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»

Тема Дизассемблирование прерывания INT 8h

Студент Фролов Евгений_____

Группа ИУ7-56Б_____

Оценка (баллы)_____

Преподаватель Рязанова Н. Ю._____

Москва — 2021 г.

Цель работы

Знакомство со средством дизассемблирования Sourcer, получение дизассемблированного кода ядра операционной системы Windows на примере обработчика прерывания INT 8h в virtual mode – специальном режиме защищенного режима (32-разрядный режим работы), который эмулирует реальный режим работы вычислительной системы на базе процессоров Intel.

Задание

Используя Sourcer получить дизассемблированный код обработчика аппаратного прерывания от системного таймера INT 8h. На основе полученного кода составить алгоритм работы обработчика INT 8h.

Листинг обработчика INT 8h

Листинг 1 – Обработчик INT 8h

```
; вызов подпрограммы sub_1
020A:0746  E8 0070              call    sub_1           ; (07B9)
020A:0746  E8 7000              db      0E8h, 70h, 00h
; сохранение значений регистров
020A:0749  06                   push    es
020A:074A  1E                   push    ds
020A:074B  50                   push    ax
020A:074C  52                   push    dx
; инициализация значений регистров
020A:074D  B8 0040              mov ax,40h
020A:0750  8E D8                mov ds,ax
020A:0752  33 C0                xor ax,ax           ; Zero register
020A:0754  8E C0                mov es,ax
; инкремент счетчиков времени
020A:0756  FF 06 006C           inc word ptr ds:[6Ch]   ; (0040:006C=0E3B7h)
020A:075A  75 04                jnz loc_1           ; Jump if not zero
020A:075C  FF 06 006E           inc word ptr ds:[6Eh]   ; (0040:006E=13h)
020A:0760           loc_1:
020A:0760  83 3E 006E 18        cmp word ptr ds:[6Eh],18h   ; (0040:006E=13h)
020A:0765  75 15                jne loc_2           ; Jump if not equal
020A:0767  81 3E 006C 00B0      cmp word ptr ds:[6Ch],0B0h  ; (0040:006C=0E3B7h)
020A:076D  75 0D                jne loc_2           ; Jump if not equal
020A:076F  A3 006E              mov word ptr ds:[6Eh],ax    ; (0040:006E=13h)
020A:0772  A3 006C              mov word ptr ds:[6Ch],ax    ; (0040:006C=0E3B7h)
020A:0775  C6 06 0070 01        mov byte ptr ds:[70h],1 ; (0040:0070=0)
020A:077A  0C 08                or  al,8
; Отправка сигнала отключения моторчика
020A:077C           loc_2:
020A:077C  50                   push    ax
020A:077D  FE 0E 0040           dec byte ptr ds:[40h]   ; (0040:0040=0A9h)
020A:0781  75 0B                jnz loc_3           ; Jump if not zero
020A:0783  80 26 003F F0        and byte ptr ds:[3Fh],0F0h  ; (0040:003F=0)
020A:0788  B0 0C                mov al,0Ch
020A:078A  BA 03F2              mov dx,3F2h
020A:078D  EE                   out dx,al           ; port 3F2h, dsk0 contrl output
; проверка возможности вызова маскируемых прерываний
020A:078E           loc_3:
020A:078E  58                   pop ax
020A:078F  F7 06 0314 0004      test    word ptr ds:[314h],4    ; (0040:0314=3200h)
020A:0795  75 0C                jnz loc_4           ; Jump if not zero
020A:0797  9F                   lahf                ; Load ah from flags
020A:0798  86 E0                xchg    ah,al
020A:079A  50                   push    ax
020A:079B  26: FF 1E 0070       call    dword ptr es:[70h]  ; (0000:0070=6ADh)
020A:07A0  EB 03                jmp short loc_5     ; (07A5)
020A:07A2  90                   nop
; вызов прерывания по таймеру
020A:07A3           loc_4:
020A:07A3  CD 1C                int 1Ch         ; Timer break (call each 18.2ms)
020A:07A5           loc_5:
020A:07A5  E8 0011              call    sub_1           ; (07B9)
; сброс контроллера прерываний
020A:07A8  B0 20                mov al,20h          ; ' '
020A:07AA  E6 20                out 20h,al          ; port 20h, 8259-1 int command
                                ;  al = 20h, end of interrupt
```

```
56   ; восстановление значений регистров
57   020A:07AC  5A                  pop dx
58   020A:07AD  58                  pop ax
59   020A:07AE  1F                  pop ds
60   020A:07AF  07                  pop es
61   ; выход из программы
62   020A:07B0  E9 FE99             jmp $-164h
63   020A:07B3  C4                  db  0C4h
64   020A:07B4  C4 0E 93E9          les cx,dword ptr ds:[93E9h] ; (0000:93E9=76h) Load 32 bit ptr
65   020A:07B8  FE                  db  0FEh
```

