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, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по лабораторной работе № 1

по курсу «Операционные системы»

на тему: «Обработчик прерывания от системного таймера»

Студент ИУ7-55Б _____ Романов С. К.
(Группа) (Подпись, дата) (И. О. Фамилия)

Преподаватель _____ Рязанова Н. Ю.
(Подпись, дата) (И. О. Фамилия)

2022 г.

Оглавление

1 Исходный дизассемблированный код **3**

1.1 Прерывание 8h 3

1.2 SUB_6 6

1.3 Прерывание 1Ch 7

2 Схема алгоритмов **9**

2.1 Схема прерывания int 8h 9

2.2 Схема подпрограммы sub_6 12

1 Исходный дизассемблированный код

1.1 Прерывание 8h

```
Temp.lst                         Sourcer Listing v3.07     6-Sep-22
   7:15 pm   Page 1

;; вызов sub_1
020A:0746  E8 0070              call    sub_6           ; (07B9)
;; Сохранение регистров ES, DS, AX, DX
020A:0749  06                   push    es
020A:074A  1E                   push    ds
020A:074B  50                   push    ax
020A:074C  52                   push    dx

;; DS = 0040
020A:074D  B8 0040              mov ax,40h
020A:0750  8E D8                mov ds,ax
;; AX = 0
020A:0752  33 C0                xor ax,ax           ; Zero register
020A:0754  8E C0                mov es,ax

;; 0040:006Ch - адрес счетчика таймера
020A:0756  FF 06 006C               inc word ptr ds:[6Ch]   ;
   (0040:006C=41B9h)
020A:075A  75 04                jnz loc_2           ; Jump if not zero

;; 0040:006Eh - старшие 2 байта счетчика таймера
020A:075C  FF 06 006E               inc word ptr ds:[6Eh]   ;
   (0040:006E=13h)
020A:0760           loc_2:

;; Проверка: 0040:006Eh == 18h (24) и 0040:006Ch == B0h (176)
;; Можно убедиться в том, что: 18h << 16 + B0h = 24 * 60 * 60 * freq,
;; где freq - кол-во раз, которое вызывается таймер в секунду.
;; Таким образом из того, что условие выполняется, следует, что прошли
    сутки.

020A:0760  83 3E 006E 18            cmp word ptr ds:[6Eh],18h   ;
   (0040:006E=13h)
020A:0765  75 15                jne loc_3           ; Jump if not
   equal
020A:0767  81 3E 006C 00B0          cmp word ptr ds:[6Ch],0B0h  ;
   (0040:006C=41B9h)
020A:076D  75 0D                jne loc_3           ; Jump if not
   equal

;; Зануление счетчика (старшего слова и младшего слова)
```

```
37
38      020A:076F  A3 006E              mov word ptr ds:[6Eh],ax    ;
           (0040:006E=13h)
39      020A:0772  A3 006C              mov word ptr ds:[6Ch],ax    ;
           (0040:006C=41B9h)
40
41      ;; Прошло более 24 часов, занесение значения 1 в 0040:0070
42
43      020A:0775  C6 06 0070 01            mov byte ptr ds:[70h],1 ;
           (0040:0070=0)
44
45      ;; AL = 8
46      020A:077A  0C 08                or  al,8
47
48      020A:077C          loc_3:
49      ;; сохранение AX
50      020A:077C  50                   push    ax
51
52      ;; Декремент счетчика отключения моторчика
53      020A:077D  FE 0E 0040               dec byte ptr ds:[40h]   ;
           (0040:0040=6Ah)
54      020A:0781  75 0B                jnz loc_4           ; Jump if not zero
55
56      ;; Установка флага отключения моторчика дисковода (1-3 биты == 0)
57      020A:0783  80 26 003F F0            and byte ptr ds:[3Fh],0F0h  ;
           (0040:003F=0)
58
59      ;; 3 строчки - посылка команды отключения дисководу
60      020A:0788  B0 0C                mov al,0Ch
61      020A:078A  BA 03F2              mov dx,3F2h
62      020A:078D  EE                   out dx,al           ; port 3F2h, dsk0
           contrl output
63
64      020A:078E          loc_4:
65      ;; Восстановление AX
66      020A:078E  58                   pop ax
67
68      ;; Проверка третьего бита (Parity Flag)
69      020A:078F  F7 06 0314 0004          test    word ptr ds:[314h],4    ;
           (0040:0314=3200h)
70      020A:0795  75 0C                jnz loc_5           ; Jump if not zero
71
72      ;; Копирование младшего байта FLAGS в ah
73      020A:0797  9F                   lahf                ; Load ah from
           flags
74      ;; Смена мест:
75      ;; теперь в ax: 08XXh - где XX - младший байт FLAGS
76      020A:0798  86 E0                xchg    ah,al
```

```
77      ;; Кладем это на стек и вызываем прерывание
78      020A:079A  50                   push    ax
79
80      ;; Вызываем 1Ch через адрес в таблице векторов. До этого мы добавили в
            стек AX, в то время как
81      ;; вызов int делает push флагов (то есть наш ax, описанный 6 строками
           выше будет как FLAGS в 1Ch)
82      020A:079B  26: FF 1E 0070           call    dword ptr es:[70h]  ;
           (0000:0070=6ADh)
83      020A:07A0  EB 03                jmp short loc_6     ; (07A5)
84      020A:07A2  90                   nop
85      020A:07A3           loc_5:
86      020A:07A3  CD 1C                int 1Ch         ; Timer break (call
           each 18.2ms)
87      020A:07A5           loc_6:
88      020A:07A5  E8 0011              call    sub_6           ; (07B9)
89
90      ;; Сброс контроллера прерываний
91      020A:07A8  B0 20                mov al,20h          ; ' '
92      020A:07AA  E6 20                out 20h,al          ; port 20h,
           8259-1 int command
93                                          ;  al = 20h, end of interrupt
94                                      ;  al = 20h, end of interrupt
95
96      ;; Восстановление регистров
97      020A:07AC  5A                   pop dx
98      020A:07AD  58                   pop ax
99      020A:07AE  1F                   pop ds
100     020A:07AF  07                   pop es
101     020A:07B0  E9 FE99              jmp $-164h
102     ...
103     ;; 07B0h - 0164h = 064Ch
104     ;; Листинг ниже
105     ...
```

