



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1 **по курсу «Операционные системы»**

«Дизассемблирование INT 8h»

Студент _____ Маслова Марина Дмитриевна
фамилия, имя, отчество

Группа _____ ИУ7-53Б

Оценка (баллы) _____

Преподаватель _____ Рязанова Наталья Юрьевна

2021 г.

Цель работы

Знакомство со средством дизассемблирования Sourcer, получение дизассемблерного кода ядра операционной системы Windows на примере обработчика прерывания Int 8h в virtual mode – специальном режиме защищенного режима, который эмулирует реальный режим работы вычислительной системы на базе процессоров Intel.

Задание

Используя Sourcer получить дизассемблерный код обработчика аппаратного прерывания от системного таймера Int 8h. На основе полученного кода составить алгоритм работы обработчика Int 8h.

Полученный дизассемблерный код

Листинг 1: Обработчик INT 8h

```
1 ; вызов подпрограммы sub_1
2 020A:0746 E8 0070          call     sub_1             ; (07B9)
3 ; сохранение значений регистров ES, DS, AX, DX
4 020A:0749 06              push     es
5 020A:074A 1E              push     ds
6 020A:074B 50              push     ax
7 020A:074C 52              push     dx
8 ; загрузка в DS 0040h
9 020A:074D B8 0040          mov     ax,40h
10 020A:0750 8E D8           mov     ds,ax
11 ; AX = 0; ES = 0
12 020A:0752 33 C0           xor     ax,ax             ; Zero register
13 020A:0754 8E C0           mov     es,ax
14 ; инкремент счетчика таймера, находящегося по адресу 0040:006C
15 ; (2 младших байт)
16 020A:0756 FF 06 006C         inc     word ptr ds:[6Ch] ; (0040:006C=2E0h)
17 020A:075A 75 04           jnz     loc_1             ; Jump if not zero
18 ; при переполнении 2 младших байт счетчика таймера
19 ; инкремент 2 старших байт счетчика таймера
20 020A:075C FF 06 006E         inc     word ptr ds:[6Eh] ; (0040:006E=16h)
21 020A:0760          loc_1: ; xref 020A:075A
22 ; проверка, прошли ли 24 часа:
23 ; 0040:006E == 18h (24) и 0040:006C == B0h (176)
24 020A:0760 83 3E 006E 18     cmp     word ptr ds:[6Eh],18h ; (0040:006E=16h)
25 020A:0765 75 15           jne     loc_2             ; Jump if not equal
26 020A:0767 81 3E 006C 00B0    cmp     word ptr ds:[6Ch],0B0h ; (0040:006C=2E0h)
27 020A:076D 75 0D           jne     loc_2             ; Jump if not equal
28 ; сброс счетчиков времени при наступлении нового дня
29 020A:076F A3 006E         mov     word ptr ds:[6Eh],ax ; (0040:006E=16h)
30 020A:0772 A3 006C         mov     word ptr ds:[6Ch],ax ; (0040:006C=2E0h)
31 ; установка флага прошедших суток по адресу 0040:0070
32 020A:0775 C6 06 0070 01     mov     byte ptr ds:[70h],1 ; (0040:0070=0)
33 ; AL = 8
34 020A:077A 0C 08           or      al,8
35 020A:077C          loc_2: ; xref 020A:0765, 076D
36 ; сохранение значения регистра AX
37 020A:077C 50              push     ax
38 ; декремент счетчика времени, оставшегося до остановки моторчика дисковод
39 020A:077D FE 0E 0040         dec     byte ptr ds:[40h] ; (0040:0040=94h)
40 020A:0781 75 0B           jnz     loc_3             ; Jump if not zero
41 ; установка флага отключения моторчика дисковод
42 020A:0783 80 26 003F F0     and     byte ptr ds:[3Fh],0F0h ; (0040:003F=0)
43 ; посылка команды в порт дисковод на отключение моторчика дисковод
44 020A:0788 B0 0C           mov     al,0Ch
```

```

45 020A:078A BA 03F2      mov dx,3F2h
46 020A:078D EE          out dx,al      ; port 3F2h, dsk0 contrl output
47 020A:078E          loc_3:                      ; xref 020A:0781
48 ; восстановление регистра AX
49 020A:078E 58          pop ax
50 ; проверка, установлен ли PF
51 020A:078F F7 06 0314 0004 test word ptr ds:[314h],4 ; (0040:0314=3200h)
52 020A:0795 75 0C      jnz loc_4      ; Jump if not zero
53 ; загрузка младшего байта регистра флагов в AH
54 020A:0797 9F          lahf                      ; Load ah from flags
55 ; обмен AH и AL
56 020A:0798 86 E0      xchg ah,al
57 ; сохранение значения регистра AX
58 020A:079A 50          push ax
59 ; вызов прерывания 1Ch через таблицу векторов прерываний;
60 020A:079B 26: FF 1E 0070 call dword ptr es:[70h] ; (0000:0070=6ADh)
61 020A:07A0 EB 03      jmp short loc_5 ; (07A5)
62 020A:07A2 90          nop
63 020A:07A3          loc_4:                      ; xref 020A:0795
64 ; вызов прерывания 1Ch
65 020A:07A3 CD 1C      int 1Ch      ; Timer break (call each 18.2ms)
66 020A:07A5          loc_5:                      ; xref 020A:07A0
67 020A:07A5 E8 0011     call sub_1    ; (07B9)
68 ; сброс контроллера прерываний
69 020A:07A8 B0 20      mov al,20h    ; ' '
70 020A:07AA E6 20      out 20h,al    ; port 20h, 8259-1 int command
71                                     ; al = 20h, end of interrupt
72 ; восстановление регистров
73 020A:07AC 5A          pop dx
74 020A:07AD 58          pop ax
75 020A:07AE 1F          pop ds
76 020A:07AF 07          pop es
77 ; (020A:07B0 - 164h = 020A:064Ch)
78 020A:07B0 E9 FE99     jmp $-164h
79 ; ...
80 ; возврат из прерывания
81 020A:06AC CF          iret                      ; Interrupt return
82
83          sub_1      proc near
84 ; сохранение значений регистров DS, AX

