



**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

Лабораторная работа № 1

Тема: Дизассемблирование INT 8h

Студент Воякин А. Я.

Группа ИУ7-54Б

Преподаватель Рязанова Н. Ю.

Москва, 2020 г.

Листинг обработчика прерываний int 8h:

```

; Вызов подпрограммы sub_1.
020C:0746 E8 0070          call     sub_1          ; (07B9)

; Сохранение аппаратного контекста. Сохраняются регистры ES, DS, AX, DX.
020C:0749 06              push     es
020C:074A 1E              push     ds
020C:074B 50              push     ax
020C:074C 52              push     dx

; Инициализация значений регистров.
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 младшей части счётчика суточного времени.
020C:0756 FF 06 006C      inc      word ptr ds:[6Ch] ; (0040:006C=0FDD8h)

; Выполнение jump, если флаг ZF равен 0.
020C:075A 75 04          jnz      loc_1          ; Jump if not zero

; Инкремент старшей части счётчика суточного времени.
020C:075C FF 06 006E      inc      word ptr ds:[6Eh] ; (0040:006E=0Dh)

; Сброс счётчика суточного времени при начале нового дня.
020C:0760          loc_1:
; Сравнение старшей части счётчика суточного времени с 24.
020C:0760 83 3E 006E 18      cmp      word ptr ds:[6Eh],18h ; (0040:006E=0Dh)
020C:0765 75 15          jne      loc_2          ; Jump if not equal
; Сравнение младшей части счётчика суточного времени с 176.
020C:0767 81 3E 006C 00B0      cmp      word ptr ds:[6Ch],0B0h ; (0040:006C=0FDD8h)
020C:076D 75 0D          jne      loc_2          ; Jump if not equal
; Обнуление счётчика суточного времени.
020C:076F A3 006E          mov      word ptr ds:[6Eh],ax ; (0040:006E=0Dh)
020C:0772 A3 006C          mov      word ptr ds:[6Ch],ax ; (0040:006C=0FDD8h)
020C:0775 C6 06 0070 01      mov      byte ptr ds:[70h],1 ; (0040:0070=0)
020C:077A 0C 08          or       al,8

020C:077C          loc_2:
020C:077C 50              push     ax
; Декремент счётчика времени до отключения моторчика дисковод.
020C:077D FE 0E 0040      dec      byte ptr ds:[40h] ; (0040:0040=0A1h)
; Выполнения jump если отключать моторчик не надо.
020C:0781 75 0B          jnz      loc_3          ; Jump if not zero
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
; Отправка команды отключения моторчика контроллеру дисковода.
020C:078D EE          out      dx,al          ; port 3F2h, disk0 contrl output
020C:078E          loc_3:
020C:078E 58              pop      ax
020C:078F F7 06 0314 0004      test     word ptr ds:[314h],4 ; (0040:0314=3200h) pf флаг чётности
020C:0795 75 0C          jnz      loc_4          ; Jump if not zero
; Копируем младший байт регистра flags в ah.
020C:0797 9F          lahf          ; Load ah from flags косв выз пр 1Ch
; Помещение старшего байта AX в AL.
020C:0798 86 E0          xchg     ah,al
020C:079A 50              push     ax
020C:079B 26 FF 1E 0070      call     dword ptr es:[70h] ; (0000:0070=6ADh) кос выз обр пр 1Ch
020C:07A0 EB 03          jmp      short loc_5     ; (07A5)
020C:07A2 90              nop

020C:07A3          loc_4:
; Вызов прерывания 1Ch.
020C:07A3 CD 1C          int      1Ch          ; Timer break (call each 18.2ms)
020C:07A5          loc_5:
; Вызов подпрограммы обслуживания прерывания.
020C:07A5 E8 0011          call     sub_1          ; (07B9)

; Сброс контроллера прерываний. Разрешение аппаратных прерываний.
020C:07A8 B0 20          mov      al,20h
020C:07AA E6 20          out      20h,al          ; port 20h, 8259-1 int command