Листинг процедуры sub_1

Листинг 2 – Процедура sub_1

```
1    sub_1 proc near
2    ;; Сохранение значений регистров, восстановление значений флагов
3    020A:07B9  1E                  push    ds
4    020A:07BA  50                  push    ax
5    020A:07BB  B8 0040             mov     ax,40h
6    020A:07BE  8E D8               mov     ds,ax
7    ;; Младший байт FLAGS в AH
8    020A:07C0  9F                  lahf                ; Load ah from flags
9    ;; Проверка старшего бита IOPL или DF? Проверка разрешены ли маскируемые прерывания
10   020A:07C1  F7 06 0314 2400     test    word ptr ds:[314h],2400h    ; (0040:0314=3200h)
11   020A:07C7  75 0C               jnz loc_7           ; Jump if not zero
12   020A:07C9  F0> 81 26 0314 FDFF            lock and word ptr ds:[314h],0FDFFh   ; (0040:0314=3200h)
13   020A:07D0          loc_6:
14   ;; AH в младший байт FLAGS. Сохраняем значение флагов, восстанавливаем значение регистров
15   020A:07D0  9E                  sahf                ; Store ah into flags
16   020A:07D1  58                  pop ax
17   020A:07D2  1F                  pop ds
18   020A:07D3  EB 03               jmp short loc_8     ; (07D8)
19   ;; Сброс IF в eflags, процессор игнорирует все прерывания, кроме NMI
20   020A:07D5          loc_7:
21   020A:07D5  FA                  cli             ; Disable interrupts
22   020A:07D6  EB F8               jmp short loc_6     ; (07D0)
23   020A:07D8          loc_8:
24   020A:07D8  C3                  retn
25   sub_1       endp
```

