Московский государственный технический университет
им. Н.Э. Баумана

УТВЕРЖДАЮ:

Большаков С.А. "__"_____2024 г.

Курсовая работа по курсу «Системное программирование»

Исходный текст программного продукта
(вид документа)

писчая бумага
(вид носителя)

44
(количество листов)

ИСПОЛНИТЕЛИ:

студенты группы ИУ5-41Б _____
Цыпышев Т.А.
"__"_____2024 г.

Москва – 2024

1. Файл tsr.lst

```
Turbo Assembler  Version 3.1        08/09/24 18:09:46       Page 1
tsr.asm



      1
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
      2                             ; tsr.asm
      3                             ;
      4                             ; Сборка:
      5                             ;  tasm.exe /l tsr.asm
      6                             ;  tlink /t /x tsr.obj
      7                             ;
      8                             ;
      9                             ; Автор:
     10                             ;  МГТУ им. Н.Э. Баумана, ИУ5-41Б,  2024 г.
     11                             ;   Цыпышев Т.А
     12
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
     13
     140000                         code segment    'code'
     15                                     assume  CS:code, DS:code
     16                                     org     100h
     170100                                 _start:
     18
     190100  E9 06DC                        jmp _initTSR ; на начало программы
     20
     21                                     ; данные
     220103  71 77 65 72 74 79  75+         ignoredChars                                    DB
                   +
     23      69 6F 70 61 73 64  66+  'qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM'  ; список игнорируемых
символов
     24      67 68 6A 6B 6C 7A  78+
     25      63 76 62 6E 6D 51  57+
     26      45 52 54 59 55 49  4F+
     27      50 41 53 44 46 47  48+
     28      4A 4B 4C 5A 58 43  56+
     29      42 4E 4D
     300137  A9 E6 E3 AA A5 AD  A3+      replaceWith           DB
'йцукенгшщзфывапролдячсмитьЙЦУКЕНГШЩЗФЫВАПРОЛДЯЧСМИТЬ'
     31      E8 E9 A7 E4 EB A2  A0+
     32      AF E0 AE AB A4 EF  E7+
     33      E1 AC A8 E2 EC 89  96+
     34      93 8A 85 8D 83 98  99+
     35      87 94 9B 82 80 8F  90+
     36      8E 8B 84 9F 97 91  8C+
     37      88 92 9C
     38      =0068                          ignoredLength                          equu     $-ignoredChars
                   +
     39                             ; длина строки ignoredChars
     40016B  00                     ignoreEnabled                          DB      0
```

```
      +
41                            ; флаг функции игнорирования ввода
42016C  7D 53 4D 22 3E 5A             translateFrom                           DB      '}SM">Z'
              +
43                            ;символы для замены (ЪЫЬЭЮЯ на англ. раскладке)
440172  9A 9B 9C 9D 9E 9F             translateTo                             DB      'ЪЫЬЭЮЯ'
      +
45                            ;символы на которые будет идти замена
46      =0006                         translateLength                         equ     $-translateTo
              +
47                            ; длина строки trasnlateFrom
480178  00                    translateEnabled                                DB      0
      +
49                            ; флаг функции перевода
50
510179  00                    signaturePrintingEnabled                DB      0
      +
52                            ; флаг функции вывода информации об авторе
53017A  00                    cursiveEnabled                          DB      0
      +
54                            ; флаг перевода символа в курсив
55
56017B  00                    cursiveSymbol                           DB 00000000b
      +
57                            ;символ, составленный из единичек (его курсивный вариант)
```

```
58017C  00                                                              DB 00000000b
59017D  00                                                              DB 00000000b
60017E  FF                                                              DB 11111111b
61017F  FF                                                              DB 11111111b
620180  3E                                                              DB 00111110b
630181  3E                                                              DB 00111110b
640182  3E                                                              DB 00111110b
650183  3E                                                              DB 00111110b
660184  3E                                                              DB 00111110b
670185  3E                                                              DB 00111110b
680186  3E                                                              DB 00111110b
690187  3E                                                              DB 00111110b
700188  00                                                              DB 00000000b
710189  00                                                              DB 00000000b
72018A  00                                                              DB 00000000b
73
74018B  92                      charToCursiveIndex              DB 'T'
      +
75                              ;символ для замены
76018C  10*(FF)                         savedSymbol                     DB 16 dup(0FFh)
      +
77                              ; переменная для хранения старого символа
78
79      =00FF                           true                                    equ     0FFh
              +
80                              ; константа истинности
81019C  ????                            old_int9hOffset                         DW      ?
              +
82                              ; адрес старого обработчика int 9h
83019E  ????                            old_int9hSegment                        DW      ?
              +
84                              ; сегмент старого обработчика int 9h
8501A0  ????                            old_int1ChOffset                        DW      ?
              +
86                              ; адрес старого обработчика int 1Ch
8701A2  ????                            old_int1ChSegment               DW      ?
              +
88                              ; сегмент старого обработчика int 1Ch
8901A4  ????                            old_int2FhOffset                        DW      ?
              +
90                              ; адрес старого обработчика int 2Fh
9101A6  ????                            old_int2FhSegment               DW      ?
              +
92                              ; сегмент старого обработчика int 2Fh
93
9401A8  00                      unloadTSR                               DB      0
      +
95                              ; 1 - выгрузить резидент
9601A9  00                      notLoadTSR                              DB      0
      +
97                              ; 1 - не загружать
9801AA  0000                            counter                         DW      0
```

```
 99      =0007                           printDelay                                      equ     7
               +
100                                     ;задержка перед выводом "подписи" в секундах
10101AC  0001                           printPos                                                DW      1
               +
102                                     ;положение  подписи на экране. 0 - верх, 1 - центр,  2 - низ
103
10401AE  B3 20 20 20 20 20  20+         signatureLine1                  DB      179, '
Цыпышев Т.А     +
105      20 20 20 20 20 20  20+  ', 179
106      20 20 96 EB AF EB  E8+
107      A5 A2 20 92 2E 80  20+
108      20 20 20 20 20 20  20+
109      20 20 20 20 20 20  20+
110      B3
111      =002B                          Line1_length                                    equ     $-
signatureLine1
11201D9  B3 20 20 20 20 20  20+         signatureLine2                  DB      179, '
ИУ5-41Б    +
113      20 20 20 20 20 20  20+  ', 179
114      20 20 88 93 35 2D  34+
```

```
Turbo Assembler Version 3.1        05/09/24 16:09:46       Page 3
tsr.asm



    115      31 81 20 20 20 20  20+
    116      20 20 20 20 20 20  20+
    117      20 20 20 20 20 20  20+
    118      20 B3
    119      =002C                          Line2_length                                    equ     $-
signatureLine2
    1200205  B3 20 20 20 20 20  20+         signatureLine3                          DB      179, '
Вариант 20         +
    121      20 20 20 20 20 20  20+  ', 179
    122      20 20 82 A0 E0 A8  A0+
