Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский институт)»

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления
КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

О т ч ё т

п о л а б о р а т о р н о й р а б о т е 1

Дисциплина: Операционные системы

Тема лабораторной работы работы: Дизассемблирование INT 8h

Студентка гр. ИУ7-51б _____ **Сушина А.Д.**
(Подпись, дата) (И.О. Фамилия)

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

Москва, 2019г

Цель лабораторной работы: знакомство со средством дизассемблирования – **sourcer** и с получением дизассемблерного кода ядра операционной системы Windows на примере обработчика прерывания **Int 8h** в **virtual mode** – специальном режиме защищенного режима, который эмулирует реальный режим работы вычислительной системы на базе процессоров Intel.

Задание:

Используя sourser **(sr.exe**) получить дизассемблерный код обработчика аппаратного прерывания от системного таймера Int 8h.

На основе полученного кода составить алгоритм работы обработчика Int 8h.

По данной лабораторной работе составляется отчет в письменном виде.

- Отчет должен содержать: полученный ассемблерный код с адресами команд и коментариями;
- Графический алгоритм работы обработчика прерывания Int 8h, структурированный и выполненный в соответствии с ГОСТ 19.701-90 ЕСПД – «Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения».

Листинг программы прерывания INT 8h

Sourcer v5.10 14-Sep-19 3:13 pm Page 1

```
; Вход в прерывание, вызов sub_1
021D:0746  E8 0070          ;*                  call    sub_1                   ; (07B9)
021D:0746  E8 70 00                             db      0E8h, 70h, 00h
; Сохранение регистров ES, DS, AX, DX
021D:0749  06                                   push    es
021D:074A  1E                                   push    ds
021D:074B  50                                   push    ax
021D:074C  52                                   push    dx
; Установка регистров DS, ES и обнуление регистра AX
021D:074D  B8 0040                              mov     ax,40h
021D:0750  8E D8                                mov     ds,ax
021D:0752  33 C0                                xor     ax,ax                   ; Zero register
021D:0754  8E C0                                mov     es,ax
; Инкремент счетчика времени, старшего и младшего байтов
021D:0756  FF 06 006C                           inc     word ptr ds:[6Ch]
021D:075A  75 04                                jnz     loc_1                   ; Jump if not zero
021D:075C  FF 06 006E                           inc     word ptr ds:[6Eh]
; Проверка наступления новых суток(прошло 24 часа)
021D:0760                   loc_1:
021D:0760  83 3E 006E 18                        cmp     word ptr ds:[6Eh],18h       ;
(0040:006E=0Fh)
```

```
021D:0765  75 15                        jne     loc_2                   ; Jump if not equal
021D:0767  81 3E 006C 00B0                      cmp     word ptr ds:[6Ch],0B0h  ;
(0040:006C=3B3Bh)
021D:076D  75 0D                        jne     loc_2                   ; Jump if not equal
```

; Обнуление счетчика времени, если наступили новые сутки

```
021D:076F  A3 006E                      mov     word ptr ds:[6Eh],ax ; (0040:006E=0Fh)
021D:0772  A3 006C                      mov     word ptr ds:[6Ch],ax ; (0040:006C=3B3Bh)
```

; Установка единицы в ячейку 0040:0070h

```
021D:0775  C6 06 0070 01                mov     byte ptr ds:[70h],1     ; (0040:0070=0)
021D:077A  0C 08                        or      al,8
```

; Декремент счетчика до отключения моторчика дисковода

```
021D:077C                       loc_2:
021D:077C  50                           push    ax
021D:077D  FE 0E 0040                   dec     byte ptr ds:[40h] ;0040:0040=36h)
021D:0781  75 0B                        jnz     loc_3                   ; Jump if not zero
```

; Посылка сигнала на отключение моторчика дисковода

```
021D:0783  80 26 003F F0                and     byte ptr ds:[3Fh],0F0h          ; (0040:003F=0)
021D:0788  B0 0C                        mov     al,0Ch
021D:078A  BA 03F2                      mov     dx,3F2h
021D:078D  EE                           out     dx,al                   ; port 3F2h, dsk0 contrl
output
```

; Проверка возможности маскируемых прерываний

```
021D:078E                       loc_3:
021D:078E  58                           pop     ax
021D:078F  F7 06 0314 0004              test    word ptr ds:[314h],4 ; (0040:0314=3200h)
021D:0795  75 0C                        jnz     loc_4                   ; Jump if not zero
021D:0797  9F                           lahf                            ; Load ah from flags
021D:0798  86 E0                        xchg    ah,al
021D:079A  50                           push    ax
021D:079B  26: FF 1E 0070               call    dword ptr es:[70h]      ; (0000:0070=6ADh)
021D:07A0  EB 03                        jmp     short loc_5             ; (07A5)
021D:07A2  90                           nop
```

; Вызов пользовательского прерывания 1Ch

```
021D:07A3                       loc_4:
021D:07A3  CD 1C                        int     1Ch                     ; Timer break (call each
18.2ms)
021D:07A5                       loc_5:
021D:07A5  E8 0011                      call    sub_1                   ; (07B9)
```

