



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»

Лабораторная работа № 1 по дисциплине "Операционные системы"

Тема Дизассемблирование INT 8h.

Студент Калашков П. А.

Группа ИУ7-56Б

Преподаватель Рязанова Н.Ю.

Москва, 2022

Листинг обработчика INT 8h

```
1  ;; вызов sub_1
2  020A:0746  E8 0070          call sub_1 ; (07B9)
3  ;; Сохранение 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  ;; DS = 0040
9  020A:074D  B8 0040          mov ax,40h
10 020A:0750  8E D8          mov ds,ax
11 ;; AX = 0
12 020A:0752  33 C0          xor ax,ax ; Zero register
13 020A:0754  8E C0          mov es,ax
14 ;; 0040:006Ch - адрес счетчика таймера
15 020A:0756  FF 06 006C      inc word ptr ds:[6Ch] ; (0040:006C=5AEBh)
16 020A:075A  75 04          jnz loc_1 ; Jump if not zero
17 ;; 0040:006Eh - старшие 2 байта счетчика таймера
18 020A:075C  FF 06 006E      inc word ptr ds:[6Eh] ; (0040:006E=2)
19 020A:0760          loc_1:
20 ;; Проверка: 0040:006Eh == 18h (24) И 0040:006Ch == B0h (176)
21 ;; Можно убедиться в том, что: 18h << 16 + B0h = 24 * 60 * 60 * freq,
22 ;; где freq - кол-во раз, которое вызывается таймер в секунду.
23 ;; Таким образом из того, что условие выполняется, следует, что прошли сутки.
24 020A:0760  83 3E 006E 18    cmp word ptr ds:[6Eh],18h ; (0040:006E=2)
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=5AEBh)
27 020A:076D  75 0D jne loc_2 ; Jump if not equal
28 ;; Зануление счетчика (старшего слова и младшего слова)
29 020A:076F  A3 006E          mov word ptr ds:[6Eh],ax ; (0040:006E=2)
30 020A:0772  A3 006C          mov word ptr ds:[6Ch],ax ; (0040:006C=5AEBh)
31 ;; Прошло более 24 часов, занесение значения 1 в 0040:0070
32 020A:0775  C6 06 0070 01    mov byte ptr ds:[70h],1 ; (0040:0070=0)
33 ;; AL = 8 (потому что ax до этого момента = 0)
34 020A:077A  0C 08          or al,8
35 020A:077C          loc_2:
36 020A:077C  50              push ax
37 ;; Декремент счетчика отключения моторчика
38 020A:077D  FE 0E 0040      dec byte ptr ds:[40h] ; (0040:0040=0F7h)
39 020A:0781  75 0B          jnz loc_3 ; Jump if not zero
