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.

Москва — 2020 г.

Листинг 1 — Код прерывания INT 8h

```
1  ; Вызывает подпрограмму sub_1
2  020A:0746  E8 0070      ;*      call    sub_1           ; (07B9)
3  020A:0746  E8 70 00             db  0E8h, 70h, 00h
4  ; Записывает регистры в стек.
5  020A:0749  06                   push    es
6  020A:074A  1E                   push    ds
7  020A:074B  50                   push    ax
8  020A:074C  52                   push    dx
9  ; Инициализирует регистры.
10 020A:074D  B8 0040              mov ax,40h
11 ; В ds помещаем начало области данных BIOS (Зубков).
12 020A:0750  8E D8                mov ds,ax
13 020A:0752  33 C0                xor ax,ax           ; Zero register
14 ; В es помещаем адрес начала таблицы векторов прерывания.
15 020A:0754  8E C0                mov es,ax
16 ; 0040:006C = 0046C - адрес 4-байтовой переменной,
17 ; располагающейся в области данных BIOS - это счетчик таймера.
18 ; Увеличивает счетчик таймера.
19 020A:0756  FF 06 006C           inc word ptr ds:[6Ch]   ; (0040:006C=0A1Dh)
20 ; JNZ - перейти, если не равно (ZF = 0) на loc_1.
21 020A:075A  75 04                jnz loc_1           ; Jump if not zero
22 ; Если счетчик равен 0, то увеличиваем часы, т.е. прошел час. ( 0040:006E
       - это часы)
23 020A:075C  FF 06 006E           inc word ptr ds:[6Eh]   ; (0040:006E=0Ah)
24 020A:0760            loc_1:
25 ; Если час не прошел, то сравниваем
26 ; 0040:006E с 24 (это часы 18h == 24)
27 020A:0760  83 3E 006E 18        cmp word ptr ds:[6Eh],18h   ;
       (0040:006E=0Ah)
28 ; Если еще не 24, то прыгаем на loc_2
29 020A:0765  75 15                jne loc_2           ; Jump if not equal
30 ; Сравниваем 0040:006C (B0h=176)
31 020A:0767  81 3E 006C 00B0      cmp word ptr ds:[6Ch],0B0h  ;
       (0040:006C=0A1Dh)
32 ; Если != 176, то прыгаем на loc_2
33 020A:076D  75 0D                jne loc_2           ; Jump if not equal
34 ; Обнуляем счетчик (если прошел день)
35 ; В ячейку 0040:0070 записывам единицу
36 ; (Для фиксации о том, что новый день наступил)
37 020A:076F  A3 006E              mov word ptr ds:[6Eh],ax    ;
       (0040:006E=0Ah)
38 020A:0772  A3 006C              mov word ptr ds:[6Ch],ax    ;
       (0040:006C=0A1Dh)
39 020A:0775  C6 06 0070 01        mov byte ptr ds:[70h],1 ; (0040:0070=0)
40 ; В младший байт регистра ax заносим 8
```

```
41 ; (т.к. ax до этого был равен 0 => 0 or 8 == 8)
42 020A:077A  0C 08                or  al,8
43 020A:077C            loc_2:
44 ; Если новый день не наступил, то
45 ; Записываем регистр ax в стек.
46 ; (Он м.б. равен 0 или 8, в зависимости от того, наступил новый день или не
       т)
47 20A:077C  50                   push    ax
48 ; Ячейка с адресом 0000:0440h содержит время, оставшееся до выключения двиг
       ателя.
49 ; Декрементируем это время.
50 020A:077D  FE 0E 0040          dec byte ptr ds:[40h]    ; (0040:0040=2Ch)
51 ; Если еще не равно нулю, то прыгаем на loc_3
52 020A:0781  75 0B               jnz loc_3           ; Jump if not zero
53 ; Если равено 0, то двигатель НГМД отключается.
54 ; Отправка сигнала отключения моторчика.
55 ; Сброс флага отключания моторчика дисковода
56 020A:0783  80 26 003F F0       and byte ptr ds:[3Fh],0F0h  ; (0040:003F=0)
57 020A:0788  B0 0C               mov al,0Ch
58 020A:078A  BA 03F2             mov dx,3F2h
59 ; Порт 3F2 - адрес порта цифрового упарвления (тип вывод).