1.2 SUB_6

```
                        sub_6       proc    near
    ;; Сохранение регистров
    020A:07B9  1E                   push    ds
    020A:07BA  50                   push    ax
    ;; DS = 40h
    020A:07BB  B8 0040              mov ax,40h
    020A:07BE  8E D8                mov ds,ax

    ;; Младший байт FLAGS в AH
    020A:07C0  9F                   lahf                ; Load ah from
       flags

    ;; Установлены ли старший бит IOPL или DF?
    020A:07C1  F7 06 0314 2400          test    word ptr
       ds:[314h],2400h    ; (0040:0314=3200h)
    020A:07C7  75 0C                jnz loc_8           ; Jump if not zero


    ;; сброс IF (Iterrupt flag) в 0040:0314h (зануление 9 бита)
    020A:07C9  F0> 81 26 0314 FDFF              lock and word ptr
       ds:[314h],0FDFFh   ; (0040:0314=3200h)

    020A:07D0           loc_7:
    ;; AH копируется в младший байт FLAGS
    020A:07D0  9E                   sahf                ; Store ah into
       flags
    020A:07D1  58                   pop ax
    020A:07D2  1F                   pop ds
    020A:07D3  EB 03                jmp short loc_9     ; (07D8)

    020A:07D5           loc_8:
    ;; Сброс IF (Iterrupt flag)
    020A:07D5  FA                   cli             ; Disable interrupts
    020A:07D6  EB F8                jmp short loc_7     ; (07D0)
    020A:07D8           loc_9:
    020A:07D8  C3                   retn
                        sub_6       endp
```

