Государственное образовательное учреждение высшего профессионального образования «Московский Государственный Технический Университет имени Н. Э. Баумана»

**Отчет**

По лабораторной работе №1 (часть 1)

По курсу «Операционные системы»

Тема: «Дизассемблирование прерывания Int 08h»

Студент: Маковская Я. Р.

Группа: ИУ7-54Б

Преподаватель: Рязанова Н. Ю.

Москва, 2019

Адрес прерывания int 08h был найден при помощи 35 прерывания DOS. Этот адрес равен 020С:0746. После этого при помощи программы Sourcer был дизассемблирован код прерывания int 08h.

**Листинг прерывания INT 08h:**

**; вызов sub\_1, которая находится по адресу 07B9**

020C:0746 E8 0070 call sub\_1 ; (07B9)

**; сохранение регистров es, ds, sx, dx**

020C:0749 06 push es

020C:074A 1E push ds

020C:074B 50 push ax

020C:074C 52 push dx

**; установка сегмента данных**

020C:074D B8 0040 mov ax, 40h

020C:0750 8E D8 mov ds, ax

020C:0752 33 C0 xor ax, ax ; Zero register

020C:0754 8E C0 mov es, ax

**; инкремент счетчика реального времени, который находится по**

**; известному адресу 0000:046Сh (4 байтовая переменная в BIOS)**

020C:0756 FF 06 006C inc word ptr ds:[6Ch]

; (0040:006C=7B67h)

020C:075A 75 04 jnz loc\_1 ; Jump if not zero

020C:075C FF 06 006E inc word ptr ds:[6Eh] ; (0040:006E=14h)

**; сброс счетчика реального времени по прошествии 24 часов**

020C:0760 loc\_1:

020C:0760 83 3E 006E 18 cmp word ptr ds:[6Eh], 18h

; (0040:006E=14h)

020C:0765 75 15 jne loc\_2 ; Jump if not equal

020C:0767 81 3E 006C 00B0 cmp word ptr ds:[6Ch], 0B0h

;(0040:006C=7B67h)

020C:076D 75 0D jne loc\_2 ; Jump if not equal

020C:076F A3 006E mov word ptr ds:[6Eh], ax

; (0040:006E=14h)

020C:0772 A3 006C mov word ptr ds:[6Ch],ax ; (0040:006C=7B67h)

**; если счетчик переполняется, потому что прошло более 24 часов с момента запуска таймера, в ячейку 0000:0470h заносится значение 1**

020C:0775 C6 06 0070 01 mov byte ptr ds:[70h], 1

; (0040:0070=0)

020C:077A 0C 08 or al, 8

**; декремент счетчика времени до остановки моторчика дисковода**

020C:077C loc\_2:

020C:077C 50 push ax

020C:077D FE 0E 0040 dec byte ptr ds:[40h] ; (0040:0040=8)

020C:0781 75 0B jnz loc\_3 ; Jump if not zero

**; отправка сигнала на отключение моторчика в порт контроллера дисковода**

020C:0783 80 26 003F F0 and byte ptr ds:[3Fh],0F0h

; (0040:003F=0)

020C:0788 B0 0C mov al,0Ch

020C:078A BA 03F2 mov dx,3F2h

020C:078D EE out dx,al ; port 3F2h, dsk0

;contrl output

020C:078E loc\_3:

020C:078E 58 pop ax

**; проверка возможности вызова прерываний**

020C:078F F7 06 0314 0004 test word ptr ds:[314h], 4

; (0040:0314=3200h)

020C:0795 75 0C jnz loc\_4 ; Jump if not zero

020C:0797 9F lahf ; Load ah from flags

020C:0798 86 E0 xchg ah,al

**; косвенный вызов пользовательского прерывания**

020C:079A 50 push ax

020C:079B 26: FF 1E 0070 call dword ptr es:[70h]

; (0000:0070=6ADh)

020C:07A0 EB 03 jmp short loc\_5 ; (07A5)

020C:07A2 90 nop

**; вызов пользовательского прерывания**

020C:07A3 loc\_4:

020C:07A3 CD 1C int 1Ch ; Timer break (call each 18.2ms)

020C:07A5 loc\_5:

020C:07A5 E8 0011 call sub\_1 ; (07B9)

**; сброс контроллера прерываний**

020C:07A8 B0 20 mov al,20h ; ' '

020C:07AA E6 20 out 20h,al ;port 20h, 8259-1 int command

; al = 20h, end of interrupt

**; восстановление регистров dx, ax, ds, es**

020C:07AC 5A pop dx

020C:07AD 58 pop ax

020C:07AE 1F pop ds

020C:07AF 07 pop es

**; переход по адресу 065F**

020C:07B0 E9 FE99 jmp $-164h

020C:07B3 C4 db 0C4h

…

**; завершение прерывания**

020C:0D93 CF iret ; Interrupt return

**Листинг подпрограммы sub\_1:**

sub\_1 proc near

**; сохранение регистрова ds, ax**

020C:07B9 1E push ds

020C:07BA 50 push ax

**; установка сегмента данных**

020C:07BB B8 0040 mov ax,40h

020C:07BE 8E D8 mov ds,ax

**; сохранение регистра флагов в ah**

020C:07C0 9F lahf ; Load ah from flags

020C:07C1 F7 06 0314 2400 test word ptr ds:[314h],2400h

; (0040:0314=3200h)

020C:07C7 75 0C jnz loc\_7 ; Jump if not zero

020C:07C9 F0> 81 26 0314 FDFF

lock and word ptr ds:[314h],0FDFFh ; (0040:0314=3200h)

**; восстановление регистра флагов из ah**

020C:07D0 loc\_6:

020C:07D0 9E sahf ; Store ah into flags

**; восстановление регистров ax и ds**

020C:07D1 58 pop ax

020C:07D2 1F pop ds

020C:07D3 EB 03 jmp short loc\_8 ; (07D8)

020C:07D5 loc\_7:

020C:07D5 FA cli ; Disable interrupts

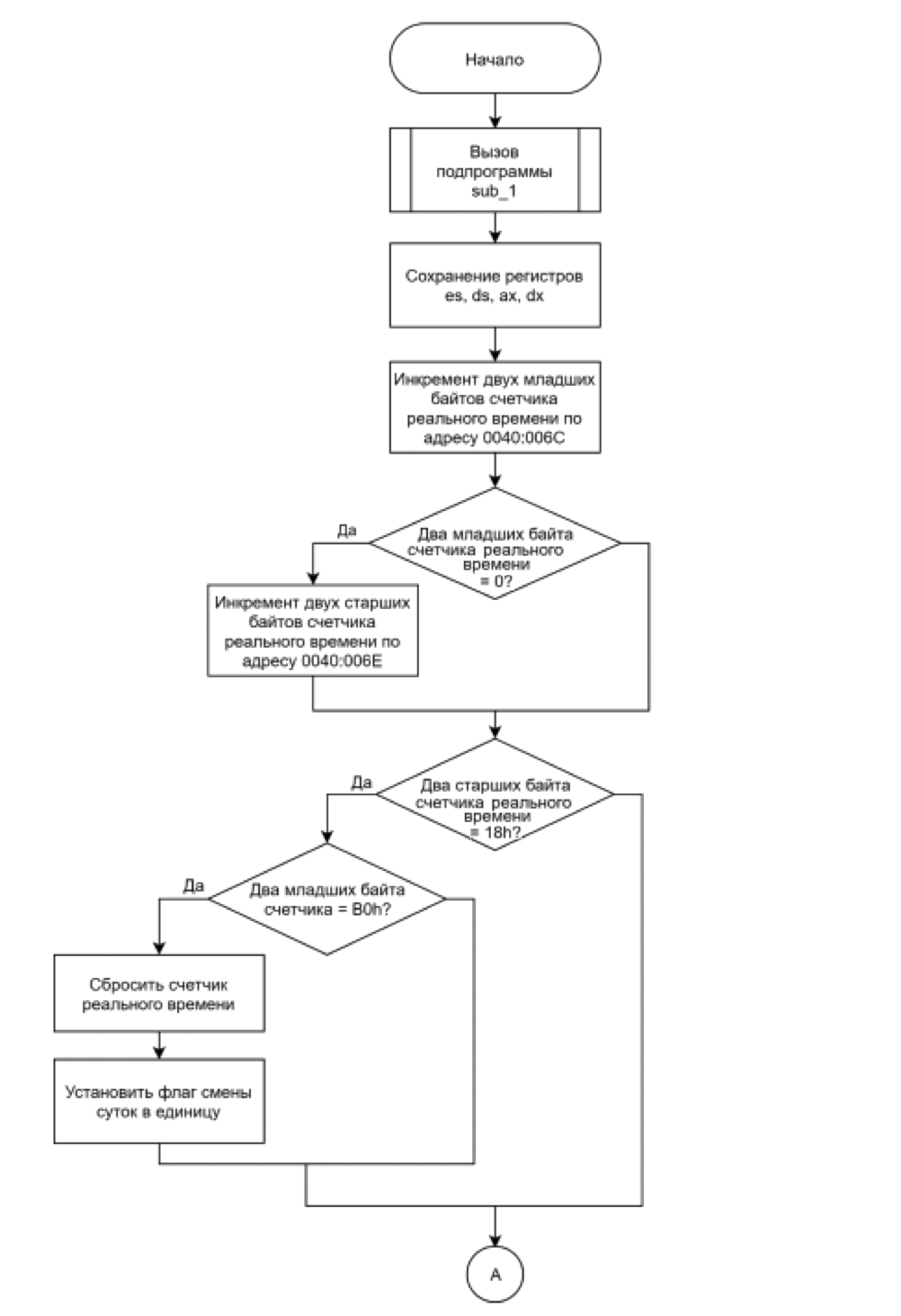
020C:07D6 EB F8 jmp short loc\_6 ; (07D0)

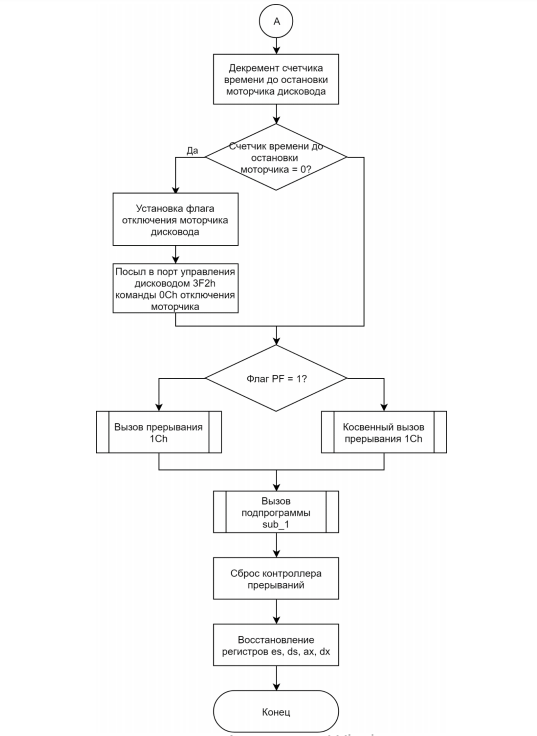
020C:07D8 loc\_8:

020C:07D8 C3 retn

sub\_1 endp

**Схема алгоритма Int 08h:**

****



**Схема алгоритма sub\_1**

