



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»

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ»

ОТЧЕТ по лабораторной работе №1

Название: Дизассемблирование INT 8h

Дисциплина: Операционные системы

Студент

Царев А.А.

Группа

ИУ7-53Б

Преподаватель

Рязанова Н.Ю.

Москва, 2022

Введение

Цель 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 sourcer (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.

Полученный дизассемблерный код

Листинг INT 8h

```
1      ; Вызов процедуры sub_2
2      020A:0746  E8 0070          call     sub_2          ; (07B9)
3
4      ; Сохранение регистров ES, DS, AX и DX
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
10     ; Запись 0040h в регистр DS
11     ; Счетчик реального времени находится в области данных BIOS (по адре
        су 0040:006C)
12     020A:074D  B8 0040          mov ax,40h
13     020A:0750  8E D8          mov ds,ax
14
15     ; Обнуление регистров AX и ES
16     020A:0752  33 C0          xor ax,ax          ; Zero register
17     020A:0754  8E C0          mov es,ax
18
19     ; Увеличение младших 2 байт счетчика реального времени на единицу
20     020A:0756  FF 06 006C      inc word ptr ds:[6Ch] ;
        (0040:006C=124Dh)
21     020A:075A  75 04          jnz loc_16          ; Jump if not
        zero
22
23     ; Увеличение старших 2 байт счетчика реального времени на единицу
24     020A:075C  FF 06 006E      inc word ptr ds:[6Eh] ;
        (0040:006E=3)
25
26     ; Проверка на то, что прошло 24 часа
27     020A:0760          loc_16:
28     020A:0760  83 3E 006E 18      cmp word ptr ds:[6Eh],18h ;
        (0040:006E=3)
29     020A:0765  75 15          jne loc_17          ; Jump if not
        equal
30     020A:0767  81 3E 006C 00B0      cmp word ptr ds:[6Ch],0B0h ;
        (0040:006C=124Dh)
31     020A:076D  75 0D          jne loc_17          ; Jump if not
        equal
32
33     ; Если прошло 24 часа, то счетчик обнуляется и устанавливается едини
        ца по адресу 0040:0070
```

```

34 020A:076F A3 006E          mov word ptr ds:[6Eh],ax      ;
    (0040:006E=3)
35 020A:0772 A3 006C          mov word ptr ds:[6Ch],ax      ;
    (0040:006C=124Dh)
36 020A:0775 C6 06 0070 01    mov byte ptr ds:[70h],1      ;
    (0040:0070=0)
37
38 ; Запись 8 в AL
39 020A:077A 0C 08          or al,8
40
41 020A:077C                loc_17:
42
43 ; Сохранение регистра AX в стеке
44 020A:077C 50             push ax
45
46 ; Уменьшаем на единицу значение счетчика времени, оставшегося до вык
    лючения моторчика
47 ; Сам счетчик находится по адресу 0040:0040
48 020A:077D FE 0E 0040      dec byte ptr ds:[40h]      ;
    (0040:0040=43h)
49 020A:0781 75 0B          jnz loc_18                ; Jump if not
    zero
50
51 ; Если счетчик равен 0, устанавливаем флаги, отвечающие за отключени
    е моторчика
52 020A:0783 80 26 003F F0    and byte ptr ds:[3Fh],0F0h  ;
    (0040:003F=0)
53
54 ; Отключение моторчика дисковод
55 020A:0788 B0 0C          mov al,0Ch
56 020A:078A BA 03F2        mov dx,3F2h
57 020A:078D EE            out dx,al                ; port 3F2h,
    dsk0 contrl output
58
59 020A:078E                loc_18:
60
61 ; Восстановление регистра AX
62 020A:078E 58             pop ax
63
64 ; Проверяем флаг четности PF
65 020A:078F F7 06 0314 0004 test word ptr ds:[314h],4
    ; (0040:0314=3200h)
66 020A:0795 75 0C          jnz loc_19                ; Jump if not
    zero
67
68 ; Запись младшего байта регистра флагов в AH