1.3 Прерывание 1Ch

```
1
2     020A:06AD  EB 9D                jmp short $-61h
3     ...
4     ;; 06AD-61h == 064Ch
5     ...
6     020A:064C           loc_1:
7     020A:064C  1E                   push    ds
8     020A:064D  50                   push    ax
9     020A:064E  B8 0040              mov ax,40h
10    020A:0651  8E D8                mov ds,ax
11    020A:0653  F7 06 0314 2400          test    word ptr
         ds:[314h],2400h     ; (0040:0314=3200h)
12    020A:0659  75 4F                jnz loc_9           ; Jump if not zero
13    020A:065B  55                   push    bp
14    020A:065C  8B EC                mov bp,sp
15    020A:065E  8B 46 0A             mov ax,[bp+0Ah]
16    020A:0661  5D                   pop bp
17    020A:0662  A9 0100              test    ax,100h
18    020A:0665  75 43                jnz loc_9           ; Jump if not zero
19    020A:0667  A9 0200              test    ax,200h
20    020A:066A  74 22                jz  loc_5           ; Jump if zero
21    020A:066C  F0> 81 0E 0314 0200            lock or  word ptr
         ds:[314h],200h ; (0040:0314=3200h)
22    020A:0673  F7 06 0314 0003          test    word ptr ds:[314h],3    ;
         (0040:0314=3200h)
23    020A:0679  75 2F                jnz loc_9           ; Jump if not zero
24    020A:067B           loc_2:
25    020A:067B  86 E0                xchg    ah,al
26    020A:067D  FC                   cld             ; Clear direction
27    020A:067E  A8 04                test    al,4
28    020A:0680  75 25                jnz loc_8           ; Jump if not zero
29    020A:0682           loc_3:
30    020A:0682  A8 08                test    al,8
31    020A:0684  75 11                jnz loc_6           ; Jump if not zero
32    020A:0686  70 19                jo  loc_7           ; Jump if
         overflow=1
33    020A:0688           loc_4:
34    020A:0688  9E                   sahf                ; Store ah into
         flags
35    020A:0689  58                   pop ax
36    020A:068A  1F                   pop ds
37    020A:068B  CA 0002              retf    2           ; Return far
38    020A:068E           loc_5:
39    020A:068E  F0> 81 26 0314 FDFF            lock and word ptr
         ds:[314h],0FDFFh    ; (020A:0314=3231h)
```

```
020A:0695  EB E4              jmp short loc_2      ; (067B)
020A:0697           loc_6:
020A:0697  70 EF              jo  loc_4            ; Jump if
   overflow=1
020A:0699  50                 push    ax
020A:069A  B0 7F              mov al,7Fh
020A:069C  04 02              add al,2
020A:069E  58                 pop ax
020A:069F  EB E7              jmp short loc_4      ; (0688)
020A:06A1           loc_7:
020A:06A1  50                 push    ax
020A:06A2  32 C0              xor al,al            ; Zero register
020A:06A4  58                 pop ax
020A:06A5  EB E1              jmp short loc_4      ; (0688)
020A:06A7           loc_8:
020A:06A7  FD                 std             ; Set direction flag
020A:06A8  EB D8              jmp short loc_3      ; (0682)
020A:06AA           loc_9:
020A:06AA  58                 pop ax
020A:06AB  1F                 pop ds
020A:06AC  CF                 iret                 ; Interrupt return
```