```

```

; Восстановление аппаратного контекста. Восстанавливаются регистры ES, DS, AX, DX. al = 20h, end of interrupt
020C:07AC 5A                pop     dx
020C:07AD 58                pop     ax
020C:07AE 1F                pop     ds
020C:07AF 07                pop     es

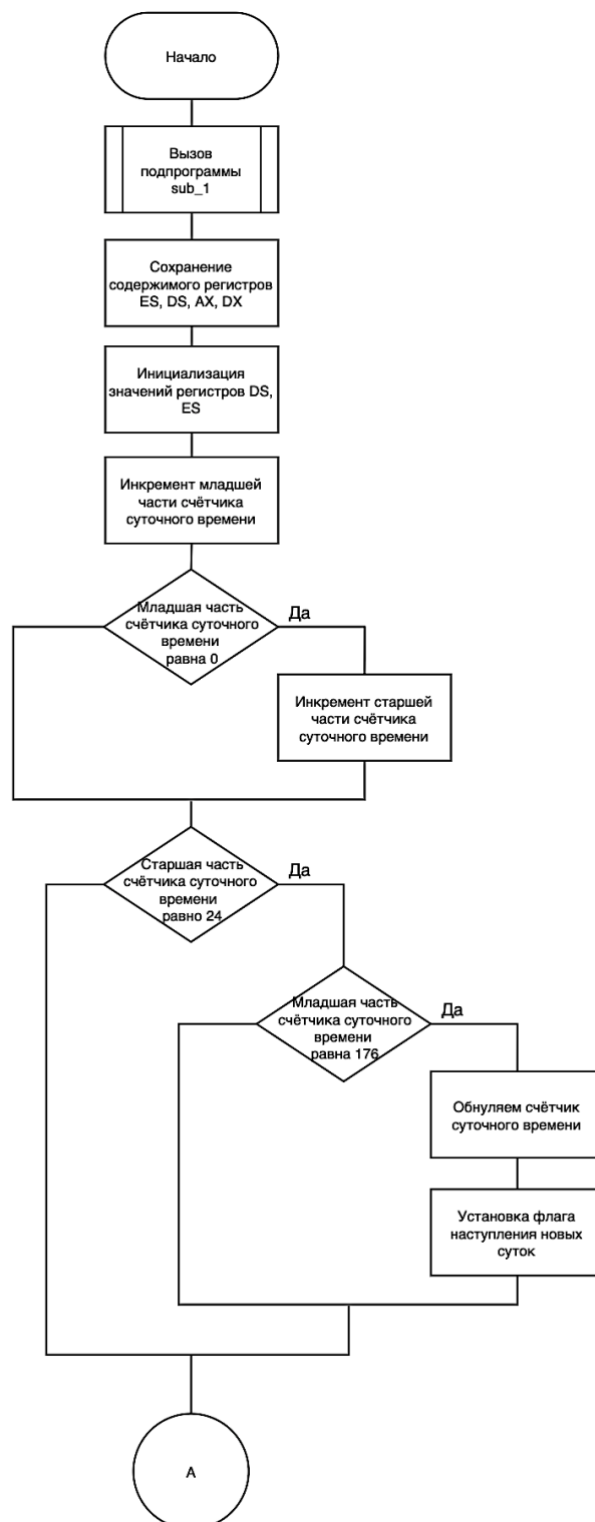
; JMP на IRET.
020C:07B0 E9 FE99        jmp     $-164h
020C:06AC CF                iret     ; Interrupt return

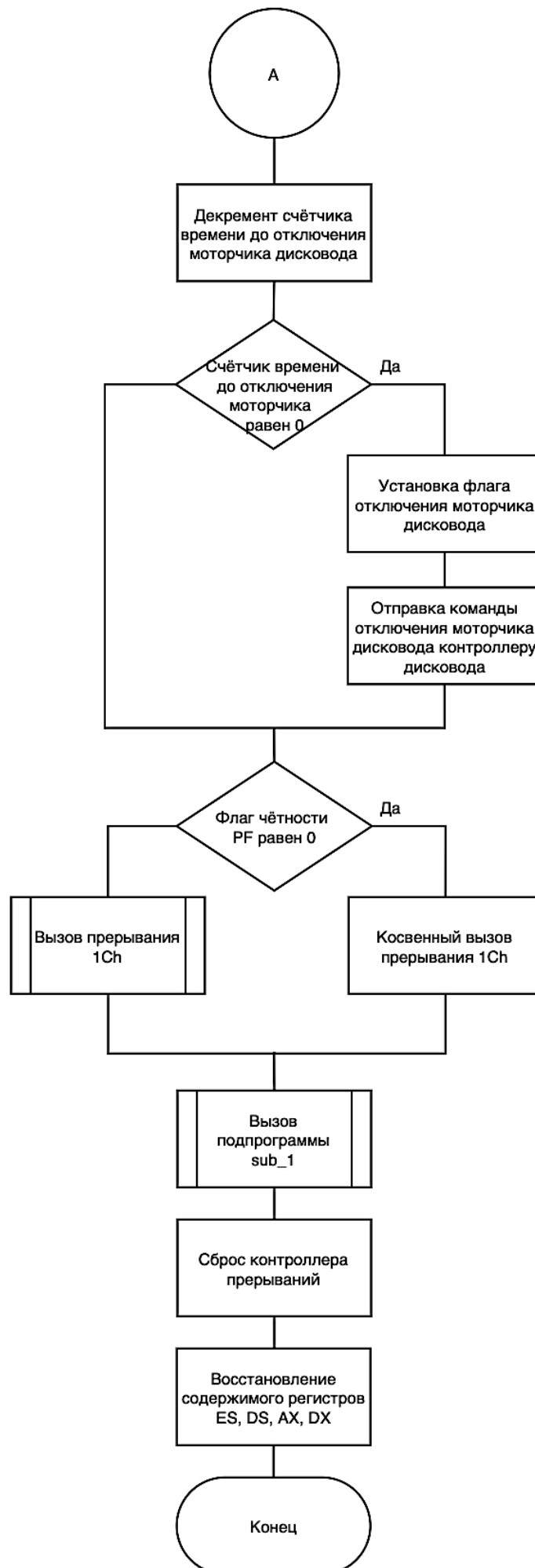
020C:07B3 C4                db      0C4h

020C:07B4 C4 0E 93E9      les     cx,dword ptr ds:[93E9h] ; * No entry point to code
020C:07B8 FE                db      0FEh ; (0000:93E9=0A7A1h) Load 32 bit ptr

```

Схема обработчика прерывания int 8h:





Листинг подпрограммы sub_1:

```

sub_1 proc near ; зап м пр
; Сохранения аппаратного контекста. Сохраняются регистры DS, AX.
020C:07B9 1E push ds
020C:07BA 50 push ax
; Загрузка адреса в DS.
020C:07BB B8 0040 mov ax,40h
020C:07BE 8E D8 mov ds,ax
; Копируем младший байт регистра flags 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) DF IOPL
020C:07C7 75 0C jnz loc_7 ; Jump if not zero
; Сброс флага IF.
020C:07C9 FD 81 26 0314 FDFF lock and word ptr ds:[314h],0FDFFh ; (0040:0314=3200h)
020C:07D0 loc_6:
; Загрузка регистра AH в младший байт регистра FLAGS.
020C:07D0 9E sahf ; Store ah into flags
; Восстановление аппаратного контекста. Восстанавливаются регистры AX, DS.
020C:07D1 58 pop ax
020C:07D2 1F pop ds
020C:07D3 EB 03 jmp short loc_8 ; (07D8)
020C:07D5 loc_7:
; Сброс флага IF.
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 подпрограммы sub_1:

