



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 «ИУ, Информатика и системы управления»

КАФЕДРА «ИУ7, Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа №1

по дисциплине

«Операционные системы»

Тема: Дизассемблирование INT 8h

Студент: Андреев А.А.

Группа: ИУ7-54Б

Преподаватель: Рязанова Н.Ю.

Москва - 2021 г.

Оглавление

Оглавление	1
1. Полученный дизассемблированный код	2
1.1. Листинг INT 8h	2
1.2. Листинг процедуры sub_04	4
2. Схема алгоритмов	6
2.1. Схема алгоритма обработчика INT 8h	6
2.2. Схема алгоритма процедуры sub_04	8

1. Полученный дизассемблированный код

1.1. Листинг INT 8h

```
1.          ; -- Вызов sub_04
2.          call sub_4          ; (07B9)
3.
4.          ; -- Сохранение регистров es, ds, ax, dx
5.          push es
6.          push ds
7.          push ax
8.          push dx
9.
10.         ; -- Загрузка в DS 0040h
11.         mov ax,40h
12.         mov ds,ax
13.
14.         ; -- AX = ES = 0
15.         xor ax,ax            ; Zero register
16.         mov es,ax
17.
18.         ; -- Инкрементирование счетчика таймера по адресу
0040:006C
19.         inc word ptr ds:[6Ch] ; (0040:006C=22A7h)
20.         jnz loc_3            ; Jump if not zero
21.
22.         ; -- Инкрементирование старших двух байта счетчика
таймера
23.         inc word ptr ds:[6Eh] ; (0040:006E=16h)
24.
25.         ; -- Проверка на то, что прошло 24 часа
26.         ; 0040H:006EH == 18H и 0040H:006H = 00B0H
27.         ; 24 * 60 * 60 * t == 18H << 16 + B0H, где количество
вызовов таймера в секунду - это t
28.         loc_3:
29.         cmp word ptr ds:[6Eh],18h ; (0040:006E=16h)
30.         jne loc_4            ; Jump if not equal
31.         cmp word ptr ds:[6Ch],0B0h ;
(0040:006C=22A7h)
32.         jne loc_4            ; Jump if not equal
33.
34.         ; -- Зануление счетчика таймера и занесение 1 в
0040H:0070 тогда, когда прошло 24 часа
35.         mov word ptr ds:[6Eh],ax ; (0040:006E=16h)
36.         mov word ptr ds:[6Ch],ax ; (0040:006C=22A7h)
37.         mov byte ptr ds:[70h],1 ; (0040:0070=0)
38.
39.         ; -- Ранее AL = 0, теперь AL = 8
```

```

40.      or    al,8
41.  loc_4:
42.      ; -- Сохранение регистра AX
43.      push  ax
44.
45.      ; -- Декрементирование счетчика отключения
      моторчика
46.      dec   byte ptr ds:[40h]      ; (0040:0040=35h)
47.      jnz   loc_5                  ; Jump if not zero
48.
49.      ; -- Установка флагов, отвечающих за отключение
      моторчика дисковода
50.      and   byte ptr ds:[3Fh],0F0h ; (0040:003F=0)
51.      mov   al,0Ch
52.      mov   dx,3F2h
53.      out   dx,al                  ; port 3F2h, dsk0 contrl
      output
54.  loc_5:
55.      ; -- Восстановление регистра AX
56.      pop   ax
57.
58.      ; -- Проверка 2 бита, Parity Flag
59.      test  word ptr ds:[314h],4   ; (0040:0314=3200h)
60.      jnz   loc_6                  ; Jump if not zero
61.
62.      ; -- Загрузка младшего байта FLAGS в регистр AH
63.      lahfh                          ; Load ah from flags
64.
65.      xchg  ah,al
66.      push  ax
67.
68.      ; -- Вызов 1CH с помощью адреса в таблице
      векторов. При вызове call на месте регистра будет лежать AX,
      который по выходу из 1CH будет установлен в FLAGS с помощью
      IRET
69.      call  dword ptr es:[70h]      ; (0000:0070=6ADh)
70.      jmp   short loc_7              ; (07A5)
71.      nop
72.  loc_6:
73.      int   1Ch                      ; Timer break (call each
      18.2ms)
74.  loc_7:
75.      call  sub_4                    ; (07B9)
76.
77.      ; -- Сброс контроллера прерываний
78.      mov   al,20h                  ; ' '
79.      out   20h,al                  ; port 20h, 8259-1 int
      command
80.      ; al = 20h, end of

```

```

interrupt
81.
82.      ; -- Восстановление регистров dx, ax, ds, es
83.      pop    dx
84.      pop    ax
85.      pop    ds
86.      pop    es
87.
88.      jmp    $-164h ; (020F:07B0H - 164h = 020A:064Ch)
89.
90.      db     0C4h
91.
92.      les     cx, dword ptr ds:[93E9h]      ;
      (0000:93E9=0E181h) Load 32 bit ptr
93.
94.      db     0FEh

```

1.2. Листинг процедуры sub_04

```

1. sub_4      proc near
2.
3.      ; -- Сохранение регистров ds, dx
4.      push   ds
5.      push   ax
6.
7.      ; -- AX = DS = 0040H
8.      mov     ax, 40h
9.      mov     ds, ax
10.
11.     ; -- Сохранение младшего байта FLAGS в AH
12.     lahf                                ; Load ah from flags
13.
14.     ; -- Проверка флага DF либо старшего бита IOPL
15.     test     word ptr ds:[314h], 2400h ;
      (0040:0314=3200h)
16.     jnz      loc_9                      ; Jump if not zero
17.
18.     ; -- Сброс Interrupt Enable Flag, 9 бит занулить
19.     lock     and     word ptr ds:[314h], 0FDFFh ;
      (0040:0314=3200h)
20. loc_8:
21.     ; -- Загрузка AH в младший байт FLAGS
22.     sahf                                ; Store ah into flags
23.     pop      ax
24.     pop      ds
25.     jmp      short loc_10                ; (07D8)
26. loc_9:
27.     cli                                ; Disable interrupts

```

```
28.      jmp     short loc_8          ; (07D0)
29.  loc_10:
30.      retn
31.  sub_4      endp
```