    123      AD E2 20 32 30 20  20+
    124      20 20 20 20 20 20  20+
    125      20 20 20 20 20 20  20+
    126      20 20 B3
    127      =002D                          Line3_length                                    equ     $-
signatureLine3
    1280232  3E 74 73 72 2E 63  6F+         helpMsg DB  '>tsr.com [/?]', 10, 13
    129      6D 20 5B 2F 3F 5D  0A+
    130      0D
    1310241  20 5B 2F 3F 5D 20  2D+                         DB ' [/?] - вывод данной справки',  10, 13
    132      20 A2 EB A2 AE A4  20+
    133      A4 A0 AD AD AE A9  20+
    134      E1 AF E0 A0 A2 AA  A8+
    135      0A 0D
    136025F  20 20 46 39 20 20  2D+                         DB '  F9  - вывод ФИО и группы по таймеру в центре
   экрана', 10, 13
    137      20 A2 EB A2 AE A4  20+
    138      94 88 8E 20 A8 20  A3+
    139      E0 E3 AF AF EB 20  AF+
    140      AE 20 E2 A0 A9 AC  A5+
    141      E0 E3 20 A2 20 E6  A5+
    142      AD E2 E0 A5 20 ED  AA+
    143      E0 A0 AD A0 0A 0D
    1440296  20 20 46 31 20 20  2D+                         DB '  F1  - включение и отключения  курсивного вывода
русского символа  +
    145      20 A2 AA AB EE E7  A5+  T', 10, 13
    146      AD A8 A5 20 A8 20  AE+
    147      E2 AA AB EE E7 A5  AD+
    148      A8 EF 20 AA E3 E0  E1+
    149      A8 A2 AD AE A3 AE  20+
    150      A2 EB A2 AE A4 A0  20+
    151      E0 E3 E1 E1 AA AE  A3+
    152      AE 20 E1 A8 AC A2  AE+
    153      AB A0 20 92 0A 0D
    15402DB  20 20 46 32 20 20  2D+                         DB '  F2  - включение и отключение  частичной
русификации клавиатуры    +
    155      20 A2 AA AB EE E7  A5+  (}SM">Z -> ЪЫЬЭЮЯ)', 10, 13
    156      AD A8 A5 20 A8 20  AE+
    157      E2 AA AB EE E7 A5  AD+
    158      A8 A5 20 E7 A0 E1  E2+
```

```
    159      A8 E7 AD AE A9 20  E0+
    160      E3 E1 A8 E4 A8 AA  A0+
    161      E6 A8 A8 20 AA AB  A0+
    162      A2 A8 A0 E2 E3 E0  EB+
    163      28 7D 53 4D 22 3E  5A+
    164      20 2D 3E 20 9A 9B  9C+
    165      9D 9E 9F 29 0A 0D
    166032E  20 20 46 33 20 20  2D+                          DB '  F3  - включение и отключение  режима замены
символов латинских    +
    167      20 A2 AA AB EE E7  A5+  букв на русские', 10, 13
    168      AD A8 A5 20 A8 20  AE+
    169      E2 AA AB EE E7 A5  AD+
    170      A8 A5 20 E0 A5 A6  A8+
    171      AC A0 20 A7 A0 AC  A5+
```

```
    172      AD EB 20 E1 A8 AC  A2+
    173      AE AB AE A2 20 AB  A0+
    174      E2 A8 AD E1 AA A8  E5+
    175      20 A1 E3 AA A2 20  AD+
    176      A0 20 E0 E3 E1 E1  AA+
    177      A8 A5 0A 0D
    178
    179      =014D                           helpMsg_length                          equ  $-helpMsg
    180037F  8E E8 A8 A1 AA A0  20+          errorParamMsg                                   DB      'Ошибка
параметров     коммандной  +
    181      AF A0 E0 A0 AC A5  E2+  строки', 10, 13
    182      E0 AE A2 20 AA AE  AC+
    183      AC A0 AD A4 AD AE  A9+
    184      20 E1 E2 E0 AE AA  A8+
    185      0A 0D
    186      =0025                           errorParamMsg_length                    equ     $-errorParamMsg
    187
    18803A4  DA 29*(C4) BF                   tableTop                                        DB      218,
Line1_length-2 dup+
    189                              (196), 191
    190      =002B                           tableTop_length                         equ     $-tableTop
    19103CF  C0 29*(C4) D9                   tableBottom                             DB      192, Line1_length-2 dup
(196), +
    192                              217
    193      =002B                           tableBottom_length                      equ  $-tableBottom
    194
    195                                      ; сообщения
    19603FA  90 A5 A7 A8 A4 A5  AD+          installedMsg                                    DB  'Резидент
загружен!$'
    197      E2 20 A7 A0 A3 E0  E3+
    198      A6 A5 AD 21 24
    199040D  90 A5 A7 A8 A4 A5  AD+          alreadyInstalledMsg             DB  'Резидент уже загружен$'
    200      E2 20 E3 A6 A5 20  A7+
    201      A0 A3 E0 E3 A6 A5  AD+
    202      24
    2030423  8D A5 A4 AE E1 E2  A0+          noMemMsg                                        DB
'Недостаточно памяти$'
    204      E2 AE E7 AD AE 20  AF+
    205      A0 AC EF E2 A8 24
    2060437  8D A5 20 E3 A4 A0  AB+          notInstalledMsg                 DB  'Не удалось загрузить
резидент$'
    207      AE E1 EC 20 A7 A0  A3+
    208      E0 E3 A7 A8 E2 EC  20+
    209      E0 A5 A7 A8 A4 A5  AD+
    210      E2 24
    211
    2120455  90 A5 A7 A8 A4 A5  AD+          removedMsg                                      DB  'Резидент выгружен'
    213      E2 20 A2 EB A3 E0  E3+
    214      A6 A5 AD
    215      =0011                           removedMsg_length                               equ     $-removedMsg
```

```
    216
    2170466  8D A5 20 E3 A4 A0  AB+          noRemoveMsg                          DB  'Не удалось выгрузить
резидент'
    218      AE E1 EC 20 A2 EB  A3+
    219      E0 E3 A7 A8 E2 EC  20+
    220      E0 A5 A7 A8 A4 A5  AD+
    221      E2
    222      =001D                          noRemoveMsg_length                   equ     $-noRemoveMsg
    223
    2240483  46 39                         f9_txt                                       DB      'F9'
    2250485  46 31                         f1_txt                                       DB      'F1'
    2260487  46 32                         f2_txt                                       DB      'F2'
    2270489  46 33                         f3_txt                                       DB      'F3'
    228      =0002                          fx_length                                    equ     $-f3_txt
```

```
    229
    230048B                             changeFx proc
    231048B  50                         push AX
    232048C  53                         push BX
    233048D  51                         push CX
    234048E  52                         push DX
    235048F  55                         push BP
    2360490  06                         push ES
    2370491  33 DB                              xor BX, BX
    238
    2390493  B4 03                              mov AH, 03h
    2400495  CD 10                              int 10h
    2410497  52                         push DX
    242
    2430498  0E                         push CS
    2440499  07                         pop ES
    245
    246049A                             _checkF9:
    247049A  BD 0483r                           lea BP, f9_txt
    248049D  B9 0002                            mov CX, fx_length
    24904A0  B7 00                              mov BH, 0
    25004A2  B6 00                              mov DH, 0
    25104A4  B2 4E                              mov DL, 78
    25204A6  B8 1301                            mov AX, 1301h
    253
    25404A9  80 3E 0179r FF                     cmp signaturePrintingEnabled, true
    25504AE  74 07                              je _greenF9
    256
    25704B0                                     _redF9:
    25804B0  B3 4F                                      mov BL, 01001111b ; red
    25904B2  CD 10                                      int 10h
    26004B4  EB 08 90                                   jmp _checkF1
    261
    26204B7                                     _greenF9:
    26304B7  BD 0483r                                   lea BP, f9_txt
    26404BA  B3 2F                                      mov BL, 00101111b ; green
    26504BC  CD 10                                      int 10h
    266
    26704BE                             _checkF1:
    26804BE  BD 0485r                           lea BP, f1_txt
    26904C1  B9 0002                            mov CX, fx_length
    27004C4  B7 00                              mov BH, 0
    27104C6  B6 01                              mov DH, 1
    27204C8  B2 4E                              mov DL, 78
    27304CA  B8 1301                            mov AX, 1301h
    274
    27504CD  80 3E 017Ar FF                     cmp cursiveEnabled, true
    27604D2  74 07                              je _greenF1
    277
```

```
27804D4                                 _redF1:
27904D4  B3 4F                                  mov BL, 01001111b ; red
28004D6  CD 10                                  int 10h
28104D8  EB 05 90                               jmp _checkF2
282
28304DB                                 _greenF1:
28404DB  B3 2F                                  mov BL, 00101111b ; green
28504DD  CD 10                                  int 10h