60 ; НГМД - накопитель на гибких магнитных дисках
61 ; Порт 3F2h работает только на запись, это порт вывода.
62 ; Мы отправляем в этот порт 0C (1100).
63 ; 2 бит поднят - разрешение работы контроллера
64 ; 3 бит поднят - разрешение прерываний и прямого доступа к памяти
65 ; 4-7 биты - значение 1 в каждом разряде вызывает включение соответствующег
       о двигателя НГМД
66 ; (Инф. https://www.frolov-lib.ru/books/bsp/v19/ch1_4.html)
67 ; Инструкция OUT выводит данные из регистра AL или AX (ИСТОЧНИК) в порт вво
       да-вывода.
68 ; Номер порта должен быть указан в ПРИЁМНИКЕ.
69 020A:078D  EE                   out dx,al           ; port 3F2h, dsk0
       contrl output
70 020A:078E            loc_3:
71 ; Если счетчик таймера не равен нулю, то
72 ; Возвращаем в ax содержимое, которое раньше положили.
73 020A:078E  58                   pop ax
74 ; Проверяем флаг PF по адресу 0040:0314.
75 ; (0100 - поднят 2 бит, он как раз отвечает за флаг PF - Parity Flag - Флаг
       чётности)
76 020A:078F  F7 06 0314 0004     test    word ptr ds:[314h],4    ;
       (0040:0314=3200h)
77 020A:0795  75 0C               jnz loc_4           ; Jump if not zero
78 ; LAHF: Загрузка флагов в регистр AH.
79 ; Команда LAHF перемещает младший байт регистра флагов EFLAGS в регистр AH.
80 020A:0797  9F                  lahf                ; Load ah from flags
```

```
81  ; Обмен ah и al.
82  020A:0798  86 E0                xchg    ah,al
83  ; Записываем ax в стек.
84  020A:079A  50                   push    ax
85  ; Косвенный вызов прерывания 1Ch (1C * 4 = 70h).
86  020A:079B  26: FF 1E 0070       call    dword ptr es:[70h]  ;
        (0000:0070=6ADh)
87  020A:07A0  EB 03                jmp short loc_5     ; (07A5)
88  020A:07A2  90                   nop
89  020A:07A3          loc_4:
90  ; Вызываем прерывание 1C.
91  ; После инициализации системы вектор INT 1Ch указывает на команду IRET,
92  ; то есть обработчик прерывания INT 1Ch ничего не делает.
93  020A:07A3  CD 1C                int 1Ch         ; Timer break (call each
        18.2ms)
94  020A:07A5          loc_5:
95  ; Вызываем подпрограмму sub_1
96  020A:07A5  E8 0011              call    sub_1       ; (07B9)
97  ; Сброс контроллера прерываний (mov  al, 20h; out  20h, al) - из методички.
98  ; Необходимо отметить, что прерывание int 1Ch вызывается обработчиком преры
        вания int 8h
99  ; до сброса контроллера прерывания, поэтому во время выполнения
100 ; прерывания int 1Ch все аппаратные прерывания запрещены.
101 ; В частности, запрещены прерывания от клавиатуры.
102 020A:07A8  B0 20              mov al,20h          ; ' '
103 ; Конец прерывания.
104 020A:07AA  E6 20              out 20h,al          ; port 20h, 8259-1 int
        command
105 ;   al = 20h, end of interrupt
106 ; Восстанавливаем значение регистров.
107 020A:07AC  5A                 pop dx
108 020A:07AD  58                 pop ax
109 020A:07AE  1F                 pop ds
110 020A:07AF  07                 pop es
111 ; Выход.
112 020A:07B0  E9 FE99            jmp $-164h
113 020A:07B3  C4                 db  0C4h
114 ;* No entry point to code
115 ; les - загружает первые 16 бит dword по адресу ds:[93E9h] в регистр CX,
116 ; А оставшиеся 16 бит загружает в ES (Т.к. lES (есть еще lDS,...))
117 020A:07B4  C4 0E 93E9         les cx,dword ptr ds:[93E9h] ;
        (0000:93E9=0A1A1h) Load 32 bit ptr
118 020A:07B8  FE                 db  0FEh
```