2. Схема алгоритмов

2.1. Схема алгоритма обработчика INT 8h

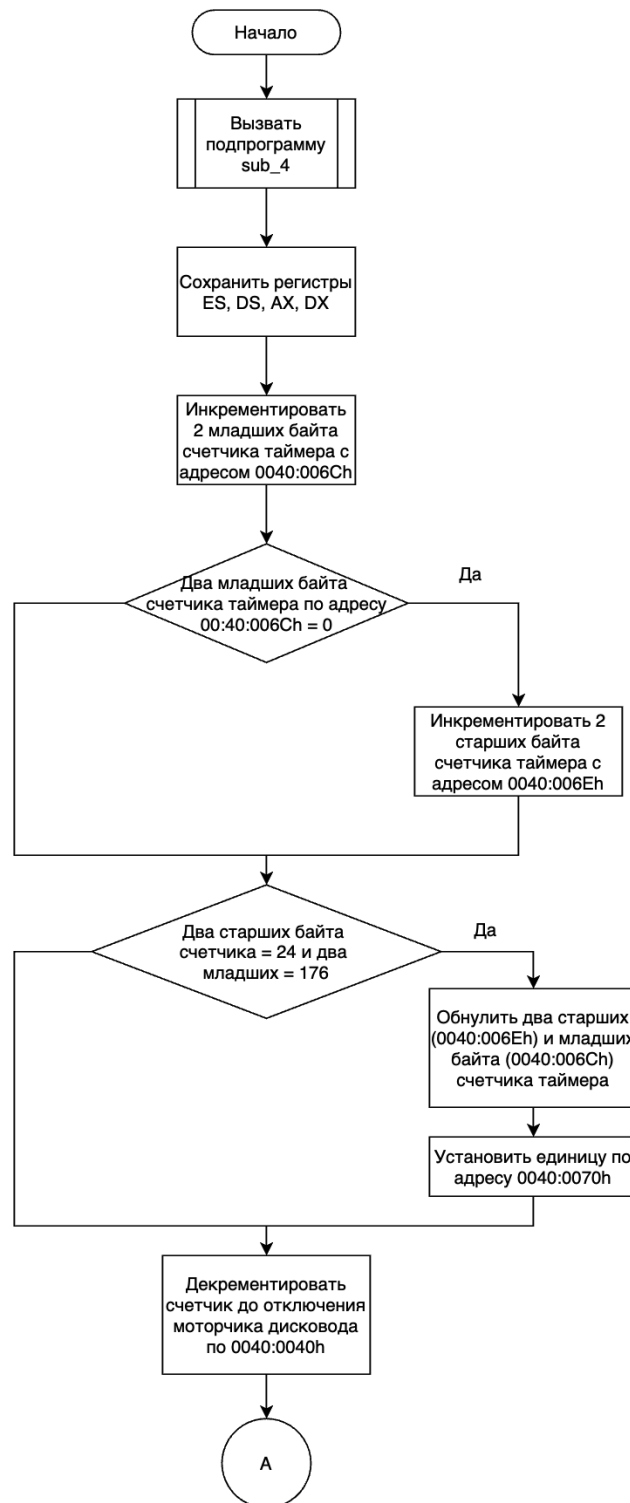


Рисунок 1 - Схема алгоритм обработчика INT 8H, часть 1

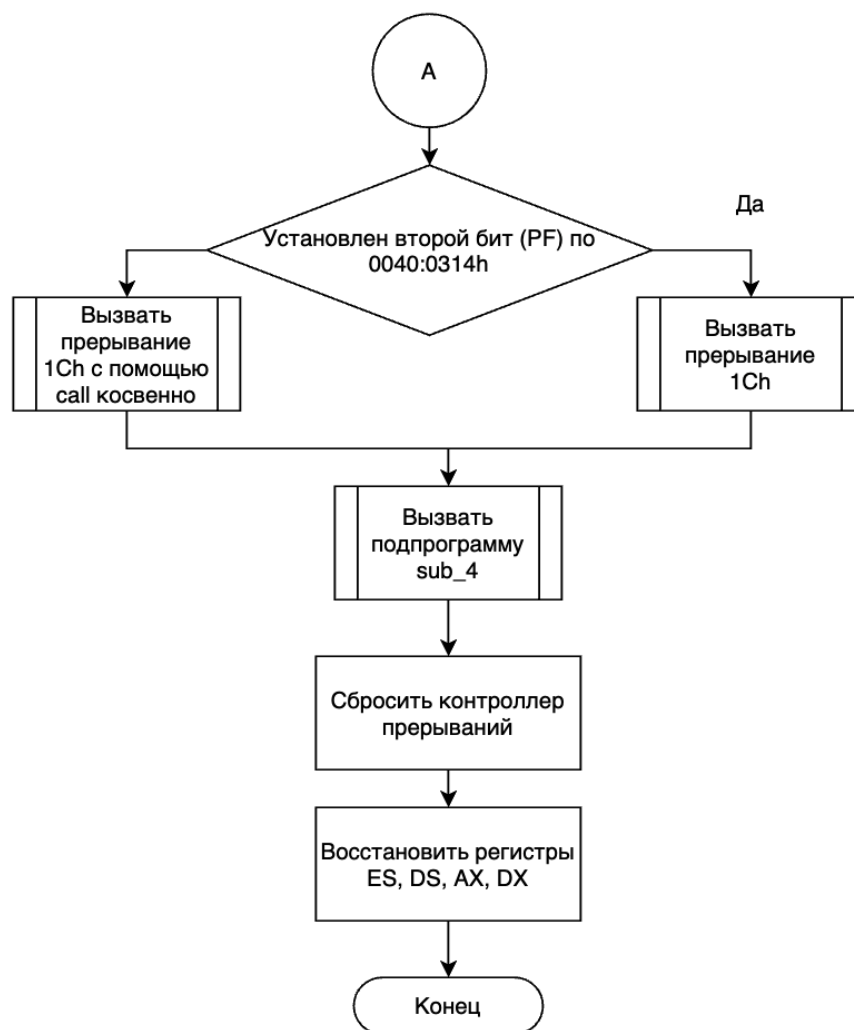