```

```
    286
    28704DF                                 _checkF2:
    28804DF  BD 0487r                               lea BP, f2_txt
    28904E2  B9 0002                                mov CX, fx_length
    29004E5  B7 00                                  mov BH, 0
    29104E7  B6 02                                  mov DH, 2
    29204E9  B2 4E                                  mov DL, 78
    29304EB  B8 1301                                mov AX, 1301h
    294
    29504EE  80 3E 0178r FF                         cmp translateEnabled, true
    29604F3  74 07                                  je _greenF2
    297
    29804F5                                         _redF2:
    29904F5  B3 4F                                          mov BL, 01001111b ; red
    30004F7  CD 10                                          int 10h
    30104F9  EB 05 90                                       jmp _checkF3
    302
    30304FC                                         _greenF2:
    30404FC  B3 2F                                          mov BL, 00101111b ; green
    30504FE  CD 10                                          int 10h
    306
    3070500                                 _checkF3:
    3080500  BD 0489r                               lea BP, f3_txt
    3090503  B9 0002                                mov CX, fx_length
    3100506  B7 00                                  mov BH, 0
    3110508  B6 03                                  mov DH, 3
    312050A  B2 4E                                  mov DL, 78
    313050C  B8 1301                                mov AX, 1301h
    314
    315050F  80 3E 016Br FF                         cmp ignoreEnabled,  true
    3160514  74 07                                  je _greenF3
    317
    3180516                                         _redF3:
    3190516  B3 4F                                          mov BL, 01001111b ; red
    3200518  CD 10                                          int 10h
    321051A  EB 05 90                                       jmp _outFx
    322
    323051D                                         _greenF3:
    324051D  B3 2F                                          mov BL, 00101111b ; green
    325051F  CD 10                                          int 10h
    326
    3270521                                 _outFx:
    3280521  5A                             pop DX
    3290522  B4 02                                  mov AH, 02h
    3300524  CD 10                                  int 10h
    331
    3320526  07                             pop ES
    3330527  5D                             pop BP
    3340528  5A                             pop DX
```

```
3350529  59                              pop CX
336052A  5B                              pop BX
337052B  58                              pop AX
338052C  C3                              ret
339052D                                  changeFx endp
340
341                                 ;новый обработчик
342052D                             new_int9h proc  far
```

```
    343                                     ; сохраняем значения всех,  изменяемых регистров в стэке
    344052D  56                     push SI
    345052E  50                     push AX
    346052F  53                     push BX
    3470530  51                     push CX
    3480531  52                     push DX
    3490532  06                     push ES
    3500533  1E                     push DS
    3510534  55                     push BP
    352                                     ; синхронизируем CS и DS
    3530535  0E                     push CS
    3540536  1F                     pop     DS
    355
    3560537  B8 0040                        mov     AX, 40h ; 40h-сегмент,где хранятся  флаги сост-я
клавиатуры, кольц.          +
    357                         буфер ввода
    358053A  8E C0                          mov     ES, AX
    359053C  E4 60                          in      AL, 60h ; записываем в AL скан-код  нажатой клавиши
    360
    361053E  26: 8A 26  0017                mov     AH, ES:[17h]     ;  флаги клавиатуры
    3620543  80 E4 0F                       and     AH, 00001111b
    3630546  80 FC 04                       cmp     AH, 00000100b   ; был ли нажат ctrl?
    3640549  75 17                          jne     _test_Fx
    365                                     ; выгрузка
    366054B  B4 FF                                  mov AH, 0FFh
    367054D  B0 01                                  mov AL, 01h
    368054F  CD 2F                                  int 2Fh
    369                                             ; завершаем обработку нажатия
    370
    3710551  E4 61                                  in      AL, 61h ;контроллер состояния клавиатуры
    3720553  0C 80                                  or      AL, 10000000b   ;пометим, что клавишу нажали
    3730555  E6 61                                  out     61h, AL
    3740557  24 7F                                  and     AL, 01111111b   ;пометим, что клавишу отпустили
    3750559  E6 61                                  out     61h, AL
    376055B  B0 20                                  mov     AL, 20h
    377055D  E6 20                                  out     20h, AL ;отправим в контроллер прерываний
признак конца       +
    378                         прерывания
    379
    380                                             ; выходим
    381055F  E9 00A4                                jmp _quit
    382
    383
    384
    3850562                                 _test_Fx:
    3860562  2C 3A                          sub AL, 58  ; в AL теперь номер функциональной клавиши
    3870564                                 _F9:
    3880564  3C 09                                  cmp AL, 9 ; F9
    3890566  75 0A                                  jne _F1
```

```
    3900568  F6 16 0179r                                    not signaturePrintingEnabled
    391056C  E8 FF1C                                        call changeFx
    392056F  EB 2E 90                                       jmp _translate_or_ignore
    3930572                                            _F1:
    3940572  3C 01                                          cmp AL, 1 ; F1
    3950574  75 0D                                          jne _F2
    3960576  F6 16 017Ar                                    not cursiveEnabled
    397057A  E8 FF0E                                        call changeFx
    398057D  E8 01F8                                        call setCursive ; перевод символа в курсив  и обратно в
зависимости    от  +
    399                            флага cursiveEnabled
```

```
    4000580  EB 1D 90                                   jmp _translate_or_ignore
    4010583                                     _F2:
    4020583  3C 02                                     cmp AL, 2 ; F2
    4030585  75 0A                                     jne _F3
    4040587  F6 16 0178r                               not translateEnabled
    405058B  E8 FEFD                                   call changeFx
    406058E  EB 0F 90                                  jmp _translate_or_ignore
    4070591                                     _F3:
    4080591  3C 03                                     cmp AL, 3 ; F3
    4090593  75 0A                                     jne _translate_or_ignore
    4100595  F6 16 016Br                               not ignoreEnabled
    4110599  E8 FEEF                                   call changeFx
    412059C  EB 01 90                                  jmp _translate_or_ignore
    413
    414                                         ;игнорирование и перевод
    415059F                                     _translate_or_ignore:
    416
    417059F  9C                             pushf
    41805A0  2E: FF 1E 019Cr                           call dword ptr CS:[old_int9hOffset] ; вызываем стандартный
       обработчик прерывания
    41905A5  B8 0040                                   mov     AX, 40h         ; 40h-сегмент,где хранятся  флаги сост-
я клавы,кольц.   +
    420                             буфер ввода
    42105A8  8E C0                                     mov     ES, AX
    42205AA  26: 8B 1E 001C                            mov     BX, ES:[1Ch]    ; адрес хвоста
    42305AF  4B                             dec     BX      ; сместимся назад к последнему
    42405B0  4B                             dec     BX      ; введённому символу
    42505B1  83 FB 1E                                  cmp     BX, 1Eh ; не вышли  ли мы за пределы буфера?