40 ;; Установка флага отключения моторчика дисковод (1-3 биты == 0)
41 020A:0783  80 26 003F F0    and byte ptr ds:[3Fh],0F0h ; (0040:003F=0)
42 ;; 3 строчки - посылка команды отключения дисководу
43 020A:0788  B0 0C          mov al,0Ch
44 020A:078A  BA 03F2        mov dx,3F2h
45 020A:078D  EE              out dx,al ; port 3F2h, dsk0 contrl output
46 020A:078E          loc_3:
47 020A:078E  58              pop ax
48 ;; Проверка 2 бита (PF)
49 020A:078F  F7 06 0314 0004  test word ptr ds:[314h],4 ; (0040:0314=3200h)
50 020A:0795  75 0C          jnz loc_4 ; Jump if not zero
51 ;; Копирование младшего байта FLAGS в ah
52 020A:0797  9F              lahf ; Load ah from flags
53 ;; Смена мест:
54 ;; теперь в ah: 08XXh - где XX - младший байт FLAGS
55 020A:0798  86 E0          xchg ah,al
56 ;; Кладем это на стек и вызываем прерывание
```

```

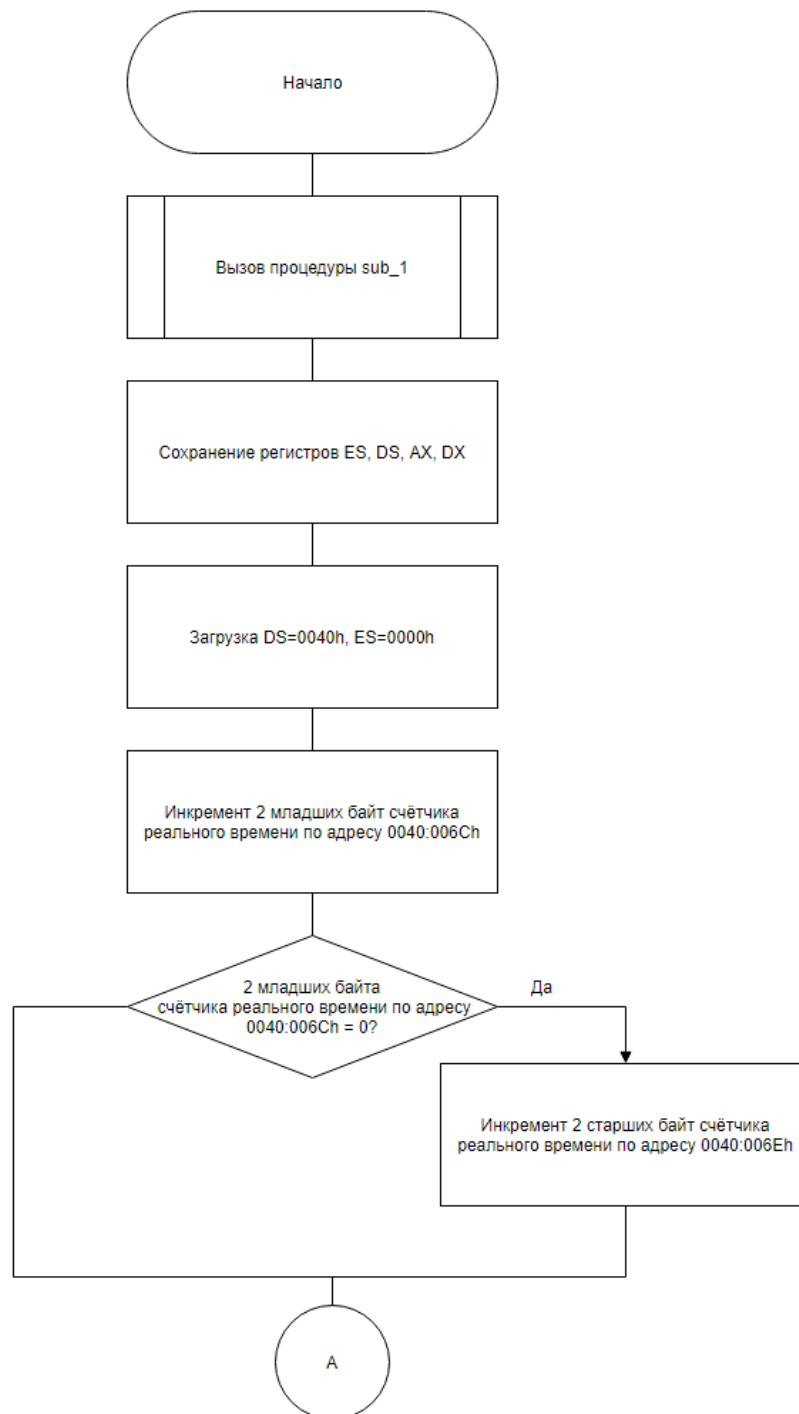
57 020A:079A 50                                push ax
58 ;; Вызываем 1Ch через адрес в таблице векторов. До этого мы добавили в стек AX,
   в то время как
59 ;; вызов int делает push флагов (то есть наш ax, описанный 6 строками выше будет
   как FLAGS в 1Ch)
60 020A:079B 26: FF 1E 0070                    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:
64 020A:07A3 CD 1C                             int 1Ch ; Timer break (call each 18.2ms)
65 020A:07A5                                loc_5:
66 020A:07A5 E8 0011                          call sub_1 ; (07B9)
67 ;; Сброс контроллера прерываний
68 ; al = 20h, end of interrupt
69 020A:07A8 B0 20                             mov al,20h ; ' '
70 020A:07AA E6 20                             out 20h,al ; port 20h, 8259-1 int command
71 ;; Восстановление регистров
72 020A:07AC 5A                             pop dx
73 020A:07AD 58                             pop ax
74 020A:07AE 1F                             pop ds
75 020A:07AF 07                             pop es
76 020A:07B0 E9 FE99 jmp $-164h ; (020A:07B0h - 164h = 020A:064Ch)
77 ;; ... -164h
78 ;; ...
79 020A:06AC CF iret ; Interrupt return

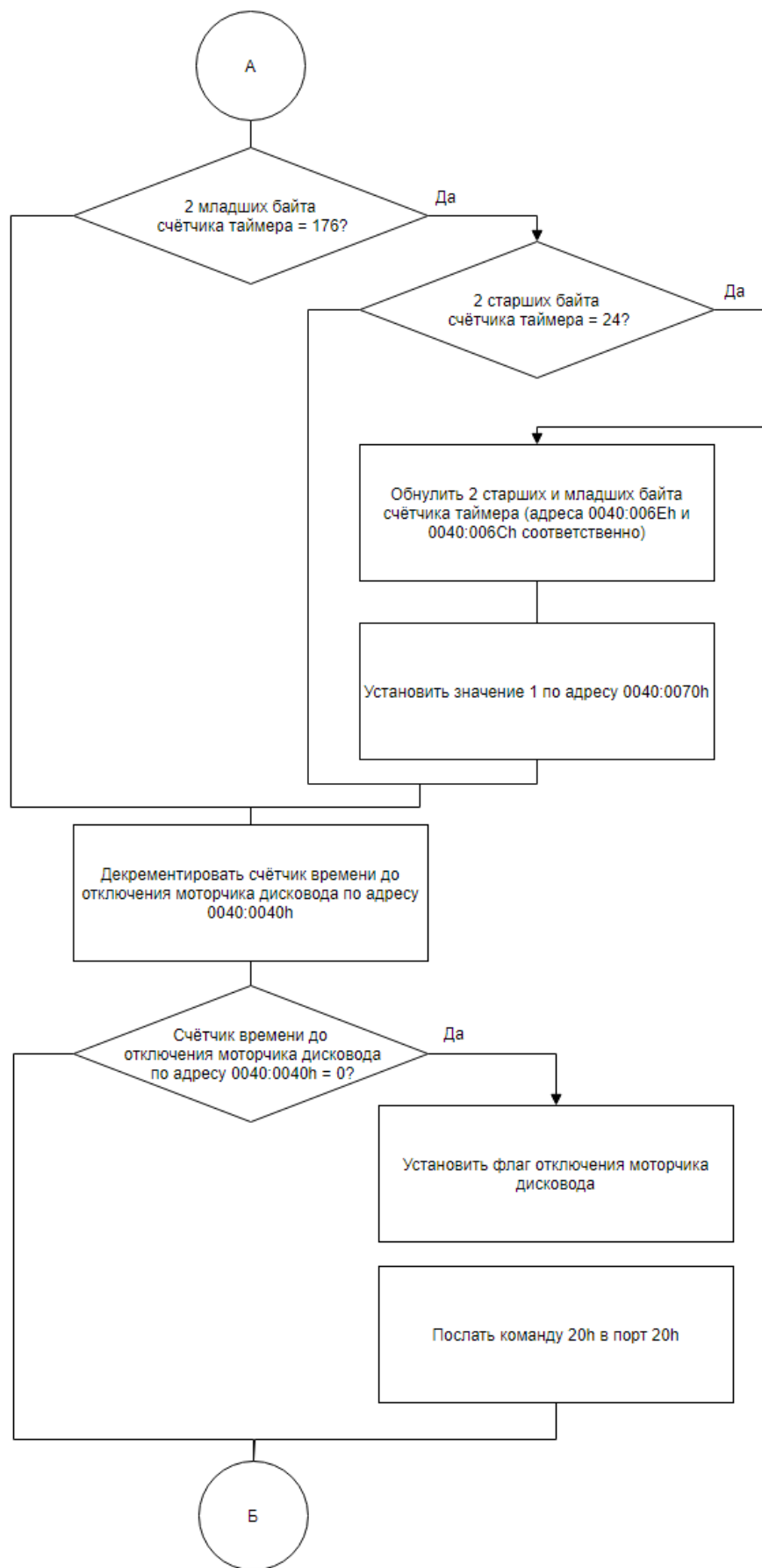
```