Рисунок 2 - Схема алгоритм обработчика INT 8H, часть 2

2.2. Схема алгоритма процедуры sub_04

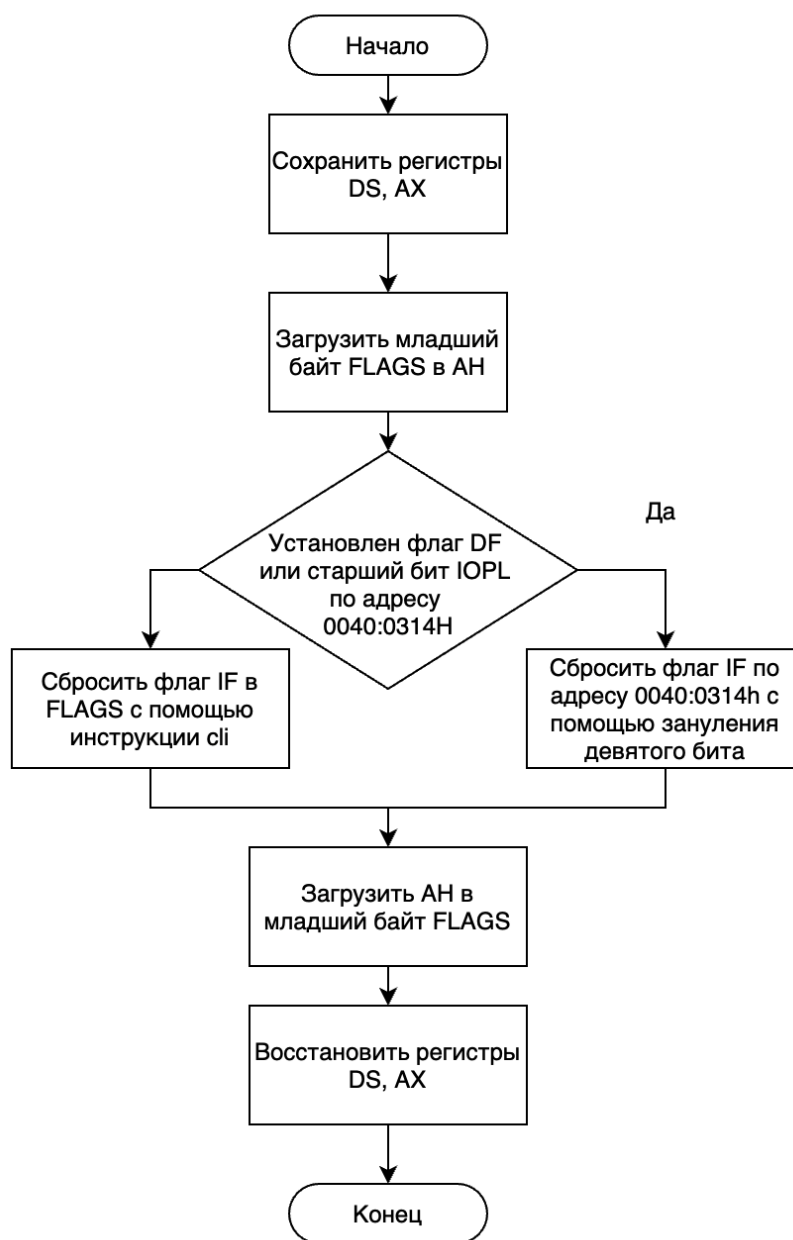


Рисунок 3 - Схема алгоритма процедуры sub_04