Схема алгоритма обработчика INT 8h

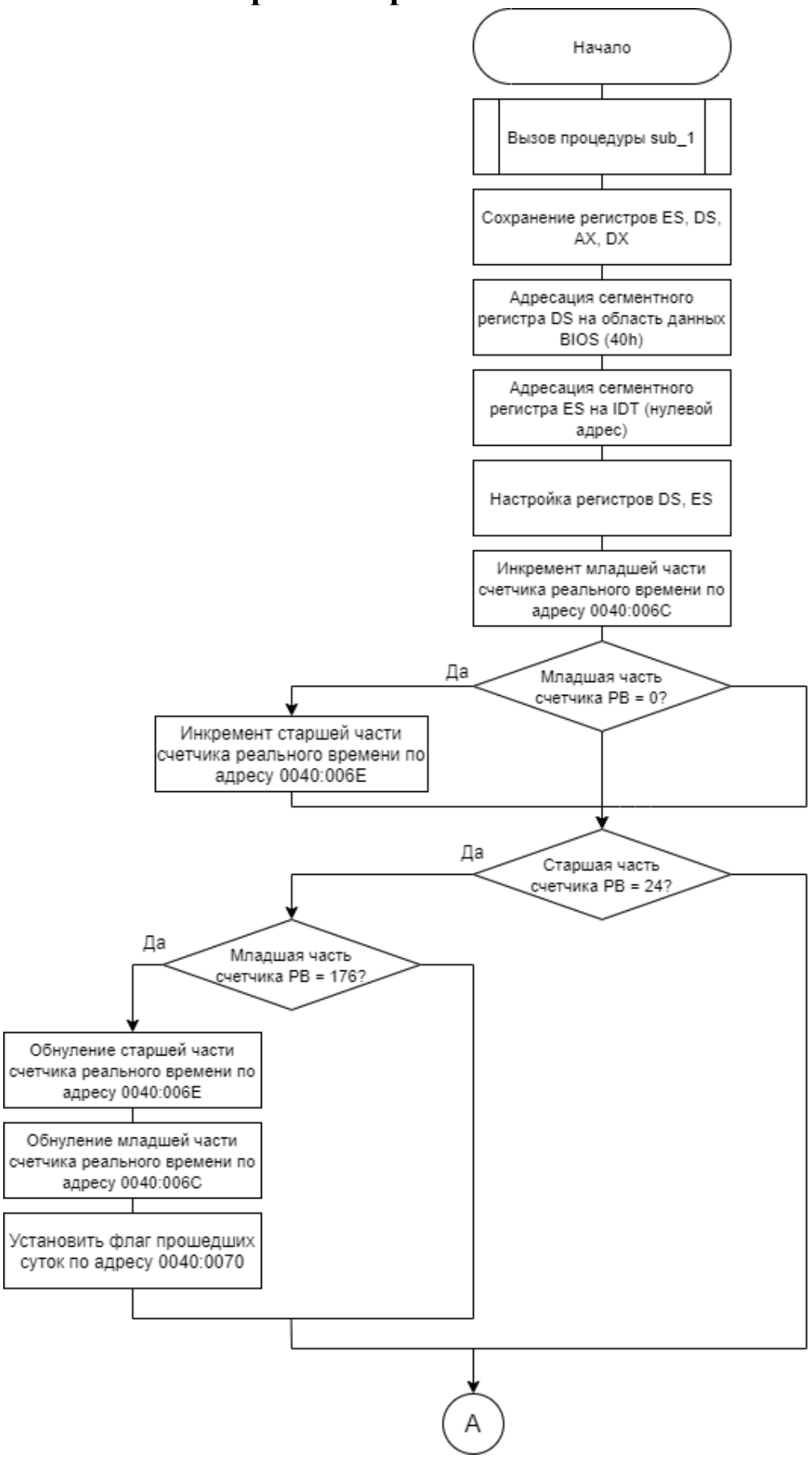

Рисунок 1 – Схема обработчика прерываний INT 8h

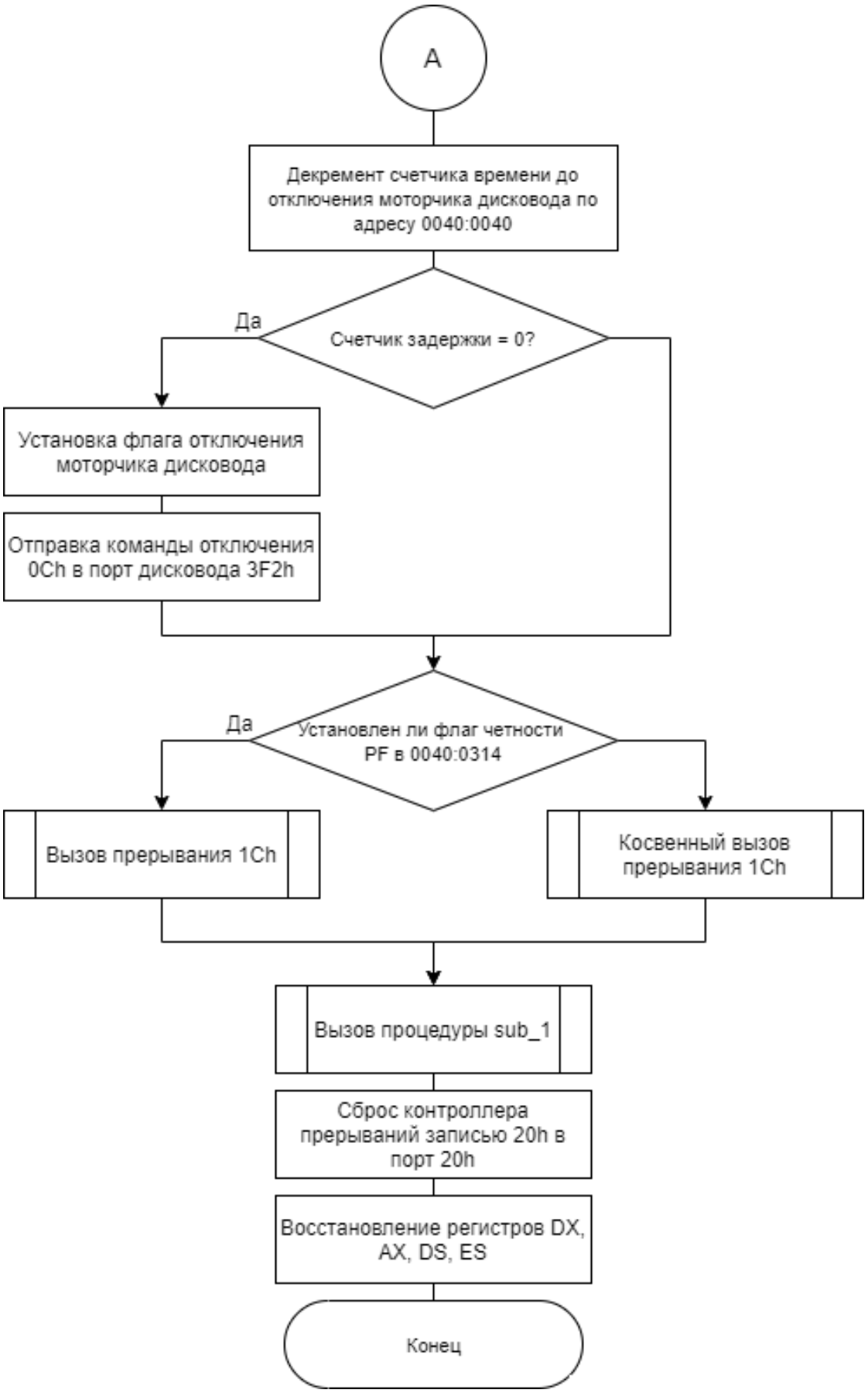

Рисунок 2 – Схема обработчика прерываний INT 8h

Схема алгоритма процедуры sub_1

Рисунок 3 – Схема процедуры sub_1

Функции прерывания от системного таймера int 8h

- **Инкремент счетчика тиков:** увеличивает на 1 текущее значение счетчика тиков. Если счетчик переполняется (с момента запуска таймера прошло более 24 часов), флаг завершения дня устанавливается в 1.
- **Декремент счётчика времени до отключения моторчика дисковода.** Когда время до отключения моторчика дисковода становится равным 0, выставляется флаг остановки моторчика и посылается команда контроллеру дисковода остановить моторчик.
- **Вызов пользовательского прерывания 1Ch.** Его стандартный обработчик состоит из одной команды IRET. Во время выполнения прерывания INT 1Ch все аппаратные прерывания запрещены.

Вывод по проделанной работе

В ходе проделанной работы мной был изучен метод получения листинга исходного кода операционной системы с помощью средства дизассемблирования Sourcer. Был изучен код обработчика прерывания от системного таймера int 8h, сформулирован алгоритм его работы с помощью схемы алгоритма, усвоены его функции. Выявленно, что обработчик прерывания int 8h реализует функции:

- инкремента счетчика системного времени и контроля за его переполнением,
- декремента счетчика времени до остановки моторчика дисковода и посыл контроллеру дисковода команды об остановке моторчика, по достижении счетчиком нуля
- вызова пользовательского прерывания 1Ch