



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"

Студент _____ Сукочева Алис

Группа _____ ИУ7-53Б

Название предприятия _____ МГТУ им. Н. Э. Баумана, каф. ИУ7

Тема _____ Исследование прерывания INT 8H

Студент:	_____	Сукочева А.
	подпись, дата	Фамилия, И.О.
Преподаватель:	_____	Рязанова Н.Ю.
	подпись, дата	Фамилия, И. О.

ТЗ:

Получение с помощью дизассемблера `sg.exe` кода прерывания `INT 8h`. Студент составляет отчет, состоящий из полученного кода `INT 8h`, алгоритма 8-го прерывания и алгоритма подпрограммы, которая вызывается в начале и в конце кода 8-го прерывания, в графическом виде по ГОСТу. Студент защищает работу.

Листинг 1 — Код прерывания `INT 8h`

1	020A:0746	E8 0070	;* call sub_1 ; (07B9)
2	020A:0746	E8 70 00	db 0E8h, 70h, 00h
3	020A:0749	06	push es
4	020A:074A	1E	push ds
5	020A:074B	50	push ax
6	020A:074C	52	push dx
7	020A:074D	B8 0040	mov ax,40h
8	020A:0750	8E D8	mov ds,ax
9	020A:0752	33 C0	xor ax,ax ; Zero register
10	020A:0754	8E C0	mov es,ax
11	020A:0756	FF 06 006C	inc word ptr ds:[6Ch] ;
		(0040:006C=0A1Dh)	
12	020A:075A	75 04	jnz loc_1 ; Jump if not zero
13	020A:075C	FF 06 006E	inc word ptr ds:[6Eh] ;
		(0040:006E=0Ah)	
14	020A:0760	loc_1:	
15	020A:0760	83 3E 006E 18	cmp word ptr ds:[6Eh],18h ;
		(0040:006E=0Ah)	
16	020A:0765	75 15	jne loc_2 ; Jump if not equal
17	020A:0767	81 3E 006C 00B0	cmp word ptr ds:[6Ch],0B0h ;
		(0040:006C=0A1Dh)	
18	020A:076D	75 0D	jne loc_2 ; Jump if not equal
19	020A:076F	A3 006E	mov word ptr ds:[6Eh],ax ;
		(0040:006E=0Ah)	
20	020A:0772	A3 006C	mov word ptr ds:[6Ch],ax ;
		(0040:006C=0A1Dh)	
21	020A:0775	C6 06 0070 01	mov byte ptr ds:[70h],1 ; (0040:0070=0)
22	020A:077A	0C 08	or al,8
23	020A:077C	loc_2:	
24	020A:077C	50	push ax
25	020A:077D	FE 0E 0040	dec byte ptr ds:[40h] ;
		(0040:0040=2Ch)	
26	020A:0781	75 0B	jnz loc_3 ; Jump if not zero
27	020A:0783	80 26 003F F0	and byte ptr ds:[3Fh],0F0h ;
		(0040:003F=0)	
28	020A:0788	B0 0C	mov al,0Ch
29	020A:078A	BA 03F2	mov dx,3F2h

```

30 020A:078D EE out dx,al ; port 3F2h, dsk0
    contrl output
31 020A:078E loc_3:
32 020A:078E 58 pop ax
33 020A:078F F7 06 0314 0004 test word ptr ds:[314h],4 ;
    (0040:0314=3200h)
34 020A:0795 75 0C jnz loc_4 ; Jump if not zero
35 020A:0797 9F lahf ; Load ah from flags
36 020A:0798 86 E0 xchg ah,al
37 020A:079A 50 push ax
38 020A:079B 26: FF 1E 0070 call dword ptr es:[70h] ;
    (0000:0070=6ADh)
39 020A:07A0 EB 03 jmp short loc_5 ; (07A5)
40 020A:07A2 90 nop
41 020A:07A3 loc_4:
42 020A:07A3 CD 1C int 1Ch ; Timer break (call each
    18.2ms)
43 020A:07A5 loc_5:
44 020A:07A5 E8 0011 call sub_1 ; (07B9)
45 020A:07A8 B0 20 mov al,20h ; ' '
46 020A:07AA E6 20 out 20h,al ; port 20h, 8259-1 int
    command
47 ; al = 20h, end of interrupt
48 020A:07AC 5A pop dx
49 020A:07AD 58 pop ax
50 020A:07AE 1F pop ds
51 020A:07AF 07 pop es
52 020A:07B0 E9 FE99 jmp $-164h
53 020A:07B3 C4 db 0C4h
54 ;* No entry point to code
55 020A:07B4 C4 0E 93E9 les cx,dword ptr ds:[93E9h] ;
    (0000:93E9=0A1A1h) Load 32 bit ptr
56 020A:07B8 FE db 0FEh
57
58 Temp.lst Sourcer v5.10 14-Sep-20 10:02 am Page 2
59
60 ; SUBROUTINE
61
62 sub_1 proc near
63 020A:07B9 1E push ds
64 020A:07BA 50 push ax
65 020A:07BB B8 0040 mov ax,40h
66 020A:07BE 8E D8 mov ds,ax
67 020A:07C0 9F lahf ; Load ah from flags
68 020A:07C1 F7 06 0314 2400 test word ptr ds:[314h],2400h ;
    (0040:0314=3200h)
69 020A:07C7 75 0C jnz loc_7 ; Jump if not zero

```

70	020A:07C9	F0> 81 26 0314 FDFF	lock and word
		ptr ds:[314h],0FDFFh ; (0040:0314=3200h)	
71	020A:07D0	loc_6:	
72	020A:07D0	9E sahf	; Store ah into flags
73	020A:07D1	58 pop ax	
74	020A:07D2	1F pop ds	
75	020A:07D3	EB 03 jmp short loc_8	; (07D8)
76	020A:07D5	loc_7:	
77	020A:07D5	FA cli	; Disable interrupts
78	020A:07D6	EB F8 jmp short loc_6	; (07D0)
79	020A:07D8	loc_8:	
80	020A:07D8	C3 retn	
81	sub_1	endp	