2 Схема алгоритмов

2.1 Схема прерывания int 8h

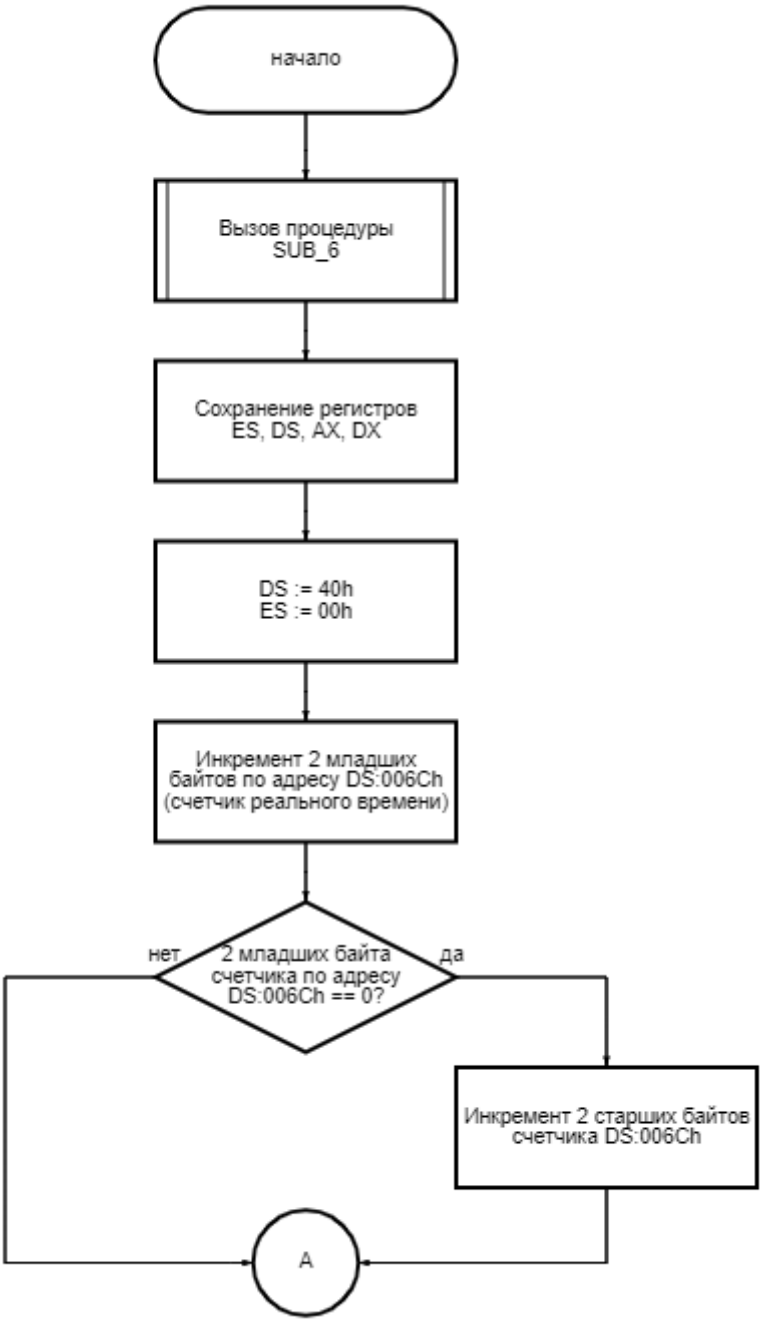

Рисунок 2.1 – Схема A