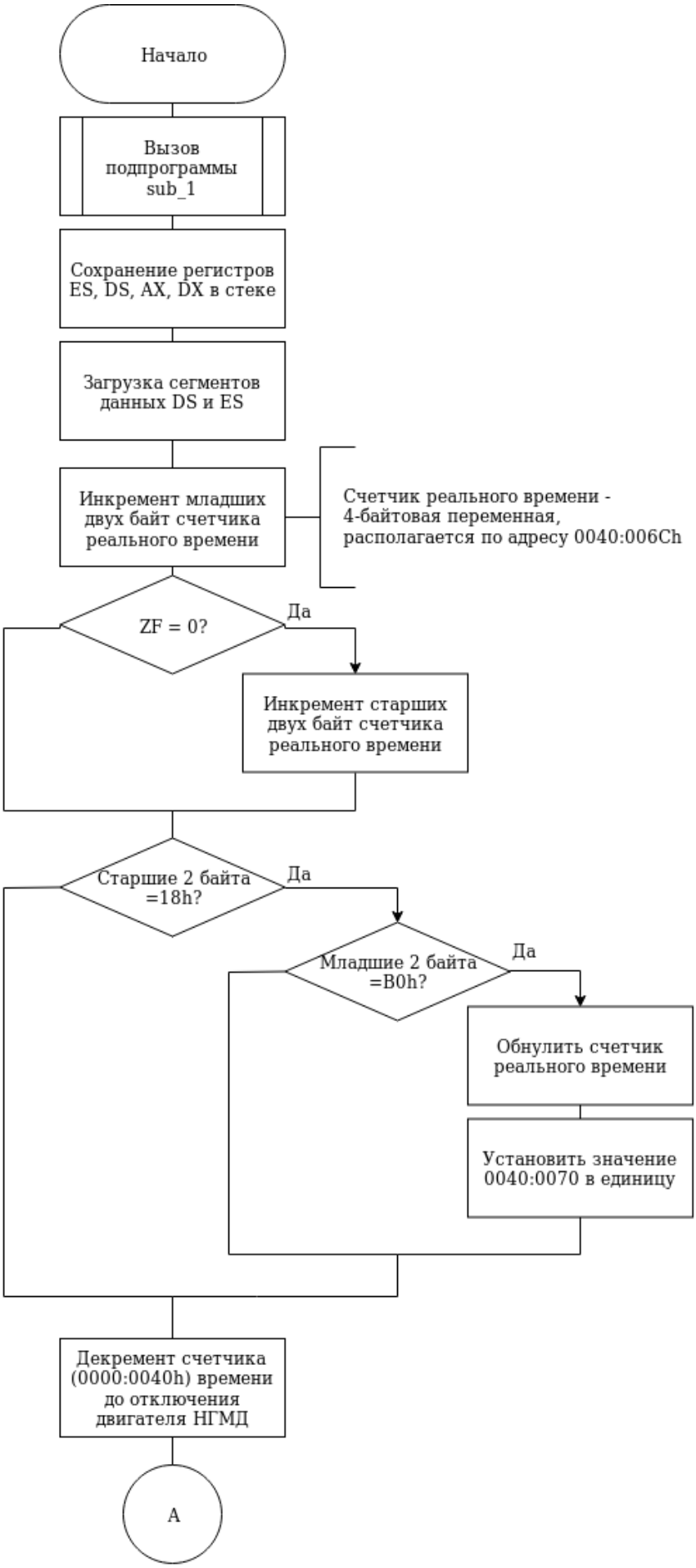
```
021D:07A8  B0 20                        mov    al,20h                ; ' '
021D:07AA  E6 20                        out    20h,al          ; port 20h, 8259-1 int command
                                                        ;  al = 20h, end of interrupt
; Восстановление значений регистров DX, AX, DS, ES
021D:07AC  5A                           pop    dx
021D:07AD  58                           pop    ax
021D:07AE  1F                           pop    ds
021D:07AF  07                           pop    es
; Переход
021D:07B0  E9 FE99                      jmp    $-164h          ;021D:064C

021D:064C                  loc_5:
021D:064C  1E                           push   ds
021D:064D  50                           push   ax
                                        ...
021D:06AA                  loc_13:
021D:06AA  58                           pop    ax
021D:06AB  1F                           pop    ds
; Возврат из обработчика прерываний
021D:06AC  CF                           iret


                                   SUBROUTINE
                           sub_1        proc   near
; Сохранение регистров DS, AX
021D:07B9  1E                           push   ds
021D:07BA  50                           push   ax
; Загрузка сегмента данных, загрузка ah
021D:07BB  B8 0040                      mov    ax,40h
021D:07BE  8E D8                        mov    ds,ax
021D:07C0  9F                           lahf                         ; Load ah from flags
; Проверка равен ли старший бит IOPL единице
021D:07C1  F7 06 0314 2400              test   word ptr ds:[314h],2400h
021D:07C7  75 0C                        jnz    loc_7                 ; Jump if not zero
; Сброс флага разрешения прерывания
021D:07C9  F0> 81 26 0314 FDFF          lock    and    word ptr ds:[314h],0FDFFh
; Восстановление регистров
```

```
021D:07D0                       loc_6:
021D:07D0  9E                                   sahf                    ; Store ah into flags
021D:07D1  58                                   pop     ax
021D:07D2  1F                                   pop     ds
021D:07D3  EB 03                                jmp     short loc_8     ; (07D8)
; Запрет маскируемых прерываний командой cli
021D:07D5                       loc_7:
021D:07D5  FA                                   cli                     ; Disable interrupts
021D:07D6  EB F8                                jmp     short loc_6     ; (07D0)
; Возврат из процедуры
021D:07D8                       loc_8:
021D:07D8  C3                                   retn
                                sub_1           endp
```