```

```

69      020A:0797  9F                                lahf                                ; Load ah from
      flags
70      020A:0798  86 E0                            xchg    ah,al
71
72      ; Сохранение регистра AX
73      020A:079A  50                                push    ax
74
75      ; Вызов обработчика прерывания 1Ch через таблицу векторов прерываний
      .
76      ; Вызов осуществляется через call, чтобы при выполнении команды iret
77      ; в регистре флагов было установлено значение регистра AX, который
78      ; мы сохранили до вызова обработчика прерываний
79      020A:079B  26: FF 1E 0070                    call    dword ptr es:[70h]  ;
      (0000:0070=6ADh)
80      020A:07A0  EB 03                            jmp short loc_20            ; (07A5)
81      020A:07A2  90                                nop
82
83      020A:07A3                                loc_19:
84      020A:07A3  CD 1C                            int 1Ch                    ; Timer break (call
      each 18.2ms)
85
86      020A:07A5                                loc_20:
87      020A:07A5  E8 0011                            call    sub_2              ; (07B9)
88
89      ; Сброс контроллера прерываний
90      020A:07A8  B0 20                            mov al,20h                ; ' '
91      020A:07AA  E6 20                            out 20h,al                ; port 20h,
      8259-1 int command
92      ; al = 20h, end of interrupt
93
94      ; Восстановление регистров DX, AX, DS и ES
95      020A:07AC  5A                                pop dx
96      020A:07AD  58                                pop ax
97      020A:07AE  1F                                pop ds
98      020A:07AF  07                                pop es
99
100     020A:07B0  E9 FE99                            jmp loc_1                  ; (064C)
101
102     ; ...
103
104     020A:064C                                loc_1:
105     020A:064C  1E                                push    ds
106     020A:064D  50                                push    ax
107
108     ; ...
109

```

```

110      020A:06AA  58                      pop ax
111      020A:06AB  1F                      pop ds
112
113      020A:06AC  CF                      ired                ; Interrupt
      return

```

Листинг процедуры sub_2

```

1      sub_2      proc      near
2
3      ; Сохранение регистров DS и AX
4      020A:07B9  1E                      push    ds
5      020A:07BA  50                      push    ax
6
7      ; Запись 0040h в регистр DS
8      020A:07BB  B8 0040                mov ax,40h
9      020A:07BE  8E D8                mov ds,ax
10
11     ; Запись младшего байта регистра флагов в AH
12     020A:07C0  9F                      lahf                ; Load ah from
      flags
13
14     ; Проверка флага DF и старшего бита флага IOPL
15     020A:07C1  F7 06 0314 2400        test    word ptr ds:[314h],2400h
      ; (0040:0314=3200h)
16     020A:07C7  75 0C                      jnz loc_22          ; Jump if not
      zero
17
18     ; Сброс флага IF в области данных BIOS (в 020A:0314)
19     ; Команда lock используется для того, чтобы следующая за ней команда
20     ; была неделимой
21     020A:07C9  F0> 81 26 0314 FDFF        lock and word ptr ds:[314
      h],0FDFFh      ; (0040:0314=3200h)
22
23     020A:07D0                      loc_21:
24
25     ; Запись регистра AH в младший байт регистра флагов
26     020A:07D0  9E                      sahf                ; Store ah into
      flags
27
28     ; Восстановление регистров AX и DS
29     020A:07D1  58                      pop ax
30     020A:07D2  1F                      pop ds
31     020A:07D3  EB 03                      jmp short loc_23      ; (07D8)
32