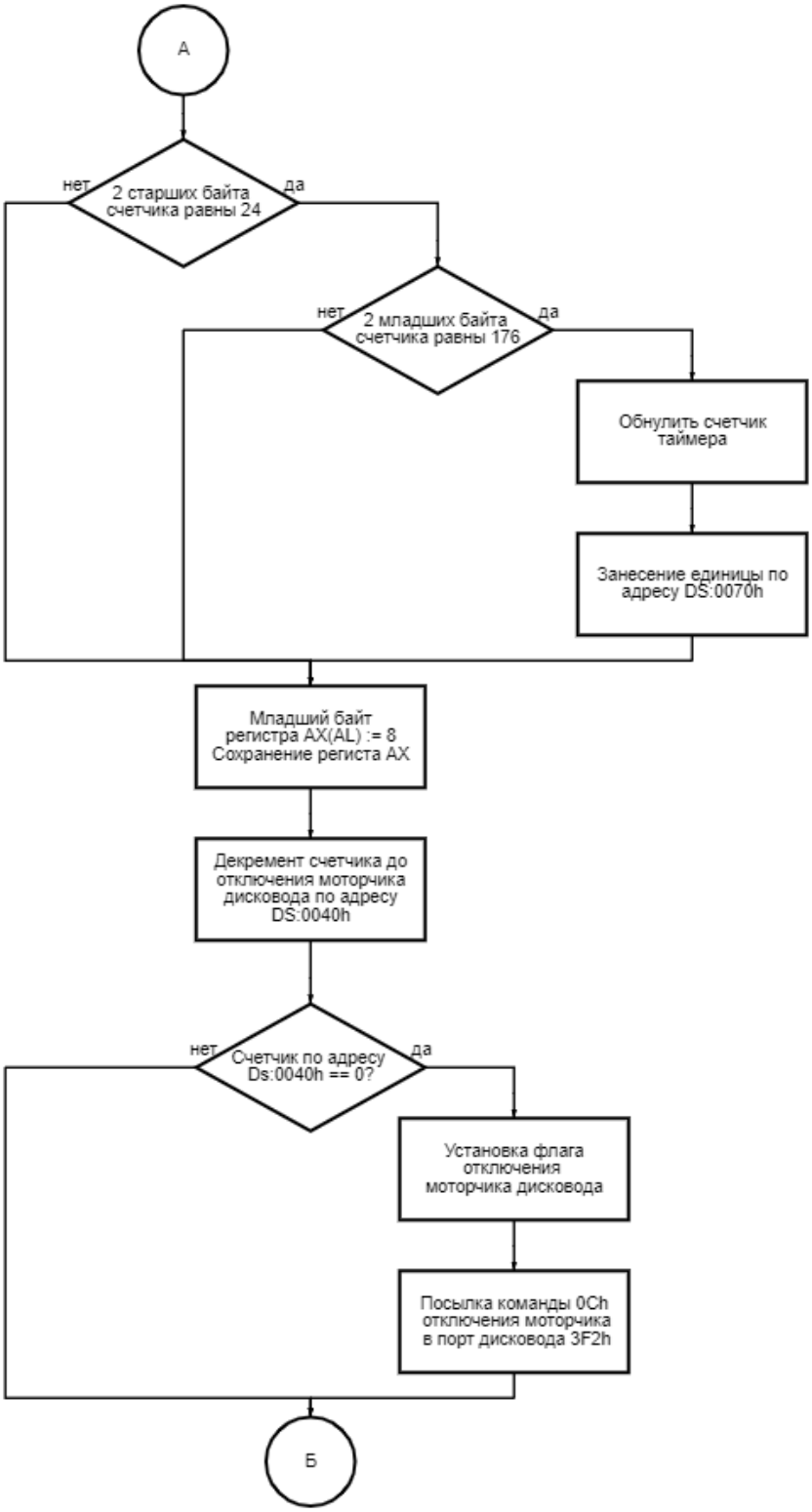

Рисунок 2.2 – Схема Б

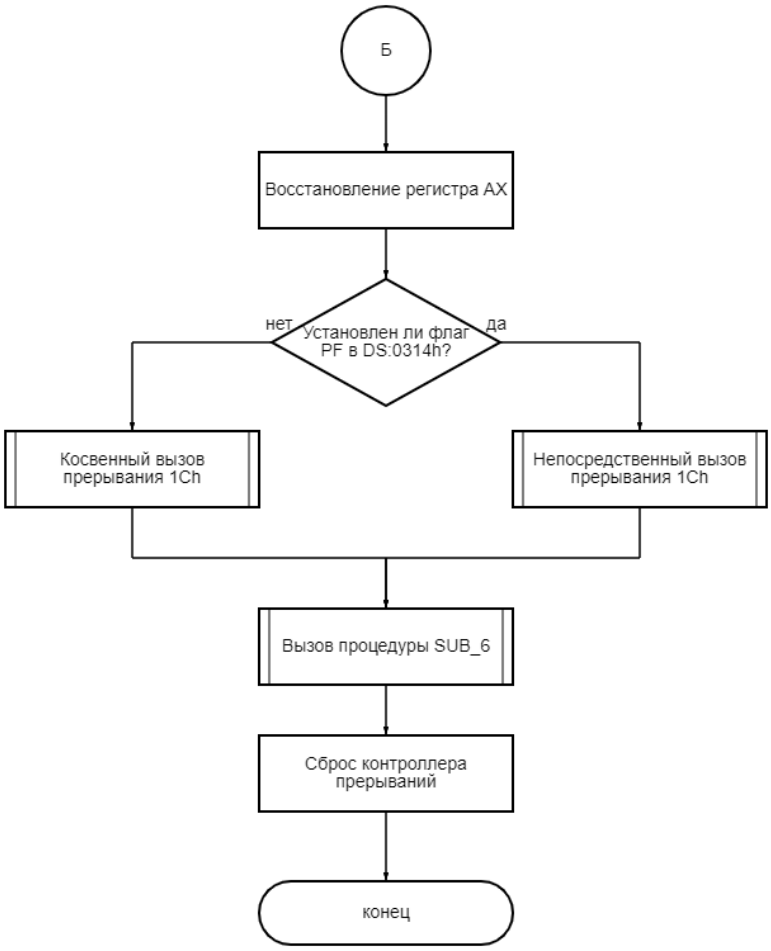