Листинг 2 — Код подпрограммы sub_1

```
1                   sub_1       proc    near
```

```
; Сохраняем флаги.
020A:07B9  1E                    push    ds
020A:07BA  50                    push    ax
;  Инициализируем регистры.
020A:07BB  B8 0040               mov ax,40h
020A:07BE  8E D8                 mov ds,ax
; lahf: Загрузка флагов в регистр AH.
; Загружает значение флагового регистра в регистр  AH.
020A:07C0  9F                    lahf                    ; Load ah from flags
; Команда TEST - логическое и без изменения операда (Меняются только флаги).
; 2400 = 10010000000000. Поднят ли флаг 10ый или 13ый?
; 10 - DF - Direction Flag - Флаг направления.
; Контролирует поведение команд обработки строк. Если установлен в 1, то строки
; обрабатываются в сторону уменьшения адресов, если сброшен в 0, то наоборот.
; 12 и 13 - IOPL - I/O Privilege Level - Уровень приоритета ввода/вывода.
020A:07C1  F7 06 0314 2400       test    word ptr ds:[314h],2400h    ; (0040:0314=3200h)
; Если не равно 0 переходим на loc_7
020A:07C7  75 0C                 jnz loc_7               ; Jump if not zero
; На все время выполнения команды, снабженной таким префиксом, будет заб-
; локирована шина данных, и если в системе присутствует другой процессор, он не
; сможет обращаться к памяти, пока не закончится выполнение команды с префик-
; сом LOCK.
; LOCK - делаеи следующую команду неделимой.
; and 2 раза обращается к памяти. 1 раз он считывает значение по адресу 0040:0314
; Затем он изменяет его и еще раз обращаяется к памяти на запись.
; Мы делаем ее неделимой, чтобы в этот промежуток, когда мы выполняем непосредственно
; Саму логическую операцию, никто не смог влезть в этот участок памяти (мы его как раз блокируем).
020A:07C9  F0> 81 26 0314 FDFF   lock    and word ptr ds:[314h],0FDFFh    ; (0040:0314=3200h)
020A:07D0                loc_6:
; Команда sahf копирует разряды 7, 6, 4, 2 и 0 регистра AH в регистр флагов процессора,
; устанавливая тем самым значения флагов SF, ZF, AF, PF и CF соответственно.
; Команда не имеет операндов.
020A:07D0  9E                    sahf                    ; Store ah into flags
; Восстанавливаем флаги.
020A:07D1  58                    pop ax
```

```
020A:07D2  1F                  pop ds
020A:07D3  EB 03               jmp short loc_8      ; (07D8)
020A:07D5          loc_7:
; cli - сбрасывает флаг IF
; Флаг IF - Interrupt Enable Flag - Флаг разрешения прерываний.
; Если сбросить этот флаг в 0, то процессор перестанет обрабатывать прерывания от внешних устройств.
; Обычно его сбрасывают на короткое время для выполнения критических участков программы.
; (маскируемые - прерывания, которые можно запрещать установкой соответствующих битов в регистре маскирования прерываний)
020A:07D5  FA                  cli                  ; Disable interrupts
020A:07D6  EB F8               jmp short loc_6      ; (07D0)
020A:07D8          loc_8:
; Конец процесса.
020A:07D8  C3                  retn
sub_1       endp
```