Схема работы обработчика INT 8h



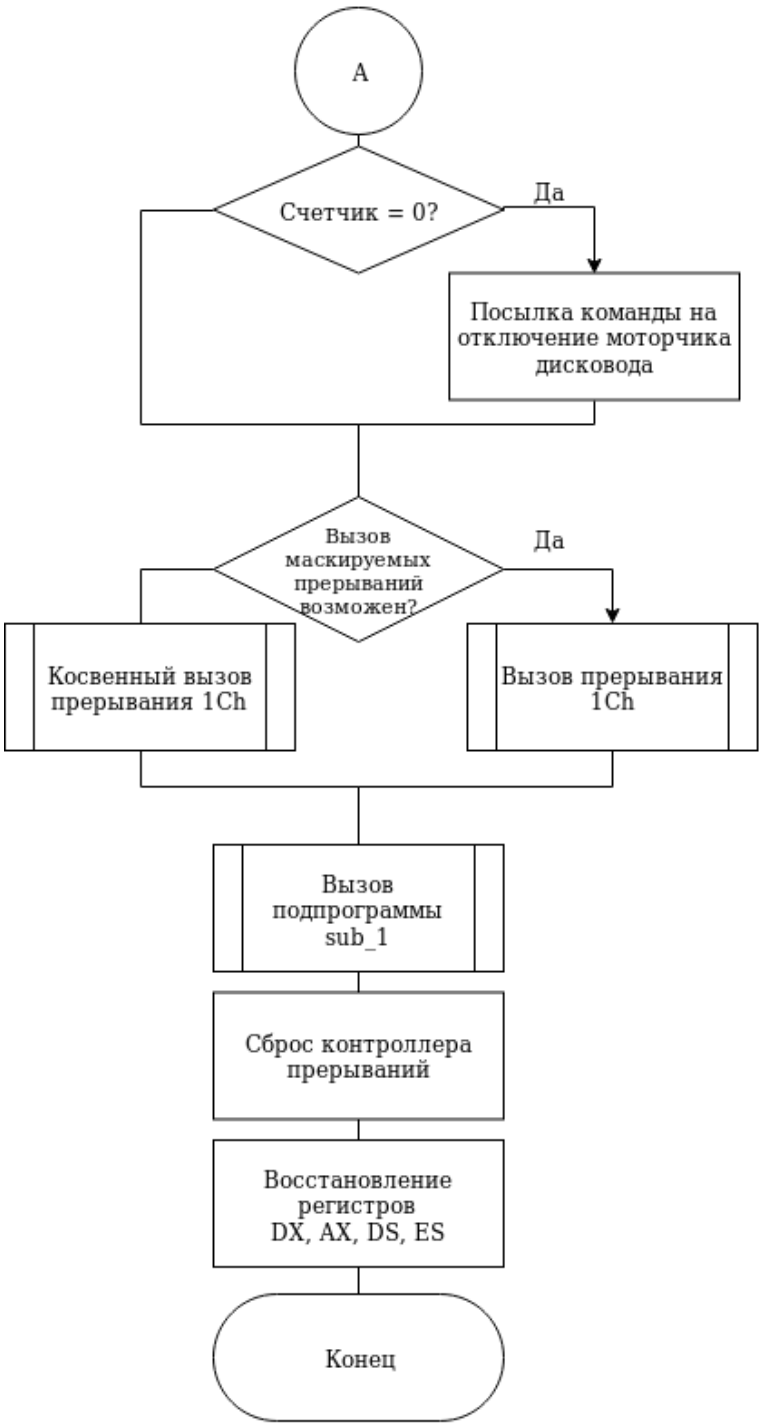

A
Счетчик = 0?
Да
Посылка команды на отключение моторчика дисковода
Вызов маскируемых прерываний возможен?
Да
Косвенный вызов прерывания 1Ch
Вызов прерывания 1Ch
Вызов подпрограммы sub_1
Сброс контроллера прерываний
Восстановление регистров DX, AX, DS, ES
Конец

Схема работы Subroutine