Рисунок 2.3 – Схема C

2.2 Схема подпрограммы sub_6

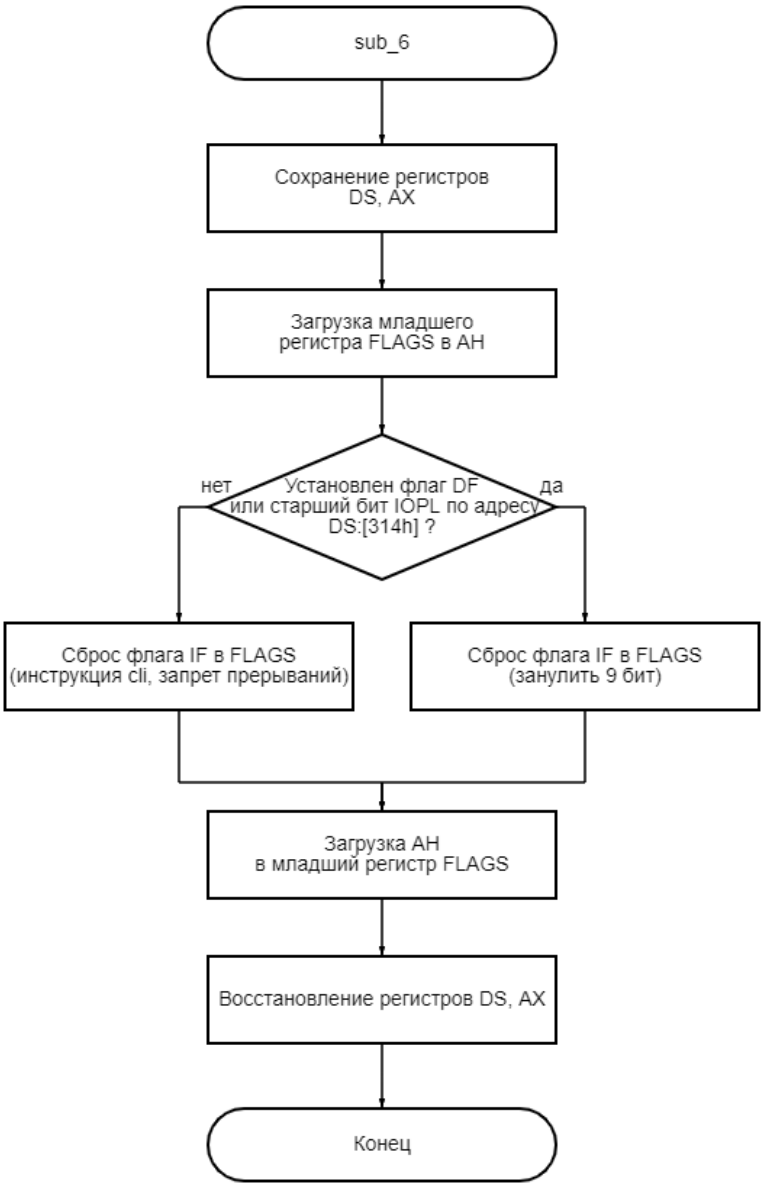

Рисунок 2.4 – sub_6