```

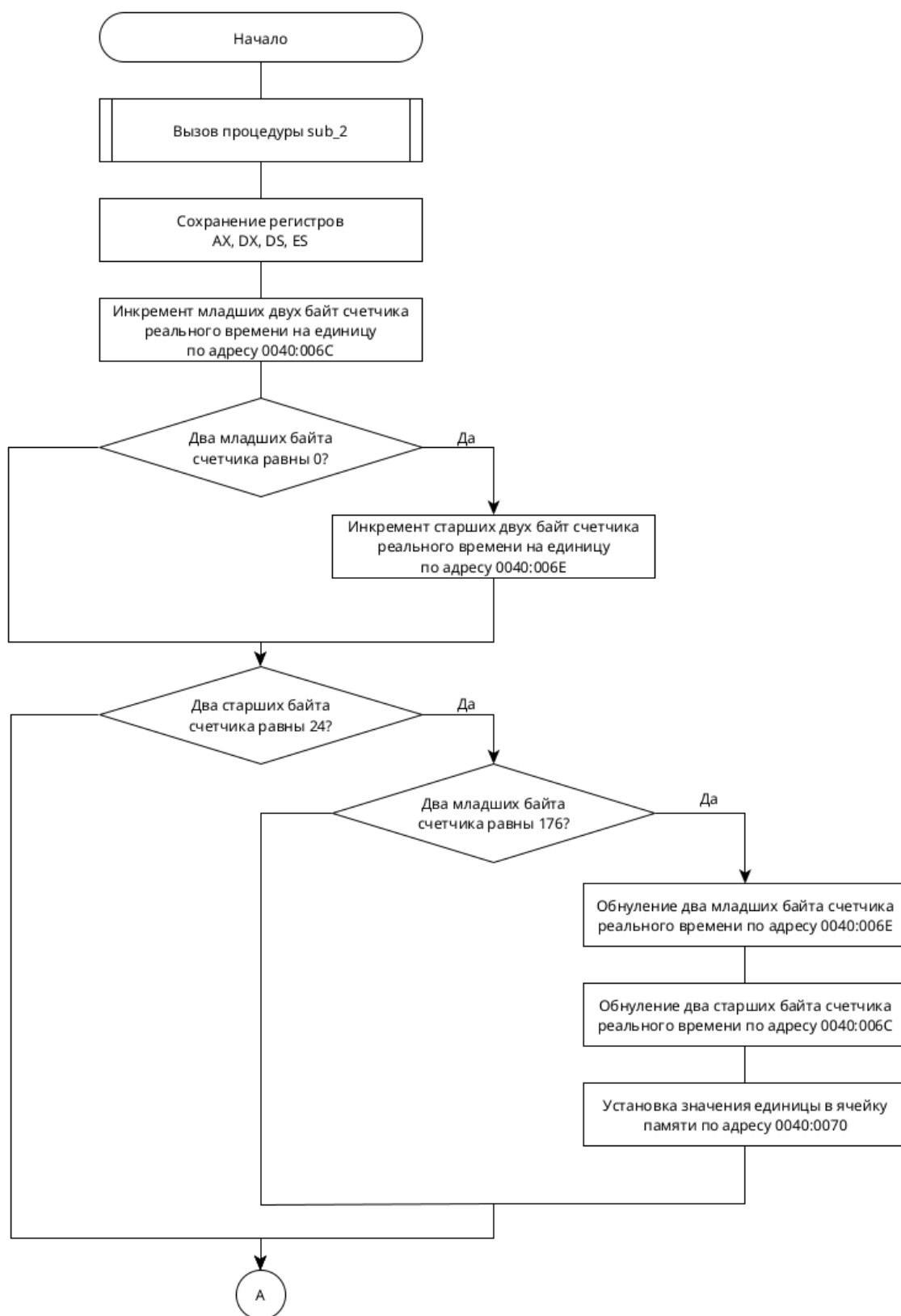
```

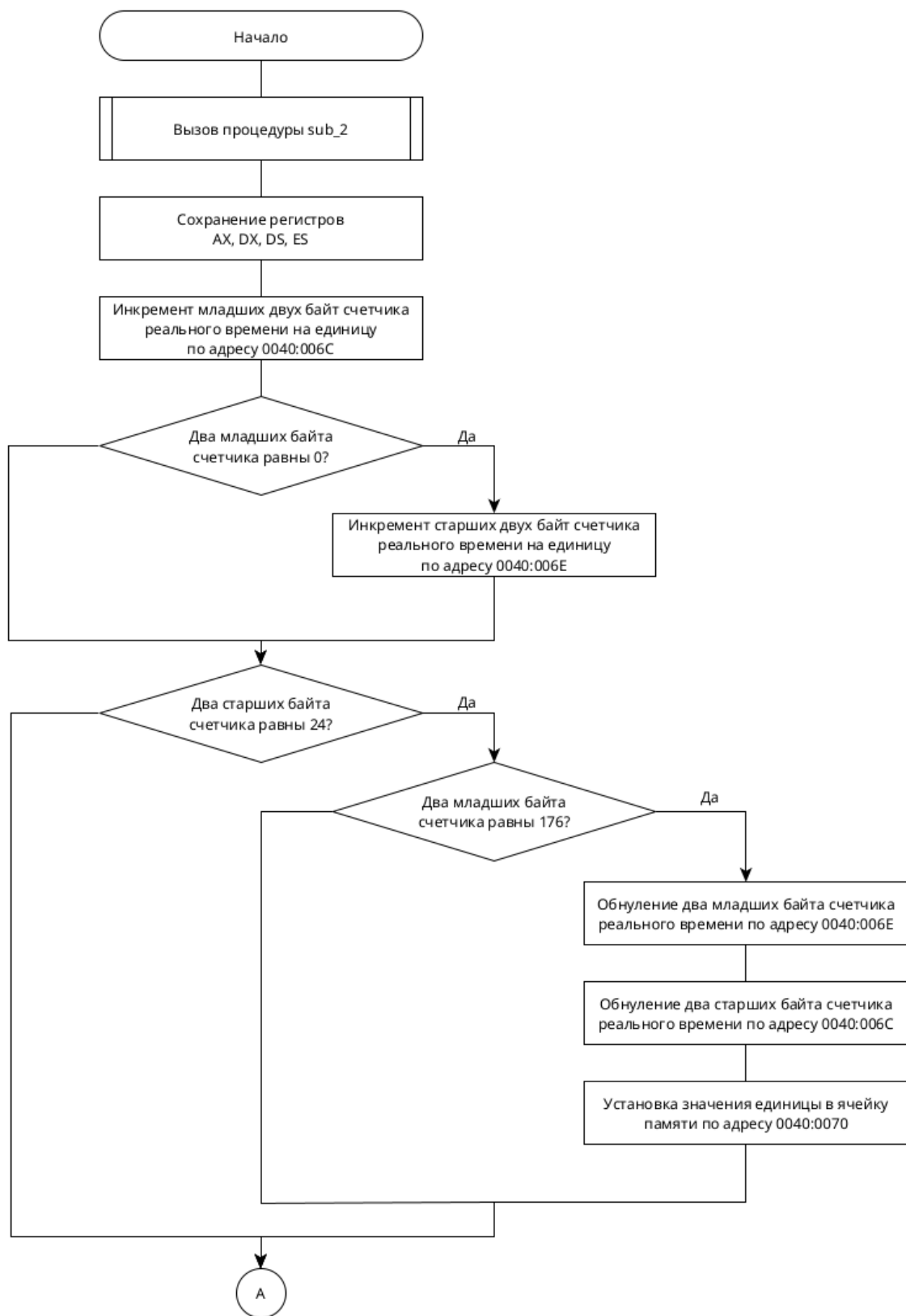
33      020A:07D5          loc_22:
34
35      ; Если установлен хотя бы один из флагов DF или IOPL, то происходит
      сброс флага IF
36      020A:07D5  FA          cli          ; Disable interrupts
37      020A:07D6  EB F8          jmp short loc_21          ; (07D0)
38      020A:07D8          loc_23:
39
40      ; Возврат из процедуры
41      020A:07D8  C3          retn
42      sub_2          endp