```

Листинг 2: Подпрограмма sub_1

```

1 020A:07BA 50                push    ax
2 ; загрузка в DS 0040h
3 020A:07BB B8 0040          mov     ax,40h
4 020A:07BE 8E D8            mov     ds,ax
5 ; загрузка младшего байта регистра флагов в AH
6 020A:07C0 9F                lahf                    ; Load ah from flags
7 ; проверка флага DF и старшего бита IOPL
8 020A:07C1 F7 06 0314 2400    test     word ptr ds:[314h],2400h
9                                     ; (0040:0314=3200h)
10 020A:07C7 75 0C            jnz     loc_7            ; Jump if not zero
11 ; сброс IF в 0040:0314h
12 020A:07C9 F0> 81 26 0314 FDFD lock and word ptr ds:[314h],0FDFDh
13                                     ; (0040:0314=3200h)
14 020A:07D0                loc_6:                    ; xref 020A:07D6
15 ; загрузка AH в младший байт регистра флагов
16 020A:07D0 9E                sahf                    ; Store ah into flags
17 ; восстановление регистров AX, DS
18 020A:07D1 58                pop     ax
19 020A:07D2 1F                pop     ds
20 020A:07D3 EB 03            jmp     short loc_ret_8 ; (07D8)
21 020A:07D5                loc_7:                    ; xref 020A:07C7
22 ; запрет маскируемых прерываний
23 020A:07D5 FA                cli                    ; Disable interrupts
24 020A:07D6 EB F8            jmp     short loc_6      ; (07D0)
25
26 020A:07D8                loc_ret_8:                ; xref 020A:07D3
27 020A:07D8 C3                retn
28                sub_1      endp

```

Алгоритм

Заключение