124D6 6683C202
                                <1>
                                          add
                                                  dx, CODEC MASTER VOL REG ; 02h ; Line Out
2377 000124DA 66EF
                                <1>
                                         out
                                                  dx, ax
2378
                                <1> ac97_vol_1:
2379 000124DC C3
                                <1>
                                         retn
2380
                                <1>
                                <1> ac97_int_handler:
2381
                                      ; 12/10/2017
2382
                                <1>
2383
                                <1>
                                          ; 10/10/2017
                                         ; 09/10/2017
2384
                                <1>
2385
                                <1>
                                        ; 13/06/2017, 13/06/2017
                                        ; 10/06/2017, 11/06/2017
2386
                                 <1>
                                         ; Interrupt Handler for AC97 (ICH) Audio Controller
2387
                                <1>
2388
                                <1>
                                        ; Note: called by 'dev_IRQ_service'
                                         ; 28/05/2017
2389
                                 <1>
2390
                                 <1>
                                         ; push eax ; * must be saved !
2391
                                 <1>
2392
                                 <1>
                                          ;push edx
2393
                                 <1>
                                          ;push ecx
                                          ;push ebx ; * must be saved !
2394
                                <1>
2395
                                <1>
                                          ;push esi
                                          ;push edi
2396
                                 <1>
2397
                                <1>
2398
                                <1>
                                          ;cmp byte [audio_busy], 1
2399
                                <1>
                                                _ac97_ih2 ; busy !
                                          ;jnb
2400
                                <1>
2401 000124DD 66BA3000
                                <1>
                                          mov
                                                   dx, GLOB STS REG
                                          add dx, [NABMBAR]
2402 000124E1 660315[7A6B0100] <1>
2403 000124E8 ED
                                <1>
                                          in eax, dx
2404
                                <1>
                                         cmp
2405 000124E9 83F8FF
                                <1>
                                                    eax, Offffffff ; -1
2406 000124EC 0F849A000000
                                <1>
                                                    _ac97_ih3 ; exit
                                          jе
2407
                                <1>
2408 000124F2 A94000000
                                <1>
                                          test
                                                    eax, 40h; PCM Out Interrupt
                                          jnz
2409 000124F7 750E
                                <1>
                                                    short _ac97_ih0
2410
                                <1>
2411 000124F9 85C0
                                         test eax, eax
                                <1>
2412 000124FB 0F848B000000
                                         jz _ac97_ih3 ; exit
                                <1>
2413
                                <1>
                                          ;mov dx, GLOB STS REG
2414
                                <1>
2415
                                <1>
                                          ; add dx, [NABMBAR]
2416 00012501 EF
                                          out dx, eax
                                <1>
2417
                                <1>
                                          jmp _ac97_ih3 ; exit
2418 00012502 E985000000
                                <1>
2419
                                <1>
2420
                                <1> _ac97_ih0:
2421 00012507 50
                                <1>
                                      push
                                                eax
2422
                                <1>
                                          ; 09/10/2017
2423 00012508 803D[746B0100]01
                                <1>
                                          cmp
                                                byte [audio_play_cmd], 1
2424 0001250F 727C
                                <1>
                                                short _ac97_ih4 ; stop command !
                                          jb
2425
                                <1>
2426
                                 <1>
                                         ;mov byte [audio_busy], 1
2427
                                <1>
2428
                                 <1>
                                          ;mov al, 10h
2429
                                 <1>
                                          ;mov dx, PO_CR_REG
2430
                                <1>
                                          ;add dx, [NABMBAR]
2431
                                <1>
                                          ;out dx, al
2432
                                <1>
                                         mov ax, 1Ch ; FIFOE (=16) +BCIS (=8) +LVBCI (=4)
mov dx, PO_SR_REG
2433 00012511 66B81C00
                                <1>
2434 00012515 66BA1600
                                <1>
2435 00012519 660315[7A6B0100] <1>
                                          add dx, [NABMBAR]
2436 00012520 66EF
                                <1>
                                         out dx, ax
2437
                                <1>
2438 00012522 66BA1400
                                <1>
                                          mov dx, PO_CIV_REG
2439 00012526 660315[7A6B0100]
                                <1>
                                          add dx, [NABMBAR]
2440 0001252D EC
                                <1>
                                          in
                                                al, dx
2441
                                <1>
2442
                                <1>
                                          ;cmp al, [audio_civ] ; [audio_flag]
2443
                                <1>
                                                short _ac97_ih2
                                          ;je
2444
                                <1>
2445 0001252E A2[756B0100]
                                <1>
                                          mov
                                                [audio_civ], al
2446 00012533 FEC8
                                <1>
                                          dec
                                                al
                                          ;inc al ; 11/06/2017
2447
                                <1>
2448 00012535 241F
                                <1>
                                               al, 1Fh
2449
                                <1>
2450 00012537 66BA1500
                                                   dx, PO LVI REG
                                <1>
                                          mov
                                         add dx, [NABMBAR]
2451 0001253B 660315[7A6B0100] <1>
2452 00012542 EE
                                <1>
                                          out dx, al
2453
                                <1>
2454
                                          ; 12/10/2017
                                <1>
2455 00012543 A0[756B0100]
                                          mov al, [audio_civ]
                                <1>
2456 00012548 FEC0
                                 <1>
                                          inc
                                                al
2457 0001254A 2401
                                               al, 1
                                <1>
                                          and
2458 0001254C A2[686B0100]
                                <1>
                                          mov [audio_flag], al
                                <1>
                                          ;; [audio_flag] : 0 = Buffer 1, 1 = Buffer 2
2459
2460
                                 <1>
2461 00012551 58
                                <1>
                                          pop
                                <1>
2462
2463 00012552 83E040
                                <1>
                                          and
                                                 eax, 40h
                                          mov dx, [NABMBAR]
2464 00012555 668B15[7A6B0100]
                                <1>
                                          add dx, GLOB STS REG
2465 0001255C 6683C230
                                <1>
2466 00012560 EF
                                 <1>
                                          out
                                               dx, eax
2467
                                <1>
                                          ;; 13/06/2017
2468
                                <1>
                                          ;mov al, 11h ; IOCE + RPBM
;mov dx, PO_CR_REG
2469
                                 <1>
2470
                                 <1>
2471
                                 <1>
                                          ; add dx, [NABMBAR]
2472
                                <1>
                                          ;out dx, al
2473
                                 <1>
2474
                                <1> ac97_tuneloop:
2475
                                <1>
                                         ; 09/10/2017
2476 00012561 8B3D[606B0100]
                                <1>
                                          mov
                                                edi, [audio_dma_buff]
2477 00012567 8B0D[646B0100]
                                <1>
                                          mov
                                                ecx, [audio_dmabuff_size]
2478 0001256D D1E9
                                 <1>
                                          shr ecx, 1; dma buff size / 2 = half buffer size
2479
                                 <1>
```

```
; 12/10/2017
                                           cmp byte [audio_flag], 0
2481 0001256F 803D[686B0100]00
                                 <1>
                                                short _ac97_ih1 ; Playing Half Buffer 2 (Current: FLAG)
2482 00012576 7702
                                 <1>
                                           jа
2483
                                 <1>
                                           ; Playing Half Buffer 1 (Current: EOL)
2484 00012578 01CF
                                 <1>
                                           add
                                                edi, ecx
2485
                                 <1> _ac97_ih1:
                                        ; Update half buffer 2 while playing half buffer 1 (next: FLAG)
2486
                                 <1>
                                          ; Update half buffer 1 while playing half buffer 2 (next: EOL)
2487
                                 <1>
2488
                                 <1>
                                                 esi, [audio_p_buffer] ; phy addr of audio buff
2489 0001257A 8B35[586B0100]
                                 <1>
2490 00012580 C1E902
                                 <1>
                                           shr
                                                ecx, 2; half buff size / 4
2491 00012583 F3A5
                                 <1>
                                           rep
                                                movsd
2492
                                 <1>
                                          ; 10/10/2017
2493
                                 <1>
2494
                                 <1>
                                          ; switch flag value
2495 00012585 8035[686B0100]01
                                 <1>
                                          xor byte [audio_flag], 1
                                          ; 12/10/2017
                                 <1>
                                           ; [audio_flag] = 0 : Playing dma half buffer 2 (even index value)
2497
                                 <1>
                                                          ; Next buffer (to update) is dma half buff 1
2498
                                 <1>
2499
                                 <1>
                                                        = 1 : Playing dma half buffer 1 (odd index value)
                                                          ; Next buffer (to update) is dma half buff 2
2500
                                 <1>
2501
                                 <1>
2502
                                 <1> _ac97_ih2:
2503
                                 <1>
                                          ;mov
                                                byte [audio_busy], 0
2504
                                 <1> _ac97_ih3:
2505
                                 <1>
                                                 edi
                                           ;pop
2506
                                 <1>
                                           ;pop esi
                                          ;pop ebx; * must be restored!
                                 <1>
2507
2508
                                 <1>
                                           ;pop
                                                 ecx
2509
                                 <1>
                                                 edx
                                           ;pop
2510
                                 <1>
                                                 eax ; * must be restored !
                                           ;pop
2511
                                 <1>
2512 0001258C C3
                                 <1>
                                           retn
2513
                                 <1>
2514
                                 <1> _ac97_ih4:
                                          ; 09/10/2017
2515
                                 <1>
2516 0001258D E818000000
                                 <1>
                                           call _ac97_stop
2517
                                 <1>
2518 00012592 58
                                 <1>
                                          pop
2519
                                 <1>
2520 00012593 83E040
                                 <1>
                                           and
                                                  eax, 40h
                                           mov dx, [NABMBAR]
2521 00012596 668B15[7A6B0100]
                                 <1>
2522 0001259D 6683C230
                                           add dx, GLOB_STS_REG
                                 <1>
2523 000125A1 EF
                                 <1>
                                          out dx, eax
2524
                                 <1>
                                           ;; 13/06/2017
2525
                                 <1>
                                          ;mov al, 11h ; IOCE + RPBM
2526
                                 <1>
                                           ;dx, PO_CR_REG
2527
                                 <1>
2528
                                 <1>
                                            ; add dx, [NABMBAR]
2529
                                 <1>
                                           ;out dx, al
2530
                                 <1>
2531
                                 <1>
                                           ; 10/10/2017
                                           ;jmp short _ac97_ih3 ; exit
2532
                                 <1>
2533 000125A2 C3
                                 <1>
2534
                                 <1>
2535
                                 <1> ac97_stop:
                                          ; 28/05/2017
                                 <1>
2537 000125A3 C605[746B0100]00
                                 <1>
                                          mov byte [audio_play_cmd], 0 ; stop !
2538
                                 <1> _ac97_stop: ; 09/10/2017
                                       ; 29/05/2017
2539
                                 <1>
2540
                                 <1>
                                          ;mov dx, [NABMBAR]
2541
                                 <1>
                                          ;add dx, PO CR REG
2542
                                 <1>
                                          ;mov al, 0
2543
                                 <1>
                                          ;out dx, al
2544
                                 <1>
2545
                                 <1>
                                           ; 11/06/2017
2546 000125AA 30C0
                                 <1>
                                          xor al, al; 0
2547 000125AC E813000000
                                 <1>
                                           call ac97_po_cmd
2548
                                 <1>
                                          ; (Ref: KolibriOS, intelac97.asm, 'stop:')
2549
                                 <1>
2550
                                 <1>
                                           ; Clear FIFOE, BCIS, LVBCI (Ref: Intel ICH hub manual)
2551 000125B1 66B81C00
                                 <1>
                                           mov
                                                ax, 1Ch
2552 000125B5 668B15[7A6B0100]
                                                   dx, [NABMBAR]
                                 <1>
                                           mov
2553 000125BC 6683C216
                                 <1>
                                           add
                                                  dx, PO_SR_REG
                                                  dx, ax
2554 000125C0 66EF
                                 <1>
                                           out
2555
                                 <1>
2556
                                 <1>
                                           ;retn
2557
                                 <1>
2558
                                 <1>
                                           ; 11/06/2017
2559 000125C2 B002
                                          mov al, RR
                                 <1>
                                 <1> ac97_po_cmd:
2560
                                          ;11/06/2017
2561
                                 <1>
                                           ; 29/05/2017
                                 <1>
                                           mov dx, [NABMBAR] add dx, PO_CR_REG
2563 000125C4 668B15[7A6B0100]
                                 <1>
2564 000125CB 6683C21B
                                 <1>
                                                                         ; PCM out control register
2565 000125CF EE
                                 <1>
                                           out dx, al
2566 000125D0 C3
                                 <1>
                                          retn
2567
                                 <1>
2568
                                 <1> ac97_pause:
                                          ; 11/06/2017
2569
                                <1>
                                           ; 29/05/2017
2570
                                 <1>
                                 <1>
2571 000125D1 B010
                                          mov al, IOCE
2572 000125D3 EBEF
                                 <1>
                                                short ac97_po_cmd
                                           jmp
2573
                                 <1>
                                 <1> reset ac97 controller:
2574
2575
                                 <1>
                                          ; 10/06/2017
2576
                                          ; 29/05/2017
                                 <1>
2577
                                         ; 28/05/2017
                                 <1>
2578
                                 <1>
                                          ; reset AC97 audio controller registers
2579 000125D5 31C0
                                         xor eax, eax
                                <1>
2580 000125D7 66BA0B00
                                          mov dx, PI_CR_REG
                                <1>
2581 000125DB 660315[7A6B0100] <1>
                                          add dx, [NABMBAR]
2582 000125E2 EE
                                <1>
                                          out
                                                 dx, al
2583
                                 <1>
2584 000125E3 66BA1B00
                                 <1>
                                          mov dx, PO_CR_REG
```

2480

```
add dx, [NABMBAR]
2585 000125E7 660315[7A6B0100] <1>
2586 000125EE EE <1>
<1>
<1>
<1>
                                                           out dx, al
2588 000125EF 66BA2B00
                                                              mov dx, MC CR REG
                                               <1>
2589 000125F3 660315[7A6B0100] <1>
                                                             add dx, [NABMBAR]
2590 000125FA EE
                                               <1>
                                                              out
                                                                      dx, al
2591
                                                <1>
2593 000125FD 66BA0B00
2592 000125FB B002
                                                             mov
                                                                         al, RR
dx, PI_CR_REG
                                               <1>
                                               <1>
                                                             add dx, [NABMBAR]
2594 00012601 660315[7A6B0100] <1>
2595 00012608 EE
                                               <1>
                                                         out dx, al
2596
                                                <1>
2597 00012609 66BA1B00
                                                               mov dx, PO_CR_REG
                                               <1>
2600
                                               <1>
                                                        mov dx, MC_CR_
add dx, [NABMBAR]
                                                              mov dx, MC_CR_REG
2601 00012615 66BA2B00
                                               <1>
2602 00012619 660315[7A6B0100] <1>
2603 00012620 EE
                                                <1>
                                                              out
                                                                      dx, al
2604
                                                <1>
2605 00012621 C3
                                                <1>
                                                              retn
2606
                                                <1>
                                                <1> ac97_reset:
2607
2608
                                                <1> ; 10/06/2017
                                                        ; 29/05/2017
; 28/05/2017
call reset_ac97_controller
2609
                                                <1>
2610
                                                <1>
2611 00012622 E8AEFFFFFF
                                               <1>
                                               <1> ; 29/05/2017 
<1> ; jmp reset_ac97_codec
2612
                                               <1>
2613
2614
                                               <1> reset_ac97_codec:
                                               <1> ; 28/05/2017 - Erdogan Tan (Ref: KolibriOS, intelac97.asm)
2615
2616 00012627 66BA2C00
                                                             mov dx, GLOB_CNT_REG; 2Ch add dx, [NABMBAR]
                                                <1>
2617 0001262B 660315[7A6B0100] <1>
                                                           in eax, dx
2618 00012632 ED
                                               <1>
2619
                                                <1>
                                              test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
test eax, 2
t
2620 00012633 A902000000
call cold_ac97codec_reset jnc short _r_ac97codec_ol
                                                                            short r ac97codec ok
                                                           ; 16/04/2017
                                                           ;xor eax, eax ; timeout error
2630
                                               <1>
                                                                 ;stc
                                               <1>
2632 00012648 C3
                                               <1>
                                                              retn
2633
                                                <1>
                                             <1> _r_ac97codec_ok:
2634
                                          <1>
                                                        xor eax, eax
2635 00012649 31C0
2636
                                               <1>
                                                                 ;moval, VIA ACLINK COO READY ; 1
2637 0001264B FEC0
                                               <1>
                                                                 inc al
2638 0001264D C3
                                               <1>
2639
                                                <1>
2640
                                                <1> warm_ac97codec_reset:
                                               ; 28/05/2017 - Erdogan Tan (Ref: KolibriOS, intelac97.asm)
2643 00012653 66BA2C00
2644 00012657 6663
                                               <1>
                                                             mov eax, 6
                                                        mov dx, GLOB_CNT_F
add dx, [NABMBAR]
                                                <1>
                                                             mov dx, GLOB_CNT_REG; 2Ch
2644 00012657 660315[7A6B0100] <1>
                                                           out dx, eax
2645 0001265E EF
                                               <1>
2646
                                                <1>
2647 0001265F B90A000000 <1>
2648 <1> w
                                                          mov
                                                                      ecx, 10 ; total 1s
2648
                                               <1> _warm_ac97c_rst_wait:
2649 00012664 51
                                               <1>
                                                       push ecx
2650 00012665 E8ECF5FFFF
                                               <1>
                                                              call delay_100ms
                                                           pop ecx
2651 0001266A 59
                                               <1>
2652
                                               <1>
2653 0001266B 66BA3000
                                               <1>
                                                             mov
                                                                       dx, GLOB_STS_REG ; 30h
                                                       add dx, [NABMBAR]
2654 0001266F 660315[7A6B0100] <1>
                                                           in
2655 00012676 ED
                                               <1>
                                                                       eax, dx
2656
                                                <1>
2657 00012677 A900030010
                                      <1>
<1>
                                                              test eax, CTRL_ST_CREADY
                                                          jnz short _warm_ac97c_rst_ok
2658 0001267C 7504
2659
                                               <1>
2660 0001267E 49
                                                               dec
                                               <1>
                                                                             ecx
                                                                         short _warm_ac97c_rst_wait
2661 0001267F 75E3
                                               <1>
                                                             jnz
2662
                                                <1>
                                                <1> _warm_ac97c_rst_fail: <1> _ stc
2663
2664 00012681 F9
                                                 <1> _warm_ac97c_rst_ok:
2665
                                                           retn
2666 00012682 C3
                                                 <1>
2667
                                                 <1>
                                                 <1> cold ac97codec reset:
2668
                                                         \frac{1}{28/05/2017} - Erdogan Tan (Ref: KolibriOS, intelac97.asm)
                                                <1>
2669
2670 00012683 B802000000
                                                                mov eax, 2
                                                <1>
                                                              mov dx, GLOB CNT REG; 2Ch
2671 00012688 66BA2C00
                                                <1>
                                                              add
2672 0001268C 660315[7A6B0100]
                                               <1>
                                                                       dx, [NABMBAR]
2673 00012693 EF
                                                <1>
                                                                       dx, eax
                                                              out
2674
                                                <1>
2675 00012694 E8BDF5FFFF
                                                           call delay_100ms ; wait 100 ms
                                                <1>
                                                             call delay_100ms; wait 100 ms call delay_100ms; wait 100 ms
2676 00012699 E8B8F5FFFF
                                                <1>
2677 0001269E E8B3F5FFFF
                                                <1>
2678 000126A3 E8AEF5FFFF
                                                <1>
                                                              call delay_100ms ; wait 100 ms
2679
                                                <1>
2680 000126A8 B910000000
                                                <1>
                                                              mov
                                                                       ecx, 16
                                                                                         ; total 20*100 \text{ ms} = 2s
                                                <1> cold ac97c rst wait:
                                                              2682 000126AD 66BA3000
                                                <1>
2683 000126B1 660315[7A6B0100]
                                               <1>
                                                               add
                                                                       dx, [NABMBAR]
2684 000126B8 ED
                                                              in
                                                <1>
                                                                       eax, dx
2685
                                                <1>
2686 000126B9 A900030010
                                                <1>
                                                              test eax, CTRL ST CREADY
2687 000126BE 750B
                                               <1>
                                                              jnz
                                                                       short _cold_ac97c_rst_ok
2688
                                                <1>
2689 000126C0 51
                                                <1>
                                                              push ecx
```

```
<1>
                                          call delay_100ms
2690 000126C1 E890F5FFF
                                           pop ecx
2691 000126C6 59
                                   <1>
2692
                                   <1>
2693 000126C7 49
                                  <1>
                                              dec
2694 000126C8 75E3
                                   <1>
                                              jnz
                                                         short _cold_ac97c_rst_wait
2695
                                   <1>
                                   <1> _cold_ac97c_rst_fail:
2696
2697 000126CA F9
                                   <1>
                                              stc
2698
                                   <1> _cold_ac97c_rst_ok:
2699 000126CB C3
                                   <1>
                                            retn
2700
                                   <1>
2701
                                    <1> sb16 current_sound_data:
                                         ; 20/08/2017
2702
                                   <1>
2703
                                   <1>
                                             ; 24/06/2017
                                            ; 22/06/2017
2704
                                    <1>
                                            ; get current sound (PCM out) data for graphics
; (for Sound Blaster 16)
2705
                                   <1>
2706
                                    <1>
2707
                                    <1>
                                            ; ebx = Physical address (on page boundary)
2708
                                    <1>
                                             ; ecx = Byte count
                                             ; [audio buff size]
2709
                                   <1>
2710
                                    <1>
2711
                                    <1>
                                             ;; mov edi, [audio_buff_size]
                                             ;mov edi, [audio_dmabuff_size]
2712
                                   <1>
2713
                                   <1>
                                              ;mov esi, [audio_dma_buff]
                                                    edi, ecx
2714 000126CC 39CF
                                   <1>
                                              cmp
2715 000126CE 7302
                                   <1>
                                              jnb
                                                    short sb16_gcd_0
2716 000126D0 89F9
                                   <1>
                                             mov ecx, edi
                                   <1> sb16_gcd_0:
2717
                                   <1> ; 20/08/2017
2718
                            Jne short sb16_gcd_1; 8 bit DMA channel
in al, 0C6h; DMA channel 5 count registe
cl> mov dl, al
cl> in al, 0C6h
cl> mov dh, al
cl> movzx eax, dx
cl> shl eax, 1; word count -> byte count
cl> jmp short sb16_gcd_2
cl> sb16_gcd_1:
cl> in al, 03h; DMA channel 1 count register
cl> mov dl, al
cl> in al, 03h; DMA channel 1 count register
cl> mov dl, al
cl> in al, 03h
2719 000126D2 803D[706B0100]10 <1>
                                             cmp byte [audio_bps], 16
2720 000126D9 750F
2721 000126DB E4C6
                                                     al, OC6h; DMA channel 5 count register
2722 000126DD 88C2
2723 000126DF E4C6
2724 000126E1 88C6
2725 000126E3 0FB7C2
2726 000126E6 D1E0
2727 000126E8 EB4E
2728
2729 000126EA E403
2730 000126EC 88C2
                                  2731 000126EE E403
2732 000126F0 88C6
2733 000126F2 0FB7C2
                                              jmp short sb16_gcd_2
2734 000126F5 EB41
                                   <1>
                                   <1> ;sb16_gcd_2:
2735
                                   <1>; cmp eax, ecx
<1>; jnb short sb16_gcd_3
2736
2737
                                              ; remain count < graphics bytes
2738
                                    <1>;
                                           mov eax, ecx; fix remain count to data size
2739
                                    <1>;
                                    <1> ;sb16_gcd_3:
2740
                                    <1>; sub edi, eax <1>; jna short sb16_gcd_4
2741
2742
2743
                                    <1>;
                                              add esi, edi ; dma buffer offset
                                    <1> ;sb16_gcd_4:
2744
                                    <1>;
2745
                                             mov edi, ebx; buffer address (for graphics)
2746
                                    <1>;
                                              mov
                                                    [u.r0], ecx
2747
                                    <1>;
                                              rep
                                                    movsb
2748
                                    <1>;
                                              retn
2749
                                    <1>
2750
                                    <1> get_current_sound_data:
                                         ; 24/06/2017
2751
                                    <1>
2752
                                    <1>
                                              ; 22/06/2017
2753
                                    <1>
                                             ; get current sound (PCM out) data for graphics
2754
                                    <1>
2755
                                    <1>
                                             ; ebx = Physical address (on page boundary)
                                             ; ecx = Byte count
2756
                                    <1>
2757
                                    <1>
                                             ; [audio_buff_size]
2758
                                    <1>
2759
                                   <1>
                                             ;mov edi, [audio_buff_size]
2760 000126F7 8B3D[646B0100]
                                   <1>
                                              mov
                                                     edi, [audio_dmabuff_size]
                                                     esi, [audio_dma_buff]
2761 000126FD 8B35[606B0100]
                                    <1>
                                              mov
                                                    byte [audio_device], 2
2762 00012703 803D[416B0100]02
                                   <1>
                                              cmp
2763 0001270A 72C0
                                   <1>
                                              jb
                                                     short sb16_current_sound_data ; = 1
                                                     edi, 1
2764 0001270C D1EF
                                    <1>
                                              shr
2765 0001270E 39CF
                                   <1>
                                              cmp
                                                     edi, ecx
2766 00012710 7302
                                   <1>
                                              jnb
                                                     short gcd 0
2767 00012712 89F9
                                    <1>
                                              mov
                                                     ecx, edi
2768
                                    <1> gcd_0:
2769 00012714 803D[416B0100]03
                                              cmp
                                                     byte [audio_device], 3
                                   <1>
                                                     short ac97_current_sound_data ; = 2
2770 0001271B 7232
                                    <1>
                                              jb
2771
                                    <1>
2772
                                    <1> vt8233_current_sound_data:
2773
                                             ; 22/06/2017
2774
                                    <1>
                                              ; 21/06/2017
2775
                                              ; get current sound (PCM out) data for graphics
                                    <1>
2776
                                    <1>
                                              ; (for VT 8233, VT 8237R)
2777
                                    <1>
                                              ; ebx = Physical address (on page boundary)
2778
                                    <1>
                                              ; ecx = Byte count
2779
                                    <1>
                                              ; [audio_buff_size]
2780
                                    <1>
2781
                                    <1>
                                              ;;mov edi, [audio_buff_size]
                                                     edi, [audio_dmabuff_size]
2782
                                    <1>
                                              ;mov
                                                     esi, [audio dma buff]
2783
                                    <1>
                                                     edi, 1
2784
                                    <1>
                                              ;shr
2785
                                    <1>
                                              ;cmp
                                                     edi, ecx
2786
                                    <1>
                                                    short vt8233_gcd_1
                                              ;jnb
2787
                                    <1>
                                              ;mov ecx, edi
2788
                                    <1> vt8233_gcd_1:
2789 0001271D BA0C000000
                                              mov edx, VIA_REG_OFFSET_CURR_COUNT
                                   <1>
2790 00012722 E88FF5FFF
                                   <1>
                                              call ctrl_io_r32
2791 00012727 89C2
                                    <1>
                                              mov
                                                     edx, eax; remain count (bits 23-0),
                                    <1>
                                                             ; SGD index (bits 31-24)
2793 00012729 81E20000001
                                                     edx, 1000000h; SGD index (0 = 1st half)
                                    <1>
2794 0001272F 7402
                                    <1>
                                              jz
                                                     short vt8233_gcd_2
```

```
<1> ; the second half of DMA buffer
<1> add eei ca'
2795
2796 00012731 01FE
2797
                               <1> vt8233_gcd_2:
2798 00012733 25FFFFFF00
                               <1> and eax, OFFFFFFh; bits 23-0
2799
                                <1> ac97 gcd 2:
                                <1> sb16_gcd_2:
2800
                               2801 00012738 39C8
2802 0001273A 7302
                            <1> ; remain count < graphics bytes
<1> mov eax, ecx; fix remain count to data size
<1> vt8233_gcd_3:
2803
2804 0001273C 89C8
2805
                               2806 0001273E 29C7
                               <1>
2807 00012740 7602
                     <1> jna short vt8233_gcd_4
<1> add esi, edi; dma buffer offset
<1> vt8233_gcd_4:
<1> mov edi. ebx : buffer address (form)
2808 00012742 01FE
2809
                               2810 00012744 89DF
2811 00012746 890D[64030300] <1>
2812 0001274C F3A4
                               <1>
                                         rep
                                               movsb
2813
                                <1> vt8233_gcd_5:
2814 0001274E C3
                                <1>
                                        retn
2815
                                <1>
2816
                                <1> ac97_current_sound_data:
                                <1> ; 23/06/2017
<1> ; 22/06/2017
2817
2818
                                        ; get current sound (PCM out) data for graphics
2819
                                <1>
                                        ; (for AC'97, ICH)
; ebx = Physical address (on page boundary)
2820
                                <1>
2821
                                <1>
                                        ; ecx = Byte count
                                <1>
2822
2823
                                <1>
                                         ; [audio_buff_size]
                                <1>
2824
                                       ;;mov edi, [audio_buff_size]
2825
                                <1>
                                         ;mov edi, [audio_dmabuff_size]
;mov esi, [audio_dma_buff]
2826
                                <1>
2827
                                <1>
2828
                                <1>
                                         shr edi, 1;
2829
                                <1>
                                         ;cmp edi, ecx
2830
                                <1>
                                         ;jnb short ac97_gcd_0
2831
                                <1>
                                         ;mov ecx, edi
                                <1> ac97_gcd_0:
2832
                                <1>
2833 0001274F 66BA1400
                                         mov dx, PO_CIV_REG; Position In Current Buff Reg
2834 00012753 660315[7A6B0100] <1>
                                         add dx, [NABMBAR]
2835 0001275A EC
                               <1>
                                         in al, dx ; current index value
                                         test al, 1
jz short
2836 0001275B A801
                                <1>
2837 0001275D 7402
                               <1>
                                               short ac97_gcd_1
                               <1> jz snort ac: <1> add esi, edi
2838 0001275F 01FE
                                <1> ac97_gcd_1:
2839
2841 00012763 66BA1800
                               <1> xor eax, eax
<1> mov dx, PO_PICB_REG; Position In Current Buff Reg
                               <1>
jmp short ac97_gcd_2
2845 00012773 EBC3
                                <1>
2846
                                <1>;
                                         cmp
                                               eax, ecx
                                          jnb short ac97_gcd_2
2847
                                <1>;
2848
                                <1>;
                                         ; remain count < graphics bytes
2849
                                <1>;
                                         mov eax, ecx; fix remain count to data size
2850
                                <1> ;ac97_gcd_2:
2851
                                <1>; sub edi, eax
2852
                                               short ac97_gcd_3
                                <1>;
                                          jna
2853
                                <1>;
                                                esi, edi ; dma buffer offset
                                          add
2854
                                <1> ;ac97_gcd_3:
2855
                                <1>; mov edi, ebx; buffer address (for graphics)
2856
                                <1>;
                                         mov
                                                [u.r0], ecx
                                <1>;
2857
                                               movsb
                                         rep
2858
                                <1>;
2859
                                <1>
                                <1> sb16_get_dma_buff_off:
2860
                                2861
2862
                                <1>
                                         ; 24/06/2017
                                        ; 22/06/2017
; get current (PCM OUT DMA buffer) pointer
2863
                                <1>
2864
                                <1>
2865
                                <1>
                                        ; (for Sound Blaster 16)
2866
                                 <1>
2867
                                <1>
                                         ;mov ecx, [audio_dmabuff_size]
2868
                                <1>
                                         ;xor ebx, ebx
                                         ;shr ecx, 1
2869
                                <1>
                                <1> sb16_gdmabo_0:
2870
                                <1> = ; 28/10/2017
                                         cmp byte [audio_bps], 16
jne short sb16_gdmabo_1; 8 bit DMA channel
2872 00012775 803D[706B0100]10
                                <1>
                                        jne
2873 0001277C 750F
                                <1>
2874
                                <1>
                                         ; 16 bit DMA channel
2875 0001277E E4C6
                                         in al, OC6h; DMA channel 5 count register
                                <1>
2876 00012780 88C2
                                <1>
                                          mov
                                                dl, al
2877 00012782 E4C6
                                <1>
                                          in
                                                al, 0C6h
2878 00012784 88C6
                                <1>
                                          mov
                                                 dh, al
2879 00012786 0FB7C2
                                <1>
                                          movzx eax, dx
2880 00012789 D1E0
                                <1>
                                          shl eax, 1; word count -> byte count
2881 0001278B EB3D
                               <1>
                                               short sb16_gdmabo_2
                                          qmŗ
2882
                                <1> sb16_gdmabo_1:
2883 0001278D E403
                                                al, 03h ; DMA channel 1 count register
                                <1>
                                          in
2884 0001278F 88C2
                               <1>
                                                dl, al
                                          mov
2885 00012791 E403
                                <1>
                                          in
                                                al, 03h
2886 00012793 88C6
                                <1>
                                          mov
                                                dh, al
2887 00012795 0FB7C2
                                <1>
                                          movzx eax, dx
2888 00012798 EB30
                                <1>
                                          jmp short sb16_gdmabo_2
2889
                                <1>
                                <1> get dma buffer offset:
2890
2891
                                <1>
                                         ; 24/06/2017
                                          ; 22/06/2017
2892
                                <1>
2893
                                <1>
                                          ; get current sound (PCM out) data for graphics
2894
                                <1>
2895
                                <1>
                                         ; ebx = Physical address (on page boundary)
2896
                                 <1>
                                         ; ecx = Byte count
2897
                                <1>
                                          ; [audio buff size]
                                 <1>
2898
2899 0001279A 8B0D[646B0100]
                                <1>
                                               ecx, [audio_dmabuff_size]
                                         mov
```

```
2900 000127A0 31DB
                                                 ebx, ebx
                                 <1>
                                         xor
2901
                                 <1> gdmabo 0:
2902 000127A2 803D[416B0100]02
                                <1> cmp
                                                byte [audio_device], 2
2903 000127A9 72CA
                                 <1>
                                           jb
                                                 short sb16_get_dma_buff_off
2904 000127AB 742A
                                 <1>
                                        jе
                                                 short ac97_get_dma_buff_off
2905
                                 <1>
2906
                                 <1> vt8233 get dma buff off:
                                       ; 24/06/2017
2907
                                 <1>
2908
                                 <1>
                                          ; 22/06/2017
                                         ; get current (PCM OUT DMA buffer) pointer
2909
                                 <1>
                                        ; (for VT 8233, VT 8237R)
2910
                                 <1>
2911
                                 <1>
2912
                                 <1>
                                         ;mov ecx, [audio_dmabuff_size]
                                      ;xor ebx, ebx shr ecx, 1
2913
                                 <1>
2914 000127AD D1E9
                                 <1>
2915
                                <1> vt8233_gdmabo_0:
                                <1> mov edx, VIA_REG_OFFSET_CURR_COUNT
2916 000127AF BA0C000000
                                      call ctrl_io_r32
2917 000127B4 E8FDF4FFF
                                <1>
2918 000127B9 89C2
                                <1>
                                         mov edx, eax; remain count (bits 23-0),
                           ; SGD index (bits 31-24)

1 and edx, 1000000h; SGD index (0 = 1st half)

2 short vt8233_gdmabo_1

3 the second half of DMA buffer

4 mov ebx, ecx
2919
2920 000127BB 81E20000001
2921 000127C1 7402
2922
2923 000127C3 89CB
2924
                                <1> vt8233_gdmabo_1:
2925 000127C5 25FFFFF00
                                <1> and eax, OFFFFFFh; bits 23-0
                                <1> sb16 gdmabo 2:
2927
                                <1> ac97_gdmabo_2:
                                <1>
2928 000127CA 29C1
                                           sub ecx, eax
jna short vt8233_gdmabo_2
2929 000127CC 7602
                                <1>
2930 000127CE 01CB
                                <1> add
                                                ebx, ecx; dma buffer offset
                                <1> vt8233 gdmabo 2:
2932 000127D0 891D[64030300]
                                <1> mov [u.r0], ebx
2933 000127D6 C3
                                <1>
2934
                                 <1>
2935
                                 <1> ac97_get_dma_buff_off:
                                 <1> ; 24/06/2017
2936
                                         ; 22/06/2017
2937
                                 <1>
                                         ; get current (PCM OUT DMA buffer) pointer
; (for AC'97, ICH)
2938
                                 <1>
2939
                                 <1>
2940
                                 <1>
                                         ; ebx = Physical address (on page boundary)
                                         ; ecx = Byte count
2941
                                 <1>
                                 <1>
2942
                                          ; [audio_buff_size]
2943
                                 <1>
                                        ;mov ecx, [audio_dmabuff_size]
;xor ebx, ebx
2944
                                 <1>
2945
                                 <1>
2946 000127D7 D1E9
                                          shr ecx, 1
                                <1>
                                <1> ac97_gdmabo_0:
2947
                                <1>
                                          mov dx, PO_CIV_REG; Position In Current Buff Reg add dx, [NABMBAR]
2948 000127D9 66BA1400
2949 000127DD 660315[7A6B0100] <1>
                                      in al, dx; current index value
2950 000127E4 EC
                                <1>
2951 000127E5 A801
                                <1>
                                          test al, 1
2952 000127E7 7402
                              <1> jz short ac97_gdmabo_1
<1> mov ebx, ecx
2953 000127E9 89CB
2954
                                <1> ac97_gdmabo_1:
                                <1> xor eax, eax <1> mov dx. PO P
2955 000127EB 31C0
2956 000127ED 66BA1800
                                          mov dx, PO PICB REG; Position In Current Buff Reg
                                <1>
2957 000127F1 660315[7A6B0100] <1>
                                          add dx, [NABMBAR]
2958 000127F8 66ED
                                 <1>
                                          in
                                                 ax, dx ; remain dwords
                                         jmp short ac97_gdmabo_2
2959 000127FA EBCE
                                <1>
2642
2643
                                  align 4
2644
2645
                                  %include 'vgadata.s'; 04/07/2016
                               1
  2
                               <1>; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.0 - vgadata.s (palette and fond data)
                               <1> ; Last Update: 04/07/2016
  5
                               <1> ; -----
  6
                               <1> ; Beginning: 16/01/2016
                               <1>; Assembler: NASM version 2.11 (trdos386.s)
  9
                               <1>: Turkish Rational DOS
 10
                               <1>; Operating System Project v2.0 by ERDOGAN TAN (Beginning: 04/01/2016)
 11
 12
 13
                               <1>; Derived from 'Plex86/Bochs VGABios' source code, vgabios-0.7a (2011)
                               <1>; by the LGPL VGABios Developers Team (2001-2008), 'vgatables.h'
 14
 15
                               <1>;
 16
                               <1> ; Oracle VirtualBox 5.0.24 VGABios Source Code
                                <1> ; ('vgabios.c', 'vgatables.h', 'vgafonts.h', 'vgarom.asm')
 17
 18
                               <1> ; Palette and font data in assembly language format:
 19
 20
                                <1>;
                                    'VBoxVgaBiosAlternative.asm'
 21
                                <1>
                                23
                               <1>
                               <1>: 04/07/2016
 24
 25
                               <1> ; COLOR DATA
 26
                               <1>
                               <1> palette0:
 28 000127FC 0000000000000000000 <1>
                                       db 000h, 000h
 28 00012805 00000000000000
                               <1>
 29 0001280C 00000000000000002A- <1>
                                       db 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 02ah, 02ah, 02ah, 02ah, 02ah, 02ah, 02ah, 02ah, 02ah
 29 00012815 2A2A2A2A2A2A2A
                               <1>
                                      db 02ah, 02ah
 30 0001281C 2A2A2A2A2A2A2A2A2A2A- <1>
 30 00012825 2A2A2A2A2A2A2A
                              <1>
 31 0001282C 2A2A2A2A2A2A2A2A2A2A- <1>
                                       db 02ah, 02ah
  31 00012835 2A2A2A2A2A2A2A
                               <1>
  32 0001283C 2A2A2A2A2A2A2A2A3F- <1>
                                       db 02ah, 02ah, 02ah, 02ah, 02ah, 02ah, 02ah, 02ah, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh
 32 00012845 3F3F3F3F3F3F3F
                              <1>
 33 0001284C 3F3F3F3F3F3F3F3F3F- <1>
                                       db 03fh, 03fh
 33 00012855 3F3F3F3F3F3F3F
                              <1>
 34 0001285C 0000000000000000000 <1>
                                       db 000h, 000h
 34 00012865 00000000000000
                              <1>
 35 0001286C 00000000000000002A- <1>
                                       db 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 02ah, 02ah, 02ah, 02ah, 02ah, 02ah, 02ah, 02ah
 35 00012875 2A2A2A2A2A2A2A
                              <1>
```