Листинг процедуры sub_1.

```
1 sub_1          proc          near
2 ;; Сохранение регистров
3 020A:07B9      1E              push ds
4 020A:07BA      50              push ax
5 020A:07BB      B8 0040         mov ax,40h
6 020A:07BE      8E D8          mov ds,ax
7 ;; Младший байт FLAGS в AH
8 020A:07C0      9F              lahf ; Load ah from flags
9 ;; Установлены ли старший бит IOPL или DF?
10 020A:07C1      F7 06 0314 2400 test word ptr ds:[314h],2400h ; (0040:0314=3200h
    )
11 020A:07C7      75 0C          jnz loc_7 ; Jump if not zero
12 ;; сброс IF в 0040:0314h (зануление 9 бита)
13 020A:07C9      F0> 81 26 0314 FDFF lock and word ptr ds:[314h],0FDFFh ;
    (0040:0314=3200h)
14 020A:07D0                      loc_6:
15 ;; AH копируется в младший байт FLAGS
16 020A:07D0      9E              sahf ; Store ah into flags
17 020A:07D1      58              pop ax
18 020A:07D2      1F              pop ds
19 020A:07D3      EB 03 jmp short loc_8 ; (07D8)
20 020A:07D5                      loc_7:
21 ;; Сброс IF
22 020A:07D5      FA              cli ; Disable interrupts
23 020A:07D6      EB F8          jmp short loc_6 ; (07D0)
24 020A:07D8                      loc_8:
25 020A:07D8      C3              retn
26 sub_1          endp
```