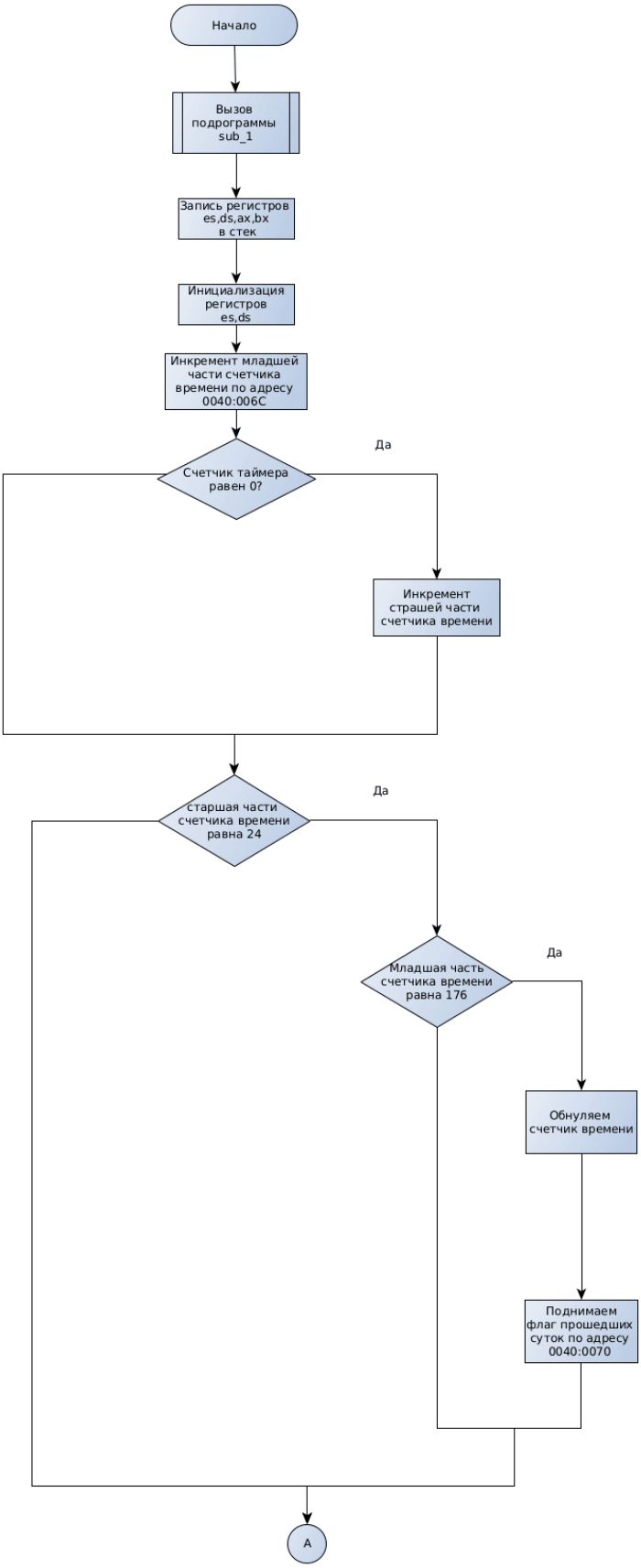

Рисунок 0.1 — Схема прерывания INT 8h

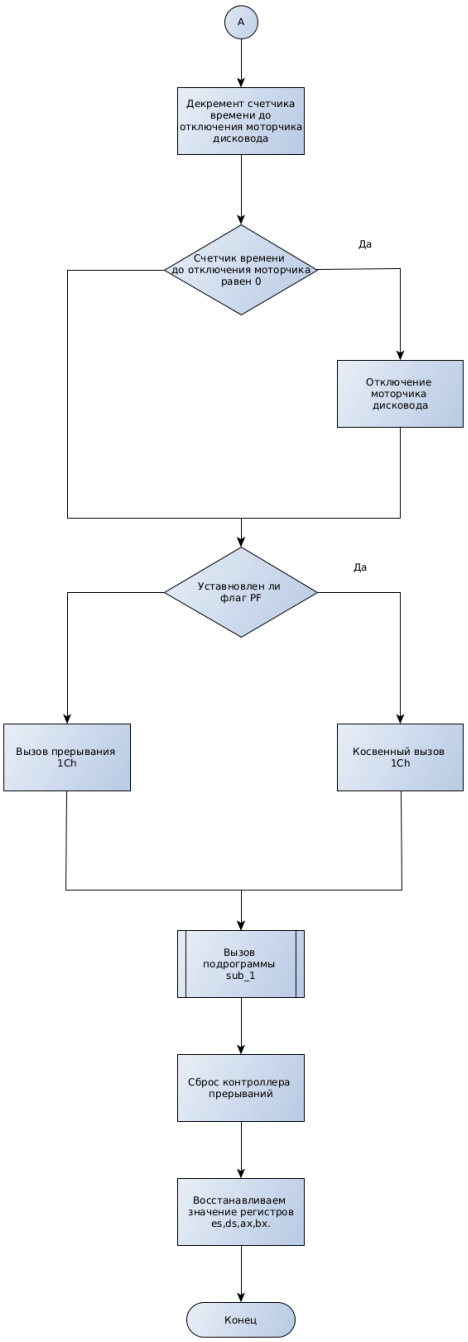