```
36 0001287C 2A2A2A2A2A2A2A2A2A2A- <1>
                                                     02ah, 02ah
36 00012885 2A2A2A2A2A2A2A
                                       <1>
37 0001288C 2A2A2A2A2A2A2A2A2A2A- <1>
                                                      02ah, 02ah
37 00012895 2A2A2A2A2A2A2A
                                       <1>
38 0001289C 2A2A2A2A2A2A2A2A3F- <1>
                                                      02ah, 02ah, 02ah, 02ah, 02ah, 02ah, 02ah, 02ah, 03fh, 
                                                 db
38 000128A5 3F3F3F3F3F3F3F
                                       <1>
39 000128AC 3F3F3F3F3F3F3F3F3F- <1>
                                                      03fh, 03fh
39 000128B5 3F3F3F3F3F3F3F
                                       <1>
                                       <1> palette1:
40
41 000128BC 00000000002A002A00- <1>
                                                      000h, 000h, 000h, 000h, 000h, 02ah, 000h, 02ah, 000h, 000h, 02ah, 02ah, 02ah, 02ah, 000h, 000h, 02ah
                                                 db
41 000128C5 002A2A2A00002A
                                       <1>
42 000128CC 002A2A15002A2A2A00- <1>
                                                      000h, 02ah, 02ah, 015h, 000h, 02ah, 02ah, 02ah, 000h, 000h, 000h, 000h, 000h, 02ah, 000h, 02ah
42 000128D5 000000002A002A
                                       <1>
43 000128DC 00002A2A2A00002A00- <1>
                                                      000h, 000h, 02ah, 02ah, 02ah, 000h, 000h, 000h, 02ah, 000h, 02ah, 015h, 000h, 02ah, 02ah, 02ah
                                                 db
43 000128E5 2A2A15002A2A2A
                                       <1>
44 000128EC 15151515153F153F15- <1>
                                                 db
                                                      015h, 015h, 015h, 015h, 015h, 03fh, 015h, 03fh, 015h, 03fh, 015h, 03fh, 03fh, 03fh, 03fh, 015h, 03fh
44 000128F5 153F3F3F15153F
                                       <1>
45 000128FC 153F3F3F153F3F3F15- <1>
                                                      015h, 03fh, 03fh, 03fh, 015h, 03fh, 03fh, 03fh, 015h, 015h, 015h, 015h, 015h, 03fh, 015h, 03fh
45 00012905 151515153F153F
                                       <1>
46 0001290C 15153F3F3F15153F15- <1>
                                                 db
                                                      015h, 015h, 03fh, 03fh, 03fh, 015h, 015h, 03fh, 015h, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh
46 00012915 3F3F3F153F3F3F
                                       <1>
47 0001291C 00000000002A002A00- <1>
                                                      000h, 000h, 000h, 000h, 000h, 002ah, 000h, 02ah, 000h, 000h, 02ah, 02ah, 02ah, 02ah, 000h, 000h, 02ah
47 00012925 002A2A2A00002A
                                       <1>
48 0001292C 002A2A15002A2A2A00- <1>
                                                      000h, 02ah, 02ah, 015h, 000h, 02ah, 02ah, 02ah, 000h, 000h, 000h, 000h, 02ah, 000h, 02ah
                                                 db
48 00012935 000000002A002A
                                       <1>
49 0001293C 00002A2A2A000002A00- <1>
                                                      000h, 000h, 02ah, 02ah, 02ah, 000h, 000h, 02ah, 000h, 02ah, 015h, 000h, 02ah, 02ah, 02ah
49 00012945 2A2A15002A2A2A
                                       <1>
50 0001294C 15151515153F153F15- <1>
                                                      015h, 015h, 015h, 015h, 015h, 03fh, 015h, 03fh, 015h, 015h, 03fh, 03fh, 03fh, 015h, 015h, 03fh
50 00012955 153F3F3F15153F
                                       <1>
51 0001295C 153F3F3F153F3F3F15- <1>
                                                      015h, 03fh, 03fh, 03fh, 015h, 03fh, 03fh, 03fh, 015h, 015h, 015h, 015h, 015h, 03fh, 015h, 03fh
51 00012965 151515153F153F
                                       <1>
52 0001296C 15153F3F3F15153F15- <1>
                                                      015h, 015h, 03fh, 03fh, 03fh, 015h, 015h, 03fh, 015h, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh
52 00012975 3F3F3F153F3F3F
                                       <1>
53
                                       <1> palette2:
54 0001297C 00000000002A002A00- <1>
                                                      000h, 000h, 000h, 000h, 000h, 002ah, 000h, 02ah, 000h, 000h, 02ah, 02ah, 02ah, 02ah, 000h, 000h, 02ah
                                                 db
54 00012985 002A2A2A00002A
                                       <1>
                                                      000h, 02ah, 02ah, 02ah, 000h, 02ah, 02ah, 02ah, 000h, 000h, 015h, 000h, 000h, 03fh, 000h, 02ah
55 0001298C 002A2A2A002A2A2A00- <1>
55 00012995 001500003F002A
                                       <1>
56 0001299C 15002A3F2A00152A00- <1>
                                                      015h, 000h, 02ah, 03fh, 02ah, 000h, 015h, 02ah, 000h, 03fh, 02ah, 02ah, 015h, 02ah, 03fh
56 000129A5 3F2A2A152A2A3F
                                       <1>
57 000129AC 00150000152A003F00- <1>
                                                      000h, 015h, 000h, 000h, 015h, 02ah, 000h, 03fh, 000h, 03fh, 02ah, 02ah, 015h, 000h, 02ah
57 000129B5 003F2A2A15002A
                                       <1>
58 000129BC 152A2A3F002A3F2A00- <1>
                                                      015h, 02ah, 02ah, 03fh, 000h, 02ah, 03fh, 02ah, 000h, 015h, 015h, 000h, 015h, 03fh, 000h, 03fh
                                                 db
58 000129C5 151500153F003F
                                       <1>
                                                      015h, 000h, 03fh, 03fh, 02ah, 015h, 015h, 02ah, 015h, 03fh, 02ah, 03fh, 015h, 02ah, 03fh
59 000129CC 15003F3F2A15152A15- <1>
59 000129D5 3F2A3F152A3F3F
                                       <1>
60 000129DC 15000015002A152A00- <1>
                                                      015h, 000h, 000h, 015h, 000h, 02ah, 015h, 02ah, 000h, 015h, 02ah, 02ah, 03fh, 000h, 000h, 03fh
                                                 db
60 000129E5 152A2A3F00003F
                                       <1>
61 000129EC 002A3F2A003F2A2A15- <1>
                                                      000h, 02ah, 03fh, 02ah, 000h, 03fh, 02ah, 02ah, 015h, 000h, 015h, 000h, 03fh, 015h, 02ah
61 000129F5 001515003F152A
                                       <1>
62 000129FC 15152A3F3F00153F00- <1>
                                                      015h, 015h, 02ah, 03fh, 03fh, 000h, 015h, 03fh, 000h, 03fh, 03fh, 02ah, 015h, 03fh, 02ah, 03fh
                                                 db
62 00012A05 3F3F2A153F2A3F
                                       <1>
63 00012A0C 15150015152A153F00- <1>
                                                      015h, 015h, 000h, 015h, 015h, 02ah, 015h, 03fh, 000h, 015h, 03fh, 02ah, 03fh, 015h, 000h, 03fh
63 00012A15 153F2A3F15003F
                                       <1>
64 00012A1C 152A3F3F003F3F2A15- <1>
                                                      015h, 02ah, 03fh, 03fh, 000h, 03fh, 03fh, 02ah, 015h, 015h, 015h, 015h, 03fh, 015h, 03fh
64 00012A25 151515153F153F
                                       <1>
65 00012A2C 15153F3F3F15153F15- <1>
                                                      015h, 015h, 03fh, 03fh, 03fh, 015h, 015h, 015h, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh
                                                 db
65 00012A35 3F3F3F153F3F3F
                                       <1>
                                       <1> palette3:
67 00012A3C 00000000002A002A00- <1>
                                                 db
                                                      000h, 000h, 000h, 000h, 000h, 002ah, 000h, 02ah, 000h, 000h, 02ah, 02ah, 02ah, 02ah, 000h, 000h, 02ah
67 00012A45 002A2A2A00002A
                                      <1>
68 00012A4C 002A2A15002A2A2A15- <1>
                                                      000h, 02ah, 02ah, 015h, 000h, 02ah, 02ah, 02ah, 015h, 015h, 015h, 015h, 03fh, 015h, 03fh
68 00012A55 151515153F153F
                                       <1>
69 00012A5C 15153F3F3F15153F15- <1>
                                                      015h, 015h, 03fh, 03fh, 03fh, 015h, 015h, 015h, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh, 03fh
69 00012A65 3F3F3F153F3F3F
                                       <1>
70 00012A6C 000000050505080808- <1>
                                                      000h, 000h, 000h, 005h, 005h, 005h, 008h, 008h, 008h, 00bh, 00bh, 00bh, 00eh, 00eh, 011h
                                                 db
70 00012A75 0B0B0B0E0E0E11
                                       <1>
71 00012A7C 111111414141818181C- <1>
                                                      011h, 011h, 014h, 014h, 014h, 018h, 018h, 018h, 01ch, 01ch, 01ch, 020h, 020h, 020h, 024h, 024h
71 00012A85 1C1C2020202424
                                       <1>
72 00012A8C 242828282D2D2D3232- <1>
                                                      024h, 028h, 028h, 028h, 02dh, 02dh, 02dh, 032h, 032h, 032h, 038h, 038h, 038h, 03fh, 03fh, 03fh
72 00012A95 323838383F3F3F
                                       <1>
73 00012A9C 00003F10003F1F003F- <1>
                                                 db
                                                      000h, 000h, 03fh, 010h, 000h, 03fh, 01fh, 000h, 03fh, 02fh, 000h, 03fh, 03fh, 000h, 03fh, 03fh
73 00012AA5 2F003F3F003F3F
                                       <1>
74 00012AAC 002F3F001F3F00103F- <1>
                                                      000h, 02fh, 03fh, 000h, 01fh, 03fh, 000h, 010h, 03fh, 000h, 000h, 03fh, 010h, 000h, 03fh, 01fh
74 00012AB5 00003F10003F1F
                                       <1>
75 00012ABC 003F2F003F3F002F3F- <1>
                                                      000h, 03fh, 02fh, 000h, 03fh, 03fh, 000h, 02fh, 03fh, 000h, 01fh, 03fh, 000h, 010h, 03fh, 000h
                                                 db
75 00012AC5 001F3F00103F00
                                       <1>
76 00012ACC 003F00003F10003F1F- <1>
                                                      000h, 03fh, 000h, 000h, 03fh, 010h, 000h, 03fh, 01fh, 000h, 03fh, 02fh, 000h, 03fh, 03fh, 000h
76 00012AD5 003F2F003F3F00
                                       <1>
77 00012ADC 2F3F001F3F00103F1F- <1>
                                                      02fh, 03fh, 000h, 01fh, 03fh, 000h, 010h, 03fh, 01fh, 01fh, 03fh, 027h, 01fh, 03fh, 02fh, 01fh
77 00012AE5 1F3F271F3F2F1F
                                       <1>
78 00012AEC 3F371F3F3F1F3F3F1F- <1>
                                                      03fh, 037h, 01fh, 03fh, 03fh, 01fh, 03fh, 03fh, 01fh, 037h, 03fh, 01fh, 02fh, 03fh, 01fh, 027h
78 00012AF5 373F1F2F3F1F27
                                       <1>
79 00012AFC 3F1F1F3F271F3F2F1F- <1>
                                                      03fh, 01fh, 01fh, 03fh, 027h, 01fh, 03fh, 02fh, 01fh, 03fh, 03fh, 01fh, 03fh, 03fh, 03fh, 03fh, 03fh, 01fh, 03fh
79 00012B05 3F371F3F3F1F37
                                       <1>
80 00012B0C 3F1F2F3F1F273F1F1F- <1>
                                                 db 03fh, 01fh, 02fh, 03fh, 01fh, 027h, 03fh, 01fh, 01fh, 03fh, 01fh, 01fh, 03fh, 027h, 01fh, 03fh
80 00012B15 3F1F1F3F271F3F
                                       <1>
81 00012B1C 2F1F3F371F3F3F1F37- <1>
                                                      02fh, 01fh, 03fh, 037h, 01fh, 03fh, 03fh, 01fh, 037h, 03fh, 01fh, 02fh, 03fh, 01fh, 027h, 03fh
81 00012B25 3F1F2F3F1F273F
                                    <1>
82 00012B2C 2D2D3F312D3F362D3F- <1>
                                                      02dh, 02dh, 03fh, 031h, 02dh, 03fh, 036h, 02dh, 03fh, 03ah, 02dh, 03fh, 03fh, 02dh, 03fh, 03fh
82 00012B35 3A2D3F3F2D3F3F
                                      <1>
83 00012B3C 2D3A3F2D363F2D313F- <1>
                                                      02dh, 03ah, 03fh, 02dh, 036h, 03fh, 02dh, 031h, 03fh, 02dh, 02dh, 03fh, 03fh, 03fh, 03fh, 03fh
83 00012B45 2D2D3F312D3F36
84 00012B4C 2D3F3A2D3F3F2D3A3F- <1>
                                                      02dh, 03fh, 03ah, 02dh, 03fh, 03fh, 02dh, 03ah, 03fh, 02dh, 03fh, 02dh, 03fh, 02dh, 03fh, 02dh
84 00012B55 2D363F2D313F2D
                                      <1>
85 00012B5C 2D3F2D2D3F312D3F36- <1>
                                                      02dh, 03fh, 02dh, 02dh, 03fh, 031h, 02dh, 03fh, 036h, 02dh, 03fh, 03h, 02dh, 03fh, 02dh
85 00012B65 2D3F3A2D3F3F2D
                                      <1>
86 00012B6C 3A3F2D363F2D313F00- <1>
                                                      03ah, 03fh, 02dh, 036h, 03fh, 02dh, 031h, 03fh, 000h, 000h, 01ch, 007h, 000h, 01ch, 00eh, 000h
86 00012B75 001C07001C0E00
                                      <1>
87 00012B7C 1C15001C1C001C1C00- <1>
                                                      01ch, 015h, 000h, 01ch, 01ch, 000h, 01ch, 01ch, 000h, 015h, 01ch, 000h, 00eh, 01ch, 000h, 007h
87 00012B85 151C000E1C0007
                                      <1>
88 00012B8C 1C00001C07001C0E00- <1>
                                                      01ch, 000h, 000h, 01ch, 007h, 000h, 01ch, 00eh, 000h, 01ch, 015h, 000h, 01ch, 01ch, 015h
88 00012B95 1C15001C1C0015
                                       <1>
89 00012B9C 1C000E1C00071C0000- <1>
                                                      01ch, 000h, 00eh, 01ch, 000h, 007h, 01ch, 000h, 000h, 01ch, 000h, 000h, 01ch, 007h, 000h, 01ch
89 00012BA5 1C00001C07001C
                                      <1>
90 00012BAC 0E001C15001C1C0015- <1>
                                                      00eh, 000h, 01ch, 015h, 000h, 01ch, 01ch, 000h, 015h, 01ch, 000h, 00eh, 01ch, 000h, 007h, 01ch
90 00012BB5 1C000E1C00071C
                                      <1>
91 00012BBC 0E0E1C110E1C150E1C- <1>
                                                 db 00eh, 00eh, 01ch, 011h, 00eh, 01ch, 015h, 00eh, 01ch, 018h, 00eh, 01ch, 01ch, 01ch, 01ch
91 00012BC5 180E1C1C0E1C1C
                                     <1>
92 00012BCC 0E181C0E151C0E111C- <1>
                                                 db 00eh, 018h, 01ch, 00eh, 015h, 01ch, 00eh, 011h, 01ch, 00eh, 00eh, 01ch, 011h, 00eh, 01ch, 015h
92 00012BD5 0E0E1C110E1C15
                                      <1>
```

00eh, 01ch, 018h, 00eh, 01ch, 01ch, 00eh, 018h, 01ch, 00eh, 015h, 01ch, 00eh, 011h, 01ch, 00eh

```
93 00012BE5 0E151C0E111C0E
                                 <1>
 94 00012BEC 0E1C0E0E1C110E1C15- <1>
                                             00eh, 01ch, 00eh, 00eh, 01ch, 011h, 00eh, 01ch, 015h, 00eh, 01ch, 018h, 00eh, 01ch, 01ch, 00eh
 94 00012BF5 0E1C180E1C1C0E
                                  <1>
                                             018h, 01ch, 00eh, 015h, 01ch, 00eh, 011h, 01ch, 014h, 014h, 01ch, 016h, 014h, 01ch, 018h, 014h
 95 00012BFC 181C0E151C0E111C14- <1>
                                         db
 95 00012C05 141C16141C1814
                                 <1>
 96 00012C0C 1C1A141C1C141C1C14- <1>
                                             01ch, 01ah, 014h, 01ch, 01ch, 014h, 01ch, 01ch, 014h, 01ah, 01ch, 014h, 016h, 016h, 016h, 016h
 96 00012C15 1A1C14181C1416
 97 00012C1C 1C14141C16141C1814- <1>
                                             01ch, 014h, 014h, 01ch, 016h, 014h, 01ch, 018h, 014h, 01ch, 01ah, 014h, 01ch, 01ch, 014h, 01ah
                                         db
 97 00012C25 1C1A141C1C141A
                                 <1>
 98 00012C2C 1C14181C14161C1414- <1>
                                             01ch, 014h, 018h, 01ch, 014h, 016h, 01ch, 014h, 014h, 01ch, 014h, 014h, 01ch, 016h, 014h, 01ch
                                         db
 98 00012C35 1C14141C16141C
                                 <1>
 99 00012C3C 18141C1A141C1C141A- <1>
                                             018h, 014h, 01ch, 01ah, 014h, 01ch, 01ch, 014h, 01ah, 01ch, 014h, 018h, 01ch, 014h, 016h, 01ch
 99 00012C45 1C14181C14161C
                                 <1>
100 00012C4C 000010040010080010- <1>
                                             000h, 000h, 010h, 004h, 000h, 010h, 008h, 000h, 010h, 00ch, 000h, 010h, 010h, 010h, 010h, 010h
                                         db
100 00012C55 0C001010001010
                                 <1>
101 00012C5C 000C10000810000410- <1>
                                             000h, 00ch, 010h, 000h, 008h, 010h, 000h, 004h, 010h, 000h, 000h, 010h, 004h, 000h, 010h, 008h
101 00012C65 00001004001008
                                  <1>
102 00012C6C 00100C001010000C10- <1>
                                             000h, 010h, 00ch, 000h, 010h, 010h, 000h, 00ch, 010h, 000h, 008h, 010h, 000h, 004h, 010h, 000h
                                         db
102 00012C75 00081000041000
                                 <1>
103 00012C7C 001000001004001008- <1>
                                             000h, 010h, 000h, 000h, 010h, 004h, 000h, 010h, 008h, 000h, 010h, 00ch, 000h, 010h, 010h, 000h
                                         db
103 00012C85 00100C00101000
                                 <1>
                                             00ch, 010h, 000h, 008h, 010h, 000h, 004h, 010h, 008h, 008h, 010h, 00ah, 008h, 010h, 00ch, 008h
104 00012C8C 0C1000081000041008- <1>
104 00012C95 08100A08100C08
                                 <1>
105 00012C9C 100E08101008101008- <1>
                                         db
                                             010h, 00eh,
                                                         008h, 010h, 010h, 008h, 010h, 010h, 008h, 00eh, 010h, 008h, 00ch, 010h, 008h, 00ah
105 00012CA5 0E10080C10080A
                                 <1>
106 00012CAC 100808100A08100C08- <1>
                                             010h, 008h, 008h, 010h, 00ah, 008h, 010h, 00ch, 008h, 010h, 00eh, 008h, 010h, 010h, 008h, 00eh
106 00012CB5 100E081010080E
                                 <1>
107 00012CBC 10080C10080A100808- <1>
                                             010h, 008h, 00ch, 010h, 008h, 00ah, 010h, 008h, 008h, 010h, 008h, 008h, 010h, 00ah, 008h, 010h
                                         db
107 00012CC5 100808100A0810
                                 <1>
108 00012CCC 0C08100E081010080E- <1>
                                             00ch, 008h, 010h, 00eh, 008h, 010h, 010h, 008h, 00eh, 010h, 008h, 00ch, 010h, 008h, 00ah, 010h
108 00012CD5 10080C10080A10
                                  <1>
109 00012CDC 0B0B100C0B100D0B10- <1>
                                             00bh, 00bh, 010h, 00ch, 00bh, 010h, 00dh, 00bh, 010h, 00fh, 00bh, 010h, 010h, 010h, 010h, 010h
                                         db
109 00012CE5 0F0B10100B1010
                                 <1>
110 00012CEC 0B0F100B0D100B0C10- <1>
                                             00bh, 00fh,
                                                         010h, 00bh, 00dh, 010h, 00bh, 00ch, 010h, 00bh, 00bh, 010h, 00ch,
                                                                                                                             00bh, 010h, 00dh
                                         db
110 00012CF5 0B0B100C0B100D
                                 <1>
                                             00bh, 010h, 00fh, 00bh, 010h, 010h, 010h, 00bh, 00fh, 010h, 00bh, 00dh, 010h, 00bh, 00ch, 010h, 00bh
111 00012CFC 0B100F0B10100B0F10- <1>
                                         db
111 00012D05 0B0D100B0C100B
                                 <1>
112 00012D0C 0B100B0B100C0B100D- <1>
                                         db
                                             00bh, 010h, 00bh, 00bh, 010h, 00ch, 00bh, 010h, 00dh, 00bh, 010h, 00fh, 00bh, 010h, 010h, 00bh
112 00012D15 0B100F0B10100B
                                 <1>
113 00012D1C 0F100B0D100B0C1000-
                                             00fh, 010h, 00bh, 00dh, 010h, 00bh, 00ch, 010h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
113 00012D25 00000000000000
                                 <1>
114 00012D2C 0000000000000000000 <1>
                                             000h, 000h
                                         db
114 00012D35 00000000000000
                                  <1>
115
                                  <1>
116
                                  <1>
                                 <1>; 04/07/2016
117
118
                                 <1>; FONT DATA
119
                                  <1>
                                  <1> CRT CHAR GEN:
120
                                 <1> vgafont8:
121
122 00012D3C 0000000000000007E- <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 07eh, 081h, 0a5h, 081h, 0bdh, 099h, 081h, 07eh
                                         db
122 00012D45 81A581BD99817E
                                 <1>
123 00012D4C 7EFFDBFFC3E7FF7E6C- <1>
                                             07eh, 0ffh, 0dbh, 0ffh, 0c3h, 0e7h, 0ffh, 07eh, 06ch, 0feh, 0feh, 0feh, 07ch, 038h, 010h, 000h
123 00012D55 FEFEFE7C381000
                                  <1>
124 00012D5C 10387CFE7C38100038- <1>
                                             010h, 038h, 07ch, 0feh, 07ch, 038h, 010h, 000h, 038h, 07ch, 038h, 0feh, 0feh, 07ch, 038h, 07ch
                                         db
124 00012D65 7C38FEFE7C387C
                                 <1>
125 00012D6C 1010387CFE7C387C00- <1>
                                             010h, 010h, 038h, 07ch, 0feh, 07ch, 038h, 07ch, 000h, 000h, 018h, 03ch, 03ch, 018h, 000h, 000h
125 00012D75 00183C3C180000
                                 <1>
126 00012D7C FFFFE7C3C3E7FFFF00- <1>
                                             Offh, Offh, Oe7h, Oc3h, Oc3h, Oe7h, Offh, Offh, O00h, O3ch, O66h, O42h, O42h, O66h, O3ch, O00h
126 00012D85 3C664242663C00
                                 <1>
127 00012D8C FFC399BDBD99C3FF0F- <1>
                                             0ffh, 0c3h, 099h, 0bdh, 0bdh, 099h, 0c3h, 0ffh, 00fh, 007h, 00fh, 07dh, 0cch, 0cch, 0cch, 078h
                                         db
127 00012D95 070F7DCCCCCC78
                                 <1>
128 00012D9C 3C6666663C187E183F- <1>
                                             03ch, 066h, 066h, 066h, 03ch, 018h, 07eh, 018h, 03fh, 033h, 03fh, 030h, 030h, 070h, 0f0h, 0e0h
128 00012DA5 333F303070F0E0
                                  <1>
129 00012DAC 7F637F636367E6C099- <1>
                                             07fh, 063h, 07fh, 063h, 063h, 067h, 0e6h, 0c0h, 099h, 05ah, 03ch, 0e7h, 0e7h, 03ch, 05ah, 099h
                                         db
129 00012DB5 5A3CE7E73C5A99
                                 <1>
130 00012DBC 80E0F8FEF8E0800002- <1>
                                             080h, 0e0h, 0f8h, 0feh, 0f8h, 0e0h, 080h, 000h, 002h, 00eh, 03eh, 0feh, 03eh, 00eh, 002h, 000h
130 00012DC5 0E3EFE3E0E0200
                                 <1>
131 00012DCC 183C7E18187E3C1866- <1>
                                             018h, 03ch, 07eh, 018h, 018h, 07eh, 03ch, 018h, 066h, 066h, 066h, 066h, 066h, 000h, 066h, 000h
                                         db
131 00012DD5 66666666006600
                                 <1>
132 00012DDC 7FDBDB7B1B1B1B003E- <1>
                                         db
                                             07fh, 0dbh, 0dbh, 07bh, 01bh, 01bh, 01bh, 000h, 03eh, 063h, 038h, 06ch, 06ch, 038h, 0cch, 078h
132 00012DE5 63386C6C38CC78
                                 <1>
133 00012DEC 000000007E7E7E0018- <1>
                                             000h, 000h, 000h, 000h, 07eh, 07eh, 07eh, 000h, 018h, 03ch, 07eh, 018h, 07eh, 03ch, 018h, 0ffh
133 00012DF5 3C7E187E3C18FF
                                  <1>
134 00012DFC 183C7E181818180018- <1>
                                             018h, 03ch, 07eh, 018h, 018h, 018h, 018h, 000h, 018h, 018h, 018h, 018h, 07eh, 03ch, 018h, 000h
                                         db
134 00012E05 1818187E3C1800
                                 <1>
135 00012E0C 00180CFE0C18000000- <1>
                                             000h, 018h, 00ch, 0feh, 00ch, 018h, 000h, 000h, 000h, 030h, 060h, 0feh, 060h, 030h, 000h, 000h
135 00012E15 3060FE60300000
                                 <1>
136 00012E1C 0000C0C0C0FE000000- <1>
                                             000h, 000h, 0c0h, 0c0h, 0c0h, 0feh, 000h, 000h, 000h, 024h, 066h, 0ffh, 066h, 024h, 000h, 000h
                                         db
136 00012E25 2466FF66240000
                                 <1>
137 00012E2C 00183C7EFFFF000000- <1>
                                             000h, 018h, 03ch, 07eh, 0ffh, 0ffh, 000h, 000h, 000h, 0ffh, 0ffh, 07eh, 03ch, 018h, 000h, 000h
                                         db
137 00012E35 FFFF7E3C180000
                                 <1>
138 00012E3C 0000000000000000030- <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 030h, 078h, 078h, 030h, 030h, 000h, 030h, 000h
138 00012E45 78783030003000
                                 <1>
139 00012E4C 6C6C6C00000000006C- <1>
                                         db
                                             06ch, 06ch, 06ch, 000h, 000h, 000h, 000h, 000h, 06ch, 06ch, 0feh, 06ch, 0feh, 06ch, 06ch, 000h
139 00012E55 6CFE6CFE6C6C00
                                 <1>
140 00012E5C 307CC0780CF8300000- <1>
                                             030h, 07ch, 0c0h, 078h, 00ch, 0f8h, 030h, 000h, 000h, 0c6h, 0cch, 018h, 030h, 066h, 0c6h, 000h
140 00012E65 C6CC183066C600
141 00012E6C 386C3876DCCC760060- <1>
                                             038h, 06ch, 038h, 076h, 0dch, 0cch, 076h, 000h, 060h, 060h, 0c0h, 000h, 000h, 000h, 000h, 000h
141 00012E75 60C00000000000
                                <1>
142 00012E7C 183060606030180060- <1>
                                             018h, 030h, 060h, 060h, 060h, 030h, 018h, 000h, 060h, 030h, 018h, 018h, 018h, 030h, 060h, 000h
142 00012E85 30181818306000
143 00012E8C 00663CFF3C66000000- <1>
                                             000h, 066h, 03ch, 0ffh, 03ch, 066h, 000h, 000h, 000h, 030h, 030h, 0fch, 030h, 030h, 000h, 000h
143 00012E95 3030FC30300000
                                 <1>
144 00012E9C 00000000030306000- <1>
                                             000h, 000h, 000h, 000h, 000h, 030h, 030h, 060h, 000h, 000h, 000h, 0fch, 000h, 000h, 000h, 000h
                                         db
144 00012EA5 0000FC0000000
                                <1>
145 00012EAC 000000000030300006- <1>
                                             000h, 000h, 000h, 000h, 000h, 030h, 030h, 030h, 000h, 006h, 018h, 030h, 060h, 0c0h, 080h, 000h
145 00012EB5 0C183060C08000
                                <1>
146 00012EBC 7CC6CEDEF6E67C0030- <1>
                                             07ch, 0c6h, 0ceh, 0deh, 0f6h, 0e6h, 07ch, 000h, 030h, 070h, 030h, 030h, 030h, 030h, 0fch, 000h
146 00012EC5 7030303030FC00
                                 <1>
147 00012ECC 78CC0C3860CCFC0078- <1>
                                             078h, Occh, 00ch, 038h, 060h, 0cch, 0fch, 000h, 078h, 0cch, 00ch, 038h, 00ch, 0cch, 078h, 000h
147 00012ED5 CC0C380CCC7800
                                 <1>
148 00012EDC 1C3C6CCCFE0C1E00FC- <1>
                                             01ch, 03ch, 06ch, 0cch, 0feh, 00ch, 01eh, 000h, 0fch, 0c0h, 0f8h, 00ch, 00ch, 0cch, 078h, 000h
148 00012EE5 C0F80C0CCC7800
                                 <1>
149 00012EEC 3860C0F8CCCC7800FC- <1>
                                             038h, 060h, 0c0h, 0f8h, 0cch, 0cch, 078h, 000h, 0fch, 0cch, 00ch, 018h, 030h, 030h, 030h, 000h
149 00012EF5 CC0C1830303000
                                <1>
150 00012EFC 78CCCC78CCC780078- <1>
                                             078h, 0cch, 0cch, 078h, 0cch, 0cch, 078h, 000h, 078h, 0cch, 0cch, 07ch, 00ch, 018h, 070h, 000h
150 00012F05 CCCC7C0C187000
                               <1>
151 00012F0C 003030000030300000- <1>
                                         db 000h, 030h, 030h, 000h, 000h, 000h, 030h, 030h, 000h, 000h, 030h, 030h, 000h, 000h, 030h, 030h, 060h
```

93 00012BDC 0E1C180E1C1C0E181C- <1>

151 00012F15 30300000303060

```
152 00012F1C 183060C06030180000- <1>
                                                      018h, 030h, 060h, 0c0h, 060h, 030h, 018h, 000h, 000h, 000h, 0fch, 000h, 000h, 0fch, 000h, 000h
152 00012F25 00FC0000FC0000
                                        <1>
                                                      060h, 030h, 018h, 00ch, 018h, 030h, 060h, 000h, 078h, 0cch, 00ch, 018h, 030h, 000h, 030h, 000h
153 00012F2C 6030180C1830600078- <1>
153 00012F35 CC0C1830003000
                                        <1>
154 00012F3C 7CC6DEDEDEC0780030- <1>
                                                      07ch, 0c6h, 0deh, 0deh, 0deh, 0c0h, 078h, 000h, 030h, 078h, 0cch, 0cch, 0fch, 0cch, 0cch, 000h
                                                  db
154 00012F45 78CCCCFCCCC00
                                        <1>
155 00012F4C FC66667C6666FC003C- <1>
                                                      0fch, 066h, 066h, 07ch, 066h, 066h, 0fch, 000h, 03ch, 066h, 0c0h, 0c0h, 0c0h, 066h, 03ch, 000h
155 00012F55 66C0C0C0663C00
156 00012F5C F86C6666666CF800FE- <1>
                                                  db
                                                      0f8h, 06ch, 066h, 066h, 066h, 06ch, 0f8h, 000h, 0feh, 062h, 068h, 078h, 068h, 062h, 0feh, 000h
156 00012F65 6268786862FE00
                                        <1>
157 00012F6C FE6268786860F0003C- <1>
                                                      0feh, 062h, 068h, 078h, 068h, 060h, 0f0h, 000h, 03ch, 066h, 0c0h, 0c0h, 0ceh, 066h, 03eh, 000h
                                                  db
157 00012F75 66C0C0CE663E00
                                        <1>
158 00012F7C CCCCCCFCCCCCC0078- <1>
                                                      Occh, Occh, Occh, Ofch, Occh, Occh, Occh, Occh, O30h, 
158 00012F85 30303030307800
                                        <1>
159 00012F8C 1E0C0C0CCCC7800E6- <1>
                                                      01eh, 00ch, 00ch, 00ch, 0cch, 0cch, 078h, 000h, 0e6h, 066h, 06ch, 078h, 06ch, 066h, 0e6h, 000h
                                                  db
159 00012F95 666C786C66E600
                                        <1>
160 00012F9C F06060606266FE00C6- <1>
                                                      0f0h, 060h, 060h, 060h, 062h, 066h, 0feh, 000h, 0c6h, 0eeh, 0feh, 0feh, 0d6h, 0c6h, 0c6h, 000h
160 00012FA5 EEFEFED6C6C600
                                        <1>
161 00012FAC C6E6F6DECEC6C60038- <1>
                                                      Oc6h, Oe6h, Of6h, Odeh, Oceh, Oc6h, Oc6h, O00h, O38h, O6ch, Oc6h, Oc6h, Oc6h, O6ch, O38h, O00h
                                                  db
161 00012FB5 6CC6C6C66C3800
                                        <1>
162 00012FBC FC66667C6060F00078- <1>
                                                      0fch, 066h, 066h, 07ch, 060h, 060h, 0f0h, 000h, 078h, 0cch, 0cch, 0cch, 0dch, 078h, 01ch, 000h
                                                  db
162 00012FC5 CCCCCDC781C00
                                        <1>
163 00012FCC FC66667C6C66E60078- <1>
                                                      0fch, 066h, 066h, 07ch, 06ch, 066h, 0e6h, 000h, 078h, 0cch, 0e0h, 070h, 01ch, 0cch, 078h, 000h
163 00012FD5 CCE0701CCC7800
                                        <1>
164 00012FDC FCB4303030307800CC- <1>
                                                  db
                                                      0fch, 0b4h, 030h, 030h, 030h, 030h, 078h, 000h, 0cch, 0cch, 0cch, 0cch, 0cch,
                                                                                                                                                     Occh, Ofch, 000h
164 00012FE5 CCCCCCCCCFC00
                                        <1>
165 00012FEC CCCCCCCCC783000C6- <1>
                                                      Occh, Occh, Occh, Occh, Occh, O78h, O30h, O00h, Oc6h, Oc6h, Oc6h, Od6h, Ofeh, Oeeh, Oc6h, O00h
165 00012FF5 C6C6D6FEEEC600
                                        <1>
166 00012FFC C6C66C38386CC600CC- <1>
                                                      0c6h, 0c6h, 06ch, 038h, 038h, 06ch, 0c6h, 000h, 0cch, 0cch, 0cch, 078h, 030h, 030h, 078h, 000h
                                                  db
166 00013005 CCCC7830307800
                                        <1>
167 0001300C FEC68C183266FE0078- <1>
                                                      0feh, 0c6h, 08ch, 018h, 032h, 066h, 0feh, 000h, 078h, 060h, 060h, 060h, 060h, 060h, 078h, 000h
    00013015 60606060607800
                                        <1>
168 0001301C C06030180C06020078- <1>
                                                      0c0h, 060h, 030h, 018h, 00ch, 006h, 002h, 000h, 078h, 018h, 018h, 018h, 018h, 078h, 000h
168 00013025 18181818187800
                                        <1>
169 0001302C 10386CC60000000000 <1>
                                                      010h, 038h,
                                                                     06ch, 0c6h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h,
                                                                                                                                                     000h, 000h, 0ffh
                                                  db
169 00013035 000000000000FF
170 0001303C 3030180000000000000 <1>
                                                      030h, 030h, 018h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 078h, 00ch, 07ch, 0cch, 076h, 000h
                                                  db
170 00013045 00780C7CCC7600
                                        <1>
171 0001304C E060607C6666DC0000- <1>
                                                  db
                                                      0e0h, 060h, 060h, 07ch, 066h, 066h, 0dch, 000h, 000h, 000h, 078h, 0cch, 0coh, 0cch, 078h, 000h
171 00013055 0078CCC0CC7800
                                        <1>
172 0001305C 1C0C0C7CCCCC760000- <1>
                                                      01ch, 00ch, 00ch, 07ch, 0cch, 0cch, 076h, 000h, 000h, 000h, 078h, 0cch, 0fch, 0c0h, 078h, 000h
172 00013065 0078CCFCC07800
                                        <1>
173 0001306C 386C60F06060F00000- <1>
                                                      038h, 06ch, 060h, 0f0h, 060h, 060h, 060h, 0f0h, 000h, 000h, 000h, 076h, 0cch, 0cch, 07ch, 00ch, 0f8h
                                                  db
173 00013075 0076CCCC7C0CF8
                                        <1>
174 0001307C E0606C766666E60030- <1>
                                                      0e0h, 060h, 06ch, 076h, 066h, 066h, 0e6h, 000h, 030h, 000h, 070h, 030h, 030h, 030h, 078h, 000h
174 00013085 00703030307800
                                        <1>
175 0001308C 0C000C0C0CCCC78E0- <1>
                                                      db
175 00013095 60666C786CE600
                                        <1>
176 0001309C 703030303030780000- <1>
                                                      070h, 030h, 030h, 030h, 030h, 030h, 078h, 000h, 000h, 000h, 0cch, 0feh, 0feh, 0d6h, 0c6h, 000h
176 000130A5 00CCFEFED6C600
                                        <1>
177 000130AC 0000F8CCCCCCC0000-
                                                      000h, 000h, 0f8h, 0cch, 0cch, 0cch, 0cch, 000h, 000h, 000h, 078h, 0cch, 0cch, 0cch, 078h, 000h
                                        <1>
                                                  db
177 000130B5 0078CCCCCC7800
                                        <1>
178 000130BC 0000DC666667C60F000- <1>
                                                  db
                                                      000h, 000h, 0dch, 066h, 066h, 07ch, 060h, 0f0h, 000h, 000h, 076h, 0cch, 0cch, 07ch, 00ch, 01eh
178 000130C5 0076CCCC7C0C1E
                                        <1>
179 000130CC 0000DC766660F00000- <1>
                                                       000h, 000h, 0dch, 076h, 066h, 060h, 0f0h, 000h, 000h, 000h, 07ch, 0c0h, 078h, 00ch, 0f8h, 000h
179 000130D5 007CC0780CF800
                                        <1>
180 000130DC 10307C303034180000- <1>
                                                      010h, 030h, 07ch, 030h, 030h, 034h, 018h, 000h, 000h, 000h, 0cch, 0cch, 0cch, 0cch, 076h, 000h
                                                  db
180 000130E5 00CCCCCCC7600
                                        <1>
181 000130EC 0000CCCCCC78300000- <1>
                                                      000h, 000h, 0cch, 0cch, 0cch, 078h, 030h, 000h, 000h, 000h, 0c6h, 0d6h, 0feh, 0feh, 06ch, 000h
181 000130F5 00C6D6FEFE6C00
                                        <1>
182 000130FC 0000C66C386CC60000- <1>
                                                      000h, 000h, 0c6h, 06ch, 038h, 06ch, 0c6h, 000h, 000h, 000h, 0cch, 0cch, 0cch, 07ch, 00ch, 0f8h
                                                  db
182 00013105 00CCCCCC7C0CF8
                                        <1>
183 0001310C 0000FC983064FC001C- <1>
                                                      000h, 000h, 0fch, 098h, 030h, 064h, 0fch, 000h, 01ch, 030h, 030h, 0e0h, 030h, 030h, 01ch, 000h
                                                  db
183 00013115 3030E030301C00
                                        <1>
184 0001311C 1818180018181800E0- <1>
                                                      018h, 018h, 018h, 000h, 018h, 018h, 018h, 000h, 0e0h, 030h, 030h, 01ch, 030h, 030h, 0e0h, 000h
                                                  db
184 00013125 30301C3030E000
                                        <1>
185 0001312C 76DC000000000000000 <1>
                                                  db
                                                      076h, 0dch, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 010h, 038h, 06ch, 0c6h, 0c6h, 0feh, 000h
185 00013135 10386CC6C6FE00
                                        <1>
186 0001313C 78CCC0CC78180C7800- <1>
                                                      078h, 0cch, 0c0h, 0cch, 078h, 018h, 00ch, 078h, 000h, 0cch, 000h, 0cch, 0cch, 0cch, 07eh, 000h
186 00013145 CC00CCCCC7E00
                                        <1>
187 0001314C 1C0078CCFCC078007E- <1>
                                                      01ch, 000h, 078h, 0cch, 0fch, 0c0h, 078h, 000h, 07eh, 0c3h, 03ch, 006h, 03eh, 066h, 03fh, 000h
                                                  db
187 00013155 C33C063E663F00
                                        <1>
188 0001315C CC00780C7CCC7E00E0- <1>
                                                      Occh, 000h, 078h, 00ch, 07ch, 0cch, 07eh, 000h, 0e0h, 000h, 078h, 00ch, 07ch, 0cch, 07eh, 000h
188 00013165 00780C7CCC7E00
189 0001316C 3030780C7CCC7E0000- <1>
                                                      030h, 030h, 078h, 00ch, 07ch, 0cch, 07eh, 000h, 000h, 000h, 078h, 0c0h, 078h, 00ch, 038h
                                                  db
189 00013175 0078C0C0780C38
                                        <1>
190 0001317C 7EC33C667E603C00CC- <1>
                                                      07eh, 0c3h, 03ch, 066h, 07eh, 060h, 03ch, 000h, 0cch, 000h, 078h, 0cch, 0fch, 0c0h, 078h, 000h
                                                  db
190 00013185 0078CCFCC07800
                                        <1>
191 0001318C E00078CCFCC07800CC- <1>
                                                      0e0h, 000h, 078h, 0cch, 0fch, 0c0h, 078h, 000h, 0cch, 000h, 070h, 030h, 030h, 030h, 078h, 000h
191 00013195 00703030307800
                                        <1>
192 0001319C 7CC6381818183C00E0- <1>
                                                  db
                                                      07ch, 0c6h, 038h, 018h, 018h, 018h, 03ch, 000h, 0e0h, 000h, 070h, 030h, 030h, 030h, 078h, 000h
192 000131A5 00703030307800
                                        <1>
193 000131AC C6386CC6FEC6C60030- <1>
                                                      0c6h, 038h, 06ch, 0c6h, 0feh, 0c6h, 0c6h, 000h, 030h, 030h, 000h, 078h, 0cch, 0fch, 0cch, 000h
193 000131B5 300078CCFCCC00
                                        <1>
194 000131BC 1C00FC607860FC0000- <1>
                                                      01ch, 000h, 0fch, 060h, 078h, 060h, 0fch, 000h, 000h, 000h, 07fh, 00ch, 07fh, 0cch, 07fh, 000h
                                                  db
194 000131C5 007F0C7FCC7F00
                                        <1>
195 000131CC 3E6CCCFECCCCE0078-
                                                  db
                                                      03eh, 06ch, 0cch, 0feh, 0cch, 0cch, 0cch, 00ch, 078h, 0cch, 000h, 078h, 0cch, 078h, 000h
                                        <1>
195 000131D5 CC0078CCCC7800
196 000131DC 00CC0078CCCC780000- <1>
                                                      000h, 0cch, 000h, 078h, 0cch, 0cch, 078h, 000h, 000h, 0e0h, 000h, 078h, 0cch, 0cch, 078h, 000h
196 000131E5 E00078CCCC7800
                                       <1>
197 000131EC 78CC00CCCCC7E0000- <1>
                                                      078h, 0cch, 000h, 0cch, 0cch, 0cch, 07eh, 000h, 000h, 0e0h, 000h, 0cch, 0cch, 0cch, 07eh, 000h
                                                  db
197 000131F5 E000CCCCCC7E00
                                       <1>
198 000131FC 00CC00CCC7C0CF8C3- <1>
                                                      000h, 0cch, 000h, 0cch, 0cch, 07ch, 00ch, 0f8h, 0c3h, 018h, 03ch, 066h, 066h, 03ch, 018h, 000h
198 00013205 183C66663C1800
                                        <1>
                                                      Occh, 000h, Occh, Occh, Occh, Occh, 078h, 000h, 018h, 018h, 07eh, 0c0h, 0c0h, 07eh, 018h, 018h
199 0001320C CC00CCCCCCC780018- <1>
199 00013215 187EC0C07E1818
                                        <1>
                                                      038h, 06ch, 064h, 0f0h, 060h, 0e6h, 0fch, 000h, 0cch, 0cch, 078h, 0fch, 030h, 0fch, 030h, 030h
200 0001321C 386C64F060E6FC00CC- <1>
200 00013225 CC78FC30FC3030
                                        <1>
201 0001322C F8CCCCFAC6CFC6C70E- <1>
                                                      0f8h, 0cch, 0cch, 0fah, 0c6h, 0cfh, 0c6h, 0c7h, 00eh, 01bh, 018h, 03ch, 018h, 018h, 048h, 070h
201 00013235 1B183C1818D870
                                        <1>
202 0001323C 1C00780C7CCC7E0038- <1>
                                                      01ch, 000h, 078h, 00ch, 07ch, 0cch, 07eh, 000h, 038h, 000h, 070h, 030h, 030h, 030h, 078h, 000h
                                                  db
202 00013245 00703030307800
                                        <1>
203 0001324C 001C0078CCCC780000- <1>
                                                      000h, 01ch, 000h, 078h, 0cch, 0cch, 078h, 000h, 000h, 01ch, 000h, 0cch, 0cch, 07eh, 000h
203 00013255 1C00CCCCC7E00
                                       <1>
204 0001325C 00F800F8CCCCCC00FC- <1>
                                                      000h, 0f8h, 000h, 0f8h, 0cch, 0cch, 0cch, 000h, 0fch, 000h, 0cch, 0ech, 0fch, 0dch, 0cch, 000h
204 00013265 00CCECFCDCCC00
                                       <1>
                                                      03ch, 06ch, 06ch, 03eh, 000h, 07eh, 000h, 000h, 038h, 06ch, 06ch, 038h, 000h, 07ch, 000h, 000h
205 0001326C 3C6C6C3E007E000038- <1>
205 00013275 6C6C38007C0000
                                       <1>
206 0001327C 30003060C0CC780000- <1>
                                                      030h, 000h, 030h, 060h, 0c0h, 0cch, 078h, 000h, 000h, 000h, 000h, 0fch, 0c0h, 0c0h, 000h, 000h
206 00013285 0000FCC0C00000
                                      <1>
207 0001328C 000000FC0C0C0000C3- <1>
                                                      000h, 000h, 000h, 0fch, 00ch, 00ch, 000h, 000h, 0c3h, 0c6h, 0cch, 0deh, 033h, 066h, 0cch, 00fh
```