    42605B4  73 03                                     jae     _go
    42705B6  BB 003C                                   mov     BX, 3Ch ; хвост вышел за пределы буфера, значит
последний введённый    +
    428                             символ
    429                                                         ; находится в конце буфера
    430
    43105B9                                 _go:
    43205B9  26: 8B 17                                 mov DX, ES:[BX] ; в DX 0 введённый  символ
    433                                         ;включен ли режим блокировки ввода?
    43405BC  80 3E 016Br FF                            cmp ignoreEnabled,  true
    43505C1  75 21                                     jne _check_translate
    436
    437                                         ; да, включен
    43805C3  BE 0000                                   mov SI, 0
    43905C6  B9 0068                                   mov CX, ignoredLength ;кол-во игнорируемых  символов
    440
    441                                         ; проверяем, присутствует ли текущий символ в списке
игнорируемых
    44205C9                                 _check_ignored:
    44305C9  3A 94 0103r                               cmp DL,ignoredChars[SI]
    44405CD  74 06                                     je _block
    44505CF  46                             inc SI
```

```
44605D0  E2 F7                          loop _check_ignored
44705D2  EB 10 90                               jmp _check_translate
448
449                                     ; блокируем
45005D5                                 _block:
45105D5  26: 89 07                      mov ES:[BX], AX
45205D8  33 C0                          xor AX, AX
45305DA  8A 84 0137r                            mov AL, replaceWith[SI]
45405DE  26: 89 07                              mov ES:[BX], AX ; замена символа
45505E1  EB 23 90                               jmp _quit
456
```

```
    45705E4                                 _check_translate:
    458                                             ; включен ли режим  перевода?
    45905E4  80 3E 0178r FF                         cmp translateEnabled, true
    46005E9  75 1B                                  jne _quit
    461
    462                                             ; да, включен
    46305EB  BE 0000                                mov SI, 0
    46405EE  B9 0006                                mov CX, translateLength ; кол-во символов для перевода
    465                                             ; проверяем, присутствует ли текущий символ в списке для
перевода
    46605F1                                         _check_translate_loop:
    46705F1  3A 94 016Cr                                    cmp DL, translateFrom[SI]
    46805F5  74 06                                          je _translate
    46905F7  46                                     inc SI
    47005F8  E2 F7                                  loop _check_translate_loop
    47105FA  EB 0A 90                               jmp _quit
    472
    473                                             ; переводим
    47405FD                                         _translate:
    47505FD  33 C0                                          xor AX, AX
    47605FF  8A 84 0172r                                    mov AL, translateTo[SI]
    4770603  26: 89 07                                      mov ES:[BX], AX ; замена символа
    478
    4790606                                 _quit:
    480                                             ; восстанавливаем все регистры
    4810606  5D                             pop BP
    4820607  1F                             pop     DS
    4830608  07                             pop     ES
    4840609  5A                             pop DX
    485060A  59                             pop CX
    486060B  5B                             pop     BX
    487060C  58                             pop     AX
    488060D  5E                             pop SI
    489060E  CF                             iret
    490060F                             new_int9h endp
    491
    492                                 ;=== Обработчик прерывания  int 1Ch ===;
    493                                 ;=== Вызывается каждые 55 мс ===;
    494060F                             new_int1Ch  proc far
    495060F  50                         push AX
    4960610  0E                         push CS
    4970611  1F                         pop DS
    498
    4990612  9C                         pushf
    5000613  2E: FF 1E  01A0r                   call dword  ptr CS:[old_int1ChOffset]
    501
    5020618  80 3E 0179r FF                     cmp signaturePrintingEnabled, true  ; если нажата управляющая клавиша (в
данном случае F)
    503061D  75 1D                              jne _notToPrint
```

```
504
505061F  81 3E 01AAr 0080                    cmp counter, printDelay*1000/55 + 1 ; если  кол-во "тактов"
   эквивалентно 5       +
506                              секундам
5070625  74 03                               je _letsPrint
508
5090627  EB 0E 90                            jmp _dontPrint
510
511062A                                      _letsPrint:
512062A  F6 16 0179r                                 not signaturePrintingEnabled
513062E  C7 06 01AAr 0000                            mov counter, 0
```

```
Turbo Assembler Version 3.1        05/09/24 16:09:46       Page 10
tsr.asm


    5140634  E8 0094                                    call printSignature
    515
    5160637                                        _dontPrint:
    5170637  83 06 01AAr 01                                add counter, 1
    518
    519063C                                    _notToPrint:
    520
    521063C  58                        pop AX
    522
    523063D  CF                        iret
    524063E                            new_int1Ch  endp
    525
    526                                ;=== Обработчик прерывания  int 2Fh ===;
    527                                ;=== Служит для:
    528                                ;===  1) проверки факта присутствия TSR в памяти (при AH=0FFh, AL=0)
    529                                ;===     будет возвращён AH='i' в случае, если TSR  уже загружен
    530                                ;===  2) выгрузки TSR из памяти (при AH=0FFh, AL=1)
    531                                ;===
    532063E                            new_int2Fh  proc
    533063E  80 FC FF                          cmp     AH, 0FFh         ;наша функция?
    5340641  75 0B                             jne     _2Fh_std         ;нет - на старый обработчик
    5350643  3C 00                             cmp     AL, 0   ;подфункция проверки, загружен ли резидент  в память?
    5360645  74 0C                             je      _already_installed
    5370647  3C 01                             cmp     AL, 1   ;подфункция выгрузки из памяти?