Схема алгоритма обработчика INT 8h





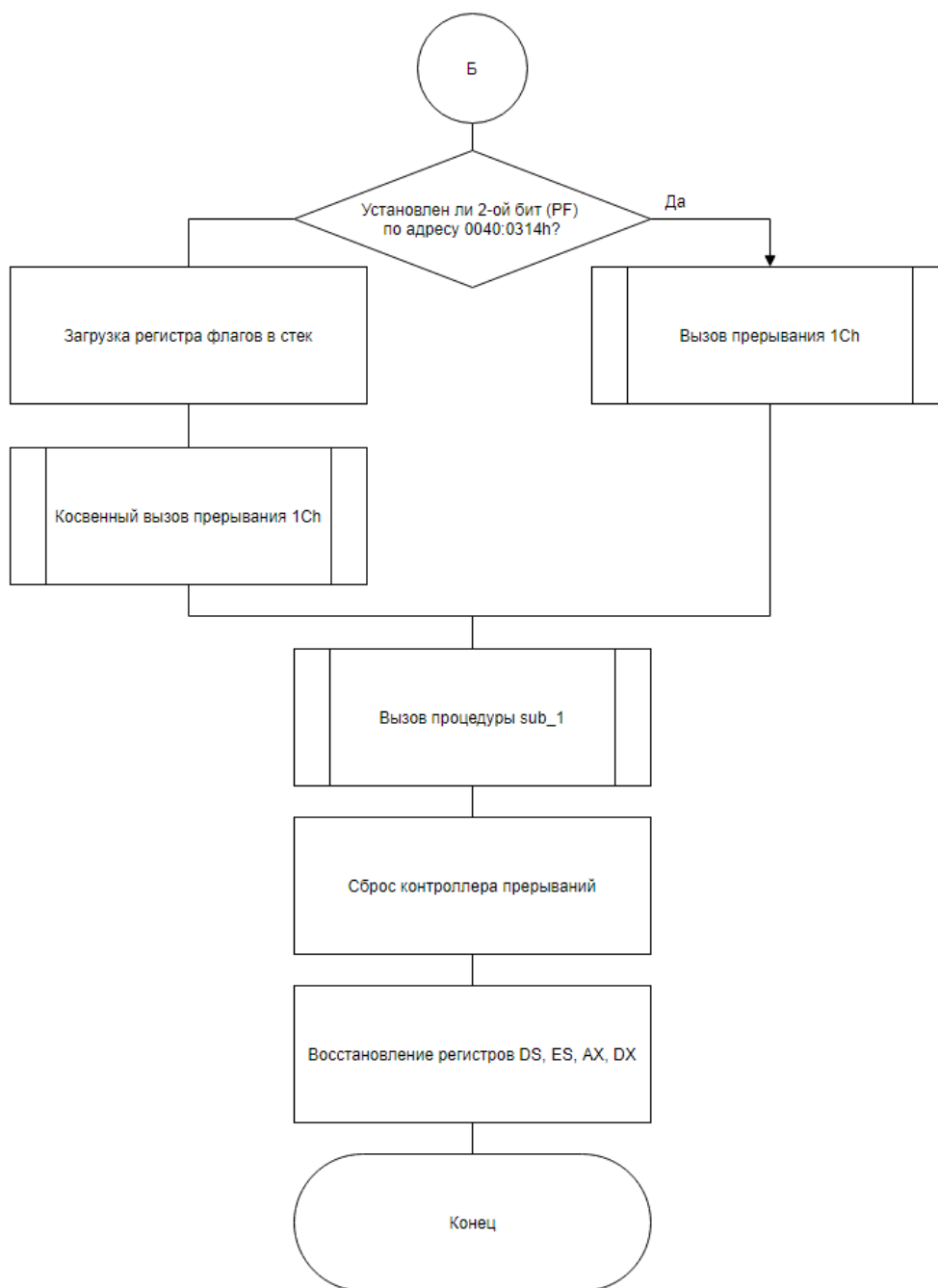


Схема алгоритма процедуры sub_1