```
208 0001329C C3C6CCDB376FCF0318- <1>
                                             0c3h, 0c6h, 0cch, 0dbh, 037h, 06fh, 0cfh, 003h, 018h, 018h, 000h, 018h, 018h, 018h, 018h, 000h
208 000132A5 18001818181800
                                 <1>
209 000132AC 003366CC6633000000- <1>
                                             000h, 033h, 066h, 0cch, 066h, 033h, 000h, 000h, 000h, 0cch, 066h, 033h, 066h, 0cch, 000h, 000h
                                         db
209 000132B5 CC663366CC0000
                                 <1>
210 000132BC 228822882288228855- <1>
                                         db
                                             022h, 088h, 022h, 088h, 022h, 088h, 022h, 088h, 055h, 0aah, 055h, 0aah, 055h, 0aah, 055h, 0aah
210 000132C5 AA55AA55AA55AA
                                 <1>
211 000132CC DB77DBEEDB77DBEE18- <1>
                                             0dbh, 077h, 0dbh, 0eeh, 0dbh, 077h, 0dbh, 0eeh, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h
211 000132D5 18181818181818
                                 <1>
212 000132DC 18181818F8181818- <1>
                                             018h, 018h
                                         db
212 000132E5 18F818F8181818
                                 <1>
213 000132EC 36363636F636363600- <1>
                                             036h, 036h, 036h, 036h, 056h, 056h, 036h, 036h, 000h, 000h, 000h, 000h, 056h, 036h, 036h, 036h
213 000132F5 000000FE363636
                                 <1>
214 000132FC 0000F818F818181836- <1>
                                             000h, 000h, 0f8h, 018h, 0f8h, 018h, 018h, 018h, 036h, 036h, 066h, 066h, 066h, 036h, 036h, 036h
                                         db
214 00013305 36F606F6363636
                                 <1>
215 0001330C 36363636363636363600- <1>
                                         db
                                             036h, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 000h, 000h, 0feh, 006h, 0f6h, 036h, 036h, 036h
215 00013315 00FE06F6363636
                                 <1>
216 0001331C 3636F606FE00000036- <1>
                                             036h, 036h, 0f6h, 006h, 0feh, 000h, 000h, 000h, 036h, 036h, 036h, 0feh, 000h, 000h, 000h
216 00013325 363636FE000000
                                 <1>
217 0001332C 1818F818F800000000- <1>
                                         db
                                             018h, 018h, 0f8h, 018h, 0f8h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 0f8h, 018h, 018h, 018h
217 00013335 000000F8181818
                                 <1>
218 0001333C 181818181F00000018- <1>
                                             018h, 018h, 018h, 018h, 01fh, 000h, 000h, 000h, 018h, 018h, 018h, 018h, 0ffh, 000h, 000h, 000h
218 00013345 181818FF000000
                                 <1>
219 0001334C 00000000FF18181818- <1>
                                             000h, 000h, 000h, 000h, 0ffh, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h
                                         db
219 00013355 1818181F181818
                                 <1>
220 0001335C 00000000FF00000018- <1>
                                             000h, 000h, 000h, 000h, 0ffh, 000h, 000h, 000h, 018h, 018h, 018h, 018h, 018h, 018h, 018h
220 00013365 181818FF181818
                                 <1>
221 0001336C 18181F181F18181836- <1>
                                             018h, 018h, 01fh, 018h, 01fh, 018h, 018h, 018h, 036h, 036h, 036h, 036h, 037h, 036h, 036h, 036h
                                         db
221 00013375 36363637363636
                                 <1>
222 0001337C 363637303F00000000- <1>
                                         db
                                             036h, 036h, 037h, 030h, 03fh, 000h, 000h, 000h, 000h, 03fh, 030h, 037h,
                                                                                                                           036h, 036h, 036h
222 00013385 003F3037363636
                                 <1>
223 0001338C 3636F700FF00000000- <1>
                                         db
                                             036h, 036h, 0f7h, 000h, 0ffh, 000h, 000h, 000h, 000h, 000h, 0ffh, 000h, 0f7h, 036h, 036h, 036h
223 00013395 00FF00F7363636
                                 <1>
224 0001339C 363637303736363600- <1>
                                             036h, 036h, 037h, 030h, 037h, 036h, 036h, 036h, 000h, 000h, 0ffh, 000h, 0ffh, 000h, 000h, 000h
                                         db
224 000133A5 00FF00FF000000
                                 <1>
225 000133AC 3636F700F736363618- <1>
                                             036h, 036h, 0f7h, 000h, 0f7h, 036h, 036h, 036h, 018h, 018h, 0ffh, 000h, 0ffh, 000h, 000h, 000h
225 000133B5 18FF00FF000000
                                 <1>
226 000133BC 36363636FF00000000- <1>
                                             036h, 036h, 036h, 036h, 0ffh, 000h, 000h, 000h, 000h, 000h, 0ffh, 000h, 0ffh, 018h, 018h, 018h
                                         db
226 000133C5 00FF00FF181818
                                 <1>
227 000133CC 00000000FF36363636- <1>
                                             000h, 000h,
                                                         000h, 000h, 0ffh, 036h, 036h, 036h, 036h, 036h, 036h, 03fh,
                                                                                                                            000h, 000h, 000h
                                         db
227 000133D5 3636363F000000
228 000133DC 18181F181F00000000- <1>
                                             018h, 018h, 01fh, 018h, 01fh, 000h, 000h, 000h, 000h, 000h, 01fh, 018h, 01fh, 018h, 018h, 018h
                                         db
228 000133E5 001F181F181818
                                 <1>
229 000133EC 000000003F36363636- <1>
                                         db
                                             000h, 000h, 000h, 000h, 03fh, 036h, 036h
229 000133F5 363636FF363636
                                 <1>
230 000133FC 1818FF18FF18181818-
                                 <1>
                                         db
                                             018h, 018h, 0ffh, 018h, 0ffh, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 000h, 000h, 000h
230 00013405 181818F8000000
                                 <1>
231 0001340C 000000001F181818FF- <1>
                                             000h, 000h, 000h, 000h, 01fh, 018h, 018h, 018h, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh
                                         db
231\ 00013415\ FFFFFFFFFFFFF
                                 <1>
232 0001341C 00000000FFFFFFFF0- <1>
                                             000h, 000h, 000h, 000h, 0ffh, 0ffh, 0ffh, 0ffh, 0f0h, 0f0h, 0f0h, 0f0h, 0f0h, 0f0h, 0f0h, 0f0h
232 00013425 F0F0F0F0F0F0F0
                                 <1>
233 0001342C 0F0F0F0F0F0F0F0FFF- <1>
                                             00fh, 00fh, 00fh, 00fh, 00fh, 00fh, 00fh, 00fh, 0ffh, 0ffh, 0ffh, 0ffh, 000h, 000h, 000h, 000h
                                         db
233 00013435 FFFFFF00000000
                                 <1>
234 0001343C 000076DCC8DC760000- <1>
                                             000h, 000h, 076h, 0dch, 0c8h, 0dch, 076h, 000h, 000h, 078h, 0cch, 0f8h, 0cch, 0f8h, 0c0h, 0c0h
234 00013445 78CCF8CCF8C0C0
235 0001344C 00FCCCC0C0C0C000000- <1>
                                             000h, 0fch, 0cch, 0c0h, 0c0h, 0c0h, 0c0h, 000h, 000h, 0feh, 06ch, 06ch, 06ch, 06ch, 06ch, 06ch, 000h
                                         db
235 00013455 FE6C6C6C6C6C00
                                 <1>
236 0001345C FCCC603060CCFC0000- <1>
                                             0fch, 0cch, 060h, 030h, 060h, 0cch, 0fch, 000h, 000h, 000h, 07eh, 0d8h, 0d8h, 0d8h, 070h, 000h
236 00013465 007ED8D8D87000
                                 <1>
237 0001346C 00666666667C60C000- <1>
                                             000h, 066h, 066h, 066h, 066h, 07ch, 060h, 0c0h, 000h, 076h, 0dch, 018h, 018h, 018h, 000h
237 00013475 76DC1818181800
                                 <1>
238 0001347C FC3078CCCC7830FC38- <1>
                                             Ofch, 030h, 078h, 0cch, 0cch, 078h, 030h, 0fch, 038h, 06ch, 0c6h, 0feh, 0c6h, 06ch, 038h, 000h
                                         db
238 00013485 6CC6FEC66C3800
                                 <1>
239 0001348C 386CC6C66C6CEE001C- <1>
                                             038h, 06ch, 0c6h, 0c6h, 06ch, 06ch, 0eeh, 000h, 01ch, 030h, 018h, 07ch, 0cch, 0cch, 078h, 000h
239 00013495 30187CCCCC7800
                                 <1>
240 0001349C 00007EDBDB7E000006- <1>
                                             000h, 000h, 07eh, 0dbh, 07eh, 000h, 000h, 000h, 00ch, 07eh, 0dbh, 07eh, 060h, 0c0h
                                         db
240 000134A5 0C7EDBDB7E60C0
                                 <1>
241 000134AC 3860C0F8C060380078- <1>
                                             038h, 060h, 0c0h, 0f8h, 0c0h, 060h, 038h, 000h, 078h, 0cch, 0cch, 0cch, 0cch, 0cch, 0cch, 000h
                                         db
241 000134B5 CCCCCCCCCC00
                                 <1>
242 000134BC 00FC00FC00FC000030- <1>
                                             000h, 0fch, 000h, 0fch, 000h, 0fch, 000h, 000h, 030h, 030h, 0fch, 030h, 030h, 000h, 0fch, 000h
                                         db
242 000134C5 30FC303000FC00
                                 <1>
243 000134CC 603018306000FC0018- <1>
                                         db
                                             060h, 030h, 018h, 030h, 060h, 000h, 0fch, 000h, 018h, 030h, 060h, 030h, 018h, 000h, 0fch, 000h
243 000134D5 3060301800FC00
                                 <1>
244 000134DC 0E1B1B181818181818- <1>
                                             00eh, 01bh, 01bh, 018h, 048h, 070h
244 000134E5 18181818D8D870
                                 <1>
245 000134EC 303000FC0030300000- <1>
                                             030h, 030h, 000h, 0fch, 000h, 030h, 030h, 000h, 000h, 076h, 0dch, 000h, 076h, 0dch, 000h, 000h
                                         db
245 000134F5 76DC0076DC0000
                                 <1>
246 000134FC 386C6C380000000000 <1>
                                             038h, 06ch, 06ch, 038h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 018h, 018h, 000h, 000h, 000h
246 00013505 00001818000000
                                 <1>
247 0001350C 00000000180000000F- <1>
                                             000h, 000h, 000h, 000h, 018h, 000h, 000h, 000h, 00fh, 00ch, 00ch, 00ch, 0ech, 06ch, 03ch, 01ch
                                         db
247 00013515 0C0C0CEC6C3C1C
                                 <1>
                                             078h, 06ch, 06ch, 06ch, 06ch, 000h, 000h, 000h, 070h, 018h, 030h, 060h, 078h, 000h, 000h, 000h
248 0001351C 786C6C6C6C00000070- <1>
                                         db
248 00013525 18306078000000
                                 <1>
249 0001352C 00003C3C3C3C000000-
                                 <1>
                                         db 000h, 000h, 03ch, 03ch, 03ch, 03ch, 000h, 000h
249 00013535 00000000000000
                                 <1>
                                 <1> vgafont14:
250
    0001353C 0000000000000000000000
                                         db 000h, 000h
                                 <1>
251 00013545 00000000000000
252 0001354C 7E81A58181BD99817E- <1>
                                             07eh, 081h, 0a5h, 081h, 081h, 0bdh, 099h, 081h, 07eh, 000h, 000h, 000h, 000h, 000h, 07eh, 0ffh
252 00013555 00000000007EFF
                                <1>
                                             Odbh, Offh, Offh, Oc3h, Oe7h, Offh, O7eh, O00h, O00h, O00h, O00h, O00h, O00h, O6ch, Ofeh, Ofeh
253 0001355C DBFFFFC3E7FF7E0000- <1>
253 00013565 000000006CFEFE
                                <1>
254 0001356C FEFE7C381000000000- <1>
                                             0feh, 0feh, 07ch, 038h, 010h, 000h, 000h, 000h, 000h, 000h, 010h, 038h, 07ch, 0feh, 07ch
254 00013575 000010387CFE7C
                                 <1>
255 0001357C 38100000000000018- <1>
                                             038h, 010h, 000h, 000h, 000h, 000h, 000h, 000h, 018h, 03ch, 03ch, 0e7h, 0e7h, 0e7h, 018h, 018h
255 00013585 3C3CE7E7E71818
                                 <1>
256 0001358C 3C000000000183C7E- <1>
                                             03ch, 000h, 000h, 000h, 000h, 000h, 018h, 03ch, 07eh, 0ffh, 0feh, 018h, 018h, 03ch, 000h
256 00013595 FFFF7E18183C00
                                 <1>
257 0001359C 0000000000000183C- <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 000h, 018h, 03ch, 03ch, 018h, 000h, 000h, 000h, 000h, 000h
257 000135A5 3C18000000000
                                 <1>
258 000135AC FFFFFFFFFFF7C3C3E7- <1>
                                             Offh, Offh, Offh, Offh, Offh, Oe7h, Oc3h, Oc3h, Oe7h, Offh, Offh, Offh, Offh, Offh, O00h, O00h
                                         db
258 000135B5 FFFFFFFFF0000
                                 <1>
259 000135BC 00003C664242663C00- <1>
                                             000h, 000h, 03ch, 066h, 042h, 042h, 066h, 03ch, 000h, 000h, 000h, 000h, 0ffh, 0ffh, 0ffh
259 000135C5 000000FFFFFFF
                                 <1>
260 000135CC C399BDBD99C3FFFFFF- <1>
                                             0c3h, 099h, 0bdh, 0bdh, 099h, 0c3h, 0ffh, 0ffh, 0ffh, 00fh, 000h, 01eh, 00eh, 01ah, 032h
260 000135D5 FF00001E0E1A32
                                <1>
261 000135DC 78CCCCCC7800000000- <1>
                                             078h, Occh, Occh, Occh, 078h, 000h, 000h, 000h, 000h, 000h, 03ch, 066h, 066h, 066h, 03ch, 018h
261 000135E5 003C6666663C18
                                <1>
262 000135EC 7E18180000000003F- <1>
                                             07eh, 018h, 018h, 000h, 000h, 000h, 000h, 03fh, 03fh, 03fh, 030h, 030h, 030h, 070h, 0f0h
262 000135F5 333F30303070F0
                                <1>
263 000135FC E00000000007F637F- <1>
                                             0e0h, 000h, 000h, 000h, 000h, 000h, 07fh, 063h, 07fh, 063h, 063h, 063h, 067h, 0e7h, 0e6h, 0c0h
```

207 00013295 C6CCDE3366CC0F

```
264 0001360C 000000001818DB3CE7- <1>
                                             000h, 000h, 000h, 000h, 018h, 018h, 018h, 03ch, 0e7h, 03ch, 0dbh, 018h, 018h, 000h, 000h, 000h
264 00013615 3CDB1818000000
                                 <1>
265 0001361C 000080C0E0F8FEF8E0- <1>
                                             000h, 000h, 080h, 0c0h, 0e0h, 0f8h, 0feh, 0f8h, 0e0h, 0c0h, 080h, 000h, 000h, 000h, 000h, 000h
                                         db
265 00013625 C0800000000000
                                 <1>
266 0001362C 02060E3EFE3E0E0602- <1>
                                         db
                                             002h, 006h, 00eh, 03eh, 0feh, 03eh, 00eh, 006h, 002h, 000h, 000h, 000h, 000h, 018h, 03ch
266 00013635 0000000000183C
                                 <1>
267 0001363C 7E1818187E3C180000- <1>
                                             07eh, 018h, 018h, 07eh, 03ch, 018h, 000h, 000h, 000h, 000h, 000h, 066h, 066h, 066h, 066h
267 00013645 00000066666666
                                 <1>
268 0001364C 6666006666000000000- <1>
                                             066h, 066h, 000h, 066h, 066h, 000h, 000h, 000h, 000h, 07fh, 0dbh, 0dbh,
                                         db
                                                                                                                            0dbh, 07bh, 01bh
268 00013655 007FDBDBDB7B1B
                                 <1>
269 0001365C 1B1B1B00000007CC6- <1>
                                             01bh, 01bh, 01bh, 000h, 000h, 000h, 000h, 07ch, 0c6h, 060h, 038h, 06ch, 0c6h, 0c6h, 06ch, 038h
269 00013665 60386CC6C66C38
                                 <1>
270 0001366C 0CC67C0000000000000 <1>
                                             00ch, 0c6h, 07ch, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 0feh, 0feh, 0feh, 000h
                                         db
270 00013675 000000FEFEFE00
                                 <1>
271 0001367C 00000000183C7E1818- <1>
                                         db
                                             000h, 000h, 000h, 000h, 018h, 03ch, 07eh, 018h, 018h, 018h, 07eh, 03ch, 018h, 07eh, 000h, 000h
271 00013685 187E3C187E0000
                                 <1>
272 0001368C 0000183C7E18181818- <1>
                                             000h, 000h, 018h, 03ch, 07eh, 018h, 018h, 018h, 018h, 018h, 018h, 000h, 000h, 000h, 000h, 000h
272 00013695 18180000000000
                                 <1>
273 0001369C 1818181818187E3C18- <1>
                                         db
                                             018h, 018h, 018h, 018h, 018h, 018h, 07eh, 03ch, 018h, 000h, 000h, 000h, 000h,
                                                                                                                            000h, 000h, 000h
273 000136A5 00000000000000
                                 <1>
274 000136AC 180CFE0C1800000000- <1>
                                             018h, 00ch, 0feh, 00ch, 018h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 030h, 060h
274 000136B5 0000000003060
                                 <1>
275 000136BC FE60300000000000000 <1>
                                             Ofeh, 060h, 030h, 000h, 0c0h, 0c0h, 0c0h
                                         db
275 000136C5 00000000C0C0C0
                                 <1>
276 000136CC FE00000000000000000 <1>
                                             0feh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 028h, 06ch, 0feh, 06ch, 028h, 000h
                                         db
276 000136D5 00286CFE6C2800
277 000136DC 00000000000001038- <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 000h, 010h, 038h, 038h, 07ch, 07ch, 0feh, 0feh, 000h, 000h
                                         db
277 000136E5 387C7CFEFE0000
                                 <1>
278 000136EC 0000000000FEFE7C7C- <1>
                                         db
                                             000h, 000h,
                                                         000h, 000h, 000h, 0feh, 0feh, 07ch, 07ch, 038h, 038h, 010h, 000h,
                                                                                                                            000h, 000h, 000h
278 000136F5 38381000000000
                                 <1>
279 000136FC 000000000000000000 <1>
                                         db
                                             000h, 000h
279 00013705 00000000000000
                                 <1>
280 0001370C 183C3C3C1818001818- <1>
                                         db
                                             018h, 03ch, 03ch, 03ch, 018h, 018h, 000h, 018h, 018h, 000h, 000h, 000h, 000h, 066h, 066h, 066h
280 00013715 00000000666666
                                 <1>
281 0001371C 2400000000000000000 <1>
                                             024h, 000h, 06ch, 06ch, 0feh, 06ch
281 00013725 0000006C6CFE6C
                                 <1>
282 0001372C 6C6CFE6C6C00000018- <1>
                                             06ch, 06ch, 0feh, 06ch, 06ch, 000h, 000h, 000h, 018h, 018h, 07ch, 0c6h, 0c2h, 0c0h, 07ch, 006h
                                         db
282 00013735 187CC6C2C07C06
                                 <1>
283 0001373C 86C67C181800000000- <1>
                                                         07ch, 018h, 018h, 000h, 000h, 000h, 000h, 0c2h,
                                                                                                                0c6h, 00ch,
                                                                                                                            018h, 030h, 066h
                                         db
                                             086h,
                                                   0c6h,
283 00013745 00C2C60C183066
284 0001374C C6000000000386C6C- <1>
                                             0c6h, 000h, 000h, 000h, 000h, 000h, 038h, 06ch, 06ch, 038h, 076h, 0dch, 0cch, 0cch, 076h, 000h
                                         db
284 00013755 3876DCCCCC7600
                                 <1>
                                             000h, 000h, 000h, 030h, 030h, 030h, 060h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
285 0001375C 000000303030600000- <1>
                                         db
285 00013765 00000000000000
                                 <1>
286 0001376C 00000C183030303030-
                                 <1>
                                         db
                                             000h, 000h, 00ch, 018h, 030h, 030h, 030h, 030h, 030h, 018h, 00ch, 000h, 000h, 000h, 000h, 000h
286 00013775 180C0000000000
                                 <1>
287 0001377C 30180C0C0C0C0C1830- <1>
                                             030h, 018h, 00ch, 00ch, 00ch, 00ch, 00ch, 018h, 030h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                         db
287 00013785 00000000000000
                                 <1>
   0001378C 663CFF3C6600000000- <1>
                                             066h, 03ch, 0ffh, 03ch, 066h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h,
288
                                                                                                                            000h, 018h, 018h
288 00013795 0000000001818
                                 <1>
289 0001379C 7E1818000000000000- <1>
                                             07eh, 018h, 018h, 000h, 000h
                                         db
289 000137A5 00000000000000
                                 <1>
290 000137AC 181818300000000000 <1>
                                             018h, 018h, 018h, 030h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 0feh, 000h, 000h, 000h
                                         db
   000137B5 000000FE000000
291 000137BC 00000000000000000000
                                             000h, 018h, 018h, 000h
                                         db
291 000137C5 00000000181800
                                 <1>
292 000137CC 0000000002060C1830- <1>
                                         db
                                             000h, 000h, 000h, 000h, 002h, 006h, 00ch, 018h, 030h, 060h, 0c0h, 080h, 000h, 000h, 000h, 000h
292 000137D5 60C08000000000
                                 <1>
   000137DC 00007CC6CEDEF6E6C6- <1>
                                             000h, 000h, 07ch, 0c6h, 0ceh, 0deh, 0f6h, 0e6h, 0c6h, 07ch, 000h, 000h,
                                                                                                                            000h, 000h, 000h
293 000137E5 C67C0000000000
                                 <1>
294 000137EC 18387818181818187E- <1>
                                             018h, 038h, 078h, 018h, 018h, 018h, 018h, 018h, 07eh, 000h, 000h, 000h, 000h, 07ch, 0c6h
                                         db
294 000137F5 00000000007CC6
                                 <1>
295 000137FC 060C183060C6FE0000- <1>
                                             006h, 00ch, 018h, 030h, 060h, 0c6h, 0feh, 000h, 000h, 000h, 000h, 000h, 07ch, 0c6h, 006h, 006h
295 00013805 0000007CC60606
                                 <1>
296 0001380C 3C0606C67C00000000- <1>
                                             03ch, 006h, 006h, 0c6h, 07ch, 000h, 000h, 000h, 000h, 000h, 00ch, 01ch, 03ch, 06ch, 0cch, 0feh
                                         db
296 00013815 000C1C3C6CCCFE
                                 <1>
297 0001381C 0C0C1E0000000000FE- <1>
                                             00ch, 00ch, 01eh, 000h, 000h, 000h, 000h, 000h, 0feh, 0c0h, 0c0h, 0c0h, 0fch, 006h, 006h, 0c6h
                                         db
297 00013825 C0C0C0FC0606C6
                                 <1>
298 0001382C 7C0000000003860C0- <1>
                                             07ch, 000h, 000h, 000h, 000h, 000h, 000h, 038h, 060h, 0c0h, 0c0h, 0fch, 0c6h, 0c6h, 0c6h, 07ch, 000h
                                         db
298 00013835 C0FCC6C6C67C00
                                 <1>
299 0001383C 00000000FEC6060C18- <1>
                                         db
                                             000h, 000h, 000h, 000h, 0feh, 0c6h, 006h, 00ch, 018h, 030h, 030h, 030h, 030h, 000h, 000h, 000h
299 00013845 30303030000000
                                 <1>
300 0001384C 00007CC6C6C6C67CC6C6- <1>
                                             000h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 07ch, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h, 000h
300 00013855 C67C0000000000
                                 <1>
301 0001385C 7CC6C6C67E06060C78- <1>
                                             07ch, 0c6h, 0c6h, 0c6h, 07eh, 006h, 006h, 00ch, 078h, 000h, 000h, 000h, 000h,
                                                                                                                            000h, 000h, 018h
                                         db
301 00013865 00000000000018
                                 <1>
302 0001386C 180000001818000000- <1>
                                             018h, 000h, 000h, 000h, 018h, 018h, 000h, 000h, 000h, 000h, 000h, 000h, 018h, 018h, 000h
302 00013875 00000000181800
                                 <1>
303 0001387C 000018183000000000- <1>
                                             000h, 000h, 018h, 018h, 030h, 000h, 000h, 000h, 000h, 000h, 006h, 00ch, 018h, 030h, 060h, 030h
                                         db
303 00013885 00060C18306030
                                 <1>
304 0001388C 180C060000000000000 <1>
                                             018h, 00ch, 006h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 07eh, 000h, 07eh, 000h
                                         db
304 00013895 00007E00007E00
                                 <1>
305 0001389C 00000000000603018- <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 060h, 030h, 018h, 00ch, 006h, 00ch, 018h, 030h, 060h, 000h
305 000138A5 0C060C18306000
                                 <1>
306 000138AC 000000007CC6C60C18- <1>
                                         db
                                             000h, 000h, 000h, 000h, 07ch, 0c6h, 0c6h, 00ch, 018h, 018h, 000h, 018h, 018h, 000h, 000h, 000h
306 000138B5 18001818000000
                                 <1>
307 000138BC 00007CC6C6DEDEDEDC- <1>
                                             000h, 000h, 07ch, 0c6h, 0c6h, 0deh, 0deh, 0deh, 0dch, 0c0h, 07ch, 000h, 000h, 000h, 000h, 000h
307 000138C5 C07C0000000000
                               <1>
308 000138CC 10386CC6C6FEC6C6C6- <1>
                                             010h, 038h, 06ch, 0c6h, 0c6h, 0feh, 0c6h, 0c6h, 0c6h, 000h, 000h, 000h, 000h, 000h, 0fch, 066h
308 000138D5 000000000FC66
                                <1>
309 000138DC 66667C666666FC0000- <1>
                                             066h, 066h, 07ch, 066h, 066h, 066h, 0fch, 000h, 000h, 000h, 000h, 000h, 03ch, 066h, 0c2h, 0c0h
309 000138E5 0000003C66C2C0
                                 <1>
310 000138EC C0C0C2663C00000000- <1>
                                             0c0h, 0c0h, 0c2h, 066h, 03ch, 000h, 000h, 000h, 000h, 006h, 066h, 066h, 066h, 066h, 066h
310 000138F5 00F86C66666666
                                 <1>
311 000138FC 666CF8000000000FE- <1>
                                             066h, 06ch, 0f8h, 000h, 000h, 000h, 000h, 000h, 0feh, 066h, 062h, 068h, 078h, 068h, 062h, 066h
                                         db
311 00013905 66626878686266
                                 <1>
312 0001390C FE000000000FE6662- <1>
                                             0feh, 000h, 000h, 000h, 000h, 000h, 0feh, 066h, 062h, 068h, 078h, 068h, 060h, 060h, 0f0h, 000h
312 00013915 6878686060F000
                                 <1>
313 0001391C 000000003C66C2C0C0- <1>
                                             000h, 000h, 000h, 000h, 03ch, 066h, 0c2h, 0c0h, 0c0h, 0deh, 0c6h, 066h, 03ah, 000h, 000h, 000h
313 00013925 DEC6663A000000
                                 <1>
314 0001392C 0000C6C6C6C6FEC6C6- <1>
                                             000h, 000h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 000h, 000h, 000h, 000h, 000h
314 00013935 C6C60000000000
                                 <1>
315 0001393C 3C181818181818183C- <1>
                                             03ch, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h, 000h, 01eh, 00ch
315 00013945 00000000001E0C
                                 <1>
316 0001394C 0C0C0C0CCCCC780000- <1>
                                             00ch, 00ch, 00ch, 00ch, 0cch, 0cch, 078h, 000h, 000h, 000h, 000h, 000h, 066h, 066h, 06ch, 06ch
                                         db
316 00013955 000000E6666C6C
                                <1>
317 0001395C 786C6C66E600000000- <1>
                                             078h, 06ch, 06ch, 066h, 0e6h, 000h, 000h, 000h, 000h, 000h, 0f0h, 060h, 060h, 060h, 060h, 060h
317 00013965 00F06060606060
                               <1>
318 0001396C 6266FE0000000000C6- <1>
                                         db 062h, 066h, 0feh, 000h, 000h, 000h, 000h, 000h, 0c6h, 0eeh, 0feh, 0feh, 0d6h, 0c6h, 0c6h, 0c6h
318 00013975 EEFEFED6C6C6C6
                                <1>
```

263 00013605 63636367E7E6C0

0c6h, 000h, 000h, 000h, 000h, 000h, 0c6h, 0e6h, 0f6h, 0feh, 0deh, 0ceh, 0c6h, 0c6h, 0c6h, 000h

```
319 00013985 FEDECEC6C6C600
                                 <1>
320 0001398C 00000000386CC6C6C6- <1>
                                             000h, 000h, 000h, 000h, 038h, 06ch, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 06ch, 038h, 000h, 000h, 000h
320 00013995 C6C66C38000000
                                 <1>
                                             000h, 000h, 0fch, 066h, 066h, 07ch, 060h, 060h, 060h, 060h, 000h, 000h, 000h, 000h, 000h
321 0001399C 0000FC6666667C6060- <1>
                                         db
321 000139A5 60F0000000000
                                 <1>
322 000139AC 7CC6C6C6C6D6DE7C0C- <1>
                                             07ch, 0c6h, 0c6h, 0c6h, 0c6h, 0d6h, 0deh, 07ch, 00ch, 00eh, 000h, 000h, 000h, 000h, 0fch, 066h
322 000139B5 0E0000000FC66
323 000139BC 66667C6C6666E60000- <1>
                                             066h, 066h, 07ch, 06ch, 066h, 066h, 0e6h, 000h, 000h, 000h, 000h, 07ch, 0c6h, 0c6h, 060h
                                         db
323 000139C5 0000007CC6C660
                                 <1>
324 000139CC 380CC6C67C00000000- <1>
                                             038h, 00ch, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h, 000h, 07eh, 07eh, 05ah, 018h, 018h, 018h
                                         db
324 000139D5 007E7E5A181818
                                 <1>
325 000139DC 18183C000000000C6- <1>
                                             018h, 018h, 03ch, 000h, 000h, 000h, 000h, 000h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h
325 000139E5 C6C6C6C6C6C6C6
                                 <1>
326 000139EC 7C0000000000C6C6C6- <1>
                                             07ch, 000h, 000h, 000h, 000h, 000h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h,
                                                                                                                           038h, 010h, 000h
                                         db
326 000139F5 C6C6C66C381000
                                 <1>
327 000139FC 00000000C6C6C6C6D6- <1>
                                             000h, 000h, 000h, 000h, 0c6h, 0c6h, 0c6h, 0c6h, 0d6h, 0d6h, 0feh, 07ch, 06ch, 000h, 000h, 000h
327 00013A05 D6FE7C6C000000
                                 <1>
328 00013A0C 0000C6C66C3838386C- <1>
                                             000h, 000h, 0c6h, 0c6h, 06ch, 038h, 038h, 038h, 06ch, 0c6h, 0c6h, 000h, 000h, 000h, 000h, 000h
                                         db
328 00013A15 C6C60000000000
                                 <1>
329 00013A1C 66666663C1818183C- <1>
                                             066h, 066h, 066h, 066h, 03ch, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h, 000h, 0feh, 0c6h
                                         db
329 00013A25 0000000000FEC6
                                 <1>
                                             08ch, 018h, 030h, 060h, 0c2h, 0c6h, 0feh, 000h, 000h, 000h, 000h, 03ch, 030h, 030h, 030h
330 00013A2C 8C183060C2C6FE0000- <1>
                                         db
330 00013A35 0000003C303030
                                 <1>
331 00013A3C 303030303C00000000- <1>
                                         db
                                             030h, 030h,
                                                         030h, 030h, 03ch, 000h, 000h, 000h, 000h, 000h, 080h, 0c0h, 0e0h,
                                                                                                                            070h, 038h, 01ch
331 00013A45 0080C0E070381C
                                 <1>
332 00013A4C 0E06020000000003C- <1>
                                             00eh, 006h, 002h, 000h, 000h, 000h, 000h, 000h, 03ch, 00ch, 00ch, 00ch, 00ch, 00ch, 00ch
332 00013A55 0C0C0C0C0C0C0C
                                 <1>
333 00013A5C 3C00000010386CC600- <1>
                                             03ch, 000h, 000h, 000h, 010h, 038h, 06ch, 0c6h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                         db
333 00013A65 00000000000000
                                 <1>
334 00013A6C 000000000000000000 <1>
                                             000h, 000h
                                         db
334 00013A75 0000000000FF00
                                 <1>
335 00013A7C 3030180000000000000 <1>
                                             030h, 030h, 018h, 000h, 000h
                                         db
335 00013A85 0000000000000
                                 <1>
336 00013A8C 000000780C7CCCC
                                                                                                                            000h, 0e0h, 060h
                             276- <1>
                                             000h, 000h,
                                                         000h, 078h, 00ch, 07ch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h,
                                         db
336 00013A95 0000000000E060
337 00013A9C 60786C6666667C0000- <1>
                                             060h, 078h, 06ch, 066h, 066h, 066h, 07ch, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 07ch
                                         db
337 00013AA5 0000000000007C
                                 <1>
338 00013AAC C6C0C0C67C00000000- <1>
                                         db
                                             0c6h, 0c0h, 0c0h, 0c6h, 07ch, 000h, 000h, 000h, 000h, 000h, 01ch, 00ch, 00ch, 03ch, 06ch, 0cch
338 00013AB5 001C0C0C3C6CCC
                                 <1>
339 00013ABC CCCC760000000000000- <1>
                                             Occh, Occh, 076h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 07ch, 0c6h, 0feh, 0c6h, 0c6h
339 00013AC5 00007CC6FEC0C6
                                 <1>
340 00013ACC 7C000000000386C64- <1>
                                             07ch, 000h, 000h, 000h, 000h, 000h, 038h, 06ch, 064h, 060h, 0f0h, 060h, 060h, 060h, 060h, 090h
                                         db
340 00013AD5 60F0606060F000
                                 <1>
341 00013ADC 000000000000076CC- <1>
                                                         000h, 000h, 000h, 000h, 000h, 076h, 0cch, 0cch, 0cch, 07ch, 00ch,
341 00013AE5 CCCC7C0CCC7800
                                 <1>
342 00013AEC 0000E060606C766666- <1>
                                             000h, 000h, 0e0h, 060h, 060h, 06ch, 076h, 066h, 066h, 066h, 0e6h, 000h, 000h, 000h, 000h, 000h
                                         db
342 00013AF5 66E60000000000
                                 <1>
343 00013AFC 18180038181818183C- <1>
                                             018h, 018h, 000h, 038h, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h, 000h, 006h, 006h
                                         db
343 00013B05 00000000000606
                                 <1>
                                             000h, 00eh, 006h, 006h, 006h, 006h, 066h, 066h, 03ch, 000h, 000h, 000h, 0e0h, 060h, 066h
344 00013B0C 000E0606060666663C-
                                 <1>
                                         db
344 00013B15 000000E0606066
                                 <1>
345 00013B1C 6C786C66E600000000- <1>
                                         db
                                             06ch, 078h, 06ch, 066h, 0e6h, 000h, 000h, 000h, 000h, 000h, 038h, 018h, 018h, 018h, 018h, 018h
345 00013B25 00381818181818
                                 <1>
346 00013B2C 18183C000000000000 <1>
                                         db
                                             018h, 018h, 03ch, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 0ech, 0feh, 0d6h, 0d6h, 0d6h
346 00013B35 0000ECFED6D6D6
                                 <1>
347 00013B3C C600000000000000000 <1>
                                             0c6h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 00ch, 066h, 066h, 066h, 066h, 066h, 000h
                                         db
347 00013B45 DC66666666600
                                 <1>
348 00013B4C 00000000000007CC6- <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h
348 00013B55 C6C6C67C000000
                                 <1>
349 00013B5C 000000000DC666666- <1>
                                             000h, 000h, 000h, 000h, 000h, 0dch, 066h, 066h, 066h, 07ch, 060h, 060h, 0f0h, 000h, 000h, 000h
                                         db
349 00013B65 7C6060F0000000
                                 <1>
350 00013B6C 00000076CCCCCC7C0C- <1>
                                             000h, 000h, 000h, 076h, 0cch, 0cch, 0cch, 07ch, 00ch, 00ch, 01eh, 000h, 000h, 000h, 000h, 000h
                                         db
350 00013B75 0C1E0000000000
351 00013B7C 00DC76666060F00000- <1>
                                             000h, 0dch, 076h, 066h, 060h, 060h, 0f0h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 07ch
                                         db
351 00013B85 000000000007C
                                 <1>
352 00013B8C C6701CC67C00000000- <1>
                                         db
                                             0c6h, 070h, 01ch, 0c6h, 07ch, 000h, 000h, 000h, 000h, 010h, 030h, 030h, 05ch, 030h, 030h
352 00013B95 00103030FC3030
                                 <1>
353 00013B9C 30361C0000000000000 <1>
                                             030h, 036h, 01ch, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 0cch, 0cch, 0cch, 0cch, 0cch
353 00013BA5 0000CCCCCCCCC
                                 <1>
354 00013BAC 7600000000000000000 <1>
                                             076h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 066h, 066h, 066h, 066h, 03ch, 018h, 000h
                                         db
354 00013BB5 66666663C1800
                                 <1>
355 00013BBC 00000000000000C6C6- <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 000h, 0c6h, 0c6h, 0d6h, 0d6h, 0feh, 06ch, 000h, 000h, 000h
355 00013BC5 D6D6FE6C000000
356 00013BCC 0000000000C66C3838- <1>
                                             000h, 000h, 000h, 000h, 000h, 006h, 06ch, 038h, 038h, 06ch, 0c6h, 000h, 000h, 000h, 000h, 000h
                                         db
356 00013BD5 6CC60000000000
                                 <1>
357 00013BDC 000000C6C6C6C67E06- <1>
                                         db
                                             000h, 000h, 000h, 0c6h, 0c6h, 0c6h, 0c6h, 07eh, 006h, 00ch, 0f8h, 000h, 000h, 000h, 000h, 000h
357 00013BE5 0CF80000000000
                                 <1>
358 00013BEC 00FECC183066FE0000- <1>
                                             000h, 0feh, 0cch, 018h, 030h, 066h, 0feh, 000h, 000h, 000h, 000h, 000h, 00eh, 018h, 018h, 018h
358 00013BF5 0000000E181818
                                 <1>
359 00013BFC 701818180E00000000- <1>
                                         db
                                             070h, 018h, 018h, 018h, 00eh, 000h, 000h, 000h, 000h, 000h, 018h, 018h, 018h, 018h, 000h, 018h
359 00013C05 00181818180018
                                 <1>
360 00013C0C 18181800000000070- <1>
                                             018h, 018h, 018h, 000h, 000h, 000h, 000h, 000h, 070h, 018h, 018h, 018h, 00eh, 018h, 018h, 018h
360 00013C15 1818180E181818
                                 <1>
361 00013C1C 7000000000076DC00- <1>
                                             070h, 000h, 000h, 000h, 000h, 000h, 076h, 0dch, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                         db
361 00013C25 00000000000000
                                 <1>
                                             000h, 000h,
                                                                                                                            000h, 000h, 000h
   00013C2C 0000000000010386C-
                                 <1>
                                         db
                                                         000h, 000h, 000h, 000h, 010h, 038h, 06ch, 0c6h, 0c6h, 0feh, 000h,
362 00013C35 C6C6FE00000000
363 00013C3C 00003C66C2C0C0C266- <1>
                                             000h, 000h, 03ch, 066h, 0c2h, 0c0h, 0c0h, 0c2h, 066h, 03ch, 00ch, 006h, 07ch, 000h, 000h, 000h
363 00013C45 3C0C067C000000
                                <1>
                                             Occh, Occh, 000h, 00ch, Occh, Occh, Occh, Occh, 076h, 000h, 000h, 000h, 000h, 000h, 018h, 030h
364 00013C4C CCCC00CCCCCCCCC76- <1>
                                         db
364 00013C55 000000000C1830
                                <1>
365 00013C5C 007CC6FEC0C67C0000- <1>
                                             000h, 07ch, 0c6h, 0feh, 0c0h, 0c6h, 07ch, 000h, 000h, 000h, 010h, 038h, 06ch, 000h, 078h
365 00013C65 000010386C0078
                                 <1>
                                             00ch, 07ch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h, 000h, 0cch, 0cch, 000h, 078h, 00ch, 07ch
366 00013C6C 0C7CCCCC7600000000- <1>
                                         db
366 00013C75 00CCCC00780C7C
                                 <1>
367 00013C7C CCCC76000000006030- <1>
                                             Occh, Occh, 076h, 000h, 000h, 000h, 000h, 060h, 030h, 018h, 000h, 078h, 00ch, 07ch, 0cch, 0cch
367 00013C85 1800780C7CCCC
                                 <1>
368 00013C8C 760000000386C3800- <1>
                                             076h, 000h, 000h, 000h, 000h, 038h, 06ch, 038h, 000h, 078h, 00ch, 07ch, 0cch, 076h, 000h
                                         db
368 00013C95 780C7CCCCC7600
                                 <1>
369 00013C9C 000000000003C6660- <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 03ch, 066h, 060h, 066h, 03ch, 00ch, 006h, 03ch, 000h, 000h
                                         db
369 00013CA5 663C0C063C0000
                                 <1>
370 00013CAC 0010386C007CC6FEC0- <1>
                                             000h, 010h, 038h, 06ch, 000h, 07ch, 0c6h, 0feh, 0c0h, 0c6h, 07ch, 000h, 000h, 000h, 000h, 000h
                                         db
370 00013CB5 C67C0000000000
                                <1>
371 00013CBC CCCC007CC6FEC0C67C- <1>
                                             Occh, Occh, 000h, 07ch, 0c6h, 0feh, 0c0h, 0c6h, 07ch, 000h, 000h, 000h, 000h, 060h, 030h, 018h
371 00013CC5 00000000603018
                                <1>
372 00013CCC 007CC6FEC0C67C0000- <1>
                                             000h, 07ch, 0c6h, 0feh, 0c0h, 0c6h, 07ch, 000h, 000h, 000h, 000h, 000h, 066h, 066h, 000h, 038h
372 00013CD5 00000066660038
                                <1>
373 00013CDC 181818183C00000000- <1>
                                             018h, 018h, 018h, 018h, 03ch, 03ch, 000h, 000h, 000h, 018h, 03ch, 066h, 000h, 038h, 018h, 018h
373 00013CE5 183C6600381818
                               <1>
374 00013CEC 18183C00000006030- <1>
                                            018h, 018h, 03ch, 000h, 000h, 000h, 000h, 060h, 030h, 018h, 000h, 038h, 018h, 018h, 018h, 018h
```

319 0001397C C6000000000C6E6F6- <1>

db

```
375 00013CFC 3C00000000C6C61038- <1>
                                                       03ch, 000h, 000h, 000h, 000h, 0c6h, 0c6h, 010h, 038h, 06ch, 0c6h, 0c6h, 0feh, 0c6h, 0c6h, 000h
375 00013D05 6CC6C6FEC6C600
376 00013D0C 0000386C3800386CC6- <1>
                                                       000h, 000h, 038h, 06ch, 038h, 000h, 038h, 06ch, 0c6h, 0c6h, 0feh, 0c6h, 0c6h, 000h, 000h, 000h
                                                  db
376 00013D15 C6FEC6C6000000
                                        <1>
377 00013D1C 18306000FE66607C60- <1>
                                                  db
                                                       018h, 030h, 060h, 000h, 0feh, 066h, 060h, 07ch, 060h, 066h, 0feh, 000h, 000h, 000h, 000h, 000h
377 00013D25 66FE0000000000
                                        <1>
378 00013D2C 0000CC76367ED8D86E- <1>
                                                       000h, 000h, 0cch, 076h, 036h, 07eh, 0d8h, 0d8h, 06eh, 000h, 000h, 000h, 000h, 03eh, 06ch
378 00013D35 0000000003E6C
                                         <1>
379 00013D3C CCCCFECCCCCCE0000- <1>
                                                       Occh, Occh, Ofeh, Occh, Occh, Occh, Occh, Ooch, O00h, O00h, O00h, O10h, O38h, O6ch, O00h, O7ch
                                                  db
379 00013D45 000010386C007C
                                        <1>
380 00013D4C C6C6C6C67C00000000- <1>
                                                       0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h, 000h, 0c6h, 0c6h, 000h, 07ch, 0c6h, 0c6h
380 00013D55 00C6C6007CC6C6
                                         <1>
381 00013D5C C6C67C000000006030- <1>
                                                       0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h, 060h, 030h, 018h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h
                                                  db
381 00013D65 18007CC6C6C6C6
                                        <1>
382 00013D6C 7C000000003078CC00- <1>
                                                  db
                                                       07ch, 000h, 000h, 000h, 000h, 030h, 078h, 0cch, 000h, 0cch, 0cch, 0cch, 0cch, 0cch, 076h, 000h
382 00013D75 CCCCCCCCC7600
                                        <1>
383 00013D7C 00000060301800CCCC- <1>
                                                       000h, 000h, 000h, 060h, 030h, 018h, 000h, 0cch, 0cch, 0cch, 0cch, 076h, 000h, 000h, 000h
383 00013D85 CCCCCC76000000
                                        <1>
384 00013D8C 0000C6C600C6C6C6C6- <1>
                                                  db
                                                       000h, 000h, 0c6h, 0c6h, 000h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07eh, 006h, 00ch, 078h, 000h, 000h, 0c6h
384 00013D95 7E060C780000C6
                                        <1>
385 00013D9C C6386CC6C6C6C6C6C38- <1>
                                                       0c6h, 038h, 06ch, 0c6h, 0c6h, 0c6h, 0c6h, 06ch, 038h, 000h, 000h, 000h, 000h, 0c6h, 0c6h, 000h
385 00013DA5 00000000C6C600
                                         <1>
                                                       0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 018h, 018h, 03ch, 066h, 060h
386 00013DAC C6C6C6C6C6C67C0000- <1>
                                                  db
386 00013DB5 000018183C6660
                                        <1>
387 00013DBC 60663C181800000000- <1>
                                                       060h, 066h, 03ch, 018h, 018h, 000h, 000h, 000h, 000h, 038h, 06ch, 064h, 060h, 0f0h, 060h, 060h
    00013DC5 386C6460F06060
                                        <1>
388 00013DCC 60E6FC00000000066- <1>
                                                       060h, 0e6h, 0fch, 000h, 000h, 000h, 000h, 000h, 066h, 066h, 03ch, 018h, 07eh, 018h, 07eh, 018h
                                                  db
388 00013DD5 663C187E187E18
                                        <1>
                                                                                                                                                       Occh, Oc6h, 000h
389 00013DDC 180000000F8CCCCF8- <1>
                                                  db
                                                       018h, 000h,
                                                                      000h, 000h, 000h, 0f8h, 0cch, 0cch, 0f8h, 0c4h, 0cch, 0deh, 0cch,
389 00013DE5 C4CCDECCCC600
                                        <1>
390 00013DEC 0000000E1B1818187E- <1>
                                                  db
                                                       000h, 000h, 000h, 00eh, 01bh, 018h, 018h, 018h, 07eh, 018h, 018h, 018h, 018h, 048h, 070h, 000h
390 00013DF5 18181818D87000
                                        <1>
391 00013DFC 0018306000780C7CCC- <1>
                                                  db
                                                       000h, 018h, 030h, 060h, 000h, 078h, 00ch, 07ch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h, 00ch
391 00013E05 CC7600000000C
                                        <1>
392 00013E0C 18300038181818183C- <1>
                                                       018h, 030h, 000h, 038h, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 018h, 030h, 060h
392 00013E15 00000000183060
                                         <1>
393 00013E1C 007CC6C6C6C67C0000- <1>
                                                       000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 018h, 030h, 060h, 000h, 0cch
                                                  db
393 00013E25 000018306000CC
                                        <1>
394 00013E2C CCCCCCC7600000000- <1>
                                                                      Occh, Occh, 076h, 000h, 000h, 000h, 000h, 076h, 0dch, 000h,
                                                                                                                                                       Odch, 066h, 066h
394 00013E35 0076DC00DC6666
395 00013E3C 66666600000076DC00- <1>
                                                       066h, 066h, 066h, 000h, 000h, 000h, 076h, 0dch, 000h, 0c6h, 0e6h, 0f6h, 0feh, 0deh, 0ceh, 0c6h
                                                  db
395 00013E45 C6E6F6FEDECEC6
                                        <1>
396 00013E4C C6000000003C6C6C3E- <1>
                                                  db
                                                       0c6h, 000h, 000h, 000h, 000h, 03ch, 06ch, 06ch, 03eh, 000h, 07eh, 000h, 000h, 000h, 000h, 000h
396 00013E55 007E0000000000
                                        <1>
397 00013E5C 000000386C6C38007C-
                                        <1>
                                                  db
                                                       000h, 000h, 000h, 038h, 06ch, 06ch, 038h, 000h, 07ch, 000h, 000h, 000h, 000h, 000h, 000h, 000h
397 00013E65 00000000000000
                                        <1>
398 00013E6C 0000303000303060C6- <1>
                                                       000h, 000h, 030h, 030h, 000h, 030h, 030h, 060h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h, 000h
                                                  db
398 00013E75 C67C0000000000
                                        <1>
399 00013E7C 00000000FEC0C0C000- <1>
                                                                      000h, 000h, 0feh, 0c0h, 0c0h, 0c0h, 000h, 000h, 000h, 000h, 000h,
                                                                                                                                                       000h, 000h, 000h
399 00013E85 00000000000000
                                        <1>
400 00013E8C 0000FE060606000000- <1>
                                                       000h, 000h, 0feh, 006h, 006h, 006h, 000h, 000h, 000h, 000h, 000h, 0c0h, 0c6h, 0c6h, 0c6h, 0d8h
                                                  db
400 00013E95 0000C0C0C6CCD8
                                        <1>
401 00013E9C 3060DC860C183E0000- <1>
                                                       030h, 060h, 0dch, 086h, 00ch, 018h, 03eh, 000h, 000h, 0c0h, 0c6h, 0c6h, 0cch, 0d8h, 030h, 066h
                                                  db
401 00013EA5 C0C0C6CCD83066
402 00013EAC CE9E3E060600000018-
                                                       Oceh, 09eh, 03eh, 006h, 006h, 000h, 000h, 000h, 018h, 018h, 000h, 018h, 018h, 03ch, 03ch, 03ch
                                        <1>
                                                  db
402 00013EB5 180018183C3C3C
                                        <1>
403 00013EBC 18000000000000036- <1>
                                                  db
                                                       018h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 036h, 06ch, 0d8h, 06ch, 036h, 000h, 000h, 000h
403 00013EC5 6CD86C36000000
                                        <1>
404 00013ECC 000000000000D86C36- <1>
                                                       000h, 000h, 000h, 000h, 000h, 000h, 00dh, 0d8h, 06ch, 036h, 06ch, 0d8h, 000h, 000h, 000h, 000h, 000h
404 00013ED5 6CD80000000000
                                        <1>
405 00013EDC 114411441144114411- <1>
                                                       011h, 044h, 055h, 0aah
                                                  db
405 00013EE5 441144114455AA
                                        <1>
406 00013EEC 55AA55AA55AA55AA55- <1>
                                                       055h, 0aah, 055h, 0aah, 055h, 0aah, 055h, 0aah, 055h, 0aah, 055h, 0aah, 0ddh, 077h, 0ddh, 077h
406 00013EF5 AA55AADD77DD77
                                         <1>
407 00013EFC DD77DD77DD77DD77DD- <1>
                                                       0ddh, 077h, 0ddh, 077h, 0ddh, 077h, 0ddh, 077h, 0ddh, 077h, 018h, 018h, 018h, 018h, 018h, 018h
                                                  db
407 00013F05 77181818181818
                                        <1>
408 00013F0C 181818181818181818- <1>
                                                       018h, 018h
                                                  db
408 00013F15 181818181818F8
                                        <1>
409 00013F1C 181818181818181818- <1>
                                                       018h, 018h
                                                  db
409 00013F25 1818F818F81818
                                        <1>
410 00013F2C 181818183636363636- <1>
                                                  db
                                                       018h, 018h, 018h, 018h, 036h, 036h
410 00013F35 3636F636363636
                                        <1>
411 00013F3C 3636000000000000000 <1>
                                                       036h, 036h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 0feh, 036h, 036h, 036h, 036h, 036h, 036h
411 00013F45 FE363636363636
                                         <1>
412 00013F4C 0000000000F818F818- <1>
                                                       000h, 000h, 000h, 000h, 000h, 0f8h, 018h, 0f8h, 018h, 018h, 018h, 018h, 018h, 036h, 036h
                                                  db
412 00013F55 18181818183636
                                        <1>
413 00013F5C 363636F606F6363636- <1>
                                                       036h, 036h, 036h, 056h, 066h, 066h, 056h, 036h, 
413 00013F65 36363636363636
                                         <1>
414 00013F6C 36363636363636363636- <1>
                                                       036h, 000h, 000h, 000h, 000h, 0feh
                                                  db
414 00013F75 360000000000FE
                                        <1>
415 00013F7C 06F63636363636363636 <1>
                                                       006h, 0f6h, 036h, 066h, 066h, 066h, 066h
                                                  db
415 00013F85 36363636F606FE
                                        <1>
416 00013F8C 00000000000363636- <1>
                                                       000h, 000h, 000h, 000h, 000h, 000h, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 06h, 090h, 000h
416 00013F95 36363636FE0000
                                        <1>
417 00013F9C 000000001818181818- <1>
                                                       000h, 000h, 000h, 000h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 000h, 000h, 000h, 000h
                                                  db
417 00013FA5 F818F800000000
                                         <1>
418 00013FAC 00000000000000000000000- <1>
                                                       000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 006h, 018h, 018h, 018h, 018h, 018h, 018h
418 00013FB5 F8181818181818
                                      <1>
419 00013FBC 181818181818181F00- <1>
                                                       018h, 018h, 018h, 018h, 018h, 018h, 018h, 01fh, 000h, 000h, 000h, 000h, 000h, 018h, 018h
419 00013FC5 00000000001818
                                        <1>
420 00013FCC 1818181818FF000000- <1>
                                                       018h, 018h, 018h, 018h, 018h, 0ffh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
420 00013FD5 0000000000000
                                        <1>
421 00013FDC 000000FF1818181818- <1>
                                                       000h, 000h, 000h, 0ffh, 018h, 018h
421 00013FE5 18181818181818
                                        <1>
422 00013FEC 181F18181818181800- <1>
                                                       018h, 01fh, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 000h, 000h, 000h, 000h, 000h, 000h, 00ffh
                                                  db
422 00013FF5 000000000000FF
                                        <1>
423 00013FFC 00000000000181818- <1>
                                                       000h, 000h, 000h, 000h, 000h, 000h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h
423 00014005 18181818FF1818
                                        <1>
424 0001400C 181818181818181818- <1>
                                                       018h, 01fh, 018h, 01fh, 018h, 018h, 018h, 018h
424 00014015 1F181F18181818
                                        <1>
                                                       018h, 018h, 036h, 036h
425 0001401C 18183636363636363636 <1>
425 00014025 37363636363636
                                        <1>
426 0001402C 363636363637303F00- <1>
                                                       036h, 036h, 036h, 036h, 036h, 037h, 030h, 03fh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
426 00014035 00000000000000
                                        <1>
427 0001403C 0000003F3037363636- <1>
                                                       000h, 000h, 000h, 03fh, 030h, 037h, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 036h
427 00014045 36363636363636
                                        <1>
428 0001404C 36F700FF0000000000- <1>
                                                       036h, 0f7h, 000h, 0ffh, 000h, 0ffh
428 00014055 000000000000FF
                                      <1>
429 0001405C 00F73636363636363636 <1>
                                                  db 000h, 0f7h, 036h, 037h, 030h, 037h
429 00014065 36363636373037
                                       <1>
```

374 00013CF5 18003818181818

```
430 0001406C 363636363636000000- <1>
                                                       036h, 036h, 036h, 036h, 036h, 036h, 000h, 000h, 000h, 000h, 000h, 0ffh, 000h, 0ffh, 000h, 000h
                                                  db
430 00014075 0000FF00FF0000
                                        <1>
431 0001407C 000000003636363636- <1>
                                                       000h, 000h, 000h, 000h, 036h, 036h, 036h, 036h, 036h, 057h, 000h, 0f7h, 036h, 036h, 036h, 036h
431 00014085 F700F736363636
                                         <1>
432 0001408C 36361818181818FF00- <1>
                                                       036h, 036h, 018h, 018h, 018h, 018h, 018h, 018h, 06fh, 000h, 06fh, 000h, 000h, 000h, 000h, 000h, 000h
                                                  db
432 00014095 FF000000000000
                                        <1>
433 0001409C 36363636363636FF00- <1>
                                                       036h, 036h, 036h, 036h, 036h, 036h, 036h, 06fh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
433 000140A5 00000000000000
434 000140AC 000000FF00FF181818- <1>
                                                       000h, 000h, 000h, 0ffh, 000h, 0ffh, 018h, 018h, 018h, 018h, 018h, 000h, 000h, 000h, 000h
                                                  db
434 000140B5 18181800000000
                                        <1>
435 000140BC 000000FF3636363636- <1>
                                                       000h, 000h, 000h, 0ffh, 036h, 036h
                                                  db
435 000140C5 36363636363636
                                        <1>
436 000140CC 363F0000000000018- <1>
                                                       036h, 03fh, 000h, 000h, 000h, 000h, 000h, 000h, 018h, 018h, 018h, 018h, 018h, 01fh, 016h
436 000140D5 181818181F181F
                                        <1>
437 000140DC 00000000000000000000000- <1>
                                                       000h, 01fh, 018h, 01fh, 018h, 018h
                                                  db
437 000140E5 00001F181F1818
                                        <1>
                                                       018h, 018h, 018h, 018h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 03fh, 036h, 036h, 036h, 036h
438 000140EC 181818180000000000- <1>
438 000140F5 00003F36363636
                                         <1>
439 000140FC 36363636363636363636- <1>
                                                       036h, 
                                                  db
439 00014105 FF363636363636
                                        <1>
                                                       018h, 018h, 018h, 018h, 018h, 018h, 0ffh, 018h, 0ffh, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h
440 0001410C 1818181818FF18FF18- <1>
                                                  db
440 00014115 18181818181818
                                         <1>
                                                       018h, 018h, 018h, 018h, 018h, 018h, 068h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
441 0001411C 1818181818F8000000- <1>
441 00014125 00000000000000
                                        <1>
442 0001412C 0000001F1818181818- <1>
                                                  db
                                                       000h, 000h, 000h, 01fh, 018h, 018h, 018h, 018h, 018h, 018h, 01fh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh
442 00014135 18FFFFFFFFFFF
                                        <1>
443 0001413C FFFFFFFFFFFFFF00- <1>
                                                       Offh, Offh, Offh, Offh, Offh, Offh, Offh, Offh, O00h, 000h, 000h, 000h, 000h, 000h, 000h, Offh
443 00014145 000000000000FF
                                        <1>
444 0001414C FFFFFFFFFFFFF0F0F0- <1>
                                                       Offh, Offh, Offh, Offh, Offh, Offh, Of0h, Of0h, Of0h, Of0h, Of0h, Of0h, Of0h, Of0h, Of0h, Of0h
                                                  db
444 00014155 F0F0F0F0F0F0F0
                                        <1>
445 0001415C F0F0F0F0F0F0F0F0F0F- <1>
                                                       0f0h, 0f0h, 0f0h, 0f0h, 00fh, 00fh
445 00014165 0F0F0F0F0F0F0F
                                         <1>
446 0001416C 0F0FFFFFFFFFFFFF- <1>
                                                       00fh, 00fh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 00fh, 000h, 000h, 000h, 000h, 000h, 000h
446 00014175 00000000000000
                                        <1>
                                                                      000h, 000h, 000h, 076h, 0dch, 0d8h, 0d8h, 0dch, 076h, 000h, 000h,
                                                                                                                                                       000h, 000h, 000h
447 0001417C 000000000076DCD8D8- <1>
                                                       000h, 000h,
                                                  db
447 00014185 DC760000000000
448 0001418C 00007CC6FCC6C6FCC0- <1>
                                                       000h, 000h, 07ch, 0c6h, 0fch, 0c6h, 0fch, 0c6h, 0c0h, 0c0h, 040h, 000h, 000h, 000h, 0feh, 0c6h
                                                  db
448 00014195 C040000000FEC6
                                        <1>
449 0001419C C6C0C0C0C0C0C000000- <1>
                                                  db
                                                       Oc6h, Oc0h, Oc0h, Oc0h, Oc0h, Oc0h, Oc0h, O00h, O00h, O00h, O00h, O00h, O00h, O00h, Ofeh, O6ch
449 000141A5 0000000000FE6C
                                         <1>
450 000141AC 6C6C6C6C6C000000000 <1>
                                                       06ch, 06ch, 06ch, 06ch, 06ch, 000h, 000h, 000h, 000h, 000h, 0feh, 0c6h, 060h, 030h, 018h, 030h
450 000141B5 00FEC660301830
                                        <1>
451 000141BC 60C6FE0000000000000 <1>
                                                       060h, 0c6h, 0feh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 07eh, 0d8h, 0d8h, 0d8h, 0d8h
                                                  db
451 000141C5 00007ED8D8D8D8
                                        <1>
452 000141CC 70000000000000066- <1>
                                                                      000h, 000h, 000h, 000h, 000h, 000h, 066h, 066h, 066h, 07ch,
452 000141D5 6666667C6060C0
                                         <1>
453 000141DC 0000000000076DC18- <1>
                                                       000h, 000h, 000h, 000h, 000h, 000h, 076h, 0dch, 018h, 018h, 018h, 018h, 018h, 000h, 000h, 000h
                                                  db
453 000141E5 18181818000000
                                        <1>
454 000141EC 00007E183C6666663C- <1>
                                                       000h, 000h, 07eh, 018h, 03ch, 066h, 066h, 066h, 03ch, 018h, 07eh, 000h, 000h, 000h, 000h, 000h
454 000141F5 187E0000000000
                                        <1>
455 000141FC 386CC6C6FEC6C66C38-
                                                       038h, 06ch, 0c6h, 0c6h, 0feh, 0c6h, 0c6h, 06ch, 038h, 000h, 000h, 000h, 000h, 000h, 038h, 06ch
                                        <1>
                                                  db
455 00014205 0000000000386C
                                        <1>
456 0001420C C6C6C6C6C6CEE0000- <1>
                                                  db
                                                       0c6h, 0c6h, 0c6h, 06ch, 06ch, 06ch, 0eeh, 000h, 000h, 000h, 000h, 01eh, 030h, 018h, 00ch
456 00014215 0000001E30180C
                                        <1>
    0001421C 3E6666663C00000000- <1>
                                                  db
                                                       03eh, 066h, 066h, 066h, 03ch, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 07eh, 0dbh, 0dbh
457 00014225 000000007EDBDB
                                        <1>
458 0001422C 7E0000000000000003- <1>
                                                       07eh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 003h, 006h, 07eh, 0dbh, 0dbh, 0f3h, 07eh, 060h
                                                  db
458 00014235 067EDBDBF37E60
                                        <1>
459 0001423C C00000000001C3060- <1>
                                                       0c0h, 000h, 000h, 000h, 000h, 000h, 01ch, 030h, 060h, 060h, 07ch, 060h, 060h, 030h, 01ch, 000h
459 00014245 607C6060301C00
                                         <1>
460 0001424C 0000000007CC6C6C6- <1>
                                                       000h, 000h, 000h, 000h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 000h, 000h, 000h
                                                  db
460 00014255 C6C6C6C6000000
                                        <1>
461 0001425C 000000FE0000FE0000- <1>
                                                       000h, 000h, 000h, 0feh, 000h, 000h, 0feh, 000h, 000h, 0feh, 000h, 000h, 000h, 000h, 000h, 000h
                                                  db
461 00014265 FE0000000000000
                                        <1>
462 0001426C 0018187E18180000FF- <1>
                                                       000h, 018h, 018h, 07eh, 018h, 018h, 000h, 000h, 0ffh, 000h, 000h, 000h, 000h, 000h, 030h, 018h
                                                  db
462 00014275 0000000003018
                                        <1>
463 0001427C 0C060C1830007E0000- <1>
                                                  db
                                                       00ch, 006h, 00ch, 018h, 030h, 000h, 07eh, 000h, 000h, 000h, 000h, 000h, 00ch, 018h, 030h, 060h
463 00014285 0000000C183060
                                        <1>
464 0001428C 30180C007E00000000- <1>
                                                       030h, 018h, 00ch, 000h, 07eh, 000h, 000h, 000h, 000h, 000h, 00eh, 01bh, 018h, 018h, 018h
464 00014295 000E1B1B181818
                                         <1>
465 0001429C 1818181818181818- <1>
                                                       018h, 048h
                                                  db
465 000142A5 1818181818D8D8
                                        <1>
466 000142AC 70000000000001818- <1>
                                                       070h, 000h, 000h, 000h, 000h, 000h, 000h, 018h, 018h, 000h, 07eh, 000h, 018h, 018h, 000h, 000h
466 000142B5 007E0018180000
                                         <1>
                                                       000h, 000h, 000h, 000h, 000h, 000h, 076h, 0dch, 000h, 076h, 0dch, 000h, 000h, 000h, 000h, 000h
467 000142BC 0000000000076DC00- <1>
                                                  db
467 000142C5 76DC0000000000
                                        <1>
                                                  db
468 000142CC 00386C6C3800000000- <1>
                                                       000h, 038h, 06ch, 06ch, 038h, 000h, 000h
468 000142D5 00000000000000
                                        <1>
469 000142DC 000000001818000000- <1>
                                                       000h, 000h, 000h, 000h, 018h, 018h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
469 000142E5 00000000000000
                                        <1>
470 000142EC 000000180000000000 <1>
                                                       000h, 000h, 000h, 018h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 00fh, 00ch, 00ch, 00ch, 00ch
                                                  db
470 000142F5 00000F0C0C0C0C
                                        <1>
471 000142FC 0CEC6C3C1C00000000- <1>
                                                       00ch, 0ech, 06ch, 03ch, 01ch, 000h, 000h, 000h, 000h, 0d8h, 06ch, 06ch, 06ch, 06ch, 06ch, 000h
471 00014305 D86C6C6C6C6C00
                                        <1>
472 0001430C 0000000000000070D8- <1>
                                                       000h, 000h, 000h, 000h, 000h, 000h, 000h, 070h, 0d8h, 030h, 060h, 0c8h, 0f8h, 000h, 000h, 000h
                                                  db
472 00014315 3060C8F8000000
                                        <1>
473 0001431C 00000000000000007C-
                                                                      000h, 000h, 000h, 000h, 000h, 000h, 07ch, 07ch, 07ch, 07ch, 07ch,
                                                                                                                                                       07ch, 000h, 000h
                                                  db
                                                       000h,
                                        <1>
473 00014325 7C7C7C7C7C0000
                                                  db 000h, 000h
474 0001432C 0000000000000000000 <1>
474 00014335 000000000000000
                                        <1>
475
                                        <1> vgafont16:
476 0001433C 0000000000000000000 <1>
                                                       000h, 000h
476 00014345 00000000000000
                                        <1>
477 0001434C 00007E81A58181BD99- <1>
                                                       000h, 000h, 07eh, 081h, 0a5h, 081h, 081h, 0bdh, 099h, 081h, 081h, 07eh, 000h, 000h, 000h, 000h
477 00014355 81817E00000000
                                        <1>
478 0001435C 00007EFFDBFFFFC3E7- <1>
                                                       000h, 000h, 07eh, 0ffh, 0dbh, 0ffh, 0ffh, 0c3h, 0e7h, 0ffh, 0ffh, 07eh, 000h, 000h, 000h, 000h
                                                  db
478 00014365 FFFF7E00000000
                                        <1>
479 0001436C 000000006CFEFEFEF- <1>
                                                       000h, 000h, 000h, 000h, 06ch, 0feh, 0feh, 0feh, 0feh, 07ch, 038h, 010h, 000h, 000h, 000h, 000h
                                                  db
479 00014375 7C381000000000
                                        <1>
480 0001437C 0000000010387CFE7C- <1>
                                                  db
                                                       000h, 000h, 000h, 000h, 010h, 038h, 07ch, 0feh, 07ch, 038h, 010h, 000h, 000h, 000h, 000h, 000h
480 00014385 3810000000000
                                        <1>
481 0001438C 000000183C3CE7E7E7- <1>
                                                       000h, 000h, 000h, 018h, 03ch, 03ch, 0e7h, 0e7h, 0e7h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h
481 00014395 18183C00000000
                                        <1>
482 0001439C 000000183C7EFFFF7E- <1>
                                                       000h, 000h, 000h, 018h, 03ch, 07eh, 0ffh, 0feh, 07eh, 018h, 018h, 03ch, 000h, 000h, 000h, 000h
                                                  db
482 000143A5 18183C00000000
                                        <1>
483 000143AC 00000000000183C3C- <1>
                                                       000h, 000h, 000h, 000h, 000h, 000h, 018h, 03ch, 03ch, 018h, 000h, 000h, 000h, 000h, 000h, 000h
                                                  db
483 000143B5 18000000000000
                                        <1>
484 000143BC FFFFFFFFFFFFFC7C3C3- <1>
                                                       Offh, Offh, Offh, Offh, Offh, Offh, Oe7h, Oc3h, Oc3h, Oe7h, Offh, Offh, Offh, Offh, Offh
                                                  db
484 000143C5 E7FFFFFFFFFF
                                      <1>
485 000143CC 0000000003C664242- <1>
                                                  db 000h, 000h, 000h, 000h, 000h, 03ch, 066h, 042h, 042h, 066h, 03ch, 000h, 000h, 000h, 000h, 000h
485 000143D5 663C0000000000
                                        <1>
```

Offh, Offh, Offh, Offh, Offh, Oc3h, O99h, Obdh, O99h, Oc3h, Offh, Offh, Offh, Offh, Offh