    5380649  74 0B                             je      _uninstall
    539064B  EB 01 90                          jmp     _2Fh_std         ;нет - на старый обработчик
    540
    541064E                            _2Fh_std:
    542064E  2E: FF 2E  01A4r                  jmp     dword ptr CS:[old_int2FhOffset] ;вызов старого обработчика
    543
    5440653                            _already_installed:
    5450653  B4 69                             mov     AH, 'i' ;вернём 'i', если резидент  загружен     в
память
    5460655  CF                                iret
    547
    5480656                            _uninstall:
    5490656  1E                        push    DS
    5500657  06                        push    ES
    5510658  52                        push    DX
    5520659  53                        push    BX
    553
    554065A  33 DB                             xor BX, BX
    555
    556                                        ; CS = ES,  для доступа к переменным
    557065C  0E                        push CS
    558065D  07                        pop ES
    559
    560065E  B8 2509                           mov     AX, 2509h
    5610661  26: 8B 16  019Cr                  mov DX, ES:old_int9hOffset          ; возвращаем вектор прерывания
```

```
5620666  26: 8E 1E  019Er           mov DS, ES:old_int9hSegment        ; на место
563066B  CD 21                        int     21h
564
565066D  B8 251C                      mov     AX, 251Ch
5660670  26: 8B 16  01A0r             mov DX, ES:old_int1ChOffset  ; возвращаем вектор прерывания
5670675  26: 8E 1E  01A2r           mov DS, ES:old_int1ChSegment        ; на место
568067A  CD 21                        int     21h
569
570067C  B8 252F                      mov     AX, 252Fh
```

```
571067F  26: 8B 16  01A4r              mov DX, ES:old_int2FhOffset  ; возвращаем вектор прерывания
5720684  26: 8E 1E  01A6r          mov DS, ES:old_int2FhSegment         ; на место
5730689  CD 21                         int     21h
574
575068B  2E: 8E 06  002C               mov     ES, CS:2Ch       ; загрузим в ES адрес окружения
5760690  B4 49                         mov     AH, 49h          ; выгрузим из памяти окружение
5770692  CD 21                         int     21h
5780694  72 0B                         jc _notRemove
579
5800696  0E                    push    CS
5810697  07                    pop     ES       ;в ES - адрес резидентной программы
5820698  B4 49                         mov     AH, 49h  ;выгрузим из памяти резидент
583069A  CD 21                         int     21h
584069C  72 03                         jc _notRemove
585069E  EB 15 90                      jmp _unloaded
586
58706A1                        _notRemove: ; не удалось выполнить  выгрузку
588                                ; вывод сообщения о неудачной выгрузке
58906A1  B4 03                         mov AH, 03h                              ; получаем позицию курсора
59006A3  CD 10                         int 10h
59106A5  BD 0466r                      lea BP, noRemoveMsg
59206A8  B9 001D                       mov CX, noRemoveMsg_length
59306AB  B3 07                         mov BL, 0111b
59406AD  B8 1301                       mov AX, 1301h
59506B0  CD 10                         int 10h
59606B2  EB 12 90                      jmp _2Fh_exit
597
59806B5                        _unloaded: ; выгрузка прошла успешно
599                                ; вывод сообщения об удачной выгрузке
60006B5  B4 03                         mov AH, 03h                              ; получаем позицию курсора
60106B7  CD 10                         int 10h
60206B9  BD 0455r                      lea BP, removedMsg
60306BC  B9 0011                       mov CX, removedMsg_length
60406BF  B3 07                         mov BL, 0111b
60506C1  B8 1301                       mov AX, 1301h
60606C4  CD 10                         int 10h
607
60806C6                        _2Fh_exit:
60906C6  5B                    pop BX
61006C7  5A                    pop     DX
61106C8  07                    pop     ES
61206C9  1F                    pop     DS
61306CA  CF                    iret
61406CB                        new_int2Fh  endp
615
616                            ;=== Процедура вывода подписи (ФИО, группа)
617                            ;=== Настраивается  значениями переменных в  начале исходника
618                            ;===
61906CB                        printSignature proc
```

```
62006CB  50                  push AX
62106CC  52                  push DX
62206CD  51                  push CX
62306CE  53                  push BX
62406CF  06                  push ES
62506D0  54                  push SP
62606D1  55                  push BP
62706D2  56                  push SI
```

```
    62806D3  57                     push DI
    629
    63006D4  33 C0                          xor AX, AX
    63106D6  33 DB                          xor BX, BX
    63206D8  33 D2                          xor DX, DX
    633
    63406DA  B4 03                          mov AH, 03h                                     ;чтение текущей позиции
курсора
    63506DC  CD 10                          int 10h
    63606DE  52                     push DX                                         ;помещаем информацию о
          +
    637                             положении курсора в стек
    638
    63906DF  83 3E 01ACr 00                 cmp printPos, 0
    64006E4  74 0E                          je _printTop
    641
    64206E6  83 3E 01ACr 01                 cmp printPos, 1
    64306EB  74 0E                          je _printCenter
    644
    64506ED  83 3E 01ACr 02                 cmp printPos, 2
    64606F2  74 0E                          je _printBottom
    647
    648                                     ;все числа  подобраны на глаз...
    64906F4                                 _printTop:
    65006F4  B6 00                                  mov DH, 0
    65106F6  B2 0F                                  mov DL, 15
    65206F8  EB 0F 90                               jmp _actualPrint
    653
    65406FB                                 _printCenter:
    65506FB  B6 09                                  mov DH, 9
    65606FD  B2 0F                                  mov DL, 15
    65706FF  EB 08 90                               jmp _actualPrint
    658
    6590702                                 _printBottom:
    6600702  B6 13                                  mov DH, 19
    6610704  B2 0F                                  mov DL, 15
    6620706  EB 01 90                               jmp _actualPrint
    663
    6640709                                 _actualPrint:
    6650709  B4 0F                                  mov AH, 0Fh                             ;чтение текущего
видеорежима. в+
    666                             BH - текущая страница
    667070B  CD 10                                  int 10h
    668
    669070D  0E                             push CS
    670070E  07                             pop ES                                  ;указываем ES на CS
    671
    672                                             ;вывод 'верхушки' таблицы
    673070F  52                             push DX
```

```
    6740710  BD 03A4r                            lea BP, tableTop                                ;помещаем в BP
указатель на    +
    675                              выводимую строку
    6760713  B9 002B                            mov CX, tableTop_length          ;в CX - длина строки
    6770716  B3 07                              mov BL, 0111b                             ;цвет выводимого текста
ref:       +
    678                              http://en.wikipedia.org/wiki/BIOS_color_attributes
    6790718  B8 1301                            mov AX, 1301h                                   ;AH=13h - номер
ф-ии, AL=01h - +
    680                              курсор перемещается при выводе каждого из символов  строки
    681071B  CD 10                              int 10h
    682071D  5A                             pop DX
    683071E  FE C6                              inc DH
    684
```

```
    685
    686                                     ;вывод первой линии
    6870720  52                         push DX
    6880721  BD 01AEr                       lea BP, signatureLine1
    6890724  B9 002B                        mov CX, Line1_length
    6900727  B3 07                          mov BL, 0111b
    6910729  B8 1301                        mov AX, 1301h
    692072C  CD 10                          int 10h
    693072E  5A                         pop DX
    694072F  FE C6                          inc DH
    695
    696                                     ;вывод второй линии
    6970731  52                         push DX
    6980732  BD 01D9r                       lea BP, signatureLine2
    6990735  B9 002C                        mov CX, Line2_length
    7000738  B3 07                          mov BL, 0111b
    701073A  B8 1301                        mov AX, 1301h
    702073D  CD 10                          int 10h
    703073F  5A                         pop DX
    7040740  FE C6                          inc DH
    705
    706                                     ;вывод третьей линии
    7070742  52                         push DX
    7080743  BD 0205r                       lea BP, signatureLine3
    7090746  B9 002D                        mov CX, Line3_length
    7100749  B3 07                          mov BL, 0111b
    711074B  B8 1301                        mov AX, 1301h
    712074E  CD 10                          int 10h
    7130750  5A                         pop DX
    7140751  FE C6                          inc DH
    715
    716                                     ;вывод 'низа' таблицы
    7170753  52                         push DX
    7180754  BD 03CFr                       lea BP, tableBottom
    7190757  B9 002B                        mov CX, tableBottom_length
    720075A  B3 07                          mov BL, 0111b
    721075C  B8 1301                        mov AX, 1301h
    722075F  CD 10                          int 10h
    7230761  5A                         pop DX
    7240762  FE C6                          inc DH
    725
    7260764  33 DB                          xor BX, BX
    7270766  5A                         pop DX                                  ;восстанавливаем из
стека      +
    728                             прежнее положение курсора
    7290767  B4 02                          mov AH, 02h                         ;меняем положение
курсора на   +
    730                             первоначальное
    7310769  CD 10                          int 10h
```

```
732076B  E8 FD1D                          call changeFx
733
734076E  5F                    pop DI
735076F  5E                    pop SI
7360770  5D                    pop BP
7370771  5C                    pop SP