Рисунок 0.2 — Схема прерывания INT 8h

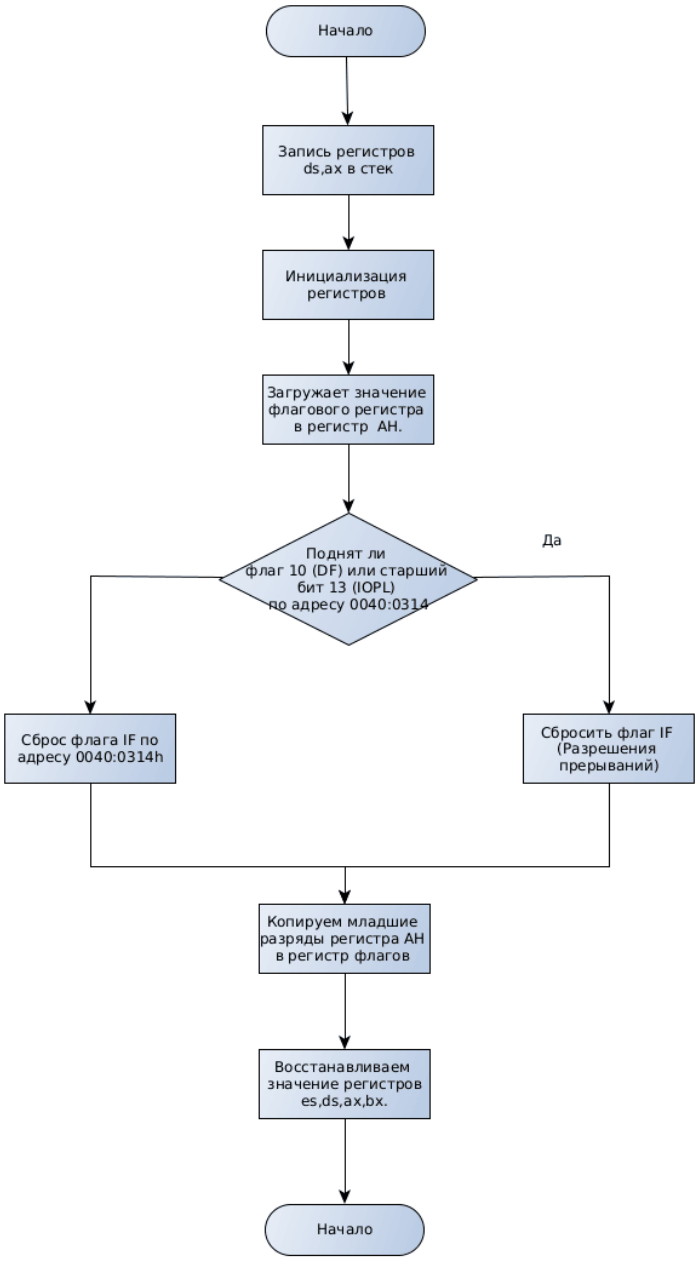

Рисунок 0.3 — Схема подпрограммы sub_1