```
486 000143E5 99C3FFFFFFFFF
                                 <1>
                                             000h, 000h, 01eh, 00eh, 01ah, 032h, 078h, 0cch, 0cch, 0cch, 078h, 000h, 000h, 000h, 000h
487 000143EC 00001E0E1A3278CCCC- <1>
487 000143F5 CCCC7800000000
                                 <1>
488 000143FC 00003C666666663C18- <1>
                                             000h, 000h, 03ch, 066h, 066h, 066h, 066h, 03ch, 018h, 07eh, 018h, 018h, 000h, 000h, 000h, 000h
                                         db
488 00014405 7E181800000000
                                 <1>
489 0001440C 00003F333F30303030- <1>
                                             000h, 000h, 03fh, 033h, 03fh, 030h, 030h, 030h, 030h, 070h, 0f0h, 0e0h, 000h, 000h, 000h, 000h
489 00014415 70F0E000000000
490 0001441C 00007F637F63636363- <1>
                                         db
                                             000h, 000h, 07fh, 063h, 07fh, 063h, 063h, 063h, 063h, 067h, 0e7h, 0e6h, 0c0h, 000h, 000h, 000h
490 00014425 67E7E6C0000000
                                 <1>
491 0001442C 0000001818DB3CE73C- <1>
                                             000h, 000h, 000h, 018h, 018h, 0dbh, 03ch, 0e7h, 03ch, 0dbh, 018h, 018h, 000h, 000h, 000h, 000h
                                         db
491 00014435 DB181800000000
                                 <1>
492 0001443C 0080C0E0F0F8FEF8F0-
                                 <1>
                                             000h, 080h, 0c0h, 0e0h, 0f0h, 0f8h, 0feh, 0f8h, 0f0h, 0e0h, 0c0h, 080h, 000h, 000h, 000h, 000h
492 00014445 E0C08000000000
                                 <1>
493 0001444C 0002060E1E3EFE3E1E- <1>
                                             000h, 002h, 006h, 00eh, 01eh, 03eh, 0feh, 03eh, 01eh, 00eh, 006h, 002h, 000h,
                                                                                                                            000h, 000h, 000h
                                         db
493 00014455 0E060200000000
                                 <1>
494 0001445C 0000183C7E1818187E- <1>
                                             000h, 000h, 018h, 03ch, 07eh, 018h, 018h, 018h, 07eh, 03ch, 018h, 000h, 000h, 000h, 000h, 000h
494 00014465 3C180000000000
                                 <1>
495 0001446C 0000666666666666666 <1>
                                             000h, 000h, 066h, 066h, 066h, 066h, 066h, 066h, 066h, 000h, 066h, 066h, 000h, 000h, 000h, 000h
                                         db
495 00014475 00666600000000
                                 <1>
496 0001447C 00007FDBDBDB7B1B1B- <1>
                                             000h, 000h, 07fh, 0dbh, 0dbh, 0dbh, 07bh, 01bh, 01bh, 01bh, 01bh, 01bh, 000h, 000h, 000h, 000h
                                         db
496 00014485 1B1B1B00000000
                                 <1>
497 0001448C 007CC660386CC6C6C6C- <1>
                                         db
                                             000h, 07ch, 0c6h, 060h, 038h, 06ch, 0c6h, 0c6h, 06ch, 038h, 00ch, 0c6h, 07ch, 000h, 000h, 000h
497 00014495 380CC67C000000
                                 <1>
498 0001449C 000000000000000FE- <1>
                                         db
                                             000h, 000h,
                                                         000h, 000h, 000h, 000h, 000h, 000h, 0feh, 0feh, 0feh, 0feh, 000h,
                                                                                                                            000h, 000h, 000h
498 000144A5 FEFEFE00000000
                                 <1>
499 000144AC 0000183C7E1818187E- <1>
                                             000h, 000h, 018h, 03ch, 07eh, 018h, 018h, 018h, 07eh, 03ch, 018h, 07eh, 000h, 000h, 000h, 000h
499 000144B5 3C187E00000000
                                 <1>
                                             000h, 000h, 018h, 03ch, 07eh, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 000h, 000h, 000h, 000h
500 000144BC 0000183C7E18181818- <1>
                                         db
500 000144C5 18181800000000
                                 <1>
501 000144CC 000018181818181818- <1>
                                             000h, 000h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 07eh, 03ch, 018h, 000h, 000h, 000h, 000h
                                         db
501 000144D5 7E3C1800000000
                                 <1>
502 000144DC 000000000180CFE0C- <1>
                                             000h, 000h, 000h, 000h, 000h, 018h, 00ch, 0feh, 00ch, 018h, 000h, 000h, 000h, 000h, 000h, 000h
                                         db
502 000144E5 1800000000000
                                 <1>
503 000144EC 0000000003060FE60- <1>
                                                                                                                            000h, 000h, 000h
                                             000h, 000h,
                                                         000h, 000h, 000h, 030h, 060h, 0feh, 060h, 030h, 000h, 000h, 000h,
                                         db
503 000144F5 30000000000000
504 000144FC 000000000000C0C0C0-
                                 <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 000h, 0c0h, 0c0h, 0c0h, 0feh, 000h, 000h, 000h, 000h, 000h
                                         db
504 00014505 FE000000000000
                                 <1>
505 0001450C 00000000002466FF66- <1>
                                         db
                                             000h, 000h, 000h, 000h, 000h, 024h, 066h, 0ffh, 066h, 024h, 000h, 000h, 000h, 000h, 000h, 000h
505 00014515 24000000000000
                                 <1>
506 0001451C 000000001038387C7C-
                                             000h, 000h, 000h, 000h, 010h, 038h, 038h, 07ch, 07ch, 0feh, 0feh, 000h, 000h, 000h, 000h, 000h
506 00014525 FEFE0000000000
                                 <1>
507 0001452C 00000000FEFE7C7C38- <1>
                                             000h, 000h, 000h, 000h, 0feh, 0feh, 07ch, 07ch, 038h, 038h, 010h, 000h, 000h, 000h, 000h, 000h
                                         db
507 00014535 3810000000000
                                 <1>
508 0001453C 000000000000000000 <1>
                                                         000h,                                                                                                                             000h, 000h, 000h
508 00014545 00000000000000
                                 <1>
509 0001454C 0000183C3C3C181818- <1>
                                             000h, 000h, 018h, 03ch, 03ch, 03ch, 018h, 018h, 018h, 000h, 018h, 018h, 000h, 000h, 000h, 000h
                                         db
509 00014555 00181800000000
                                 <1>
510 0001455C 006666662400000000- <1>
                                             000h, 066h, 066h, 066h, 024h, 000h, 000h
510 00014565 00000000000000
                                 <1>
511 0001456C 0000006C6CFE6C6C6C-
                                             000h, 000h, 000h, 06ch, 06ch, 0feh, 06ch, 06ch, 06ch, 06ch, 06ch, 000h, 000h, 000h, 000h
                                 <1>
                                         db
511 00014575 FE6C6C00000000
                                 <1>
512 0001457C 18187CC6C2C07C0606- <1>
                                         db
                                             018h, 018h, 07ch, 0c6h, 0c2h, 0c0h, 07ch, 006h, 006h, 086h, 0c6h, 07ch, 018h, 018h, 000h, 000h
512 00014585 86C67C18180000
                                 <1>
513 0001458C 00000000C2C60C1830-
                                 <1>
                                         db
                                             000h, 000h, 000h, 000h, 0c2h, 0c6h, 00ch, 018h, 030h, 060h, 0c6h, 086h, 000h,
513 00014595 60C68600000000
                                 <1>
514 0001459C 0000386C6C3876DCCC- <1>
                                             000h, 000h, 038h, 06ch, 06ch, 038h, 076h, 0dch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h
                                         db
514 000145A5 CCCC760000000
                                 <1>
515 000145AC 003030306000000000 <1>
                                             000h, 030h, 030h, 030h, 060h, 000h, 000h
515 000145B5 00000000000000
                                 <1>
516 000145BC 00000C183030303030- <1>
                                             000h, 000h, 00ch, 018h, 030h, 030h, 030h, 030h, 030h, 030h, 018h, 00ch, 000h, 000h, 000h, 000h
                                         db
516 000145C5 30180C00000000
                                 <1>
517 000145CC 000030180C0C0C0C0C- <1>
                                             000h, 000h, 030h, 018h, 00ch, 00ch, 00ch, 00ch, 00ch, 00ch, 018h, 030h, 000h, 000h, 000h, 000h
                                         db
517 000145D5 0C183000000000
                                 <1>
518 000145DC 000000000663CFF3C- <1>
                                             000h, 000h, 000h, 000h, 000h, 066h, 03ch, 0ffh, 03ch, 066h, 000h, 000h, 000h, 000h, 000h, 000h
                                         db
518 000145E5 66000000000000
                                 <1>
519 000145EC 00000000018187E18- <1>
                                         db
                                             000h, 000h, 000h, 000h, 000h, 018h, 018h, 07eh, 018h, 018h, 000h, 000h, 000h, 000h, 000h, 000h
519 000145F5 1800000000000
                                 <1>
520 000145FC 0000000000000000000 <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 018h, 018h, 018h, 030h, 000h, 000h, 000h
520 00014605 18181830000000
                                 <1>
521 0001460C 0000000000000FE00- <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 000h, 0feh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                         db
521 00014615 00000000000000
                                 <1>
522 0001461C 000000000000000000 <1>
                                             000h, 018h, 018h, 000h, 000h, 000h, 000h
522 00014625 00181800000000
                                             000h, 000h, 000h, 000h, 002h, 006h, 00ch, 018h, 030h, 060h, 0c0h, 080h, 000h, 000h, 000h, 000h
523 0001462C 0000000002060C1830- <1>
                                         db
523 00014635 60C08000000000
                                 <1>
                                         db
524 0001463C 00003C66C3C3DBDBC3- <1>
                                             000h, 000h, 03ch, 066h, 0c3h, 0c3h, 0dbh, 0dbh, 0c3h, 0c3h, 066h, 03ch, 000h, 000h, 000h, 000h
524 00014645 C3663C00000000
                                 <1>
525 0001464C 000018387818181818-
                                             000h, 000h, 018h, 038h, 078h, 018h, 018h, 018h, 018h, 018h, 018h, 07eh, 000h, 000h, 000h, 000h
                                 <1>
525 00014655 18187E00000000
                                 <1>
526 0001465C 00007CC6060C183060- <1>
                                         db
                                             000h, 000h, 07ch, 0c6h, 006h, 00ch, 018h, 030h, 060h, 0c0h, 0c6h, 0feh, 000h, 000h, 000h, 000h
526 00014665 C0C6FE00000000
                                 <1>
527 0001466C 00007CC606063C0606- <1>
                                             000h, 000h, 07ch, 0c6h, 006h, 006h, 03ch, 006h, 006h, 006h, 0c6h, 07ch, 000h, 000h, 000h, 000h
527 00014675 06C67C00000000
                                 <1>
528 0001467C 00000C1C3C6CCCFE0C- <1>
                                             000h, 000h, 00ch, 01ch, 03ch, 06ch, 0cch, 0feh, 00ch, 00ch, 00ch, 01eh, 000h, 000h, 000h, 000h
                                         db
528 00014685 0C0C1E00000000
                                 <1>
                                                                                                                            000h, 000h, 000h
529 0001468C 0000FEC0C0C0FC0606-
                                 <1>
                                         db
                                             000h,
                                                         Ofeh, OcOh, OcOh, OcOh, Ofch, OO6h, OO6h, OC6h, O7ch, OO0h,
529 00014695 06C67C00000000
530 0001469C 00003860C0C0FCC6C6- <1>
                                             000h, 000h, 038h, 060h, 0c0h, 0c0h, 0fch, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h
530 000146A5 C6C67C00000000
                                 <1>
                                             000h, 000h, 0feh, 0c6h, 006h, 006h, 00ch, 018h, 030h, 030h, 030h, 030h, 000h, 000h, 000h, 000h
531 000146AC 0000FEC606060C1830- <1>
                                         db
531 000146B5 30303000000000
                                 <1>
532 000146BC 00007CC6C6C67CC6C6- <1>
                                             000h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h
532 000146C5 C6C67C00000000
                                 <1>
533 000146CC 00007CC6C6C6C67E0606- <1>
                                         db
                                             000h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 07eh, 006h, 006h, 006h, 00ch, 078h, 000h, 000h, 000h, 000h
533 000146D5 060C7800000000
                                 <1>
534 000146DC 000000001818000000- <1>
                                         db
                                             000h, 000h, 000h, 000h, 018h, 018h, 000h, 000h, 000h, 018h, 018h, 000h, 000h, 000h, 000h
534 000146E5 18180000000000
                                 <1>
                                             000h, 000h, 000h, 000h, 018h, 018h, 000h, 000h, 000h, 018h, 018h, 030h, 000h, 000h, 000h
535 000146EC 000000001818000000- <1>
                                         db
535 000146F5 18183000000000
                                 <1>
536 000146FC 000000060C18306030- <1>
                                             000h, 000h, 000h, 006h, 00ch, 018h, 030h, 060h, 030h, 018h, 00ch, 006h, 000h, 000h, 000h, 000h
                                         db
536 00014705 180C0600000000
                                 <1>
537 0001470C 00000000007E00007E- <1>
                                             000h, 000h, 000h, 000h, 000h, 07eh, 000h, 07eh, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                         db
537 00014715 00000000000000
                                 <1>
                                             000h, 000h, 000h, 060h, 030h, 018h, 00ch, 006h, 00ch, 018h, 030h, 060h, 000h, 000h, 000h
538 0001471C 0000006030180C060C- <1>
                                         db
538 00014725 18306000000000
                                 <1>
539 0001472C 00007CC6C60C181818- <1>
                                             000h, 000h, 07ch, 0c6h, 0c6h, 00ch, 018h, 018h, 018h, 000h, 018h, 018h, 000h, 000h, 000h, 000h
539 00014735 00181800000000
                                 <1>
540 0001473C 0000007CC6C6DEDEDE- <1>
                                             000h, 000h, 000h, 07ch, 0c6h, 0c6h, 0deh, 0deh, 0deh, 0dch, 0c0h, 07ch, 000h, 000h, 000h, 000h
                                         db
540 00014745 DCC07C00000000
                                 <1>
541 0001474C 000010386CC6C6FEC6- <1>
                                             000h, 000h, 010h, 038h, 06ch, 0c6h, 0c6h, 0feh, 0c6h, 0c6h, 0c6h, 0c6h, 000h, 000h, 000h, 000h
```

486 000143DC FFFFFFFFFC399BDBD- <1>

```
542 0001475C 0000FC6666667C6666- <1>
                                             000h, 000h, 0fch, 066h, 066h, 066h, 07ch, 066h, 066h, 066h, 066h, 0fch, 000h, 000h, 000h, 000h
542 00014765 6666FC00000000
543 0001476C 00003C66C2C0C0C0C0- <1>
                                             000h, 000h, 03ch, 066h, 0c2h, 0c0h, 0c0h, 0c0h, 0c0h, 0c2h, 066h, 03ch, 000h, 000h, 000h, 000h
                                         db
543 00014775 C2663C00000000
                                 <1>
544 0001477C 0000F86C6666666666 <1>
                                         db
                                             000h, 000h, 0f8h, 06ch, 066h, 066h, 066h, 066h, 066h, 066h, 06ch, 0f8h, 000h, 000h, 000h, 000h
544 00014785 666CF800000000
                                 <1>
545 0001478C 0000FE666268786860- <1>
                                             000h, 000h, 0feh, 066h, 062h, 068h, 078h, 068h, 060h, 062h, 066h, 0feh, 000h, 000h, 000h, 000h
545 00014795 6266FE00000000
                                 <1>
546 0001479C 0000FE666268786860- <1>
                                             000h, 000h, 0feh, 066h, 062h, 068h, 078h, 068h, 060h, 060h, 060h, 0f0h, 000h,
                                                                                                                           000h, 000h, 000h
                                         db
546 000147A5 6060F000000000
                                 <1>
547 000147AC 00003C66C2C0C0DEC6- <1>
                                             000h, 000h, 03ch, 066h, 0c2h, 0c0h, 0c0h, 0deh, 0c6h, 066h, 03ah, 000h, 000h, 000h, 000h
547 000147B5 C6663A00000000
                                 <1>
548 000147BC 0000C6C6C6C6FEC6C6- <1>
                                         db
                                             000h, 000h, 0c6h, 0c6h, 0c6h, 0c6h, 0feh, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 000h, 000h, 000h, 000h
548 000147C5 C6C6C600000000
                                 <1>
549 000147CC 00003C181818181818- <1>
                                         db
                                             000h, 000h, 03ch, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h
549 000147D5 18183C00000000
                                 <1>
550 000147DC 00001E0C0C0C0C0CCC-
                                <1>
                                             550 000147E5 CCCC7800000000
                                 <1>
551 000147EC 0000E666666C78786C- <1>
                                         db
                                             000h, 000h,
                                                         0e6h, 066h, 066h, 06ch, 078h, 078h, 06ch, 066h, 066h, 0e6h, 000h,
                                                                                                                           000h, 000h, 000h
551 000147F5 6666E600000000
                                 <1>
552 000147FC 0000F060606060606060- <1>
                                             000h, 000h, 0f0h, 060h, 060h, 060h, 060h, 060h, 060h, 062h, 066h, 0feh, 000h, 000h, 000h, 000h
552 00014805 6266FE00000000
                                 <1>
553 0001480C 0000C3E7FFFDBC3C3- <1>
                                             000h, 000h, 0c3h, 0e7h, 0ffh, 0ffh, 0dbh, 0c3h, 0c3h, 0c3h, 0c3h, 0c3h, 000h, 000h, 000h, 000h
                                         db
553 00014815 C3C3C300000000
                                 <1>
554 0001481C 0000C6E6F6FEDECEC6- <1>
                                             000h, 000h, 0c6h, 0e6h, 0f6h, 0feh, 0deh, 0ceh, 0c6h, 0c6h, 0c6h, 0c6h, 000h, 000h, 000h, 000h
                                         db
554 00014825 C6C6C600000000
555 0001482C 00007CC6C6C6C6C6C6C- <1>
                                             000h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h
                                         db
                                 <1>
555 00014835 C6C67C00000000
556 0001483C 0000FC6666667C6060- <1>
                                         db
                                             000h,
                                                   000h,
                                                         Ofch, 066h, 066h, 066h, 07ch, 060h, 060h, 060h, 060h, 0f0h, 000h,
                                                                                                                           000h, 000h, 000h
556 00014845 6060F000000000
                                 <1>
   0001484C 00007CC6C6C6C6C6C6C6 <1>
                                         db
                                             000h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0d6h, 0deh, 07ch, 00ch, 00eh, 000h, 000h
557 00014855 D6DE7C0C0E0000
                                 <1>
558 0001485C 0000FC6666667C6C66- <1>
                                         db
                                             000h, 000h, 0fch, 066h, 066h, 066h, 07ch, 06ch, 066h, 066h, 066h, 0e6h, 000h, 000h, 000h, 000h
558 00014865 6666E600000000
                                 <1>
559 0001486C 00007CC6C660380C06- <1>
                                             000h, 000h, 07ch, 0c6h, 0c6h, 060h, 038h, 00ch, 006h, 0c6h, 07ch, 000h, 000h, 000h, 000h
559 00014875 C6C67C00000000
                                 <1>
560 0001487C 0000FFDB9918181818- <1>
                                             000h, 000h, 0ffh, 0dbh, 099h, 018h, 018h, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h
                                         db
560 00014885 18183C00000000
                                 <1>
561 0001488C 0000C6C6C6C6C6C6C6C- <1>
                                                                                                                           000h, 000h, 000h
                                             000h,
                                                   000h,
                                                         0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h,
                                         db
561 00014895 C6C67C00000000
562 0001489C 0000C3C3C3C3C3C3C3C3- <1>
                                             000h, 000h, 0c3h, 0c3h, 0c3h, 0c3h, 0c3h, 0c3h, 0c3h, 066h, 03ch, 018h, 000h, 000h, 000h, 000h
                                         db
562 000148A5 663C1800000000
                                 <1>
563 000148AC 0000C3C3C3C3C3DBDB- <1>
                                         db
                                             000h, 000h, 0c3h, 0c3h, 0c3h, 0c3h, 0c3h, 0dbh, 0dbh, 0ffh, 066h, 066h, 000h, 000h, 000h, 000h
563 000148B5 FF666600000000
                                 <1>
564 000148BC 0000C3C3663C18183C-
                                 <1>
                                         db
                                             000h, 000h, 0c3h, 0c3h, 066h, 03ch, 018h, 018h, 03ch, 066h, 0c3h, 0c3h, 000h, 000h, 000h, 000h
564 000148C5 66C3C300000000
                                 <1>
565 000148CC 0000C3C3C3663C1818- <1>
                                         db
                                             000h, 000h, 0c3h, 0c3h, 0c3h, 066h, 03ch, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h
565 000148D5 18183C00000000
                                 <1>
566 000148DC 0000FFC3860C183060- <1>
                                                   000h, 0ffh, 0c3h, 086h, 00ch, 018h, 030h, 060h, 0c1h, 0c3h, 0ffh, 000h,
                                                                                                                           000h, 000h, 000h
566 000148E5 C1C3FF00000000
                                 <1>
567 000148EC 00003C303030303030- <1>
                                             000h, 000h, 03ch, 030h, 030h, 030h, 030h, 030h, 030h, 030h, 030h, 03ch, 000h, 000h, 000h, 000h
                                         db
567 000148F5 30303C00000000
                                 <1>
568 000148FC 00000080C0E070381C- <1>
                                             000h, 000h, 000h, 080h, 0c0h, 0e0h, 070h, 038h, 01ch, 00eh, 006h, 002h, 000h,
                                                                                                                           000h, 000h, 000h
                                         db
   00014905 0E060200000000
569 0001490C 00003C0C0C0C0C0CC-
                                             000h, 000h, 03ch, 00ch, 00ch, 00ch, 00ch, 00ch, 00ch, 00ch, 00ch, 00ch, 03ch, 000h, 000h, 000h, 000h
                                 <1>
                                         db
569 00014915 0C0C3C00000000
                                 <1>
570 0001491C 10386CC60000000000 <1>
                                         db
                                             010h, 038h, 06ch, 0c6h, 000h, 000h
570 00014925 00000000000000
                                 <1>
571 0001492C 0000000000000000000 <1>
                                             000h, 571 00014935 00000000FF0000
                                 <1>
572 0001493C 303018000000000000 <1>
                                             030h, 030h, 018h, 000h, 000h
                                         db
572 00014945 00000000000000
                                 <1>
573 0001494C 000000000780C7CCC- <1>
                                             000h, 000h, 000h, 000h, 000h, 076h, 00ch, 07ch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h
573 00014955 CCCC7600000000
                                 <1>
574 0001495C 0000E06060786C6666- <1>
                                             000h, 000h, 0e0h, 060h, 060h, 078h, 06ch, 066h, 066h, 066h, 07ch, 000h, 000h, 000h, 000h
                                         db
574 00014965 66667C00000000
                                 <1>
575 0001496C 0000000007CC6C0C0- <1>
                                             000h, 000h, 000h, 000h, 000h, 07ch, 0c6h, 0c0h, 0c0h, 0c0h, 0c6h, 07ch, 000h, 000h, 000h, 000h
                                         db
575 00014975 C0C67C00000000
                                 <1>
576 0001497C 00001C0C0C3C6CC0
                             CCC- <1>
                                         db
                                             000h, 000h, 01ch, 00ch, 00ch, 03ch, 06ch, 0cch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h
576 00014985 CCCC7600000000
                                 <1>
577 0001498C 0000000007CC6FEC0- <1>
                                         db
                                             000h, 000h, 000h, 000h, 000h, 07ch, 0c6h, 0feh, 0c0h, 0c0h, 0c6h, 07ch, 000h, 000h, 000h, 000h
577 00014995 C0C67C00000000
                                 <1>
578 0001499C 0000386C6460F06060- <1>
                                             000h, 000h, 038h, 06ch, 064h, 060h, 0f0h, 060h, 060h, 060h, 060h, 0f0h, 000h, 000h, 000h, 000h
578 000149A5 6060F000000000
                                 <1>
579 000149AC 000000000076CCCCCC- <1>
                                             000h, 000h, 000h, 000h, 000h, 076h, 0cch, 0cch, 0cch, 0cch, 0cch, 07ch, 00ch,
                                                                                                                           Occh, 078h, 000h
                                         db
579 000149B5 CCCC7C0CCC7800
                                 <1>
580 000149BC 0000E060606C766666- <1>
                                             000h, 000h, 0e0h, 060h, 060h, 06ch, 076h, 066h, 066h, 066h, 0e6h, 0e6h, 000h, 000h, 000h, 000h
580 000149C5 6666E600000000
                                 <1>
581 000149CC 000018180038181818- <1>
                                             000h, 000h, 018h, 018h, 000h, 038h, 018h, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h
                                         db
581 000149D5 18183C00000000
                                 <1>
582 000149DC 00000606000E060606- <1>
                                             000h, 000h, 006h, 006h, 000h, 00eh, 006h, 006h, 006h, 006h, 006h, 006h, 066h, 066h, 03ch, 000h
                                         db
582 000149E5 06060666663C00
                                 <1>
583 000149EC 0000E06060666C7878- <1>
                                             000h, 000h, 0e0h, 060h, 060h, 066h, 06ch, 078h, 078h, 06ch, 066h, 0e6h, 000h, 000h, 000h, 000h
583 000149F5 6C66E600000000
                                 <1>
584 000149FC 000038181818181818- <1>
                                         db
                                             000h, 000h,
                                                         038h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h
584 00014A05 18183C00000000
                                 <1>
585 00014A0C 0000000000E6FFDBDB- <1>
                                             000h, 000h, 000h, 000h, 000h, 0e6h, 0ffh, 0dbh, 0dbh, 0dbh, 0dbh, 000h, 000h, 000h, 000h
585 00014A15 DBDBDB00000000
                               <1>
586 00014A1C 000000000DC666666- <1>
                                             000h, 000h, 000h, 000h, 000h, 0dch, 066h, 066h, 066h, 066h, 066h, 000h, 000h, 000h, 000h
                                         db
586 00014A25 66666600000000
                                 <1>
587 00014A2C 0000000007CC6C6C6- <1>
                                             000h, 000h, 000h, 000h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h
587 00014A35 C6C67C00000000
                                 <1>
588 00014A3C 000000000DC666666- <1>
                                             000h, 000h, 000h, 000h, 000h, 00ch, 06ch, 066h, 066h, 066h, 066h, 07ch, 060h, 060h, 0f0h, 000h
588 00014A45 66667C6060F000
                                 <1>
589 00014A4C 000000000076CCCCCC- <1>
                                             000h, 000h, 000h, 000h, 000h, 076h, 0cch, 0cch, 0cch, 0cch, 0cch, 07ch, 00ch, 00ch, 01eh, 000h
                                         db
589 00014A55 CCCC7C0C0C1E00
                                 <1>
590 00014A5C 000000000DC766660- <1>
                                             000h, 000h, 000h, 000h, 000h, 0dch, 076h, 066h, 060h, 060h, 060h, 0f0h, 000h, 000h, 000h, 000h
                                         db
590 00014A65 6060F000000000
                                 <1>
591 00014A6C 0000000007CC66038- <1>
                                         db
                                             000h, 000h, 000h, 000h, 000h, 07ch, 0c6h, 060h, 038h, 00ch, 0c6h, 07ch, 000h, 000h, 000h, 000h
591 00014A75 0CC67C00000000
                                 <1>
592 00014A7C 0000103030FC303030- <1>
                                             000h, 000h, 010h, 030h, 030h, 05ch, 030h, 030h, 030h, 030h, 036h, 01ch, 000h, 000h, 000h, 000h
592 00014A85 30361C00000000
                                 <1>
593 00014A8C 0000000000CCCCCCCC- <1>
                                             000h, 000h, 000h, 000h, 000h, 00ch, 0cch, 0cch, 0cch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h
                                         db
593 00014A95 CCCC760000000
                                 <1>
594 00014A9C 000000000C3C3C3C3- <1>
                                             000h, 000h, 000h, 000h, 000h, 0c3h, 0c3h, 0c3h, 0c3h, 066h, 03ch, 018h, 000h, 000h, 000h, 000h
                                         db
594 00014AA5 663C1800000000
                                 <1>
595 00014AAC 000000000C3C3C3DB- <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 0c3h, 0c3h, 0c3h, 0dbh, 0dbh, 0ffh, 066h, 000h, 000h, 000h, 000h
                                         db
595 00014AB5 DBFF6600000000
                               <1>
                                         db 000h, 000h, 000h, 000h, 000h, 000h, 0c3h, 066h, 03ch, 018h, 03ch, 066h, 0c3h, 000h, 000h, 000h, 000h
596 00014ABC 0000000000C3663C18- <1>
596 00014AC5 3C66C300000000
                                <1>
```

541 00014755 C6C6C600000000

000h, 000h, 000h, 000h, 000h, 006h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07eh, 006h, 00ch, 0f8h, 000h

```
597 00014AD5 C6C67E060CF800
                                 <1>
                                             000h, 000h, 000h, 000h, 000h, 0feh, 0cch, 018h, 030h, 060h, 0c6h, 0feh, 000h, 000h, 000h, 000h
598 00014ADC 0000000000FECC1830- <1>
598 00014AE5 60C6FE00000000
                                 <1>
599 00014AEC 00000E181818701818- <1>
                                             000h, 000h, 00eh, 018h, 018h, 018h, 070h, 018h, 018h, 018h, 018h, 00eh, 000h, 000h, 000h, 000h
                                         db
599 00014AF5 18180E00000000
                                 <1>
600 00014AFC 000018181818001818- <1>
                                             000h, 000h, 018h, 018h, 018h, 018h, 000h, 018h, 018h, 018h, 018h, 018h, 000h, 000h, 000h, 000h
600 00014B05 18181800000000
601 00014B0C 0000701818180E1818- <1>
                                         db
                                             000h, 000h, 070h, 018h, 018h, 018h, 00eh, 018h, 018h, 018h, 018h, 070h, 000h, 000h, 000h, 000h
601 00014B15 1818700000000
                                 <1>
602 00014B1C 000076DC0000000000- <1>
                                             000h, 000h, 076h, 0dch, 000h, 000h
                                         db
602 00014B25 00000000000000
                                 <1>
603 00014B2C 0000000010386CC6C6- <1>
                                             000h, 000h, 000h, 000h, 010h, 038h, 06ch, 0c6h, 0c6h, 0feh, 000h, 000h, 000h, 000h, 000h
603 00014B35 C6FE0000000000
                                 <1>
604 00014B3C 00003C66C2C0C0C0C2- <1>
                                             000h, 000h, 03ch, 066h, 0c2h, 0c0h, 0c0h, 0c0h, 0c2h, 066h, 03ch, 00ch, 006h, 07ch, 000h, 000h
                                         db
604 00014B45 663C0C067C0000
                                 <1>
605 00014B4C 0000CC0000CCCCCCCC <1>
                                             000h, 000h, 0cch, 000h, 000h, 0cch, 0cch, 0cch, 0cch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h
605 00014B55 CCCC7600000000
                                 <1>
                                             000h, 00ch, 018h, 030h, 000h, 07ch, 0c6h, 0feh, 0c0h, 0c0h, 0c6h, 07ch, 000h, 000h, 000h, 000h
606 00014B5C 000C1830007CC6FEC0- <1>
                                         db
606 00014B65 C0C67C00000000
                                 <1>
607 00014B6C 0010386C00780C7CCC- <1>
                                             000h, 010h, 038h, 06ch, 000h, 078h, 00ch, 07ch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h
                                         db
   00014B75 CCCC7600000000
                                 <1>
608 00014B7C 0000CC0000780C7CCC- <1>
                                         db
                                             000h, 000h, 0cch, 000h, 000h, 078h, 00ch, 07ch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h
608 00014B85 CCCC760000000
                                 <1>
609 00014B8C 0060301800780C70
                             CCC- <1>
                                         db
                                             000h, 060h,
                                                         030h, 018h, 000h, 078h, 00ch, 07ch, 0cch, 0cch, 076h, 000h,
                                                                                                                            000h, 000h, 000h
609 00014B95 CCCC7600000000
                                 <1>
610 00014B9C 00386C3800780C7CCC- <1>
                                             000h, 038h, 06ch, 038h, 000h, 078h, 00ch, 07ch, 0cch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h
610 00014BA5 CCCC7600000000
                                 <1>
611 00014BAC 000000003C66606066- <1>
                                         db
                                             000h, 000h, 000h, 000h, 03ch, 066h, 060h, 060h, 066h, 03ch, 00ch, 006h, 03ch, 000h, 000h, 000h
611 00014BB5 3C0C063C000000
                                 <1>
612 00014BBC 0010386C007CC6FEC0- <1>
                                             000h, 010h, 038h, 06ch, 000h, 07ch, 0c6h, 0feh, 0c0h, 0c0h, 0c6h, 07ch, 000h, 000h, 000h, 000h
612 00014BC5 C0C67C00000000
                                 <1>
613 00014BCC 0000C600007CC6FEC0- <1>
                                             000h, 000h, 0c6h, 000h, 000h, 07ch, 0c6h, 0feh, 0c0h, 0c0h, 0c6h, 07ch, 000h, 000h, 000h, 000h
                                         db
613 00014BD5 C0C67C00000000
                                 <1>
                                                                                                                            000h, 000h, 000h
614 00014BDC 00603018007CC6FEC0- <1>
                                             000h, 060h,
                                                         030h, 018h, 000h, 07ch, 0c6h, 0feh, 0c0h, 0c0h, 0c6h, 07ch, 000h,
                                         db
614 00014BE5 C0C67C00000000
                                             000h, 000h, 066h, 000h, 000h, 038h, 018h, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h
615 00014BEC 000066000038181818- <1>
                                         db
615 00014BF5 18183C00000000
                                 <1>
616 00014BFC 00183C660038181818- <1>
                                         db
                                             000h, 018h, 03ch, 066h, 000h, 038h, 018h, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h
616 00014C05 18183C00000000
                                 <1>
617 00014C0C 006030180038181818-
                                             000h, 060h, 030h, 018h, 000h, 038h, 018h, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h
617 00014C15 18183C00000000
                                 <1>
618 00014C1C 00C60010386CC6C6FE- <1>
                                             000h, 0c6h, 000h, 010h, 038h, 06ch, 0c6h, 0c6h, 0feh, 0c6h, 0c6h, 0c6h, 000h, 000h, 000h, 000h
                                         db
618 00014C25 C6C6C600000000
                                 <1>
                                                         038h, 000h, 038h, 06ch, 0c6h, 0c6h, 0feh, 0c6h, 0c6h, 0c6h, 000h,
619 00014C2C 386C3800386CC6C6FE- <1>
                                                                                                                            000h, 000h, 000h
619 00014C35 C6C6C600000000
                                 <1>
620 00014C3C 18306000FE666607C60- <1>
                                             018h, 030h, 060h, 000h, 0feh, 066h, 060h, 07ch, 060h, 060h, 066h, 0feh, 000h, 000h, 000h, 000h
                                         db
620 00014C45 6066FE00000000
                                 <1>
621 00014C4C 0000000006E3B1B7E- <1>
                                             000h, 000h, 000h, 000h, 000h, 06eh, 03bh, 01bh, 07eh, 0d8h, 0dch, 077h, 000h, 000h, 000h, 000h
621 00014C55 D8DC7700000000
                                 <1>
622 00014C5C 00003E6CCCCCFECCCC
                                             000h, 000h, 03eh, 06ch, 0cch, 0cch, 0feh, 0cch, 0cch, 0cch, 0cch, 0cch, 000h, 000h, 000h, 000h
                                 <1>
                                         db
622 00014C65 CCCCE00000000
                                 <1>
623 00014C6C 0010386C007CC6C6C6- <1>
                                         db
                                             000h, 010h, 038h, 06ch, 000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h
623 00014C75 C6C67C00000000
                                 <1>
624 00014C7C 0000C600007CC6C6C6-
                                 <1>
                                         db
                                             000h, 000h, 0c6h, 000h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h,
624 00014C85 C6C67C00000000
                                 <1>
625 00014C8C 00603018007CC6C6C6- <1>
                                             000h, 060h, 030h, 018h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h
                                         db
625 00014C95 C6C67C00000000
                                 <1>
626 00014C9C 003078CC00CCCCCCC- <1>
                                             000h, 030h, 078h, 0cch, 000h, 0cch, 0cch, 0cch, 0cch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h
626 00014CA5 CCCC7600000000
                                 <1>
627 00014CAC 0060301800CCCCCCCC <1>
                                             000h, 060h, 030h, 018h, 000h, 0cch, 0cch, 0cch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h
                                         db
627 00014CB5 CCCC7600000000
                                 <1>
628 00014CBC 0000C60000C6C6C6C6- <1>
                                             000h, 000h, 0c6h, 000h, 000h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07eh, 006h, 00ch, 078h, 000h
                                         db
628 00014CC5 C6C67E060C7800
                                 <1>
629 00014CCC 00C6007CC6C6C6C6C6C <1>
                                             000h, 0c6h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h
                                         db
629 00014CD5 C6C67C00000000
                                 <1>
630 00014CDC 00C600C6C6C6C6C6C6C6 <1>
                                         db
                                             000h, 0c6h, 000h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h
630 00014CE5 C6C67C00000000
                                 <1>
631 00014CEC 0018187EC3C0C0C0C3- <1>
                                             000h, 018h, 018h, 07eh, 0c3h, 0c0h, 0c0h, 0c0h, 0c3h, 07eh, 018h, 018h, 000h, 000h, 000h, 000h
631 00014CF5 7E181800000000
                                 <1>
632 00014CFC 00386C6460F0606060- <1>
                                             000h, 038h, 06ch, 064h, 060h, 0f0h, 060h, 060h, 060h, 060h, 0e6h, 0fch, 000h, 000h, 000h, 000h
                                         db
632 00014D05 60E6FC00000000
                                 <1>
633 00014D0C 0000C3663C18FF18FF- <1>
                                             000h, 000h, 0c3h, 066h, 03ch, 018h, 0ffh, 018h, 0ffh, 018h, 018h, 018h, 000h, 000h, 000h, 000h
633 00014D15 18181800000000
                                 <1>
634 00014D1C 00FC66667C62666F66- <1>
                                             000h, 0fch, 066h, 066h, 07ch, 062h, 066h, 066h, 066h, 066h, 066h, 0f3h, 000h, 000h, 000h, 000h
                                         db
634 00014D25 6666F30000000
                                 <1>
635 00014D2C 000E1B1818187E1818- <1>
                                             000h, 00eh, 01bh, 018h, 018h, 018h, 07eh, 018h, 018h, 018h, 018h, 018h, 048h, 070h, 000h, 000h
                                         db
635 00014D35 181818D8700000
                                 <1>
636 00014D3C 0018306000780C7CCC-
                                 <1>
                                             000h, 018h, 030h, 060h, 000h, 078h, 00ch, 07ch, 0cch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h
636 00014D45 CCCC7600000000
                                 <1>
637 00014D4C 000C18300038181818- <1>
                                         db
                                             000h, 00ch, 018h, 030h, 000h, 038h, 018h, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h
637 00014D55 18183C00000000
                                 <1>
638 00014D5C 00183060007CC6C6C6- <1>
                                             000h, 018h, 030h, 060h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h
638 00014D65 C6C67C00000000
                                 <1>
639 00014D6C 0018306000CCCCCC
                             CCC- <1>
                                             000h, 018h, 030h, 060h, 000h, 0cch, 0cch, 0cch, 0cch, 0cch, 0cch, 076h, 000h, 000h, 000h, 000h
                                         db
639 00014D75 CCCC7600000000
                                 <1>
                                             000h, 000h,
640 00014D7C 000076DC00DC666666-
                                 <1>
                                         db
                                                         076h, 0dch, 000h, 0dch, 066h, 066h, 066h, 066h, 066h, 000h,
                                                                                                                            000h, 000h,
640 00014D85 66666600000000
641 00014D8C 76DC00C6E6F6FEDECE- <1>
                                             076h, 0dch, 000h, 0c6h, 0e6h, 0f6h, 0feh, 0deh, 0ceh, 0c6h, 0c6h, 0c6h, 000h, 000h, 000h
641 00014D95 C6C6C600000000
                                 <1>
                                             000h, 03ch, 06ch, 06ch, 03eh, 000h, 07eh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
642 00014D9C 003C6C6C3E007E0000- <1>
                                         db
642 00014DA5 00000000000000
                                 <1>
643 00014DAC 00386C6C38007C0000- <1>
                                             000h, 038h, 06ch, 06ch, 038h, 000h, 07ch, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
643 00014DB5 0000000000000
                                 <1>
                                             000h, 000h, 030h, 030h, 000h, 030h, 030h, 060h, 0c0h, 0c6h, 0c6h, 07ch, 000h, 000h, 000h, 000h
644 00014DBC 0000303000303060C0- <1>
                                         db
644 00014DC5 C6C67C00000000
                                 <1>
645 00014DCC 00000000000FEC0C0- <1>
                                         db
                                             000h, 000h, 000h, 000h, 000h, 000h, 000h, 0feh, 0c0h, 0c0h, 0c0h, 0c0h, 000h, 000h, 000h, 000h, 000h
645 00014DD5 C0C00000000000
                                 <1>
646 00014DDC 00000000000FE0606- <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 006h, 006h, 006h, 006h, 006h, 000h, 000h, 000h, 000h, 000h
                                         db
646 00014DE5 06060000000000
                                 <1>
647 00014DEC 00C0C0C2C6CC183060- <1>
                                             000h, 0c0h, 0c0h, 0c2h, 0c6h, 0cch, 018h, 030h, 060h, 0ceh, 09bh, 006h, 00ch, 01fh, 000h, 000h
                                         db
647 00014DF5 CE9B060C1F0000
                                 <1>
648 00014DFC 00C0C0C2C6CC183066- <1>
                                             000h, 0c0h, 0c0h, 0c2h, 0c6h, 0cch, 018h, 030h, 066h, 0ceh, 096h, 03eh, 006h, 006h, 000h, 000h
                                         db
648 00014E05 CE963E06060000
                                 <1>
649 00014E0C 00001818001818183C- <1>
                                         db
                                             000h, 000h, 018h, 018h, 000h, 018h, 018h, 018h, 03ch, 03ch, 03ch, 018h, 000h, 000h, 000h, 000h
649 00014E15 3C3C1800000000
                                 <1>
650 00014E1C 000000000366CD86C- <1>
                                             000h, 000h, 000h, 000h, 000h, 036h, 06ch, 0d8h, 06ch, 036h, 000h, 000h, 000h, 000h, 000h, 000h
650 00014E25 36000000000000
                                 <1>
                                             000h, 000h, 000h, 000h, 000h, 000h, 0d8h, 06ch, 036h, 06ch, 0d8h, 000h, 000h, 000h, 000h, 000h, 000h
651 00014E2C 000000000D86C366C- <1>
                                         db
651 00014E35 D8000000000000
                                <1>
652 00014E3C 114411441144114411- <1>
                                             011h, 044h, 011h, 044h
```

597 00014ACC 0000000000C6C6C6C6C <1>

db