7380772  07                    pop ES
7390773  5B                    pop BX
7400774  59                    pop CX
7410775  5A                    pop DX
```

```
    7420776  58                     pop AX
    743
    7440777  C3                     ret
    7450778                         printSignature endp
    746
    747                             ;=== Функция, которая в зависимости от флага cursiveEnabled меняет  начертание
символа с курсива+
    748                             на обычное и наоброт
    749                             ;=== Сама смена происходит  в процедуре changeFont,  а здесь подготавливаются
данные
    7500778                         setCursive  proc
    7510778  06                     push ES ; сохраняем регистры
    7520779  50                     push AX
    753077A  0E                     push CS
    754077B  07                     pop ES
    755
    756077C  80 3E 017Ar FF                 cmp cursiveEnabled, true
    7570781  75 30                          jne _restoreSymbol
    758                                     ; если флаг равен true, выполняем замену символа на курсивный вариант,
    759                                     ; предварительно сохраняя старый символ в savedSymbol
    760
    7610783  E8 004C                        call saveFont
    7620786  8A 0E 018Br                    mov CL, charToCursiveIndex
    763078A                         _shifTtable:
    764                                     ; мы получаем в BP  таблицу всех символов. адрес указывает на символ 0
    765                                     ; поэтому нуэно совершить сдвиг 16*X - где  X - код символа
    766078A  83 C5 10                       add BP, 16
    767078D  E2 FB                          loop _shiftTable
    768
    769                                     ; при savefont смещается регистр ES
    770                                     ; поэтому приходится делать такие махинации, чтобы
    771                                     ; записать  полученный элемент в savedSymbol
    772                                     ; swap(ES,  DS) и сохранение старого значения DS
    773078F  1E                     push DS
    7740790  58                     pop AX
    7750791  06                     push ES
    7760792  1F                     pop DS
    7770793  50                     push AX
    7780794  07                     pop ES
    7790795  50                     push AX
    780
    7810796  8B F5                          mov SI, BP
    7820798  BF 018Cr                       lea DI, savedSymbol
    783                                     ; сохраняем в переменную savedSymbol
    784                                     ; таблицу нужного символа
    785079B  B9 0010                        mov CX, 16
    786                                     ; movsb из  DS:SI в ES:DI
    787079E  F3> A4                         rep movsb
    788                                     ; исходные  позиции сегментов возвращены
```

```
78907A0  1F                    pop DS ; восстановление DS
790
791                                    ; заменим написание символа на курсив
79207A1  B9 0001                       mov CX, 1
79307A4  B6 00                         mov DH, 0
79407A6  8A 16 018Br                   mov DL, charToCursiveIndex
79507AA  BD 017Br                      lea BP, cursiveSymbol
79607AD  E8 0015                       call changeFont
79707B0  EB 10 90                      jmp _exitSetCursive
798
```

```
Turbo Assembler Version 3.1        05/09/24 16:09:46       Page 15
tsr.asm


    79907B3                         _restoreSymbol:
    800                                     ; если флаг равен 0, выполняем замену курсивного символа на старый
       вариант
    801
    80207B3  B9 0001                        mov CX, 1
    80307B6  B6 00                          mov DH, 0
    80407B8  8A 16 018Br                    mov DL, charToCursiveIndex
    80507BC  BD 018Cr                       lea bp, savedSymbol
    80607BF  E8 0003                        call changeFont
    807
    80807C2                         _exitSetCursive:
    80907C2  58                     pop AX
    81007C3  07                     pop ES
    81107C4  C3                     ret
    81207C5                         setCursive  endp
    813
    814                             ;=== Функция смены  начертания символа (курсив/нормальное)
    815                             ;===
    816                             ; *** входные данные
    817                             ; DL = номер символа для замены
    818                             ; CX = Кол-во символов заменяемых изображений символов
    819                             ; (начиная  с символа указанного в DX)
    820                             ; ES:bp = адрес таблицы
    821                             ;
    822                             ; *** описание работы процедуры
    823                             ; Происходит вызов  int 10h (видеосервис)
    824                             ; с функцией AH = 11h (функции знакогенератора)
    825                             ; Параметр  AL = 0 сообщает, что будет заменено изображение
    826                             ; символа для текущего шрифта
    827                             ; В случаях, когда  AL = 1 или 2, будет заменено изображение
    828                             ; только для опредленного шрифта (8x14 и 8x8 соответственно)
    829                             ; Параметр  BH = 0Eh сообщает, что на опредление каждого изображения символа
    830                             ; расходуется по 14 байт (режим 8x14 бит как раз 14 байт)
    831                             ; Параметр  BL = 0 - блок шрифта для загрузки (от 0  до 4)
    832                             ;
    833                             ; *** результат
    834                             ; изображение указанного(ых) символа(ов) будет заменено
    835                             ; на предложенное пользователем.
    836                             ; Изменению подвергнутся все символы, находящиеся на экране,
    837                             ; то есть если изображение  заменено, старый вариант нигде уже не проявится
    838
    83907C5                         changeFont  proc
    84007C5  50                     push AX
    84107C6  53                     push BX
    84207C7  B8 1100                        mov AX, 1100h
    84307CA  BB 1000                        mov BX, 1000h
    84407CD  CD 10                          int 10h
    84507CF  58                     pop AX
    84607D0  5B                     pop BX
```

```
84707D1  C3                    ret
84807D2                        changeFont  endp
849
850                            ;=== Функция сохранения нормального начертания символа
851                            ;===
852                            ; *** входные данные
853                            ; BH - тип  возвращаемой символьной  таблицы
854                            ;   0 - таблица из  int 1fh
855                            ;   1 - таблица из  int 44h
```

```
    856                             ;   2-5 - таблица из 8x14,  8x8, 8x8 (top),  9x14
    857                             ;   6 - 8x16
    858                             ;
    859                             ; *** описание работы процедуры
    860                             ; Происходит вызов  int 10h (видеосервис)
    861                             ; с функцией AH = 11h (функции знакогенератора)
    862                             ; Параметр  AL = 30 - подфункция получения информации о EGA
    863                             ;
    864                             ; *** результат
    865                             ; в ES:BP находится таблица символов (полная)
    866                             ; в CX находится байт на символ
    867                             ; в DL количество экранных  строк
    868                             ; ВАЖНО! Происходит сдвиг регистра  ES
    869                             ; ( ES становится равным C000h )
    870
    87107D2                         saveFont proc
    87207D2  50                     push AX
    87307D3  53                     push BX
    87407D4  B8 1130                        mov AX, 1130h
    87507D7  BB 0600                        mov BX, 0600h
    87607DA  CD 10                          int 10h
    87707DC  58                     pop AX
    87807DD  5B                     pop BX
    87907DE  C3                     ret
    88007DF                         saveFont endp
    881
    882
    883                             ;=== Отсюда начинается выполнение основной  части программы ===;
    884                             ;===
    88507DF                         _initTSR:                               ; старт резидента
    88607DF  B4 03                          mov AH, 03h
    88707E1  CD 10                          int 10h
    88807E3  52                     push DX
    88907E4  B4 00                          mov AH,00h                                      ; установка видеорежима
(83h  текст    +
    890                             80x25  16/8  CGA,EGA  b800   Comp,RGB,Enhanced), без очистки экрана
    89107E6  B0 83                          mov AL,83h
    89207E8  CD 10                          int 10h
    89307EA  5A                     pop DX
    89407EB  B4 02                          mov AH, 02h
    89507ED  CD 10                          int 10h
    896
    897
    89807EF  E8 0095                    call commandParamsParser
    89907F2  B8 3509                        mov AX,3509h                    ; получить  в ES:BX вектор 09
    90007F5  CD 21                      int 21h                 ; прерывания
    901
    90207F7  80 3E 01A9r FF                 cmp notLoadTSR, true                    ; если была выведена справка
    90307FC  74 0E                          je _exit_tmp                                    ; просто
```

выходим

```
    904                                 ; === Удаление резидента из памяти  ===
    905                                 ;cmp unloadTSR, true
    906                                  ;      je _removingOnParameter
    907                                  ;      jmp _notRemovingNow
    90807FE                             _removingOnParameter:
    90907FE  B4 FF                              mov AH, 0FFh
    9100800  B0 00                              mov AL, 0
    9110802  CD 2F                                      int 2Fh
    9120804  80 FC 69                           cmp AH, 'i'  ; проверка того, загружена ли  уже программа
```

```
9130807  74 62                                 je _remove
914
9150809                                  _notRemovingNow:
916
9170809  EB 04 90                        jmp _tmp
918
919080C                                  _exit_tmp:
920080C  EB 77 90                               jmp _exit
921
922080F                                  _tmp:
923080F  06                         push ES
9240810  A1 002C                        mov AX, DS:[2Ch]                ; psp
9250813  8E C0                          mov ES, AX
9260815  B4 49                          mov AH, 49h                     ; хватит памяти чтоб остаться
9270817  CD 21                          int 21h                 ; резидентом?
9280819  07                     pop ES
929081A  72 62                          jc _notMem                      ; не хватило -  выходим
930
931                                      ;== int 09h ==;
932
933081C  2E: 89 1E  019Cr               mov     word ptr CS:old_int9hOffset, BX
9340821  2E: 8C 06  019Er               mov     word ptr CS:old_int9hSegment, ES
9350826  B8 2509                        mov AX, 2509h                   ; установим вектор на 09
9360829  BA 052Dr                       mov DX, offset  new_int9h            ; прерывание
937082C  CD 21                          int 21h
938
939                                      ;== int 1Ch ==;
940082E  B8 351C                        mov AX,351Ch                    ; получить  в ES:BX вектор 1C
9410831  CD 21                          int 21h                 ; прерывания
9420833  2E: 89 1E  01A0r               mov     word ptr CS:old_int1ChOffset, BX
9430838  2E: 8C 06  01A2r               mov     word ptr CS:old_int1ChSegment, ES
944083D  B8 251C                        mov AX, 251Ch                   ; установим вектор  на 1C
9450840  BA 060Fr                       mov DX, offset new_int1Ch            ; прерывание
9460843  CD 21                          int 21h
947
948                                      ;== int 2Fh ==;
9490845  B8 352F                        mov AX,352Fh                    ; получить  в ES:BX вектор 1C
9500848  CD 21                          int 21h                 ; прерывания
951084A  2E: 89 1E  01A4r               mov     word ptr CS:old_int2FhOffset, BX
952084F  2E: 8C 06  01A6r               mov     word ptr CS:old_int2FhSegment, ES
9530854  B8 252F                        mov AX, 252Fh                   ; установим вектор  на 2F
9540857  BA 063Er                       mov DX, offset new_int2Fh            ; прерывание
955085A  CD 21                          int 21h
956
957085C  E8 FC2C                        call changeFx
958085F  BA 03FAr                       mov DX, offset  installedMsg         ; выводим что все ок
9590862  B4 09                          mov AH, 9
9600864  CD 21                          int 21h
9610866  BA 07DFr                       mov DX, offset  _initTSR       ; остаемся в памяти резидентом
```

```
9620869  CD 27                    int 27h                 ; и выходим
963                               ; конец основной программы
964086B                        _remove: ;  выгрузка программы из памяти
965086B  B4 FF                          mov AH, 0FFh
966086D  B0 01                          mov AL, 1
967086F  CD 2F                          int 2Fh
9680871  EB 12 90                       jmp _exit
9690874                        _alreadyInstalled:
```

```
Turbo Assembler Version 3.1        05/09/24 16:09:46       Page 18
tsr.asm



    9700874  B4 09                          mov AH, 09h
    9710876  BA 040Dr                       lea DX, alreadyInstalledMsg
    9720879  CD 21                          int 21h
    973087B  EB 08 90                       jmp _exit
    974087E                          _notMem:                            ; не хватает памяти, чтобы остаться
резидентом
    975087E  BA 0423r                    mov DX, offset  noMemMsg
    9760881  B4 09                       mov AH, 9
    9770883  CD 21                       int 21h
    9780885                          _exit:                               ; выход
    9790885  CD 20                       int 20h
    980
    981                              ;=== Процедура проверки параметров  ком. строки ===;
    982                              ;===
    9830887                          commandParamsParser proc
    9840887  0E                      push CS
    9850888  07                      pop ES
    9860889  C6 06 01A9r 00                mov notLoadTSR, 0
    987
    988088E  BE 0080                        mov SI, 80h                     ;SI=смещение командной строки.
    9890891  AC                      lodsb                                          ;Получим кол-во символов.
    9900892  0A C0                          or AL, AL                               ;Если 0 символов введено,
    9910894  74 3A                          jz _exitHelp                            ;то все в порядке.
    992
    9930896                                 _nextChar:
    994
    9950896  46                      inc SI                                         ;Теперь SI  указывает на первый
символ  +
    996                              строки.