```
653 00014E4C 55AA55AA55AA55AA55- <1>
                                                                055h, 0aah, 055h, 0aah
653 00014E55 AA55AA55AA55AA
                                                <1>
654 00014E5C DD77DD77DD77DD77DD- <1>
                                                                 0ddh, 077h, 0ddh, 077h
654 00014E65 77DD77DD77DD77
                                               <1>
655 00014E6C 181818181818181818 <1>
                                                           db
                                                                 018h, 018h
655 00014E75 18181818181818
                                                <1>
656 00014E7C 18181818181818F818-
                                                                 018h, 018h
656 00014E85 18181818181818
                                                <1>
657 00014E8C 1818181818F818F818- <1>
                                                                 018h, 018h
                                                           db
657 00014E95 18181818181818
                                                <1>
658 00014E9C 36363636363636F636- <1>
                                                                 036h, 036h
658 00014EA5 36363636363636
                                                <1>
659 00014EAC 00000000000000FE36- <1>
                                                                 000h, 000h, 000h, 000h, 000h, 000h, 000h, 0feh, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 036h
                                                           db
659 00014EB5 36363636363636
                                                <1>
660 00014EBC 0000000000F818F818- <1>
                                                           db
                                                                 000h, 000h, 000h, 000h, 000h, 0f8h, 018h, 0f8h, 018h, 018h, 018h, 018h, 018h, 018h, 018h
660 00014EC5 18181818181818
                                                <1>
661 00014ECC 3636363636F606F636- <1>
                                                                 036h, 036h, 036h, 036h, 036h, 056h, 056h, 066h, 066h, 036h, 
661 00014ED5 36363636363636
                                               <1>
                                                                 036h, 
662 00014EDC 36363636363636363636 <1>
                                                           db
662 00014EE5 36363636363636
                                                <1>
663 00014EEC 000000000FE06F636- <1>
                                                                 000h, 000h, 000h, 000h, 000h, 00feh, 006h, 0f6h, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 036h
663 00014EF5 36363636363636
                                                <1>
664 00014EFC 3636363636F606FE00- <1>
                                                                 036h, 036h, 036h, 036h, 036h, 0f6h, 006h, 0feh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                                           db
664 00014F05 00000000000000
                                                <1>
665 00014F0C 36363636363636FE00- <1>
                                                                 036h, 036h, 036h, 036h, 036h, 036h, 036h, 06eh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
665 00014F15 00000000000000
                                                <1>
666 00014F1C 1818181818F818F800- <1>
                                                                 018h, 018h, 018h, 018h, 018h, 018h, 018h, 068h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
666 00014F25 0000000000000
                                               <1>
667 00014F2C 0000000000000F818- <1>
                                                           db
                                                                 000h, 000h,
                                                                                  000h, 000h, 000h, 000h, 000h, 0f8h, 018h, 018h, 018h, 018h, 018h,
                                                                                                                                                                                  018h, 018h, 018h
667 00014F35 18181818181818
                                               <1>
668 00014F3C 181818181818181F00- <1>
                                                           db
                                                                 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 01fh, 000h, 000h, 000h, 000h, 000h, 000h, 000h
668 00014F45 00000000000000
                                                <1>
669 00014F4C 181818181818FF00- <1>
                                                                 018h, 018h, 018h, 018h, 018h, 018h, 018h, 0ffh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                                           db
669 00014F55 00000000000000
                                                <1>
670 00014F5C 0000000000000FF18- <1>
                                                                 000h, 000h, 000h, 000h, 000h, 000h, 000h, 0ffh, 018h, 018h, 018h, 018h, 018h, 018h, 018h
670 00014F65 18181818181818
                                                <1>
671 00014F6C 181818181818181F18- <1>
                                                                 018h, 018h
                                                           db
671 00014F75 18181818181818
                                                <1>
672 00014F7C 0000000000000FF00- <1>
                                                                 000h, 000h,
                                                                                  000h, 000h, 000h, 000h, 000h, 0ffh, 000h, 000h, 000h, 000h, 000h,
                                                                                                                                                                                  000h, 000h, 000h
672 00014F85 00000000000000
673 00014F8C 18181818181818FF18- <1>
                                                                 018h, 018h
                                                           db
673 00014F95 18181818181818
                                                <1>
674 00014F9C 18181818181F181F18- <1>
                                                           db
                                                                 018h, 018h, 018h, 018h, 018h, 01fh, 018h, 01fh, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h
674 00014FA5 18181818181818
                                                <1>
675 00014FAC 363636363636363736-
                                               <1>
                                                           db
                                                                 036h, 036h
675 00014FB5 36363636363636
                                                <1>
676 00014FBC 363636363637303F00- <1>
                                                                 036h, 036h, 036h, 036h, 036h, 037h, 030h, 03fh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                                           db
676 00014FC5 00000000000000
                                                <1>
677 00014FCC 0000000003F303736- <1>
                                                                 000h, 000h, 000h, 000h, 000h, 03fh, 030h, 037h, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 036h
677 00014FD5 36363636363636
                                                <1>
678 00014FDC 3636363636F700FF00- <1>
                                                                 036h, 036h, 036h, 036h, 036h, 0f7h, 000h, 0ffh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                                           db
678 00014FE5 00000000000000
                                                <1>
679 00014FEC 000000000FF00F736- <1>
                                                                 000h, 000h, 000h, 000h, 000h, 0ffh, 000h, 0f7h, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 036h
679 00014FF5 36363636363636
                                                <1>
680 00014FFC 363636363637303736-
                                                                 036h, 036h, 036h, 036h, 036h, 037h, 030h, 037h, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 036h
                                               <1>
                                                           db
680 00015005 36363636363636
                                                <1>
681 0001500C 0000000000FF00FF00- <1>
                                                           db
                                                                 000h, 000h, 000h, 000h, 000h, 0ffh, 000h, 0ffh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
681 00015015 00000000000000
                                                <1>
682 0001501C 3636363636F700F736- <1>
                                                                 036h, 036h, 036h, 036h, 036h, 0f7h, 000h, 0f7h, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 036h
682 00015025 36363636363636
                                                <1>
683 0001502C 1818181818FF00FF00- <1>
                                                                 018h, 018h, 018h, 018h, 018h, 0ffh, 000h, 0ffh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                                           db
683 00015035 00000000000000
                                                <1>
684 0001503C 36363636363636FF00- <1>
                                                                 036h, 036h, 036h, 036h, 036h, 036h, 036h, 06fh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
684 00015045 00000000000000
                                                <1>
685 0001504C 000000000FF00FF18- <1>
                                                                 000h, 000h, 000h, 000h, 000h, 00ffh, 000h, 0ffh, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h
                                                           db
685 00015055 18181818181818
                                                <1>
686 0001505C 00000000000000FF36- <1>
                                                                 000h, 000h, 000h, 000h, 000h, 000h, 000h, 0ffh, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 036h
                                                           db
686 00015065 36363636363636
                                                <1>
687 0001506C 363636363636363F00- <1>
                                                                 036h, 036h, 036h, 036h, 036h, 036h, 036h, 03fh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                                           db
687 00015075 00000000000000
                                                <1>
688 0001507C 18181818181F181F00- <1>
                                                           db
                                                                 018h, 018h, 018h, 018h, 018h, 01fh, 018h, 01fh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
688 00015085 00000000000000
                                                <1>
689 0001508C 0000000001F181F18- <1>
                                                                 000h, 000h, 000h, 000h, 000h, 01fh, 018h, 01fh, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h
689 00015095 18181818181818
                                                <1>
                                                                 000h, 000h, 000h, 000h, 000h, 000h, 000h, 03fh, 036h, 036h, 036h, 036h, 036h, 036h, 036h, 036h
690 0001509C 00000000000003F36- <1>
                                                           db
690 000150A5 36363636363636
                                                <1>
691 000150AC 36363636363636FF36- <1>
                                                                 036h, 036h
691 000150B5 36363636363636
                                                <1>
692 000150BC 1818181818FF18FF18- <1>
                                                                 018h, 018h
                                                           db
692 000150C5 18181818181818
                                               <1>
693 000150CC 18181818181818F800- <1>
                                                                 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 068h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                                           db
693 000150D5 00000000000000
                                                <1>
694 000150DC 00000000000001F18- <1>
                                                                 000h, 000h, 000h, 000h, 000h, 000h, 000h, 01fh, 018h, 018h, 018h, 018h, 018h, 018h, 018h
694 000150E5 18181818181818
                                               <1>
695 000150EC FFFFFFFFFFFFFFFF <1>
                                                           db
                                                                Offh, Offh
695 000150F5 FFFFFFFFFFFFF
                                                <1>
696 000150FC 00000000000000FFFF- <1>
                                                                 000h, 000h, 000h, 000h, 000h, 000h, 000h, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh, 0ffh
696 00015105 FFFFFFFFFFFFF
                                            <1>
697 0001510C F0F0F0F0F0F0F0F0F0F0- <1>
                                                                 0f0h, 0f0h
697 00015115 F0F0F0F0F0F0F0
                                               <1>
698 0001511C 0F0F0F0F0F0F0F0F0F- <1>
                                                                 00fh, 00fh
698 00015125 0F0F0F0F0F0F0F
                                               <1>
699 0001512C FFFFFFFFFFFFF0000- <1>
                                                                 Offh, Offh, Offh, Offh, Offh, Offh, Offh, O00h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
699 00015135 00000000000000
                                               <1>
700 0001513C 000000000076DCD8D8- <1>
                                                                 000h, 000h, 000h, 000h, 000h, 076h, 0dch, 0d8h, 0d8h, 0d8h, 0dch, 076h, 000h, 000h, 000h
                                                           db
700 00015145 D8DC7600000000
                                               <1>
                                                                 000h, 000h, 078h, 0cch, 0cch, 0cch, 0d8h, 0cch, 0c6h, 0c6h, 0c6h, 0c6h, 000h, 000h, 000h, 000h
701 0001514C 000078CCCCCCD8CCC6- <1>
701 00015155 C6C6CC00000000
                                               <1>
702 0001515C 0000FEC6C6C0C0C0C0- <1>
                                                           db
                                                                 000h, 000h, 0feh, 0c6h, 0c6h, 0c0h, 0c0h, 0c0h, 0c0h, 0c0h, 0c0h, 0c0h, 000h, 000h, 000h, 000h
702 00015165 C0C0C000000000
                                               <1>
703 0001516C 00000000FE6C6C6C6C- <1>
                                                                 000h, 000h, 000h, 000h, 0feh, 06ch, 06ch, 06ch, 06ch, 06ch, 06ch, 06ch, 000h, 000h, 000h, 000h
703 00015175 6C6C6C00000000
                                               <1>
704 0001517C 000000FEC660301830- <1>
                                                                 000h, 000h, 000h, 0feh, 0c6h, 060h, 030h, 018h, 030h, 060h, 0c6h, 0feh, 000h, 000h, 000h, 000h
                                                           db
704 00015185 60C6FE00000000
                                               <1>
705 0001518C 00000000007ED8D8D8- <1>
                                                                 000h, 000h, 000h, 000h, 000h, 07eh, 0d8h, 0d8h, 0d8h, 0d8h, 070h, 000h, 000h, 000h, 000h
                                                           db
705 00015195 D8D87000000000
                                               <1>
706 0001519C 000000006666666666 <1>
                                                                000h, 000h, 000h, 000h, 066h, 066h, 066h, 066h, 066h, 07ch, 060h, 060h, 0c0h, 000h, 000h, 000h
                                                           db
706 000151A5 7C6060C0000000
                                             <1>
707 000151AC 0000000076DC181818- <1>
                                                           db 000h, 000h, 000h, 000h, 076h, 0dch, 018h, 018h, 018h, 018h, 018h, 000h, 000h, 000h, 000h
707 000151B5 18181800000000
                                               <1>
```

652 00014E45 44114411441144

```
708 000151BC 0000007E183C666666- <1>
                                                       000h, 000h, 000h, 07eh, 018h, 03ch, 066h, 066h, 066h, 03ch, 018h, 07eh, 000h, 000h, 000h, 000h
                                                  db
708 000151C5 3C187E00000000
                                        <1>
709 000151CC 000000386CC6C6FEC6- <1>
                                                       000h, 000h, 000h, 038h, 06ch, 0c6h, 0c6h, 0feh, 0c6h, 0c6h, 06ch, 038h, 000h, 000h, 000h, 000h
709 000151D5 C66C3800000000
                                         <1>
                                                       000h, 000h, 038h, 06ch, 0c6h, 0c6h, 0c6h, 06ch, 06ch, 06ch, 06ch, 0eeh, 000h, 000h, 000h, 000h
710 000151DC 0000386CC6C6C6C6C6C- <1>
                                                  db
710 000151E5 6C6CEE00000000
                                        <1>
711 000151EC 00001E30180C3E6666- <1>
                                                       000h, 000h, 01eh, 030h, 018h, 00ch, 03eh, 066h, 066h, 066h, 03ch, 000h, 000h, 000h, 000h
711 000151F5 66663C00000000
712 000151FC 0000000007EDBDBDB- <1>
                                                  db
                                                       000h, 000h, 000h, 000h, 000h, 07eh, 0dbh, 0dbh, 07eh, 000h, 000h, 000h, 000h, 000h, 000h
712 00015205 7E000000000000
                                        <1>
713 0001520C 00000003067EDBDBF3- <1>
                                                       000h, 000h, 000h, 003h, 006h, 07eh, 0dbh, 0dbh, 0f3h, 07eh, 060h, 0c0h, 000h, 000h, 000h, 000h
                                                  db
713 00015215 7E60C000000000
                                        <1>
714 0001521C 00001C3060607C6060- <1>
                                                       000h, 000h, 01ch, 030h, 060h, 060h, 07ch, 060h, 060h, 060h, 030h, 01ch, 000h, 000h, 000h, 000h
714 00015225 60301C00000000
                                        <1>
715 0001522C 0000007CC6C6C6C6C6C6 <1>
                                                       000h, 000h, 000h, 07ch, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 0c6h, 000h,
                                                                                                                                                        000h, 000h, 000h
                                                  db
715 00015235 C6C6C600000000
                                        <1>
716 0001523C 00000000FE0000FE00- <1>
                                                       000h, 000h, 000h, 000h, 0feh, 000h, 0feh, 000h, 0feh, 000h, 0feh, 000h, 000h, 000h, 000h, 000h
716 00015245 00FE0000000000
                                         <1>
717 0001524C 0000000018187E1818- <1>
                                                       000h, 000h, 000h, 000h, 018h, 018h, 07eh, 018h, 018h, 000h, 000h, 0ffh, 000h, 000h, 000h, 000h
                                                  db
717 00015255 0000FF0000000
                                        <1>
718 0001525C 00000030180C060C18- <1>
                                                       000h, 000h, 000h, 030h, 018h, 00ch, 006h, 00ch, 018h, 030h, 000h, 07eh, 000h, 000h, 000h, 000h
                                                  db
718 00015265 30007E00000000
                                        <1>
719 0001526C 0000000C1830603018- <1>
                                                       000h, 000h, 000h, 00ch, 018h, 030h, 060h, 030h, 018h, 00ch, 000h, 07eh, 000h, 000h, 000h, 000h
                                                  db
719 00015275 0C007E00000000
                                        <1>
720 0001527C 00000E1B1B18181818- <1>
                                                  db
                                                       000h, 000h,
                                                                      00eh, 01bh, 01bh, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h,
                                                                                                                                                        018h, 018h, 018h
720 00015285 18181818181818
                                        <1>
721 0001528C 1818181818181818D8- <1>
                                                       018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 018h, 0d8h, 0d8h, 070h, 000h, 000h, 000h, 000h
721 00015295 D8D87000000000
                                        <1>
722 0001529C 000000001818007E00- <1>
                                                  db
                                                       000h, 000h, 000h, 000h, 018h, 018h, 000h, 07eh, 000h, 018h, 018h, 000h, 000h, 000h, 000h, 000h
722 000152A5 18180000000000
                                         <1>
723 000152AC 00000000076DC0076- <1>
                                                       000h, 000h, 000h, 000h, 000h, 076h, 0dch, 000h, 076h, 0dch, 000h, 000h, 000h, 000h, 000h, 000h
723 000152B5 DC000000000000
                                         <1>
724 000152BC 00386C6C3800000000- <1>
                                                       000h, 038h, 06ch, 06ch, 038h, 000h, 000h
                                                  db
                                        <1>
724 000152C5 00000000000000
725 000152CC 00000000000001818- <1>
                                                                                                                                                        000h, 000h, 000h
                                                       000h, 000h,
                                                                      000h, 000h, 000h, 000h, 000h, 018h, 018h, 000h, 000h, 000h, 000h,
                                                  db
725 000152D5 00000000000000
                                        <1>
726 000152DC 00000000000000018-
                                        <1>
                                                       000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 018h, 000h, 000h, 000h, 000h, 000h, 000h
                                                  db
726 000152E5 00000000000000
                                        <1>
727 000152EC 000F0C0C0C0C0CEC6C- <1>
                                                  db
                                                       000h, 00fh, 00ch, 00ch, 00ch, 00ch, 00ch, 0ech, 06ch, 06ch, 03ch, 01ch, 000h, 000h, 000h, 000h
727 000152F5 6C3C1C00000000
                                         <1>
728 000152FC 00D86C6C6C6C6C0000- <1>
                                                       000h, 0d8h, 06ch, 06ch, 06ch, 06ch, 06ch, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
728 00015305 00000000000000
                                        <1>
729 0001530C 0070D83060C8F80000- <1>
                                                       000h, 070h, 0d8h, 030h, 060h, 0c8h, 0f8h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h
                                                  db
729 00015315 00000000000000
                                        <1>
730 0001531C 000000007C7C7C7
                                                       000h, 000h, 000h, 000h, 07ch, 07ch, 07ch, 07ch, 07ch, 07ch, 07ch, 000h, 000h,
                                    7C- <1>
730 00015325 7C7C0000000000
                                         <1>
731 0001532C 0000000000000000000 <1>
                                                       000h, 000h
                                                  db
731 00015335 00000000000000
                                        <1>
                                        <1> vgafont14alt:
732
733 0001533C 1D000000002466FF66- <1>
                                                       01dh, 000h, 000h, 000h, 000h, 024h, 066h, 0ffh, 066h, 024h, 000h, 000h, 000h,
                                                                                                                                                       000h, 000h, 022h
733 00015345 24000000000022
                                        <1>
734 0001534C 006363632200000000- <1>
                                                       000h, 063h, 063h, 063h, 022h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 02bh, 000h
                                                  db
734 00015355 00000000002B00
                                        <1>
735 0001535C 0000181818FF181818- <1>
                                                       000h, 000h, 018h, 018h, 018h, 0ffh, 018h, 018h, 018h, 000h, 000h, 000h, 000h, 02dh, 000h, 000h
735 00015365 000000002D0000
736 0001536C 00000000FF00000000-
                                                       000h, 000h, 000h, 000h, 0ffh, 000h, 000h, 000h, 000h, 000h, 000h, 000h, 04dh, 000h, 000h, 0c3h
                                        <1>
                                                  db
736 00015375 0000004D0000C3
                                        <1>
737 0001537C E7FFDBC3C3C3C3C3C300- <1>
                                                       0e7h, 0ffh, 0dbh, 0c3h, 0c3h, 0c3h, 0c3h, 0c3h, 0c3h, 000h, 000h, 000h, 054h, 000h, 000h, 0ffh, 0dbh
737 00015385 0000540000FFDB
                                        <1>
    0001538C 9918181818183C0000- <1>
                                                       099h, 018h, 018h, 018h, 018h, 018h, 03ch, 000h, 000h, 000h, 056h, 000h, 000h, 0c3h, 0c3h, 0c3h
738 00015395 00560000C3C3C3
                                        <1>
739 0001539C C3C3C3663C18000000- <1>
                                                       0c3h, 0c3h, 0c3h, 066h, 03ch, 018h, 000h, 000h, 000h, 057h, 000h, 000h, 0c3h, 0c3h, 0c3h, 0c3h
                                                  db
739 000153A5 570000C3C3C3C3
                                        <1>
740 000153AC DBDBFF666600000058- <1>
                                                       Odbh, Odbh, Offh, O66h, O66h, O00h, O00h, O00h, O58h, O00h, O00h, Oc3h, Oc3h, O66h, O3ch, O18h
740 000153B5 0000C3C3663C18
                                         <1>
741 000153BC 3C66C3C30000005900- <1>
                                                       03ch, 066h, 0c3h, 0c3h, 000h, 000h, 000h, 059h, 000h, 000h, 0c3h, 0c3h, 0c3h, 066h, 03ch, 018h
                                                  db
741 000153C5 00C3C3C3663C18
                                        <1>
742 000153CC 18183C0000005A0000- <1>
                                                       018h, 018h, 03ch, 000h, 000h, 000h, 05ah, 000h, 000h, 0ffh, 0c3h, 086h, 00ch, 018h, 030h, 061h
                                                  db
742 000153D5 FFC3860C183061
                                        <1>
743 000153DC C3FF0000006D000000- <1>
                                                       0c3h, 0ffh, 000h, 000h, 000h, 06dh, 000h, 000h, 000h, 000h, 000h, 0e6h, 0ffh, 0dbh, 0dbh
                                                  db
743 000153E5 0000E6FFDBDBDB
                                        <1>
744 000153EC DB0000007600000000- <1>
                                                  db
                                                       0dbh, 000h, 000h, 000h, 076h, 000h, 000h, 000h, 000h, 000h, 0c3h, 0c3h, 0c3h, 066h, 03ch, 018h
744 000153F5 00C3C3C3663C18
                                        <1>
745 000153FC 000000770000000000 <1>
                                                       000h, 000h, 000h, 077h, 000h, 000h, 000h, 000h, 000h, 0c3h, 0c3h, 0dbh, 0dbh, 0ffh, 066h, 000h
745 00015405 C3C3DBDBFF6600
                                         <1>
746 0001540C 00009100000006E3B- <1>
                                                       000h, 000h, 091h, 000h, 000h, 000h, 000h, 06eh, 03bh, 01bh, 07eh, 0d8h, 0dch,
                                                                                                                                                       077h, 000h, 000h
                                                  db
746 00015415 1B7ED8DC770000
                                        <1>
747 0001541C 009B0018187EC3C0C0- <1>
                                                       000h, 09bh, 000h, 018h, 018h, 07eh, 0c3h, 0c0h, 0c0h, 0c3h, 07eh, 018h, 018h, 000h, 000h, 000h
747 00015425 C37E1818000000
                                         <1>
748 0001542C 9D0000C3663C18FF18- <1>
                                                       09dh, 000h, 000h, 0c3h, 066h, 03ch, 018h, 0ffh, 018h, 0ffh, 018h, 018h, 000h, 000h, 000h, 09eh
                                                  db
748 00015435 FF18180000009E
                                        <1>
749 0001543C 00FC66667C62666F66- <1>
                                                       000h, 0fch, 066h, 066h, 07ch, 062h, 066h, 06fh, 066h, 066h, 0f3h, 000h, 000h, 000h, 0f1h, 000h
                                                  db
749 00015445 66F3000000F100
                                        <1>
                                                       000h, 018h, 018h, 018h, 0ffh, 018h, 018h, 018h, 000h, 0ffh, 000h, 000h, 000h, 0ffh, 000h, 000h
750 0001544C 00181818FF18181800- <1>
750 00015455 FF000000F60000
                                        <1>
751 0001545C 18180000FF00001818- <1>
                                                  db
                                                       018h, 018h, 000h, 000h, 0ffh, 000h, 000h, 018h, 018h, 000h, 000h, 000h, 000h
751
    00015465 00000000
                                         <1>
                                        <1> vgafont16alt:
753 00015469 1D00000000002466FF- <1>
                                                  db 01dh, 000h, 000h, 000h, 000h, 000h, 024h, 066h, 0ffh, 066h, 024h, 000h, 000h, 000h, 000h, 000h
753 00015472 66240000000000
                                        <1>
754 00015479 003000003C66C3C3DB- <1>
                                                       000h, 030h, 000h, 000h, 03ch, 066h, 0c3h, 0c3h, 0dbh, 0dbh, 0c3h, 0c3h, 066h, 03ch, 000h, 000h
754 00015482 DBC3C3663C0000
                                        <1>
755 00015489 00004D0000C3E7FFFF- <1>
                                                       000h, 000h, 04dh, 000h, 000h, 0c3h, 0e7h, 0ffh, 0ffh, 0dbh, 0c3h, 0c3h, 0c3h, 0c3h, 0c3h, 0c0h
755 00015492 DBC3C3C3C3C300
                                        <1>
756 00015499 000000540000FFDB99- <1>
                                                  db
                                                       000h, 000h, 000h, 054h, 000h, 000h, 0ffh, 0dbh, 099h, 018h, 018h, 018h, 018h, 018h, 03ch
756 000154A2 1818181818183C
                                        <1>
757 000154A9 00000000560000C3C3- <1>
                                                  db
                                                       000h, 000h, 000h, 000h, 056h, 000h, 000h, 0c3h, 
757 000154B2 C3C3C3C3C3663C
                                        <1>
758 000154B9 180000000570000C3- <1>
                                                       018h, 000h, 000h, 000h, 000h, 000h, 057h, 000h, 000h, 0c3h, 0c3h, 0c3h, 0c3h, 0c3h, 0dbh, 0dbh, 0ffh
                                                  db
758 000154C2 C3C3C3C3DBDBFF
                                        <1>
759 000154C9 66660000000580000- <1>
                                                       066h, 066h, 000h, 000h, 000h, 000h, 058h, 000h, 000h, 0c3h, 0c3h, 066h, 03ch, 018h, 018h, 03ch
                                                  db
759 000154D2 C3C3663C18183C
                                        <1>
760 000154D9 66C3C300000005900- <1>
                                                       066h, 0c3h, 0c3h, 000h, 000h, 000h, 000h, 059h, 000h, 000h, 0c3h, 0c3h, 0c3h, 066h, 03ch, 018h
760 000154E2 00C3C3C3663C18
                                        <1>
761 000154E9 1818183C00000005A- <1>
                                                       018h, 018h, 018h, 03ch, 000h, 000h, 000h, 000h, 05ah, 000h, 000h, 0ffh, 0c3h, 086h, 00ch, 018h
761 000154F2 0000FFC3860C18
                                        <1>
762 000154F9 3060C1C3FF00000000- <1>
                                                       030h, 060h, 0c1h, 0c3h, 0ffh, 000h, 000h, 000h, 000h, 06dh, 000h, 000h, 000h, 000h, 000h, 0e6h
762 00015502 6D0000000000E6
                                        <1>
                                                       Offh, Odbh, Odbh, Odbh, Odbh, Odbh, Oo0h, O00h, O00h, O00h, O76h, O00h, O00h, O00h, O00h
763 00015509 FFDBDBDBDBDBDB000000- <1>
763 00015512 00760000000000
                                       <1>
764 00015519 C3C3C3C3663C180000- <1>
                                                       0c3h, 0c3h, 0c3h, 0c3h, 066h, 03ch, 018h, 000h, 000h, 000h, 000h, 077h, 000h, 000h, 000h, 000h
```

```
764 00015522 00007700000000
                                <1>
765 00015529 00C3C3C3DBDBFF6600- <1>
                                        db 000h, 0c3h, 0c3h, 0c3h, 0dbh, 0dbh, 0ffh, 066h, 000h, 000h, 000h, 000h, 078h, 000h, 000h, 000h
765 00015532 00000078000000
766 00015539 0000C3663C183C66C3- <1>
                                            000h, 000h, 0c3h, 066h, 03ch, 018h, 03ch, 066h, 0c3h, 000h, 000h, 000h, 000h, 091h, 000h, 000h
766 00015542 00000000910000
                              <1>
767 00015549 0000006E3B1B7ED8DC- <1>
                                        db
                                            000h, 000h, 000h, 06eh, 03bh, 01bh, 07eh, 0d8h, 0dch, 077h, 000h, 000h, 000h, 000h, 09bh, 000h
767 00015552 7700000009B00
                              <1>
768 00015559 18187EC3C0C0C0C37E- <1>
                                            018h, 018h, 07eh, 0c3h, 0c0h, 0c0h, 0c0h, 0c3h, 07eh, 018h, 018h, 000h, 000h, 000h, 000h, 09dh
768 00015562 1818000000009D
                               <1>
769 00015569 0000C3663C18FF18FF- <1>
                                            000h, 000h, 0c3h, 066h, 03ch, 018h, 0ffh, 018h, 0ffh, 018h, 018h, 018h, 000h, 000h, 000h, 000h
                                        db
769 00015572 18181800000000
                              <1>
770 00015579 9E00FC666667C62666F- <1>
                                            09eh, 000h, 0fch, 066h, 066h, 07ch, 062h, 066h, 06fh, 066h, 066h, 066h, 0f3h, 000h, 000h
770 00015582 666666F3000000
                              <1>
771 00015589 00AB00C0C0C2C6CC18- <1>
                                        db 000h, 0abh, 000h, 0c0h, 0c0h, 0c2h, 0c6h, 0cch, 018h, 030h, 060h, 0ceh, 09bh, 006h, 00ch, 01fh
771 00015592 3060CE9B060C1F
                               <1>
772 00015599 0000AC00C0C0C2C6CC- <1>
                                        db 000h, 000h, 0ach, 000h, 0c0h, 0c0h, 0c2h, 0c6h, 0cch, 018h, 030h, 066h, 0ceh, 096h, 03eh, 006h
772 000155A2 183066CE963E06
                                <1>
773 000155A9 06000000
                                <1>
                                        db 006h, 000h, 000h, 000h
2646
                                      align 2
2647 000155AD 90
2648
2649
                                      ; EPOCH Variables
2650
                                      ; 13/04/2015 - Retro UNIX 386 v1 Beginning
2651
                                      ; 09/04/2013 epoch variables
2652
                                      ; Retro UNIX 8086 v1 Prototype: UNIXCOPY.ASM, 10/03/2013
2653
2654 000155AE B207
                                      year:
                                                   dw 1970
2655 000155B0 0100
                                      month:
                                                   dw 1
2656 000155B2 0100
                                      day: dw 1
2657 000155B4 0000
                                      hour:
                                                   dw 0
2658 000155B6 0000
                                      minute: dw 0
2659 000155B8 0000
                                      second: dw 0
2660
                                      DMonth:
2661
2662 000155BA 0000
                                            dw 0
2663 000155BC 1F00
                                            dw 31
2664 000155BE 3B00
                                            dw 59
2665 000155C0 5A00
                                            dw 90
2666 000155C2 7800
                                            dw 120
2667 000155C4 9700
                                            dw 151
2668 000155C6 B500
                                            dw 181
2669 000155C8 D400
                                            dw 212
2670 000155CA F300
                                            dw 243
2671 000155CC 1101
                                            dw 273
2672 000155CE 3001
                                            dw 304
2673 000155D0 4E01
                                            dw 334
2674
2675
                                       ; 20/02/2017
                                       KERNELFSIZE equ $ ; 04/07/2016
2676
2677
2678
                                      bss_start:
2679
2680
                                      ABSOLUTE bss start
2681
2682 000155D2 <res 00000006>
                                      alignb 8 ; 25/12/2016
2683
2684
                                            ; 15/04/2016
2685
                                            ; TRDOS 386 (TRDOS v2.0)
                                                  80 interrupts
2686
2687
                                            ; 11/03/2015
                                            ; Interrupt Descriptor Table (20/08/2014)
2688
2689
                                       idt:
2690
                                            ;resb 64*8; INT 0 to INT 3Fh
                                            ; 15/04/2016
2691
2692 000155D8 <res 00000280>
                                            resb 80*8; INT 0 to INT 4Fh
2693
2694
                                       idt_end:
2695
2696
                                       ;alignb 4
2697
2698
                                       task_state segment:
2699
                                           ; 24/03/2015
2700 00015858 <res 00000002>
                                       tss.link: resw 1
2701 0001585A <res 00000002>
                                                resw 1
2702
                                       ; tss offset 4
                                      tss.esp0: resd 1
2703 0001585C <res 00000004>
2704 00015860 <res 00000002>
                                       tss.ss0:
                                                  resw 1
2705 00015862 <res 00000002>
                                                resw 1
2706 00015864 <res 00000004>
                                       tss.esp1: resd 1
2707 00015868 <res 00000002>
                                       tss.ss1:
2708 0001586A <res 00000002>
                                                resw 1
                                       tss.esp2: resd 1
2709 0001586C <res 00000004>
2710 00015870 <res 00000002>
                                       tss.ss2:
                                                   resw 1
2711 00015872 <res 00000002>
                                                resw 1
2712
                                      ; tss offset 28
                                      tss.CR3: resd 1
2713 00015874 <res 00000004>
2714 00015878 <res 00000004>
                                      tss.eip:
                                                  resd 1
2715 0001587C <res 00000004>
                                      tss.eflags: resd 1
2716
                                      ; tss offset 40
2717 00015880 <res 00000004>
                                                  resd 1
                                      tss.eax:
2718 00015884 <res 00000004>
                                      tss.ecx:
                                                  resd 1
2719 00015888 <res 00000004>
                                      tss.edx:
                                                  resd 1
2720 0001588C <res 00000004>
                                      tss.ebx:
                                                  resd 1
2721 00015890 <res 00000004>
                                      tss.esp:
                                                  resd 1
                                      tss.ebp:
2722 00015894 <res 00000004>
                                                   resd 1
                                      tss.esi:
2723 00015898 <res 00000004>
                                                  resd 1
2724 0001589C <res 00000004>
                                      tss.edi:
                                                  resd 1
                                      ; tss offset 72
2725
2726 000158A0 <res 00000002>
                                      tss.ES:
                                                 resw 1
2727 000158A2 <res 00000002>
                                                resw 1
2728 000158A4 <res 00000002>
                                      tss.CS:
                                                 resw 1
                                                resw 1
2729 000158A6 <res 00000002>
2730 000158A8 <res 00000002>
                                      tss.SS:
                                                 resw 1
2731 000158AA <res 00000002>
                                                 resw 1
                                                 resw 1
2732 000158AC <res 00000002>
                                      tss.DS:
2733 000158AE <res 00000002>
                                                resw 1
```

```
resw 1
2734 000158B0 <res 00000002>
2735 000158B2 <res 00000002>
                                              resw 1
2736 000158B4 <res 00000002>
                                     tss.GS:
                                                    resw 1
2737 000158B6 <res 00000002>
                                              resw 1
2738 000158B8 <res 00000002>
                                     tss.LDTR: resw 1
2739 000158BA <res 00000002>
                                             resw 1
2740
                                     ; tss offset 100
2741 000158BC <res 00000002>
                                    resw 1 tss.IOPB: resw 1
2742 000158BE <res 00000002>
2743
                                     ; tss offset 104
2744
                                     tss_end:
2745
                                     k_page_dir: resd 1 ; Kernel's (System) Page Directory address
2746 000158C0 <res 00000004>
2747
                                                    ; (Physical address = Virtual address)
2748 000158C4 <res 00000004>
                                     memory_size: resd 1 ; memory size in pages
                                     free pages: resd 1 ; number of free pages
2749 000158C8 <res 00000004>
2750 000158CC <res 00000004>
                                     next_page: resd 1 ; offset value in M.A.T. for
2751
                                                    ; first free page search
                                                 resd 1 ; offset value in M.A.T. which
2752 000158D0 <res 00000004>
                                     last_page:
2753
                                                   ; next free page search will be
2754
                                                     ; stopped after it. (end of M.A.T.)
2755 000158D4 <res 00000004>
                                     first page: resd 1; offset value in M.A.T. which
2756
                                                    ; first free page search
2757
                                                    ; will be started on it. (for user)
2758 000158D8 <res 00000004>
                                    mat_size:
                                                 resd 1 ; Memory Allocation Table size in pages
2759
2760
                                     ; 02/09/2014 (Retro UNIX 386 v1)
                                     ; 04/12/2013 (Retro UNIX 8086 v1)
2761
2762 000158DC <res 00000002>
                                     CRT_START: resw 1
                                                               ; starting address in regen buffer
                                                       ; NOTE: active page only
2763
2764 000158DE <res 00000010>
                                     CURSOR_POSN: resw 8 ; cursor positions for video pages
2765
                                     ACTIVE PAGE:
2766 000158EE <res 00000001>
                                                     resb 1 ; current tty
                                    ptty:
2767
                                     ; 01/07/2015 - 29/01/2016
2768 000158EF <res 00000001>
                                                     resb 1 ; current color attribute
                                     ccolor:
2769
                                     ; 26/10/2015
2770
                                     ; 07/09/2014
2771 000158F0 <res 00000014>
                                                 resw ntty+2; Character buffer (multiscreen)
                                     ttychr:
2772
                                     ; 18/05/2015 (03/06/2013 - Retro UNIX 8086 v1 feature only!)
2773
2774 00015904 <res 00000004>
                                                 resd 1 ; present time (for systime & sysmdate)
                                    p_time:
2775
2776
                                     ; 18/05/2015 (16/08/2013 - Retro UNIX 8086 v1 feature only !)
2777
                                     ; (open mode locks for pseudo TTYs)
2778
                                     ; [ major tty locks (return error in any conflicts) ]
2779 00015908 <res 00000014>
                                     ttyl:
                                                 resw ntty+2; opening locks for TTYs.
2780
                                     ; 15/04/2015 (Retro UNIX 386 v1)
2781
2782
                                     ; 22/09/2013 (Retro UNIX 8086 v1)
                                                resb ntty+2; wait channel list (0 to 9 for TTYs)
2783 0001591C <res 0000000A>
                                     ; 15/04/2015 (Retro UNIX 386 v1)
2784
2785
                                     ;; 12/07/2014 -> sp_init set comm. parameters as 0E3h
2786
                                     ;; 0 means serial port is not available
2787
                                     ;;comprm: ; 25/06/2014
2788 00015926 <res 00000001>
                                     com1p:
                                                resb 1 ;;0E3h
2789 00015927 <res 00000001>
                                     com2p:
                                                 resb 1 ;;0E3h
2790
2791
                                     ; 17/11/2015
2792
                                     ; request for response (from the terminal)
2793 00015928 <res 00000002>
                                               resw 1
                                     req resp:
2794
                                     ; 07/11/2015
2795 0001592A <res 00000001>
                                     ccomport:
                                               resb 1 ; current COM (serial) port
2796
                                                     ; (0 = COM1, 1 = COM2)
2797
                                     ; 09/11/2015
2798 0001592B <res 00000001>
                                                     resb 1 ; 'query or response' sign (u9.s, 'sndc')
                                     comgr:
2799
                                     ; 07/11/2015
2800 0001592C <res 00000002>
                                     rchar:
                                                     resw 1 ; last received char for COM 1 and COM 2
2801 0001592E <res 00000002>
                                                     resw 1 ; last sent char for COM 1 and COM 2
                                     schar:
2802
2803
                                     ; 22/08/2014 (RTC)
                                     ; (Packed BCD)
2804
2805 00015930 <res 00000001>
                                     time seconds: resb 1
2806 00015931 <res 00000001>
                                     time_minutes: resb 1
2807 00015932 <res 00000001>
                                     time_hours: resb 1
                                     date_wday:
2808 00015933 <res 00000001>
                                                  resb 1
2809 00015934 <res 00000001>
                                     date day:
                                                  resb 1
2810 00015935 <res 00000001>
                                     date month: resb 1
2811 00015936 <res 00000001>
                                     date_year:
                                                  resb 1
2812 00015937 <res 00000001>
                                     date century: resb 1
2813
2814
                                     ; 24/01/2016
2815 00015938 <res 00000004>
                                     RTC LH:
                                                       resd 1
                                     RTC WAIT_FLAG: resb 1
2816 0001593C <res 00000001>
2817 0001593D <res 00000001>
                                     USER FLAG:
2818
                                     ; 19/05/2016
2819
                                     ;RTC second:
2820 0001593E <res 00000001>
                                    RTC_2Hz:
                                                   resb 1; from 2Hz interrupt to 1Hz timer event function
2821
2822
                                     %include 'diskbss.s'; UNINITIALIZED DISK (BIOS) DATA
                                 1
                                 <1>; TRDOS386.ASM (TRDOS 386 Kernel) - v2.0.0 - diskbss.s
  2
  3
                                 <1> ; Last Update: 24/01/2016
   4
  5
                                 <1> ; Beginning: 24/01/2016
  6
  7
                                 <1> ; -----
  8
                                 <1>; Assembler: NASM version 2.11 (trdos386.s)
  9
                                 10
                                 <1>; Turkish Rational DOS
                                 <1>; Operating System Project v2.0 by ERDOGAN TAN (Beginning: 04/01/2016)
  11
                                 <1> ;
  12
  13
                                 <1>; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
 14
                                 <1>; diskbss.inc (10/07/2015)
 15
                                 <1>;
                                 <1>; Derived from 'IBM PC-XT-286' BIOS source code (1986)
 16
```

tss.FS:

```
17
 18
                            <1>
                            <1>; Retro UNIX 386 v1 Kernel - DISKBSS.INC
 19
 20
                            <1> ; Last Modification: 10/07/2015
 21
                            <1> ; (Unnitialized Disk Parameters Data section for 'DISKIO.INC')
 22
                            <1>
 23 0001593F <res 00000001>
                           <1> alignb 2
 24
                           <1>
 25
                           <1> ;-----
                           <1>; TIMER DATA AREA :
 26
 27
                            <1> ;------
 28
                            <1>
 29
                           <1> TIMER_LH: ; 16/02/205
                                                             ; LOW WORD OF TIMER COUNT
                           <1> TIMER_LOW: resw 1
 30 00015940 <res 00000002>
                                                             ; HIGH WORD OF TIMER COUNT
                           <1> TIMER_HIGH: resw 1 <1> TIMER_OFL: resb 1 <1>
 31 00015942 <res 00000002>
 32 00015944 <res 00000001>
                                                                ; TIMER HAS ROLLED OVER SINCE LAST READ
 34
                           <1> ;------
                            <1>; DISKETTE DATA AREAS :
 35
                           <1> ;-----
 36
 37
                           <1>
 38 00015945 <res 00000001>
                           <1> SEEK STATUS: resb 1
 39 00015946 <res 00000001>
                           <1> MOTOR_STATUS: resb 1
 40 00015947 <res 00000001>
                           <1> MOTOR_COUNT: resb 1
 41 00015948 <res 00000001>
                           <1> DSKETTE STATUS: resb 1
 42 00015949 <res 00000007>
                           <1> NEC_STATUS: resb 7
                           <1>
                           <1> ;-----
 44
                           <1>; ADDITIONAL MEDIA DATA
 45
                           <1> ;-----
 46
 47
                           <1>
 48 00015950 <res 00000001>
                           <1> LASTRATE:
                                        resb 1
 49 00015951 <res 00000001>
                           <1> HF_STATUS: resb
 50 00015952 <res 00000001>
                           <1> HF ERROR: resb 1
 51 00015953 <res 00000001>
                           <1> HF_INT_FLAG: resb 1
 52 00015954 <res 00000001>
                           <1> HF_CNTRL: resb
                                              1
 53 00015955 <res 00000004>
                           <1> DSK STATE: resb 4
 54 00015959 <res 00000002>
                           <1> DSK_TRK: resb 2
 55
                           <1>
                                 56
                           <1> ;--
                           <1> ; FIXED DISK DATA AREAS
 57
                            <1> ;-----
 58
 59
                           <1>
                           60 0001595B <res 00000001>
 61 0001595C <res 00000001>
 62 0001595D <res 00000001>
 64
 65
                           <1>
 66
 67 0001595E <res 00000002>
                           <1> alignb 4
 68
                           <1>
                           <1>; HF_TBL_VEC: resd 1
 69
                                                         ; Primary master disk param. tbl. pointer
 70
                           <1>; HF1_TBL_VEC: resd 1
                                                         ; Primary slave disk param. tbl. pointer
 71
                           <1> HF_TBL_VEC: ; 22/12/2014
                           <1> HDPM_TBL_VEC: resd 1 <1> HDPS_TBL_VEC: resd 1
 72 00015960 <res 00000004>
                                                             ; Primary master disk param. tbl. pointer
 73 00015964 <res 00000004>
                                                             ; Primary slave disk param. tbl. pointer
                           <1> HDSM_TBL_VEC: resd 1
<1> HDSS_TBL_VEC: resd 1
 74 00015968 <res 00000004>
                                                             ; Secondary master disk param. tbl. pointer
 75 0001596C <res 00000004>
                           <1> HDSS_TBL_VEC:
                                                              ; Secondary slave disk param. tbl. pointer
 76
                           <1>
 77
                           <1> ; 03/01/2015
 78 00015970 <res 00000001>
                           <1> LBAMode:
                                              resb 1
 79
                           <1>
                           80
2823
2824
                               ;;; Real Mode Data (10/07/2015 - BSS)
2825
                               ;alignb 2
2826
2827
2828
                               ; 10/01/2016
                               %include 'trdoskx.s'; UNINITIALIZED KERNEL (Logical Drive & FS) DATA
2829
                            1
  2
                            <1>; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.2) - UNINITIALIZED DATA : trdoskx.s
                            <1> ; ------
                            <1>; Last Update: 30/08/2020
  4
  5
                            <1> ; -----
                            <1>; Beginning: 04/01/2016
  6
                            <1>; Assembler: NASM version 2.11 (trdos386.s)
  8
  9
 1.0
                            <1>; Derived from TRDOS Operating System v1.0 (8086) source code by Erdogan Tan
                            <1>; TRDOS2.ASM (09/11/2011)
 12
 13
                            <1> ; DRV_INIT.ASM [26/09/2009] Last Update: 07/08/2011
 14
                            <1>; MAINPROG.ASM [17/01/2004] Last Update: 09/11/2011
                            <1>; DIR.ASM [17/01/2004] Last Update: 09/10/2011
 15
                            <1>; CMD INTR.ASM [29/01/2005] Last update: 09/11/2011
 16
                            <1>; DRV_FAT.ASM [07/07/2009] Last update: 21/08/2011
 17
 18
                            <1>
 19 00015971 <res 00000003>
                            <1> alignb 4
 20
                            <1>
 21
                            <1>; MAINPROG.ASM
 22 00015974 <res 00000004>
                            <1> MainProgCfg_FileSize: resd 1 ; 14/04/2016
 23 00015978 <res 00000004>
                            <1> MainProgCfg_LineOffset: resd 1 ; 14/04/2016
                            <1>
 25 0001597C <res 00000004>
                           <1> Current_VolSerial: resd 1
 26
                            <1>
 27 00015980 <res 00000004>
                           <1> Current_Dir_FCluster: resd 1
 28
                            <1>
                           <1> Current Dir Level: resb 1
 29 00015984 <res 00000001>
 30 00015985 <res 00000001>
                            <1> Current_FATType: resb 1
 31 00015986 <res 00000001>
                            <1> Current_Drv: resb 1
 32 00015987 <res 00000001>
                           <1> Current_Dir_Drv: resb 1 ; '?'
 33 00015988 <res 00000001>
                                              resb 1 ; ':'
                            <1>
 34 00015989 <res 00000001>
                           <1> Current_Dir_Root: resb 1 ; '/'
```

```
36 000159E4 <res 00000001>
                                  <1> End Of Current Dir Str: resb 1
 37 000159E5 <res 00000001>
                                  <1> Current_Dir_StrLen: resb 1
                                  <1>
 39 000159E6 <res 00000001>
                                  <1> CursorColumn:
                                                         resb 1
 40 000159E7 <res 00000001>
                                  <1> CmdArgStart:
                                                      resb 1
                                  <1>
                                  <1> ; 03/02/2016
 42
 43 000159E8 <res 0000004E>
                                  <1> Remark:
                                                         resb 78
 44
                                  <1>
 45 00015A36 <res 00000050>
                                  <1> CommandBuffer:
                                                         resb 80
 46
                                  <1>
                                  <1> TextBuffer: resb 256
 47 00015A86 <res 00000100>
 48
                                  <1>
 49
                                  <1> MasterBootBuff:
                                  <1> MasterBootCode: resb 1BEh
 50 00015B86 <res 000001BE>
 51 00015D44 <res 00000040>
                                  <1> PartitionTable: resb 64
                                  <1> MBIDCode: resw 1
 52 00015D84 <res 00000002>
 53
                                  <1>
 54
                                  <1> PTable Buffer:
 55 00015D86 <res 00000040>
                                  <1> PTable_hd0: resb 64
 56 00015DC6 <res 00000040>
                                  <1> PTable hd1: resb 64
 57 00015E06 <res 00000040>
                                  <1> PTable_hd2: resb 64
 58 00015E46 <res 00000040>
                                  <1> PTable_hd3: resb 64
                                  <1>; 15/07/2020
 59
 60
                                  <1> ;PTable_ep0: resb 64
 61
                                  <1> ;PTable ep1: resb 64
                                  <1> ;PTable_ep2: resb 64
 62
 63
                                  <1> ;PTable_ep3: resb 64
                                  <1>
 64
 65
                                  <1> ; 13/08/2020
 66 00015E86 <res 00000001>
                                  <1> scount: resb 1 ; 16/05/2016 (diskio.s, 'int33h:')
 67 00015E87 <res 00000001>
                                  <1>
                                          resb 1
 68 00015E88 <res 00000001>
                                  <1>
                                            resb 1
 69 00015E89 <res 00000001>
                                  <1>
                                           resb 1
 70
                                  <1>
 71 00015E8A <res 00000001>
                                  <1> HD LBA yes: resb 1
                                  <1> PP_Counter: resb 1
 72 00015E8B <res 00000001>
 73 00015E8C <res 00000001>
                                  <1> EP_Counter: resb 1
                                  <1>; \overline{1}3/08/2020
 75 00015E8D <res 00000001>
                                  <1> LD_Counter: resb 1
 76
                                  <1>
 77
                                  <1>; 30/08/2020
 78 00015E8E <res 00000004>
                                  <1> MBR_EP_StartSector: resd 1
                                                ; Extd partition start sector as in MBR
 79
                                  <1>
                                  <1> EP_StartSector: resd 1 ; next extd partition start sector
 80 00015E92 <res 00000004>
                                         ; 15/07/2020
                                  <1>
 81
                                                      ;resd 1
 82
                                  <1>
 83
                                  <1>
                                                      ;resd 1
 84
                                  <1>
 85
                                  <1>; 20/07/2020
 86 00015E96 <res 00000200>
                                  <1> DOSBootSectorBuff: resb 512
 87
                                  <1> ; 15/07/2020
 88
                                  <1> ;DOSBootSectorBuff: resb 446 ; 1BEh
                                  <1> ; MiniPartitionTable: resb 64 ; 40h
 89
 90
                                  <1> ; MiniPartitionMagic: resw 1 ; 02h
 92
                                  <1> FAT_BuffDescriptor:
 93 00016096 <res 00000004>
                                  <1> FAT_CurrentCluster: resd 1
                                  <1> FAT BuffValidData: resb 1
 94 0001609A <res 00000001>
 95 0001609B <res 00000001>
                                  <1> FAT_BuffDrvName: resb 1
 96 0001609C <res 00000002>
                                  <1> FAT BuffOffset: resw 1
 97 0001609E <res 00000004>
                                  <1> FAT_BuffSector: resd 1
                                  <1>
 99 000160A2 <res 00000004>
                                  <1> FAT_ClusterCounter: resd 1
100 000160A6 <res 00000004>
                                  <1> LastCluster: resd 1
101
                                  <1>
                                  <1> ; 16/05/2016
102
103
                                  <1> ;; 18/03/2016 (TRDOS v2.0)
104
                                  <1> ;ClusterBuffer_Valid: resb 1
105
                                  <1>
106
                                  <1> Dir BuffDescriptor:
                                  <1> DirBuff DRV: resb 1
107 000160AA <res 00000001>
108 000160AB <res 00000001>
                                  <1> DirBuff_FATType: resb 1
                                  <1> DirBuff_ValidData: resb 1
<1> DirBuff_CurrentEntry: resw 1
109 000160AC <res 00000001>
110 000160AD <res 00000002>
111 000160AF <res 00000002>
                                  <1> DirBuff LastEntry: resw 1
                                  <1> DirBuff_Cluster: resd 1
112 000160B1 <res 00000004>
113 000160B5 <res 00000002>
                                  <1> DirBuffer_Size: resw 1
                                  <1> ;DirBuff_EntryCounter: resw 1
114
115
                                  <1>
116
                                  <1> ; 01/02/2016
117
                                  <1>; these are on (real mode) segment 8000h and later
                                                       resb 1536 ; 3 sectors
118
                                  <1> ; FAT Buffer:
                                                       resb 512*32
119
                                  <1> ; Dir Buffer:
120
                                  <1> ; Logical_DOSDisks: resb 6656 ; 26 * 256 bytes
121
                                  <1> ; 18/01/2016
122
123
                                  <1>
                                  <1> FreeClusterCount: resd 1
124 000160B7 <res 00000004>
125
                                  <1>
126 000160BB <res 00000004>
                                  <1> VolSize_Unit1:
                                                      resd 1
                                  <1> VolSize_Unit2: resd 1
127 000160BF <res 00000004>
                                  <1>
129 000160C3 <res 00000004>
                                  <1> Vol_Tot_Sec_Str_Start:
                                                                   resd 1
130 000160C7 <res 0000000A>
                                  <1> Vol_Tot_Sec_Str: resb 10
131 000160D1 <res 00000001>
                                  <1> Vol_Tot_Sec_Str_End:
                                                                   resb 1
132 000160D2 <res 00000001>
                                  <1> resb 1
133 000160D3 <res 00000004>
                                  <1> Vol_Free_Sectors_Str_Start: resd 1
134 000160D7 <res 0000000A>
                                  <1> Vol Free Sectors Str: resb 10
                                  <1> Vol_Free_Sectors_Str_End:
135 000160E1 <res 00000001>
                                  <1>
137
                                  <1>; 10/02/2016
138 000160E2 <res 00000001>
                                  <1> RUN_CDRV: resb 1 ; CMD_INTR.ASM ; 09/11/2011
139
```

<1> Current\_Directory: resb 90

35 0001598A <res 0000005A>

```
<1>; 24/01/2016
141 000160E3 <res 00000080>
                                 <1> PATH Array:
                                                     resb 128 ; DIR.ASM ; 09/10/2011
142
                                 <1> ; 06/02/2016
143 00016163 <res 00000004>
                                 <1> CCD_DriveDT: resd 1 ; DIR.ASM ; (word)
144 00016167 <res 00000001>
                                 <1> CCD Level: resb 1 ; DIR.ASM
145 00016168 <res 00000001>
                                 <1> Last Dir Level:
                                                       resb 1 ; DIR.ASM
146
                                 <1>;
147 00016169 <res 00000002>
                                 <1> CDLF AttributesMask: resw 1 ; DIR.ASM
148 0001616B <res 00000004>
                                 <1> CDLF FNAddress: resd 1 ; DIR.ASM (word)
149 0001616F <res 00000002>
                                 <1> CDLF_DEType: resw 1 ; DIR.ASM
                                 <1> ;
150
151 00016171 <res 00000001>
                                  <1> CD COMMAND: resb 1 ; DIR.ASM
152
                                 <1>
153 00016172 <res 00000002>
                                 <1> alignb 4
154
                                 <1>
                                 <1>; 29/01/2016
155
156 00016174 <res 00000001>
                                 <1> Program_Exit:
                                                        resb 1 ; CMD INTR.ASM ; 09/11/2011
157
                                 <1>
                                 <1> ;alignb 4
158
159
                                 <1>; 23/02/2016
160 00016175 <res 00000001>
                                 <1> disk_rw_op: resb 1 ; 0 = disk read, 1 = disk write
                                 <1>; disk rw spt:
161
                                                        resb 1 ; sectors per track (<= 63) /// (<256)
                                 <1>; 31/\overline{0}1/\overline{2}016
162
163 00016176 <res 00000001>
                                 <1> retry_count:
                                                        resb 1 ; DISK_IO.ASM ; 20/07/2011 (CHS_RetryCount)
164 00016177 <res 00000001>
                                 <1> disk_rw_err:
                                                        resb 1 ; DISK_IO.ASM ; (Disk_IO_err_code)
165 00016178 <res 00000004>
                                 <1> sector_count:
                                                        resd 1 ; DISK_IO.ASM ; (Disk_RW_SectorCount)
166
                                 <1>
167
                                 <1>; 06/02/2016 (long name)
168 0001617C <res 00000002>
                                 <1> FDE_AttrMask:
                                                           resw 1 ; DIR.ASM
169 0001617E <res 00000002>
                                 <1> AmbiguousFileName: resw 1 ; DIR.ASM
170 00016180 <res 00000001>
                                 <1> PreviousAttr:
                                                           resb 1 ; DIR.ASM
                                 <1>;
171
172 00016181 <res 00000001>
                                 <1> LongNameFound: resb 1
                                                                 ; DIR.ASM
173 00016182 <res 00000001>
                                 <1> LFN_EntryLength: resb 1 ; DIR.ASM
                                                      resb 1 ; DIR.ASM
174 00016183 <res 00000001>
                                 <1> LFN CheckSum:
175 00016184 <res 00000084>
                                 <1> LongFileName:
                                                      resb 132 ; DIR.ASM
176
                                 <1>
177
                                 <1> ; PATH_Array_Ptr: resw 1 ; DIR.ASM
                                  <1> PATH CDLevel:
178 00016208 <res 00000001>
                                                     resb 1 ; DIR.ASM
179 00016209 <res 00000001>
                                 <1> PATH_Level: resb 1 ; DIR.ASM
180
                                 <1>
181
                                 <1>; 07/02/2016
                                 <1> Dir_File_Name:
182 0001620A <res 0000000D>
                                                        resb 13 ; DIR.ASM ; 09/10/2011
                                 <1>
183
                                 <1>; 10/02/2016
184
                                 <1> Dir_Entry_Name:
185 00016217 <res 0000000D>
                                                        resb 13 ; DIR.ASM
186
                                 <1>
                                 <1> alignb 2
187
188
                                 <1>
                                 <1> AttributesMask: resw 1 ; CMD INTR.ASM ; 09/11/2011
189 00016224 <res 00000002>
190
                                 <1>
191
                                 <1>; 10/02/2016 (128 bytes -> 126 bytes)
                                 <1> ; 08/02/2016
192
193
                                 <1> ;FFF Structure (128 bytes) ; DIR.ASM ; 09/10/2011
194 00016226 <res 00000001>
                                 <1> FindFile_Drv:
                                                                 resb 1
195 00016227 <res 00000041>
                                 <1> FindFile_Directory: resb 65
196 00016268 <res 0000000D>
                                 <1> FindFile Name:
                                                                resb 13
                                 <1> FindFile_LongNameEntryLength:
197
198 00016275 <res 00000001>
                                  <1> FindFile_LongNameYes:
                                                                 resb 1 ; Sign for longname procedures
                                 <1>; Above 8\overline{0} bytes form
199
200
                                 <1> ;TR-DOS Source/Destination File FullName Format/Structure
201 00016276 <res 00000002>
                                 <1> FindFile AttributesMask: resw 1
                                 <1> FindFile_DirEntry: resb 32
202 00016278 <res 00000020>
203 00016298 <res 00000004>
                                 <1> FindFile_DirFirstCluster: resd 1
                                 <1> FindFile_DirCluster:
204 0001629C <res 00000004>
                                                                resd 1
205 000162A0 <res 00000002>
                                  <1> FindFile_DirEntryNumber: resw 1
206 000162A2 <res 00000002>
                                 <1> FindFile MatchCounter:
                                                                resw 1
                                 <1> FindFile_Reserved: resw 1 ; 06/03/2016
207 000162A4 <res 00000002>
                                 <1>
209 000162A6 <res 00000004>
                                 <1> First Path Pos: resd 1 ; DIR.ASM ; 09/10/2011
                                 <1> Last Slash Pos: resd 1 ; DIR.ASM
210 000162AA <res 00000004>
211
                                 <1>
                                 <1>; 10/02/2016
212
                                 <1> File Count:
213 000162AE <res 00000002>
                                                     resw 1
                                                               ; DIR.ASM ; 09/10/2011
214 000162B0 <res 00000002>
                                                     resw 1
                                 <1> Dir_Count:
215 000162B2 <res 00000004>
                                 <1> Total_FSize:
                                                      resd 1
216 000162B6 <res 00000004>
                                 <1> TFS_Dec_Begin:
                                                     resd 1
217 000162BA <res 0000000A>
                                 <1>
                                                      resb 10
218 000162C4 <res 00000001>
                                  <1> TFS Dec End:
219
                                 <1>
220 000162C5 <res 00000001>
                                 <1> PrintDir_RowCounter: resb 1
                                  <1>
222 000162C6 <res 00000002>
                                 <1> alignb 4
                                  <1> ; 15/02/2015 ('show' command variables)
223
224 000162C8 <res 00000004>
                                 <1> Show FDT: resd 1
225 000162CC <res 00000004>
                                 <1> Show LDDDT: resd 1
                                 <1> Show Cluster:
226 000162D0 <res 00000004>
                                                        resd 1
227 000162D4 <res 00000004>
                                 <1> Show_FileSize:
                                                       resd 1
228 000162D8 <res 00000004>
                                 <1> Show_FilePointer: resd 1
229 000162DC <res 00000002>
                                 <1> Show ClusterPointer: resw 1
230 000162DE <res 00000002>
                                 <1> Show ClusterSize: resw 1
231 000162E0 <res 00000001>
                                 <1> Show_RowCount:
                                                      resb 1
232
                                 <1>
                                 <1> alignb 4
233 000162E1 <res 00000003>
234
                                 <1> ; 21/02/2016
235 000162E4 <res 00000004>
                                 <1> DelFile_FNPointer: resd 1 ; ; CMD_INTR.ASM (word) ; 09/11/2011
236
                                 <1> ; 27/02/2016
                                 <1>; DIR.ASM (09/10/2011)
237
238 000162E8 <res 00000004>
                                 <1> DelFile_FCluster: resd 1
239 000162EC <res 00000002>
                                 <1> DelFile EntryCounter: resw 1
                                 <1> DelFile LNEL:
240 000162EE <res 00000001>
                                                               resb 1
241 000162EF <res 00000001>
                                 <1> resb 1
242
                                 <1>
                                 <1> ; DIR.ASM
243
244 000162F0 <res 00000004>
                                 <1> mkdir_DirName_Offset:
                                                               resd 1
```

140

```
245 000162F4 <res 00000004>
                                 <1> mkdir_FFCluster: resd 1
246 000162F8 <res 00000004>
                                 <1> mkdir LastDirCluster:
                                                              resd 1
247 000162FC <res 00000004>
                                 <1> mkdir_FreeSectors: resd 1
248 00016300 <res 00000002>
                                 <1> mkdir_attrib:
                                 <1> mkdir_SecPerClust: resb 1
249 00016302 <res 00000001>
250 00016303 <res 00000001>
                                 <1> mkdir_add_new_cluster: resb 1
251 00016304 <res 0000000D>
                                 <1> mkdir Name:
                                                       resb 13
                                 <1> resw <math>\overline{1}; 01/03/2016
252 00016311 <res 00000002>
253
                                 <1>; 27/02/2016
                                 <1> RmDir MultiClusters:
254 00016313 <res 00000001>
                                                               resb 1
255 00016314 <res 00000004>
                                 <1> RmDir_DirEntryOffset:
                                                               resd 1 ; 01/03/2016 (word -> dword)
256 00016318 <res 00000004>
                                 <1> RmDir_ParentDirCluster: resd 1
                                 <1> RmDir DirLastCluster: resd 1
257 0001631C <res 00000004>
                                 <1> RmDir PreviousCluster: resd 1
258 00016320 <res 00000004>
259
                                 <1>; 22/02/2016
                                 <1> UPDLMDT CDirLevel: resb 1
260 00016324 <res 00000001>
261 00016325 <res 00000004>
                                 <1> UPDLMDT CDirFCluster:
                                                               resd 1
                                 <1>
262
263 00016329 <res 00000003>
                                  <1> alignb 4
                                 <1> ; DRV FAT.ASM ; 21/08/2011
264
265 0001632C <res 00000004>
                                 <1> gffc_next_free_cluster: resd 1
266 00016330 <res 00000004>
                                  <1> gffc_first_free_cluster: resd 1
267 00016334 <res 00000004>
                                 <1> gffc_last_free_cluster: resd 1
                                 <1> ;29/04/2016
269
                                  <1> Cluster_Index: ; resd 1
270
                                 <1> ; 22/02/2016
272 00016338 <res 00000004>
                                 <1> ClusterValue:
                                                        resd 1
273
                                 <1>; 04/03/2016
274 0001633C <res 00000001>
                                 <1> Attributes: resb 1
275
                                 <1> ;; CFS_error: resb 1 ;; 01/03/2016
276 0001633D <res 00000001>
                                 <1> resb 1
277 0001633E <res 00000001>
                                 <1> CFS_OPType: resb 1
278 0001633F <res 00000001>
                                 <1> CFS_Drv:
                                                resb 1
                                 <1> CFS_CC:
279 00016340 <res 00000004>
                                                      resd 1
280 00016344 <res 00000004>
                                 <1> CFS_FAT32FSINFOSEC: resd 1
281 00016348 <res 00000004>
                                 <1> CFS FAT32FC: resd 1
282
                                 <1>
                                  <1> ; 27/02/2016
283
284
                                 <1> ;alignb 4
285 0001634C <res 00000004>
                                 <1> glc_prevcluster: resd 1 ; DRV_FAT.ASM (21/08/2011)
                                  <1>; 2\overline{2}/10/2016
286
287 00016350 <res 00000004>
                                 <1> glc_index: resd 1 ; Last Cluster Index (22/10/2016)
288
                                 <1>
                                 <1> ; DIR.ASM
289
290 00016354 <res 00000002>
                                  <1> DLN_EntryNumber: resw 1
                                 <1> DLN 40h:
291 00016356 <res 00000001>
                                                  resb 1
                                 <1>; 28/02/2016
292
293 00016357 <res 00000001>
                                 <1> TCC_FATErr: resb 1 ; DRV_FAT.ASM
294
                                 <1>
295
                                 <1> alignb 4
296
                                 <1>; DIR.ASM (09/10/2011)
297 00016358 <res 00000002>
                                 <1> LCDE_EntryIndex: resw 1 ; LCDE_EntryOffset
298 0001635A <res 00000002>
                                 <1> LCDE_ClusterSN: resw 1
299 0001635C <res 00000004>
                                 <1> LCDE_Cluster:
                                                         resd 1
300 00016360 <res 00000004>
                                 <1> LCDE_ByteOffset: resd 1
301
                                 <1>
302
                                 <1> ;alignb4
303
                                  <1>; 06/03/2016 (word -> dword)
                                 <1>; CMD_INTR.ASM (01/08/2010)
304
305 00016364 <res 00000004>
                                 <1> SourceFilePath:
306 00016368 <res 00000004>
                                 <1> DestinationFilePath: resd 1
307
                                 <1>
308
                                 <1> ;alignb 4
309
                                  <1>; 06/03/2016
310
                                  <1>; FILE.ASM (09/10/2011)
                                 <1> ;Source File Structure (same with 'Find File' Structure)
                                 <1> SourceFile_Drv:
312 0001636C <res 00000001>
                                                                      resb 1
313 0001636D <res 00000041>
                                 <1> SourceFile_Directory:
                                                                      resb 65
314 000163AE <res 0000000D>
                                 <1> SourceFile_Name:
                                                               resb 13
                                 <1> SourceFile LongNameEntryLength:
315
316 000163BB <res 00000001>
                                 <1> SourceFile_LongNameYes:
                                                                     resb 1 ; Sign for longname procedures
                                 <1>; Above 80 bytes
317
318
                                 <1> ;is TR-DOS Source File FullName Format/Structure
319 000163BC <res 00000002>
                                 <1> SourceFile_AttributesMask:
                                                                  resw 1
320 000163BE <res 00000020>
                                 <1> SourceFile_DirEntry:
                                                                      resb 32
321 000163DE <res 00000004>
                                 <1> SourceFile DirFirstCluster:
                                                                     resd 1
322 000163E2 <res 00000004>
                                 <1> SourceFile_DirCluster:
                                                                      resd 1
323 000163E6 <res 00000002>
                                  <1> SourceFile_DirEntryNumber:
                                                                      resw 1
                                 <1> SourceFile MatchCounter: resw 1
324 000163E8 <res 00000002>
325
                                 <1> ; 16/03/2016
326 000163EA <res 00000001>
                                  <1> SourceFile SecPerClust:
                                                                      resb 1
327 000163EB <res 00000001>
                                                                      resb 1
                                  <1> SourceFile_Reserved:
                                  <1> ; Above is 128 bytes
329
                                 <1>
                                 <1> ;Destination File Structure (same with 'Find File' Structure)
330
331 000163EC <res 00000001>
                                  <1> DestinationFile Drv:
                                                                     resb 1
332 000163ED <res 00000041>
                                 <1> DestinationFile_Directory:
                                                                      resb 65
                                 <1> DestinationFile_Name:
333 0001642E <res 0000000D>
                                                                      resb 13
                                 <1> DestinationFile LongNameEntryLength:
334
335 0001643B <res 00000001>
                                 <1> DestinationFile_LongNameYes: resb 1 ; Sign for longname procedures
                                  <1> ; Above 80 bytes
336
                                 <1> ;is TR-DOS Destination File FullName Format/Structure
337
338 0001643C <res 00000002>
                                 <1> DestinationFile_AttributesMask: resw 1
                                 <1> DestinationFile DirEntry: resb 32
339 0001643E <res 00000020>
                                 <1> DestinationFile DirFirstCluster: resd 1
340 0001645E <res 00000004>
341 00016462 <res 00000004>
                                 <1> DestinationFile DirCluster: resd 1
342 00016466 <res 00000002>
                                 <1> DestinationFile_DirEntryNumber: resw 1
                                 <1> DestinationFile MatchCounter: resw 1
343 00016468 <res 00000002>
                                 <1> ; 16/03/2016
344
                                                                     resb 1
                                 <1> DestinationFile SecPerClust:
345 0001646A <res 00000001>
346 0001646B <res 00000001>
                                 <1> DestinationFile Reserved: resb 1
347
                                 <1> ; Above is 128 bytes
348
                                 <1>
349
                                 <1>; 24/04/2016
```

```
<1> resw 1
                                 <1>
                                 <1>; 10/03/2016
352
353
                                 <1> ; FILE.ASM
354 0001646E <res 00000001>
                                 <1> move cmd phase:
                                                          resb 1
                                 <1> msftdf sf df drv: resb 1
355 0001646F <res 00000001>
356 00016470 <res 00000004>
                                 <1> msftdf drv offset: resd 1
357
                                 <1>
358
                                 <1> ; 11/03/2016
359
                                 <1>; DRV FAT.ASM (21/08/2011)
360 00016474 <res 00000004>
                                 <1> FAT_anc_LCluster: resd 1
361 00016478 <res 00000004>
                                 <1> FAT_anc_FFCluster: resd 1
362
                                 <1>
363
                                 <1> ;alignb 4
364
                                 <1>
                                 <1> ; 14/03/2016
365
366
                                 <1>; TRDOS 386 = TRDOS v2.0 feature only !
                                 <1> ; 'allocate_memory_block' in 'memory.s'
367
                                                        resd 1 ; page count (for contiguous allocation)
368 0001647C <res 00000004>
                                 <1> mem ipg count:
369 00016480 <res 00000004>
                                                       resd 1 ; page count (for count down)
                                 <1> mem_pg_count:
370 00016484 <res 00000004>
                                 <1> mem_aperture:
                                                      resd 1 ; contiguous free pages (current)
371 00016488 <res 00000004>
                                 <1> mem max aperture: resd 1 ; maximum value of contiguous free pages
372 0001648C <res 00000004>
                                 <1> mem_pg_pos: resd 1 ; mem. position (page #) of current aperture
373 00016490 <res 00000004>
                                 <1> mem_max_pg_pos: resd 1 ; mem. position (page #) of max. aperture
374
                                 <1>
375
                                 <1> ; 15/03/2016
                                 <1> ; FILE.ASM ('copy_source_file_to_destination_file')
377 00016494 <res 00000001>
                                 <1> copy_cmd_phase:
                                                           resb 1
378 00016495 <res 00000001>
                                 <1> csftdf_rw_err:
                                                             resb 1
379 00016496 <res 00000001>
                                 <1> DestinationFileFound: resb 1
380 00016497 <res 00000001>
                                 <1> csftdf_cdrv:
381 00016498 <res 00000004>
                                 <1> csftdf_filesize:
                                                           resd 1
                                 <1>; TRDO\overline{S}386 (TRDOS v2.0)
382
383 0001649C <res 00000004>
                                 <1> csftdf sf mem addr: resd 1
384 000164A0 <res 00000004>
                                 <1> csftdf_sf_mem_bsize: resd 1
385
                                 <1>;
                                 <1>
387 000164A4 <res 00000004>
                                 <1> csftdf_sf_cluster:
                                                           resd 1 ; 16/03/2016
388 000164A8 <res 00000004>
                                 <1> csftdf_df_cluster:
                                                           resd 1
                                 <1>; 16/0\overline{3}/2\overline{0}16
390 000164AC <res 00000004>
                                 <1> csftdf_r_size:
                                                           resd 1
                                 <1> csftdf_w_size:
<1> csftdf_sf_rbytes:
                                                           resd 1
391 000164B0 <res 00000004>
392 000164B4 <res 00000004>
                                                           resd 1
393 000164B8 <res 00000004>
                                 <1> csftdf_df_wbytes:
                                                           resd 1
394 000164BC <res 00000001>
                                 <1> csftdf_percentage:
                                                           resb 1
395
                                 <1> ; 17/03/2016
                                                           resb 1
396 000164BD <res 00000001>
                                 <1> csftdf videopage:
397 000164BE <res 00000002>
                                 <1> csftdf_cursorpos:
                                                           resw 1
398 000164C0 <res 00000004>
                                 <1> csftdf_sf_drv_dt:
                                                           resd 1
399 000164C4 <res 00000004>
                                 <1> csftdf_df_drv_dt:
                                                           resd 1
400
                                 <1>
401
                                 <1> ; 21/03/2016
                                 <1>; 20/03/2016
402
403
                                 <1> ; FILE.ASM
404 000164C8 <res 00000004>
                                 <1> createfile Name Offset: resd 1
405 000164CC <res 00000004>
                                 <1> createfile_FreeSectors: resd 1
406 000164D0 <res 00000004>
                                 <1> createfile size:
                                                             resd 1
                                 <1> createfile_FFCluster: resd 1 ; 11/03/2016
407 000164D4 <res 00000004>
408 000164D8 <res 00000004>
                                 <1> createfile_LastDirCluster: resd 1
409 000164DC <res 00000004>
                                 <1> createfile_Cluster: resd 1
410 000164E0 <res 00000004>
                                 <1> createfile_PCluster:
                                 <1> createfile_attrib: resb 1
<1> createfile_SecPerClust: resb 1
411 000164E4 <res 00000001>
412 000164E5 <res 00000001>
413 000164E6 <res 00000002>
                                 <1> createfile_DirIndex:
414 000164E8 <res 00000004>
                                 <1> createfile_CCount: resd 1
415 000164EC <res 00000002>
                                 <1> createfile_BytesPerSec: resw 1 ; 23/03/2016
416 000164EE <res 00000001>
                                 <1> createfile wfc:
                                                                resb 1
                                 <1> createfile_UpdatePDir:
417 000164EF <res 00000001>
                                                               resb 1 ; 31/03/2016
418
                                 <1>
419
                                 <1>; alignb 4
420
                                 <1>
                                 <1> ; 11/04/2016
421
422 000164F0 <res 00000002>
                                 <1> env_var_length:
                                                           resw 1
423
                                 <1>
424 000164F2 <res 00000002>
                                 <1> alignb 4
425
                                 <1>
426
                                 <1> ; 25/04/2016
427 000164F4 <res 00000001>
                                 <1> readi.valid: resb 1 ; valid data (>0 = valid for readi)
                                 <1> readi.drv: resb 1 ; drive number (0, 1,2,3,4..)
428 000164F5 <res 00000001>
429 000164F6 <res 00000001>
                                 <1> readi.spc: resb 1 ; sectors per cluster for 'readi' drive
430 000164F7 <res 00000001>
                                 <1> readi.s_index: resb 1 ; sector index in current cluster (buffer)
                                 <1> readi.sector:
431 000164F8 <res 00000004>
                                                        resd 1 ; current disk sector
432 000164FC <res 00000002>
                                 <1> readi.bpc: resw 1 ; bytes per cluster - 1
433 000164FE <res 00000002>
                                 434 00016500 <res 00000004>
                                 <1> readi.cluster: resd 1 ; current cluster number
435 00016504 <res 00000004>
                                 <1> readi.c index:
                                                       resd 1; cluster index of the current cluster (0,1,2,3...)
436 00016508 <res 00000004>
                                 <1> readi.fclust:
                                                       resd 1 ; first cluster of the current cluster
437 0001650C <res 00000004>
                                 <1> readi.fs_index: resd 1 ; sector index in disk/file section (for Singlix FS)
438
                                                     resd 1 ; readi sector buffer address
                                 <1> ; readi.buffer:
439
                                 <1>
                                 <1>; alignb 4
440
441
                                 <1>
442 00016510 <res 00000001>
                                 <1> writei.valid:
                                                        resb 1 ; valid data (>0 = valid for writei)
443 00016511 <res 00000001>
                                 <1> writei.drv: resb 1 ; drive number (0, 1,2,3,4..)
444 00016512 <res 00000001>
                                 <1> writei.spc: resb 1 ; sectors per cluster for 'writei' drive
445 00016513 <res 00000001>
                                 <1> writei.s_index: resb 1 ; sector index in current cluster (buffer)
446 00016514 <res 00000004>
                                 <1> writei.sector: resd 1 ; current disk sector
                                 <1> writei.bpc: resw 1 ; bytes per cluster - 1 \,
447 00016518 <res 00000002>
448 0001651A <res 00000002>
                                 <1> writei.offset:
                                                      resw 1 ; byte offset in cluster buffer
449 0001651C <res 00000004>
                                 <1> writei.cluster: resd 1 ; current cluster number
450 00016520 <res 00000004>
                                 <1> writei.c_index: resd 1 ; cluster index of the current cluster (0,1,2,3..)
451 00016524 <res 00000004>
                                 <1> writei.fclust: resd 1 ; first cluster of the current cluster
                                 <1> writei.fs index: resd 1 ; sector index in disk/file section (for Singlix FS)
452 00016528 <res 00000004>
                                 <1> ;writei.buffer: resd 1 ; writei sector buffer address
453
454 0001652C <res 00000004>
                                 <1> writei.lclust: resd 1 ; writei last cluster (mget w) ; 23/10/2016
```

350 0001646C <res 00000002>

```
<1> writei.l_index:
                                                      resd 1 ; writei last cluster index (mget_w) ; 23/10/2016
456 00016534 <res 00000001>
                                 <1> writei.ofn: resb 1; open file number (to be written); 23/10/2016
457
                                 <1>
458 00016535 <res 00000003>
                                 <1> alignb 4
459
                                 <1>
460
                                 <1>; 29/04/2016
461 00016538 <res 00000004>
                                 <1> Run CDirFC: resd 1
462 0001653C <res 00000001>
                                 <1> Run_Auto_Path:
                                                       resb 1
463 0001653D <res 00000001>
                                 <1> Run_Manual_Path: resb 1 ; 0 -> auto path sequence needed
464 0001653E <res 00000001>
                                 <1> EXE ID:
                                                  resb 1
465 0001653F <res 00000001>
                                 <1> EXE_dot:
                                                  resb 1
466
                                 <1>
467
                                 <1>; 06/05/2016
468 00016540 <res 00000004>
                                 <1> mainprog_return_addr: resd 1
469 00016544 <res 00000004>
                                 <1> last_error: resd 1 ; this will be used to return error code to MainProg
<1> ; 'lasterror' keyword will be used later to get the
470
                                                        ; last error code/number/status.
471
                                 <1>
                                 <1> ; 12/05/2016
472
                                 <1> video eax: resd 1 ; eax return value of video function
473 00016548 <res 00000004>
474
                                 <1>
475
                                 <1>; 01/06/2016
476 0001654C <res 00000004>
                                 <1> user buffer: resd 1 ; 'diskio.s' (INT 33h, Function 08h, floppy disk type)
477
                                 <1>
478
                                 <1> ; 21/05/2016 - TRDOS 386 ('swap/switch', 'rswap', [u.pri])
                                 <1> priority: resb 1 ; running priority level of process (0,1,2)
479 00016550 <res 00000001>
480
                                                         ; (run queue which is process comes from)
                                 <1> ; 22/05/2016 - TRDOS 386 ('set_run_sequence', 'rtc_int', 'u_timer')
481
482 00016551 <res 00000001>
                                 <1> p_change: resb 1 ; process change status (for timer events)
                                  <1>; 23/05/2016 - TRDOS 386 ('clock')
483
                                 <1> multi_tasking: resb 1 ; Multi Tasking status (0 = disabled, >0 = enabled)
484 00016552 <res 00000001>
485
                                 <1>
                                                        ; (EBX will return with user buffer addr or disk type)
486
                                 <1> ; 07/06/2016
                                 <1> timer_events:
                                                        resb 1 ; number of (active) timer events, <= 16
487 00016553 <res 00000001>
488
                                 <1>
489
                                 <1> : 24/06/2016
490 00016554 <res 00000001>
                                 <1> w_str_cmd: resb 1; WRITE_STRING command (0,1,2,3); video.s
491 00016555 <res 00000001>
                                 <1> p crt mode: resb 1 ; previous video mode (=3 or 0), backup mark/sign
                                 <1> ; 26/06/2016
492
493 00016556 <res 00000001>
                                  <1> p_crt_page: resb 1 ; previous active page (for 'set_mode')
                                 <1>; 04/07/2016
494
495 00016557 <res 00000001>
                                 <1> noclearmem: resb 1 ; if set, 'SET MODE' (INT 31h) function (AH = 4)
                                                        ; will not clear the video memory
496
                                 <1>
497
                                                        ; (usable for graphics modes only)
                                 <1>
                                 <1> alignb 2
498
499 00016558 <res 00000002>
                                 <1> CRT_LEN:
                                                resw 1 ; length of regen buffer in bytes
500 0001655A <res 00000010>
                                 <1> cursor_pposn:
                                                        resw 8 ; cursor positions backup
                                 <1>
                                 <1>; 10/07/2016 ('VGA_FONT_SETUP', INT 43H address for x86 real mode bios)
502
                                 <1> VGA INT43H: resd 1; 0 = default (not configured by user)
503 0001656A <res 00000004>
                                                        ; OFFFFFFFF = user defined fonts
504
                                 <1>
505
                                 <1>
                                                        ; address:
506
                                 <1>
                                                               vgafont8
                                                        ;
507
                                 <1>
                                                               vgafont16
                                                        ;
                                                               vgafont14
508
                                 <1>
509
                                 <1>
510
                                 <1>; 25/07/2016
511 0001656E <res 00000001>
                                 <1> VGA_MTYPE: resb 1 ; 0=CTEXT,1=MTEXT,2=CGA,3=PLANAR1,4=PLANAR4,5=LINEAR
512
                                 <1>
513
                                 <1>; 23/10/2016
514 0001656F <res 00000001>
                                 <1> setfmod
                                                        resb 1; update last modification date&time sign (if >0)
515
                                 <1>
                                                        ; (it is Open File Number + 1, if > 0)
516
                                 <1> alignb 4
517
                                 <1>
                                 <1> ; 16/10/2016
518
                                 <1> FFF_UBuffer: resd 1 \, ; User's buffer address for FFF & FNF system calls
519 00016570 <res 00000004>
                                 <1> ; 15/10/2016
520
521 00016574 <res 00000001>
                                 <1> FFF Valid: resb 1 ; Find First File Structure validation byte
522
                                 <1>
                                                        ; 0 = invalid (Find Next File can't use FFF struct)
                                                        ; >0 = valid, return type for FFF and Find Next File
523
                                 <1>
524
                                 <1>
                                                        ; 24 = basic parameters, 24 bytes
525
                                 <1>
                                                         ; 128 = entire FFF structure/table, 128 bytes
526
                                 <1>; 16/10/2016 (FFF Attrib: resw 1)
527 00016575 <res 00000001>
                                 <1> FFF_Attrib: resb 1; Find First File attributes for Find Next File (LB)
528 00016576 <res 00000001>
                                 <1> FFF_RType: resb 1 ; FFF return type (0 = Basic, >0 = complete) (HB)
                                 <1>; 1\overline{6}/10/2016 - 05/10/2016 (Set Working Path)
529
                                                        resb 1; Set Working Path - Invalid File Name
530 00016577 <res 00000001>
                                 <1> SWP_inv_fname:
531 00016578 <res 00000002>
                                 <1> SWP Mode: resw 1; Set Working Path - Mode
532 0001657A <res 00000001>
                                 <1> SWP_DRV:
                                                 resb 1; Set Working Path - Drive
533 0001657B <res 00000001>
                                 <1> SWP_DRV_chg: resb 1; Set Working Path - Drive Change
534
                                 <1>
                                 <1> ; 27/02/2017
535
536 0001657C <res 00000001>
                                  <1> fpready: resb 1; '80387 fpu is ready' flag
537
                                 <1>
                                 <1>; 08/10/2016
538
539 0001657D <res 00000009>
                                 <1> device name:
                                                     resb 9 ; capitalized (and zero padded) device canem
                                                       ; (example: "TTY0",0,0,0,0,0")
540
                                 <1>
                                 <1>
542 00016586 <res 00000002>
                                 <1> alignb 4
543
                                 <1>
                                 <1>; 08/10/2016
544
545
                                 <1>; 07/10/2016
546
                                 <1>; Table of kernel devices (which do not use installable device drivers)
547
                                 <1>; has been coded into KERNEL (trdosk9.s)
                                 <1> : 07/10/2016
548
549
                                 <1>; 8 installable device drivers available to install (NUMIDEV)
550 00016588 <res 00000020>
                                 <1> IDEV_PGDIR: resd NUMIDEV
551
                                 <1>
                                                        ; Page directories of installable device drivers
552
                                 <1>
553
                                 <1>
                                                        ; Note: Virtual start address is always 400000h
554
                                 <1>
                                                        ; (end of the 1st 4MB). [org 400000h]
555
                                 <1>
                                                        ; Segments: KCODE, KDATA
556
                                 <1>
                                                        ; Method: call 400000h (after changing page dir)
557
                                 <1>
                                                        ; Query code located at the start (400000h).
558
                                 <1>
                                                        ; Query code returns with
559
                                 <1>
                                                         ; eax = device type and driver version
```

455 00016530 <res 00000004>