    997
    9980897  80 3C 0D                       cmp [SI], BYTE ptr  13
    999089A  74 34                          je _exitHelp
   1000
   1001
   1002089C  AD                             lodsw                                  ;Получаем два символа
   1003089D  3D 3F2F                               cmp AX, '?/'                            ;Это '/?' (данные
расположены в      +
   1004                              обратном порядк, т.е. AL:AH вместо  AH:AL)
   100508A0  74 03                              je _question
   1006
   1007                                         ;cmp AH, '/'
   1008                                         ;je _errorParam
   1009
   101008A2  EB 2C 90                           jmp _exitHelp
   1011
   101208A5                                 _question:
   1013                                         ; вывод строки помощи
   101408A5  B4 03                                      mov AH,03
   101508A7  CD 10                                      int 10h
```

```
   101608A9  BD 0232r                                    lea BP, helpMsg
   101708AC  B9 014D                                     mov CX, helpMsg_length
   101808AF  B3 07                                       mov BL, 0111b
   101908B1  B8 1301                                     mov AX, 1301h
   102008B4  CD 10                                       int 10h
   1021                                          ; конец вывода строки помощи
   102208B6  F6 16 01A9r                         not notLoadTSR          ;флаг того, что необходимо  не
загружать резидент
   102308BA  EB DA                               jmp _nextChar
   1024
   102508BC                                  _finishTSR:
   1026
```

```
Turbo Assembler Version 3.1        05/09/24 16:09:46       Page 19
tsr.asm


   102708BC  EB 12 90                    jmp _exitHelp
   1028
   102908BF                              _errorParam:
   1030                                          ;вывод строки
   103108BF  B4 03                                       mov AH,03
   103208C1  CD 10                                       int 10h
   103308C3  BD 037Fr                                    lea BP, CS:errorParamMsg
   103408C6  B9 0025                                     mov CX, errorParamMsg_length
   103508C9  B3 07                                       mov BL, 0111b
   103608CB  B8 1301                                     mov AX, 1301h
   103708CE  CD 10                                       int 10h
   1038                                          ;конец вывода строки
   103908D0                              _exitHelp:
   104008D0  C3                        ret
   104108D1                            commandParamsParser endp
   1042
   104308D1                            code ends
   1044                                end _start
```

```
Turbo Assembler Version 3.1        05/09/24 16:09:46       Page 20
Symbol Table



Symbol Name                      Type   Value

??DATE                           Text   "05/09/24"
??FILENAME                       Text   "tsr     "
??TIME                           Text   "16:09:46"
??VERSION                        Number 030A
@CPU                             Text   0101H
@CURSEG                          Text   CODE
@FILENAME                        Text   TSR
@WORDSIZE                        Text   2
ALREADYINSTALLEDMSG              Byte   CODE:040D
CHANGEFONT                       Near   CODE:07C5
CHANGEFX                         Near   CODE:048B
CHARTOCURSIVEINDEX               Byte   CODE:018B
COMMANDPARAMSPARSER              Near   CODE:0887
COUNTER                          Word   CODE:01AA
CURSIVEENABLED                   Byte   CODE:017A
CURSIVESYMBOL                    Byte   CODE:017B
ERRORPARAMMSG                    Byte   CODE:037F
ERRORPARAMMSG_LENGTH             Number 0025
F1_TXT                           Byte   CODE:0485
F2_TXT                           Byte   CODE:0487
F3_TXT                           Byte   CODE:0489
F9_TXT                           Byte   CODE:0483
FX_LENGTH                        Number 0002
HELPMSG                          Byte   CODE:0232
HELPMSG_LENGTH                   Number 014D
IGNOREDCHARS                     Byte   CODE:0103
IGNOREDLENGTH                    Number 0068
IGNOREENABLED                    Byte   CODE:016B
INSTALLEDMSG                     Byte   CODE:03FA
LINE1_LENGTH                     Number 002B
LINE2_LENGTH                     Number 002C
LINE3_LENGTH                     Number 002D
NEW_INT1CH                       Far    CODE:060F
NEW_INT2FH                       Near   CODE:063E
NEW_INT9H                        Far    CODE:052D
NOMEMMSG                         Byte   CODE:0423
NOREMOVEMSG                      Byte   CODE:0466
NOREMOVEMSG_LENGTH               Number 001D
NOTINSTALLEDMSG                  Byte   CODE:0437
NOTLOADTSR                       Byte   CODE:01A9
OLD_INT1CHOFFSET                 Word   CODE:01A0
OLD_INT1CHSEGMENT                Word   CODE:01A2
OLD_INT2FHOFFSET                 Word   CODE:01A4
OLD_INT2FHSEGMENT                Word   CODE:01A6
OLD_INT9HOFFSET                  Word   CODE:019C
OLD_INT9HSEGMENT                 Word   CODE:019E
```

```
PRINTDELAY                      Number 0007
PRINTPOS                        Word   CODE:01AC
PRINTSIGNATURE                  Near   CODE:06CB
REMOVEDMSG                      Byte   CODE:0455
REMOVEDMSG_LENGTH               Number 0011
REPLACEWITH                     Byte   CODE:0137
SAVEDSYMBOL                     Byte   CODE:018C
SAVEFONT                        Near   CODE:07D2
```

Turbo Assembler Version 3.1 05/09/24 16:09:46 Page 21
Symbol Table

```
SETCURSIVE                      Near   CODE:0778
SIGNATURELINE1                  Byte   CODE:01AE
SIGNATURELINE2                  Byte   CODE:01D9
SIGNATURELINE3                  Byte   CODE:0205
SIGNATUREPRINTINGENABLED        Byte   CODE:0179
TABLEBOTTOM                     Byte   CODE:03CF
TABLEBOTTOM_LENGTH              Number 002B
TABLETOP                        Byte   CODE:03A4
TABLETOP_LENGTH                 Number 002B
TRANSLATEENABLED                Byte   CODE:0178
TRANSLATEFROM                   Byte   CODE:016C
TRANSLATELENGTH                 Number 0006
TRANSLATETO                     Byte   CODE:0172
TRUE                            Number 00FF
UNLOADTSR                       Byte   CODE:01A8
_2FH_EXIT                       Near   CODE:06C6
_2FH_STD                        Near   CODE:064E
_ACTUALPRINT                    Near   CODE:0709
_ALREADYINSTALLED               Near   CODE:0874
_ALREADY_INSTALLED              Near   CODE:0653
_BLOCK                          Near   CODE:05D5
_CHECKF1                        Near   CODE:04BE
_CHECKF2                        Near   CODE:04DF
_CHECKF3                        Near   CODE:0500
_CHECKF9                        Near   CODE:049A
_CHECK_IGNORED                  Near   CODE:05C9
_CHECK_TRANSLATE                Near   CODE:05E4
_CHECK_TRANSLATE_LOOP           Near   CODE:05F1
_DONTPRINT                      Near   CODE:0637
_ERRORPARAM                     Near   CODE:08BF
_EXIT                           Near   CODE:0885
_EXITHELP                       Near   CODE:08D0
_EXITSETCURSIVE                 Near   CODE:07C2
_EXIT_TMP                       Near   CODE:080C
_F1                             Near   CODE:0572
_F2                             Near   CODE:0583
_F3                             Near   CODE:0591
_F9                             Near   CODE:0564
_FINISHTSR                      Near   CODE:08BC
_GO                             Near   CODE:05B9
_GREENF1                        Near   CODE:04DB
_GREENF2                        Near   CODE:04FC
_GREENF3                        Near   CODE:051D
_GREENF9                        Near   CODE:04B7
_INITTSR                        Near   CODE:07DF
_LETSPRINT                      Near   CODE:062A
_NEXTCHAR                       Near   CODE:0896
_NOTMEM                         Near   CODE:087E
_NOTREMOVE                      Near   CODE:06A1
```

```
_NOTREMOVINGNOW                 Near   CODE:0809
_NOTTOPRINT                     Near   CODE:063C
_OUTFX                          Near   CODE:0521
_PRINTBOTTOM                    Near   CODE:0702
_PRINTCENTER                    Near   CODE:06FB
_PRINTTOP                       Near   CODE:06F4
_QUESTION                       Near   CODE:08A5
_QUIT                           Near   CODE:0606
```

```
Turbo Assembler Version 3.1        05/09/24 16:09:46       Page 22
Symbol Table



_REDF1                           Near   CODE:04D4
_REDF2                           Near   CODE:04F5
_REDF3                           Near   CODE:0516
_REDF9                           Near   CODE:04B0
_REMOVE                          Near   CODE:086B
_REMOVINGONPARAMETER             Near   CODE:07FE
_RESTORESYMBOL                   Near   CODE:07B3
_SHIFTTABLE                      Near   CODE:078A
_START                           Near   CODE:0100
_TEST_FX                         Near   CODE:0562
_TMP                             Near   CODE:080F
_TRANSLATE                       Near   CODE:05FD
_TRANSLATE_OR_IGNORE             Near   CODE:059F
_UNINSTALL                       Near   CODE:0656
_UNLOADED                        Near   CODE:06B5

Groups & Segments                Bit Size Align  Combine Class

CODE                             16  08D1 Para    none    CODE
```