```
560
                                 <1>
                                                                   AL = Device Type minor
561
                                  <1>
                                                                   AH = Device Type major
                                                         ;
                                                                   Byte 16-23 : Version minor
562
                                 <1>
563
                                 <1>
                                                                  Byte 24-31: Version major - 1
                                                                             (0:0 -> 1.0)
564
                                 <1>
565
                                 <1>
                                                             ebx = initialization code address
566
                                 <1>
                                                             ecx = configuration table address
567
                                 <1>
                                                             edx = description table address
568
                                 <1>
                                                             esi = device (default) name address (ASCIIZ)
                                                         ;
                                                                (name has "/DEV/" prefix)
569
                                 <1>
                                                         ;
570
                                 <1>
                                                             edi = dispatch table address
571
                                 <1>
                                                                 (for calling kernel-device functions)
                                                         ;
572
                                 <1>
                                                             ebp = address table address
                                                         ; Initialization code returns with
573
                                 <1>
574
                                 <1>
                                                             eax = open code address
                                                         ;
                                                             ecx = close code address
575
                                 <1>
                                                            ebx = read code address
576
                                 <1>
                                                         ;
577
                                                             edx = write code address
                                 <1>
578
                                 <1>
                                                             esi = IOCTL code address
                                                             edi = dispatch table address
579
                                 <1>
                                                         ;
580
                                 <1>
                                                             ebp = address table address
                                                         ; Address Table:
581
                                  <1>
                                                              Offset 0 : open code address
582
                                 <1>
583
                                 <1>
                                                              Offset 4 : read code address
584
                                 <1>
                                                              Offset 8 : write code address
585
                                 <1>
                                                              Offset 12 : close code address
586
                                 <1>
                                                              Offset 16 : IOCTL code address
                                                         ;
                                                              Offset 20 : initialization code address
587
                                 <1>
                                                              Offset 24 : description table address
588
                                 <1>
                                                         ;
                                                              Offset 28 : configuration table address
589
                                 <1>
                                                         ;
590
                                 <1>
                                                              Offset 32 : device name address
591
                                  <1>
                                                              Offset 36 : dispatch table address
                                                         ;
592
                                 <1>
                                                                    (for calling kernel-device functions)
593
                                 <1>
594 000165A8 <res 00000040>
                                 <1> IDEV_NAME: resb 8*NUMIDEV
595
                                 <1>
                                                         ; 8 byte names of installable device drivers
                                 <1>
597 000165E8 <res 00000008>
                                 <1> IDEV_TYPE: resb NUMIDEV ; Driver type of installable device drivers
598 000165F0 <res 00000008>
                                 <1> IDEV_FLAGS: resb NUMIDEV ; Device access parameters for installable
                                                              ; device drivers (These values are set while
                                 <1>
600
                                 <1>
                                                          ; the device driver is being loaded.)
601 000165F8 <res 00000020>
                                 <1> IDEV OADDR: resd NUMIDEV; open function addr for installable dev driver
602 00016618 <res 00000020>
                                 <1> IDEV_CADDR: resd NUMIDEV ; close function addr for installable dev driver
603 00016638 <res 00000020>
                                 <1> IDEV_RADDR: resd NUMIDEV ; read function addr for installable dev driver
604 00016658 <res 00000020>
                                 <1> IDEV WADDR: resd NUMIDEV ; write function addr for installable dev driver
605
                                 <1>
606
                                 <1>; 08/10/2016
607
                                 <1>; 07/10/2016
608
                                 <1> ; Device Open and Access parameters
                                 <1> DEV_ACCESS: resb NUMOFDEVICES ; bit 0 = accessable by normal users
609 00016678 <res 0000001E>
610
                                 <1>
                                                                    ; bit 1 = read access permission
611
                                  <1>
                                                                    ; bit 2 = write access permission
                                                                     ; bit 3 = IOCTL permission to users
612
                                 <1>
613
                                 <1>
                                                                     ; bit 4 = block device if it is set
614
                                 <1>
                                                                     ; bit 5 = 16 bit or 1024 byte data
615
                                 <1>
                                                                     ; bit 6 = 32 bit or 2048 byte data
                                                                     ; bit 7 = installable device driver
                                 <1>
                                 <1> DEV_R_OWNER: resb NUMOFDEVICES
617 00016696 <res 0000001E>
                                                                       ; Reading owner no (u.uid) of devices
618 000166B4 <res 0000001E>
                                 <1> DEV_R_OPENCOUNT: resb NUMOFDEVICES ; Reading open count
                                 <1> DEV W OWNER: resb NUMOFDEVICES ; Writing owner no (u.uid) of devices
619 000166D2 <res 0000001E>
620 000166F0 <res 0000001E>
                                 <1> DEV_W_OPENCOUNT: resb NUMOFDEVICES ; Writing open count
621 0001670E <res 0000001E>
                                 <1> DEV DRIVER: resb NUMOFDEVICES ; device driver number (1 to 7Fh)
                                                                    ; *if bit 7 is set (80 to FFh)
622
                                 <1>
623
                                 <1>
                                                                     ; *if it is installable device driver
                                                                     ; *index (0 to 7Fh)
624
                                 <1>
625
                                 <1>
                                                                     ; otherwise it is kernel device index
                                 <1> DEV OPENMODE:
626 0001672C <res 0000001E>
                                                          resb NUMOFDEVICES ; 1 = read mode
627
                                 <1>
                                                                     ; 2 = write mode
                                 <1>
                                                                     ; 3 = read & write
628
629
                                 <1>
                                                                    ; 0 = not open (free)
                                                       \verb"resd NUMOFDEVICES"; pointers to name addresses of drivers"
630 0001674A <res 00000078>
                                 <1> DEV NAME PTR:
631
                                  <1>
                                                                    ; Address base: KDEV_NAME+
                                                                     ; or IDEV_NAME+
632
                                 <1>
633 000167C2 <res 00000078>
                                 <1> DEV R POINTER:
                                                          resd NUMOFDEVICES ; reading pointer, writing pointer
                                                          resd NUMOFDEVICES ; sector number if block device
634 0001683A <res 00000078>
                                 <1> DEV_W_POINTER:
                                                                     ; character offset if char device
635
                                 <1>
636 000168B2 <res 00000002>
                                 <1> alignb 4
637
                                 <1>
638
                                  <1>; 06/10/2016
                                 <1> ; Open File Parameters
639
640 000168B4 <res 00000028>
                                 <1> OF_FCLUSTER: resd OPENFILES ; First clusters of open files
                                                 resb OPENFILES ; Logical DOS drive numbers of open files resb OPENFILES ; Open mode (1 = read, 2 = write, 3 = r&w)
641 000168DC <res 0000000A>
                                  <1> OF_DRIVE:
                                 <1> OF MODE:
642 000168E6 <res 0000000A>
643 000168F0 <res 0000000A>
                                  <1> OF STATUS: resb OPENFILES ; (bit 0 = read, bit 1 = write)
                                                       resb OPENFILES ; Open counts of open files
                                 <1> OF OPENCOUNT:
644 000168FA <res 0000000A>
645 00016904 <res 00000028>
                                 <1> OF_POINTER: resd OPENFILES
                                                                   ; File seek/read/write pointer
                                                 resd OPENFILES
646 0001692C <res 00000028>
                                 <1> OF SIZE:
                                                                      ; File sizes of open files (in bytes)
647 00016954 <res 00000028>
                                 <1> OF_DIRFCLUSTER: resd OPENFILES ; Directory First Clusters of open files
648 0001697C <res 00000028>
                                 <1> OF DIRCLUSTER:
                                                         resd OPENFILES ; Directory (Entry) Clusters of open files
649 000169A4 <res 00000028>
                                 <1> OF VOLUMEID: resd OPENFILES ; Vol ID for removable drives of open files
                                 <1> OF_CCLUSTER: resd OPENFILES ; Current clusters of open files
650 000169CC <res 00000028>
                                 <1> OF CCINDEX: resd OPENFILES ; Cluster index numbers of current clusters
651 000169F4 <res 00000028>
                                 <1> ; 24/10/2016
652
653 00016A1C <res 00000014>
                                 <1> OF_DIRENTRY: resw OPENFILES ; Directory entry index no. in dir cluster
                                                              ; Sector index = entry index / 16
654
                                 <1>
                                 <1> ;alignb 2
655
656
                                 <1>
657 00016A30 <res 00000060>
                                 <1> DTA:
                                                                      ; Find First File data transfer area
                                                  resd 24
658
                                 <1>
                                 <1> ; 19/12/2016
659
660 00016A90 <res 00000001>
                                                             ; Timer callback method flag for 'systimer'
                                 <1> tcallback: resb 1
                                                             ; Timer interrupt type flag for 'systimer'
661 00016A91 <res 00000001>
                                 <1> trtc:
                                                  resb 1
662
                                 <1> ; 20/02/2017
663 00016A92 <res 00000001>
                                 <1> no_page_swap:
                                                        resb 1 ; Swap lock for Signal Response Byte pages
                                 <1>;; 15/01/2017
664
```

```
<1> ; 02/01/2017
665
666
                                <1> ;;intflg: resb 1
                                                            ; software interrupt in progress signal
667
                                <1>
                                                             ; (for timer interrupt)
                                <1>
668
669 00016A93 <res 00000001>
                                <1> alignb 4
670
                                <1> ; 13/04/2017
671 00016A94 <res 0000001E>
                                <1> DEV INTR: resb NUMOFDEVICES ; Device Interrupt (IRQ) number + 1
                                                            ; (0= not available, 1= IRQ 0, 16= IRQ 15)
672
                                <1>
673 00016AB2 <res 00000040>
                                <1> DEV_INT_HNDLR:
                                                      resd 16
                                                                         ; Device Interrupt Handler addr, if > 0
674
                                <1>
675
                                <1>
676
                                <1> ;alignb 4
677
                                <1>
                                <1> ; 26/02/2017 ; IRQ Callback parameters ('syscalbac')
678
679
                                <1> ;Index: ; 0 to 8
                                         0 = IRQ3, 1 = IRQ4, 2 = IRQ5, 3 = IRQ7
680
                                <1>;
                                         4 = IRQ9, 5 = IRQ10, 6 = IRQ11, 7 = IRQ12, 8 = IRQ13
                                <1> IRQ.owner: resb 9
682 00016AF2 <res 00000009>
                                                            ; owner, 0 = free, >0 = [u.uno]
683 00016AFB <res 00000009>
                                <1> IRQ.dev:
                                               resb 9
                                                            ; 0 = default/kernel, >0 = device number
                                                                  ; 0 = Signal Response Byte, 1 = Callback
                                <1> IRQ.method: resb 9
684 00016B04 <res 00000009>
685 00016B0D <res 00000009>
                                <1> IRQ.srb: resb 9
                                                                   ; Signal Response/Return Byte value
686 00016B16 <res 00000024>
                                <1> IRQ.addr: resd 9
                                                            ; Rignal Response Byte address (physical)
687
                                <1>
                                                            ; or Callback service address (virtual)
                                <1> ; 28/02/2017
                                <1> IRQ_cr3: resd 1 ; for saving cr3 register in IRQ handler
689 00016B3A <res 00000004>
690 00016B3E <res 00000001>
                                <1> IRQnum:
                                                resb 1 ; IRQ number for IRQ handler (trdosk8.s)
                                <1>
692
                                <1> ; 10/04/2017
                                <1> ; 03/04/2017
693
                                <1>; UNINITIALIZED AUDIO DATA
694
695 00016B3F <res 00000001>
                                <1> alignb 4
696 00016B40 <res 00000001>
                                <1> audio_pci: resb 1
697 00016B41 <res 00000001>
                                <1> audio_device: resb 1
698 00016B42 <res 00000001>
                                <1> audio_mode: resb 1
699 00016B43 <res 00000001>
                                <1> audio_intr: resb 1
700 00016B44 <res 00000001>
                                <1> audio_busy: resb 1 ; Busy flag for audio irq ; 21/04/2017
 701 00016B45 <res 00000001>
                                <1> audio reserved: resb 1
702 00016B46 <res 00000002>
                                <1> audio_io_base: resw 1
                                                                   ; Base I/O address of audio device
 703 00016B48 <res 00000004>
                                <1> audio_dev_id:
                                                      resd 1; BUS/DEV/FN; 00000000BBBBBBBBDDDDDFFF00000000
704 00016B4C <res 00000004>
                                                      resd 1
                                <1> audio vendor:
705 00016B50 <res 00000004>
                                <1> audio_stats_cmd: resd 1
                                <1>;
707 00016B54 <res 00000004>
                                <1> audio_buffer:
                                                      resd 1; virtual address of user's audio buffer
708 00016B58 <res 00000004>
                                <1> audio_p_buffer: resd 1; Physical address of user's audio buffer
                                <1> audio_buff_size: resd 1 ; user's audio buffer size (half buffer size)
 709 00016B5C <res 00000004>
                                <1> audio_dma_buff: resd 1 ; dma buffer address
710 00016B60 <res 00000004>
711 00016B64 <res 00000004>
                                <1> audio dmabuff size: resd 1 ; dma buffer size (2 * half buffer size)
                                <1> audio_flag: resb 1 ; dma buffer flag (1st half = 0, 2nd half = 1)
712 00016B68 <res 00000001>
713 00016B69 <res 00000001>
                                <1> audio_user: resb 1; user number of the owner
                                <1> audio_cb_mode: resb 1; 0 = signal response byte method
714 00016B6A <res 00000001>
715
                                <1>
                                                     ; 1 = callback method
 716
                                <1>
                                                      ; 2 = s.r.b. method with auto increment
                                <1> audio_srb: resb 1; signal response byte value
717 00016B6B <res 00000001>
718 00016B6C <res 00000004>
                                <1> audio_cb_addr: resd 1 ; callback service address or s.r.b. address
719
                                <1>
                                                      ; (s.r.b. addr is physical, cbs addr is virtual)
720
                                <1>
721 00016B70 <res 00000001>
                                <1> audio bps: resb 1 ; selected mode: 8 bit, 16 bit
722 00016B71 <res 00000001>
                                <1> audio_stmo: resb 1; selected mode: mono /stereo
 723 00016B72 <res 00000002>
                                <1> audio_freq: resw 1; sampling rate
724
                                <1>
725
                                <1> ; 21/04/2017
                                <1> audio_play_cmd: resb 1 ; Play/Stop command (1 = play, 0 = stop)
<1> audio_civ: ; 28/05/2017 ; Current Buffer Index (AC'97)
 726 00016B74 <res 00000001>
727
728 00016B75 <res 00000001>
                                <1> audio_flag_eol:
                                                      resb 1 ; End of Link status (vt8233, EOL/FLAG)
729
                                <1>
730
                                <1> audio_master_volume:
 731 00016B76 <res 00000001>
                                <1> audio_master_volume_1: resb 1 ; sound volume (lineout) left channel
732 00016B77 <res 00000001>
                                <1> audio_master_volume_r: resb 1 ; sound volume (lineout) right channel
733
                                <1>
734
                                <1> alignb 4
735
                                <1>; 28/05/2017
                                <1>; AC'97 Audio Controller Base Adress Registers
 736
                                <1> NAMBAR:
737 00016B78 <res 00000002>
                                              resw 1 ; Native Audio Mixer Base Address
738 00016B7A <res 00000002>
                                <1> NABMBAR:
                                                resw 1; Native Audio Bus Mastering Base Address
739
                                <1>
740
                                <1> ;alignb 4
741
                                <1> ; 21/04/2017
742 00016B7C <res 00000400>
                                <1> audio bdl buff:
                                                     resd 32*8 ; VT8233 (AC97) BDL Buffer Size
                                 <1> ; 12/05/2017
 743
744 00016F7C <res 00000004>
                                <1> base_addr: resd 1; 'direct_memory_access' (memory.s)
745
                                <1>
 746
                                 <1>; 28/08/2017
                                <1>; 20/08/2017
748 00016F80 <res 00000001>
                                               resb 1 ;
                                <1>
 749 00016F81 <res 00000001>
                                <1> dma user:
                                               resb 1; user number for sysdma
 750 00016F82 <res 00000001>
                                <1> dma_channel: resb 1; dma channel for sysdma
 751 00016F83 <res 00000001>
                                <1> dma mode: resb 1 ; dma mode for sysdma
 752 00016F84 <res 00000004>
                                <1> dma_addr:
                                               resd 1; dma buffer physical addr for sysdma
753 00016F88 <res 00000004>
                                <1> dma_size:
                                                resd 1 ; dma buffer size (in bytes) for sysdma
                                <1> dma start: resd 1 ; dma start address for sysdma
 754 00016F8C <res 00000004>
755 00016F90 <res 00000004>
                                <1> dma_count: resd 1 ; dma count (in bytes) for sysdma
                                <1>
757 00016F94 <res 0000906C>
                                <1> alignb 65536
                                <1> ; 09/08/2017
758
 759
                                <1> ; 12/05/2017
760 00020000 <res 00010000>
                                <1> sb16_dma_buffer: resb 65536 ; DMA buffer for sb16 audio playing.
2830
                                    ; 24/01/2016
2831
                                    %include 'ubss.s' ; UNINITIALIZED KERNEL (USER) DATA
                                1
  2
                                <1> ; TRDOS386.ASM (TRDOS 386 Kernel - v2.0.0) - UNINITIALIZED USER DATA : ubss.s
  3
                                4
                                <1>; Last Update: 28/02/2017
                                <1> ; ------
  5
  6
                                 <1> ; Beginning: 24/01/2016
```

```
<1>; Assembler: NASM version 2.11 (trdos386.s)
  9
                              <1>; Derived from 'Retro UNIX 386 Kernel - v0.2.1.0' source code by Erdogan Tan
 10
 11
                              <1>; ux.s (04/12/2015)
                              12
 1.3
                              <1>
 14
                              <1> ; Retro UNIX 386 v1 Kernel - ux.s
                              <1>; Last Modification: 04/12/2015
 15
 16
                              <1>;
                              <1>; ////// RETRO UNIX 386 V1 SYSTEM DEFINITIONS ///////////
 17
 18
                              <1> ; (Modified from
 19
                                      Retro UNIX 8086 v1 system definitions in 'UNIX.ASM', 01/09/2014)
 20
                              <1>; ((UNIX.ASM (RETRO UNIX 8086 V1 Kernel), 11/03/2013 - 01/09/2014))
                              <1> ; ------
 21
 22
                              <1>; Derived from UNIX Operating System (v1.0 for PDP-11)
                              <1>; (Original) Source Code by Ken Thompson (1971-1972)
 23
                              <1> ; <Bell Laboratories (17/3/1972)>
 25
                              <1> ; <Preliminary Release of UNIX Implementation Document>
 26
                              <1>; (Section E10 (17/3/1972) - ux.s)
                              27
 28
                              <1>
 29
                              <1> alignb 2
 30
                              <1>
 31
                              <1> inode:
                                     ; 11/03/2013.
 32
                              <1>
 33
                              <1>
                                       ; Derived from UNIX v1 source code 'inode' structure (ux).
                              <1>
 35
                              <1>
 36 00030000 <res 00000002>
                              <1>
                                      i.flgs:
                                                   resw 1
 37 00030002 <res 00000001>
                                     i.nlks:
                              <1>
                                                   resb 1
                                     i.uid: resb 1
 38 00030003 <res 00000001>
                              <1>
                              <1>
                                       ;i.size: resw 1 ; size
                                     resw 1 ; 29/04/2016
 40 00030004 <res 00000002>
                              <1>
 41 00030006 <res 00000010>
                              <1>
                                     i.dskp:
                                                resw 8 ; 16 bytes
                                     i.ctim:
 42 00030016 <res 00000004>
                              <1>
                                                   resd 1
 43 0003001A <res 00000004>
                                      i.mtim:
                              <1>
                                                   resd 1
 44 0003001E <res 00000002>
                                     i.rsvd: resw 1 ; Reserved (ZERO/Undefined word for UNIX v1.)
                              <1>
 45
                              <1>
 46
                              <1> I_SIZE
                                            equ $ - inode
 47
                              <1>
 48
                              <1> process:
                                   ; 19/12/2016
 49
                              <1>
                                      ; 21/05/2016
 50
                              <1>
                                      ; 19/05/2016 - TRDOS 386 (TRDOS v2.0)
 51
                              <1>
                                     ; 06/05/2015 - Retro UNIX 386 v1
; 11/03/2013 - 05/02/2014 (Retro UNIX 8086 v1)
 52
                              <1>
 53
                              <1>
                              <1>
                                      ;Derived from UNIX v1 source code 'proc' structure (ux).
 55
                              <1>
 56
                              <1>
 57 00030020 <res 00000020>
                                       p.pid: resw nproc
                              <1>
 58 00030040 <res 00000020>
                              <1>
                                      p.ppid: resw nproc
                                       p.break: resw nproc
 59 00030060 <res 00000020>
                              <1>
 60 00030080 <res 00000010>
                              <1>
                                        p.ttyc: resb nproc; console tty in Retro UNIX 8086 v1.
 61 00030090 <res 00000010>
                              <1>
                                     p.waitc: resb nproc; waiting channel in Retro UNIX 8086 v1.
 62 000300A0 <res 00000010>
                              <1>
                                     p.link:
                                                   resb nproc
 63 000300B0 <res 00000010>
                              <1>
                                      p.stat:
                                                   resb nproc
                              <1>
                                       ; 06/05/2015 (Retro UNIX 386 v1 feature only !)
                              <1>
 65
 66 000300C0 <res 00000040>
                              <1>
                                      p.upage: resd nproc ; Physical address of the process's
                                                      ; 'user' structure
                              <1>
 67
 68
                              <1>
                                      ; 21/05/2016
 69
                              <1>
                                      ; 19/05/2016 (TRDOS 386 feature only!)
 70 00030100 <res 00000010>
                              <1>
                                       p.timer: resb nproc; number of timer events of the processs
 71
                              <1>
 72
                              <1>
                                       ; 19/12/2016
 73 00030110 <res 00000040>
                              <1>
                                       p.tcb: resd nproc ; timer callback service address (if > 0)
 74
                              <1>
 75
                              <1> P_SIZE
                                            equ $ - process
 76
                              <1>
 77
                              <1>; fsp table (original UNIX v1)
 78
                              <1>;
 79
                              <1> ;Entry
                                           15
 80
                              <1>;
                                          |---|--
 81
                              <1> ; 1
                              <1>;
                                                i-number of open file
 82
                                          |r/w|
 83
                              <1>;
                                                 device number
 84
                              <1>;
 85
                              <1>;
                                         _____
                                      (*) | offset pointer, i.e., r/w pointer to file
 86
                              <1>;
 87
                              <1>;
                              <1> ;
                                          | flag that says | number of processes
 88
                                                             | that have file open
                              <1>;
                                             file deleted
 89
 90
                              <1>;
 91
                              <1>; 2
 92
                              <1>;
 93
                              <1>;
 94
                              <1>;
 95
                              <1>;
                              <1>;
 96
 97
                              <1>;
                              <1> ;
 98
                              <1>; 3
 99
100
                              <1>;
101
                              <1>;
102
                              <1>; (*) Retro UNIX 386 v1 modification: 32 bit offset pointer
103
                              <1>
104
                              <1>
                              <1> ; 15/04/2015
105
106 00030150 <res 000001F4>
                              <1> fsp: resb nfiles * 10 ; 11/05/2015 (8 -> 10)
107 00030344 <res 00000002>
                              <1> idev: resw 1 ; device number is 1 byte in Retro UNIX 8086 v1 !
108 00030346 <res 00000002>
                              <1> cdev: resw 1 ; device number is 1 byte in Retro UNIX 8086 v1 !
109
                              <1> ; 18/05/2015
110
                              <1>; 26/04/2013 device/drive parameters (Retro UNIX 8086 v1 feature only!)
                              <1>; 'UNIX' device numbers (as in 'cdev' and 'u.cdrv')
111
112
                              <1>; 0 -> root device (which has Retro UNIX 8086 v1 file system)
```

8

```
1 -> mounted device (which has Retro UNIX 8086 v1 file system)
113
114
                                 <1>; 'Retro UNIX 8086 v1' device numbers: (for disk I/O procedures)
                                         0 -> fd0 (physical drive, floppy disk 1), physical drive number = 0
115
116
                                          1 -> fd1 (physical drive, floppy disk 2), physical drive number = 1
                                 <1>;
                                          2 -> hd0 (physical drive, hard disk 1), physical drive number = 80h
117
118
                                 <1>;
                                          3 -> hd1 (physical drive, hard disk 2), physical drive number = 81h
119
                                 <1>;
                                         4 -> hd2 (physical drive, hard disk 3), physical drive number = 82h
                                        5 -> hd3 (physical drive, hard disk 4), physical drive number = 83h
                                 <1>;
120
121 00030348 <res 00000001>
                                 <1> rdev: resb 1 ; root device number ; Retro UNIX 8086 v1 feature only!
                                 <1>
122
                                                 ; as above, for physical drives numbers in following table
                                 <1> mdev: resb 1 ; mounted device number ; Retro UNIX 8086 v1 feature only!
123 00030349 <res 00000001>
124
                                 <1> ; 15/04/2015
125 0003034A <res 00000001>
                                 <1> active:
                                                 resb 1
126 0003034B <res 00000001>
                                 <1> resb 1 ; 09/06/2015
                                 <1> mnti: resw 1
127 0003034C <res 00000002>
128 0003034E <res 00000002>
                                 <1> mpid: resw 1
129 00030350 <res 00000002>
                                 <1> rootdir: resw 1
130
                                 <1>
131
                                 <1>; 21/05/2016 - TRDOS 386 (TRDOS v2.0) - priority levels, 3 run queues
132
                                 <1> rung:
                                                                                                    ; 2
                                 <1> runq_event: resw 1 ; high priority, 'run for event'
133 00030352 <res 00000002>
134 00030354 <res 00000002>
                                 <1> runq normal: resw 1 ; normal/regular priority, 'run as regular' ; 1
135 00030356 <res 00000002>
                                 <1> runq_background: resw 1 ; low priority, 'run on background'
                                 <1> imod: resb 1 <1> smod: resb 1
137 00030358 <res 00000001>
138 00030359 <res 00000001>
139 0003035A <res 00000001>
                                 <1> mmod: resb 1
140 0003035B <res 00000001>
                                 <1> sysflg:
                                                 resb 1
                                 <1>
142
                                 <1> alignb 4
143
                                 <1>
                                 <1> user:
144
                                          ; 13/01/2017
145
                                 <1>
146
                                 <1>
                                          ; 19/12/2016
                                         ; 21/05/2016 - TRDOS 386 (TRDOS v2.0)
147
                                 <1>
                                                         [u.pri] usage method modification
148
                                 <1>
                                          ; 04/12/2015
149
                                 <1>
                                         ; 18/10/2015
                                 <1>
150
                                          ; 12/10/2015
151
                                 <1>
152
                                 <1>
                                          ; 21/09/2015
153
                                 <1>
                                          ; 24/07/2015
154
                                 <1>
                                          ; 16/06/2015
                                 <1>
                                          ; 09/06/2015
155
156
                                 <1>
                                          ; 11/05/2015
                                          ; 16/04/2015 (Retro UNIX 386 v1 - 32 bit modifications)
157
                                 <1>
158
                                 <1>
                                          ; 10/10/2013
159
                                 <1>
                                          ; 11/03/2013.
160
                                 <1>
                                           ; Derived from UNIX v1 source code 'user' structure (ux).
161
                                 <1>
                                 <1>
163 0003035C <res 00000004>
                                 <1>
                                           u.sp: resd 1 ; esp (kernel stack at the beginning of 'sysent')
                                          u.usp: resd 1; esp (kernel stack points to user's registers)
u.r0: resd 1; eax
164 00030360 <res 00000004>
                                 <1>
165 00030364 <res 00000004>
                                 <1>
166 00030368 <res 00000002>
                                 <1>
                                           u.cdir:
                                                         resw 1
167 0003036A <res 0000000A>
                                 <1>
                                          u.fp: resb 10
168 00030374 <res 00000004>
                                 <1>
                                          u.fofp:
169 00030378 <res 00000004>
                                 <1>
                                          u.dirp:
                                                          resd 1
170 0003037C <res 00000004>
                                 <1>
                                          u.namep: resd 1
                                          u.off: resd 1
u.base:
171 00030380 <res 00000004>
                                 <1>
172 00030384 <res 00000004>
                                 <1>
                                                        resd 1
173 00030388 <res 00000004>
                                 <1>
                                          u.count: resd 1
174 0003038C <res 00000004>
                                 <1>
                                          u.nread: resd 1
175 00030390 <res 00000004>
                                 <1>
                                          u.break: resd 1 ; break
176 00030394 <res 00000002>
                                 <1>
                                                         resw 1
177
                                 <1>
                                          ; 10/01/2017 (TRDOS 386, relocation and dword alignment)
178
                                 <1>
                                           ; tty number (rtty, rcvt, wtty)
179 00030396 <res 00000001>
                                 <1>
                                          u.ttyn: resb 1 ; 28/07/2013 - Retro Unix 8086 v1 feature only!
180 00030397 <res 00000001>
                                 <1>
                                          u.resb: resb 1; 10/01/2017 (TRDOS 386, temporary)
181 00030398 <res 00000010>
                                 <1>
                                           u.dirbuf: resb 16; 04/12/2015 (10 -> 16)
182
                                 <1>
                                                    resw 1 ; 14/02/2014
                                          ;u.pri:
                                           u.quant: resb 1 ; Retro UNIX 8086 v1 Feature only ! (uquant)
183 000303A8 <res 00000001>
                                 <1>
184 000303A9 <res 00000001>
                                 <1>
                                           u.pri: resb 1; Modification: 21/05/2016 (priority levels: 0, 1, 2)
185 000303AA <res 00000002>
                                           u.intr:
                                 <1>
                                                         resw 1
186 000303AC <res 00000002>
                                 <1>
                                          u.quit:
                                                         resw 1
                                 <1>
                                                         resw 1 ; 10/10/2013
187
                                          ;u.emt:
188
                                 <1>
                                          ;u.ilgins: resw 1 ; 10/01/2017
189 000303AE <res 00000002>
                                 <1>
                                          u.cdrv: resw 1 ; cdev
                                          u.uid: resb 1 ; uid
190 000303B0 <res 00000001>
                                 <1>
                                           u.ruid: resb 1
191 000303B1 <res 00000001>
                                 <1>
192 000303B2 <res 00000001>
                                                         resb 1
                                 <1>
                                          u.bsys:
193 000303B3 <res 00000001>
                                 <1>
                                           u.uno: resb 1
194 000303B4 <res 00000004>
                                 <1>
                                             u.upage: resd 1 ; 16/04/2015 - Retro Unix 386 v1 feature only !
                                           u.pgdir: resd 1; 09/03/2015 (page dir addr of process)
195 000303B8 <res 00000004>
                                 <1>
196 000303BC <res 00000004>
                                 <1>
                                           u.ppgdir: resd 1; 06/05/2015 (page dir addr of the parent process)
197 000303C0 <res 00000004>
                                 <1>
                                           u.pbase: resd 1 ; 20/05/2015 (physical base/transfer address)
198 000303C4 <res 00000002>
                                           u.pcount: resw 1; 20/05/2015 (byte -transfer- count for page)
                                 <1>
                                 <1>
                                           ;u.pncount: resw 1
200
                                                 ; 16/06/2015 (byte -transfer- count for page, 'namei', 'mkdir')
                                 <1>
201
                                 <1>
                                           ;u.pnbase: resd 1
                                                 ; 16/06/2015 (physical base/transfer address, 'namei', 'mkdir')
202
                                 <1>
                                                         ; 09/06/2015
203
                                 <1>
                                           u.kcall: resb 1; The caller is 'namei' (dskr) or 'mkdir' (dskw) sign
204 000303C6 <res 00000001>
                                 <1>
205 000303C7 <res 00000001>
                                           u.brwdev: resb 1 ; Block device number for direct I/O (bread & bwrite)
                                 <1>
                                                         ; 24/07/2015 - 24/06/2015
                                 <1>
                                           ;u.args: resd 1 ; arguments list (line) offset from start of [u.upage]
207
                                 <1>
208
                                 <1>
                                                         ; (arg list/line is from offset [u.args] to 4096 in [u.upage])
                                                         ; ([u.args] points to argument count -argc- address offset)
209
                                 <1>
210
                                 <1>
                                                         ; 24/06/2015
                                           ;u.core: resd 1 ; physical start address of user's memory space (for sys exec)
211
                                 <1>
                                           ;u.ecore: resd 1 ; physical end address of user's memory space (for sys exec)
212
                                 <1>
                                           ; last error number
213
                                 <1>
214 000303C8 <res 00000004>
                                 <1>
                                           u.error: resd 1 ; 28/07/2013 - 09/03/2015
215
                                 <1>
                                                         ; Retro UNIX 8086/386 v1 feature only!
                                 <1>
                                                        ; 21/09/2015 (debugging - page fault analyze)
217 000303CC <res 00000004>
                                 <1>
                                           u.pfcount: resd 1 ; page fault count for (this) process (for sys geterr)
```

```
218
                                  <1>
                                                  ; 19/12/2016 (TRDOS 386)
219 000303D0 <res 00000004>
                                  <1>
                                            u.tcb: resd 1 ; Timer callback address/flag which will be used by timer int
                                                ; 13/01/2017 (TRDOS 386)
220
                                  <1>
221 000303D4 <res 00000001>
                                  <1>
                                            u.t_lock: resb 1 ; Timer interrupt (callback) lock (unlocked by 'sysrele')
                                            u.t_mode: resb 1 ; running mode during timer interrupt (0= system, 0FFh= user)
222 000303D5 <res 00000001>
                                  <1>
223
                                  <1>
                                                  ; 26/02/2017 (TRDOS 386)
224 000303D6 <res 00000001>
                                                          resb 1 ; Count of IRQ callback services (IRQs in use)
                                  <1>
                                            u.irqc:
                                                 ; 28/02/2017 (TRDOS 386)
225
                                  <1>
226 000303D7 <res 00000001>
                                  <1>
                                            u.irqwait: resb 1 ; IRQ waiting for callback service flag (IRQ number, If > 0)
227 000303D8 <res 00000001>
                                            u.r lock: resb 1 ; 'IRQ callback service is in progress' flag (IRQ lock)
                                  <1>
228 000303D9 <res 00000001>
                                  <1>
                                            u.r_mode: resb 1 ; running mode during hadware interrupt
229
                                  <1>
                                                  ; 27/02/2017 (TRDOS 386)
                                            u.fpsave: resb 1 ; TRDOS 386, 'save/restore FPU registers' flag
230 000303DA <res 00000001>
                                  <1>
231 000303DB <res 00000001>
                                  <1> alignb 4
232 000303DC <res 0000005E>
                                  <1>
                                            u.fpregs: resb 94; 94 byte area for saving and restoring FPU registers
233
                                  <1>
                                  <1> alignb 4
234 0003043A <res 00000002>
235
                                  <1>
236
                                  <1> U_SIZE
                                                   equ $ - user
237
                                  <1>
238
                                  <1>; 18/10/2015 - Retro UNIX 386 v1 (local variables for 'namei' and 'sysexec')
                                  <1> pcore: resd 1 ; physical start address of user's memory space (for sys exec)
<1> ecore: resd 1 ; physical address of user's stack/last page (for sys exec)
 239 0003043C <res 00000004>
240 00030440 <res 00000004>
 241 00030444 <res 00000004>
                                  <1> nbase: resd 1; physical base address for 'namei' & 'sysexec'
242 00030448 <res 00000002>
                                  243 0003044A <res 00000002>
                                  <1> argc: resw 1; argument count for 'sysexec'
244 0003044C <res 00000004>
                                  <1> argv: resd 1; argument list (recent) address for 'sysexec'
245
                                  <1>
246
                                  <1> ; 03/06/2015 - Retro UNIX 386 v1 Beginning
                                  <1>; 07/04/2013 - 31/07/2013 - Retro UNIX 8086 v1
247
248 00030450 <res 00000001>
                                  <1> rw:
                                            resb 1 ;; Read/Write sign (iget)
249
                                  <1>
                                  <1> ;alignb 4
250
251
                                  <1>
252
                                  <1> ; 24/04/2016
253 00030451 <res 00000004>
                                  <1> ii:
                                                resd 1 ; first cluster of the program file
254 00030455 <res 00000004>
                                  <1> i.size:
                                                       resd 1; size of the program file
2832
2833 00030459 <res 00000003>
                                      alignb 4
2834
                                      ; 23/05/2016 (TRDOS 386)
2835
2836
                                      ; 14/10/2015 (Retro UNIX 386 v1, 'unix386.s')
                                                   \operatorname{resd} 1 ; \operatorname{cr} 3 \operatorname{register} content at the beginning of the timer
2837 0003045C <res 00000004>
                                      cr3reg:
2838
                                                    ; (or RTC) interrupt handler.
2839
                                      ; 10/12/2016 (callback)
2840
2841
                                      ; 10/06/2016
                                      ; 19/05/2016
2842
                                      ; 18/05/2016 - TRDOS 386 feature only !
2843
2844 00030460 <res 00000100>
                                      timer_set: resd 16*4 ; 256 bytes memory space for 16 timer events
2845
                                           ; Timer Event Structure: (max. 16 timer events, 16*16 bytes)
2846
                                                    Owner:
                                                                       resb 1; 0 = free
                                            ;
                                                                       ;>0 = process number (u.uno)
2847
                                            ;
2848
                                                   Callback: resb 1; 0 = response byte address (phy)
2849
                                                                       1 = callback address (virtual)
2850
                                                                   resb 1 ; 0 = Timer interrupt (or none)
                                                   Interrupt:
2851
                                                                     ; 1 = Real Time Clock interrupt
2852
                                                                  resb 1; 0 to 255, signal return value
                                                   Response:
2853
                                                   Count Limit: resd 1 ; count of ticks (total/set)
                                                   Current Count: resd 1 ; count of ticks (current)
2854
                                            ;
2855
                                                   Response Addr: resd 1 ; response byte (pointer) address
2856
                                                                       ; (or callback -user service- address)
2857
2858
                                      ;; Memory (swap) Data (11/03/2015)
2859
                                      ; 09/03/2015
2860 00030560 <res 00000002>
                                      swpq_count: resw 1 ; count of pages on the swap queue
2861 00030562 <res 00000004>
                                      swp drv: resd 1; logical drive description table address of the swap drive/disk
2862 00030566 <res 00000004>
                                      swpd_size: resd 1 ; size of swap drive/disk (volume) in sectors (512 bytes).
                                      swpd_free: resd 1 ; free page blocks (4096 bytes) on swap disk/drive (logical)
2863 0003056A <res 00000004>
2864 0003056E <res 00000004>
                                      swpd_next: resd 1 ; next free page block
2865 00030572 <res 00000004>
                                      swpd_last: resd 1 ; last swap page block
2866
2867 00030576 <res 00000002>
                                      alignb 4
2868
2869
                                      ; 10/07/2015
2870
                                      ; 28/08/2014
2871 00030578 <res 00000004>
                                      error_code: resd 1
                                      ; 29/08/2014
2872
2873 0003057C <res 00000004>
                                      FaultOffset:
                                                         resd 1
                                      ; 21/09/2015
2874
2875 00030580 <res 00000004>
                                      PF_Count: resd 1; total page fault count
                                                                ; (for debugging - page fault analyze)
2876
2877
                                                           'page_fault_handler' (memory.inc)
2878
                                                         ; 'sysgeterr' (u9.s)
2879
2880
                                      ; 29/04/2016 (TRDOS 386 = TRDOS v2.0)
                                      ; 22/08/2015 (Retro UNIX 386 v1)
2881
2882
                                      buffer:
2883 00030584 <res 00000008>
                                           resb 8
2884
                                      readi_buffer:
2885 0003058C <res 00000200>
                                            resb 512
2886 0003078C <res 00000008>
                                            resb 8
                                      writei buffer:
2888 00030794 <res 00000200>
                                            resb 512
2889
                                      ; 24/10/2016
2890 00030994 <res 00000008>
                                           resb 8
2891
                                      rw buffer:
2892 0003099C <res 00000800>
                                                  2048 ; general purposed, r/w sector buffer
2893
2894
                                      bss_end:
2895
2896
                                      ; 27/12/2013
2897
                                      _end: ; end of kernel code
```