```

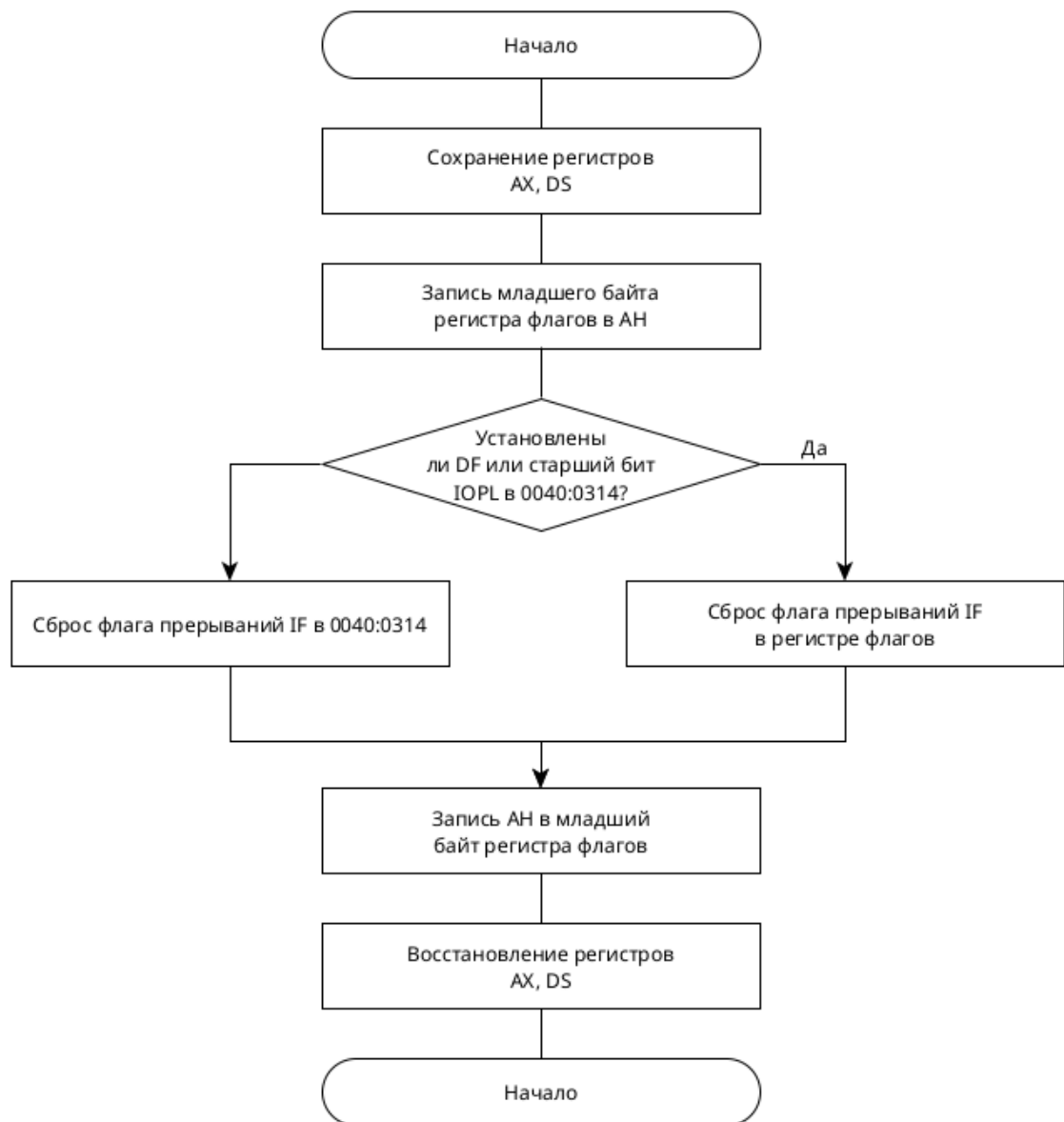
1 Схемы алгоритмов

1.1 Схема алгоритма INT 8h





1.2 Схема алгоритма процедуры sub_2



2 Вывод

Во время выполнения данной лабораторной работы я познакомился со средством дизассемблирования – `sourceg` 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.

Функции обработчика прерывания `INT 8h`:

- Инкремент счетчика реального времени. Обработчик каждый раз увеличивает на единицу текущее значение 4-байтовой переменной, располагающейся в области данных BIOS по адресу `0040:006Ch` - счетчик таймера. Если этот счетчик переполнится из-за того что прошло более 24 часов с момента запуска таймера, в ячейку `0000:0470h` заносится значение 1.
- Декремент времени до отключения моторчика дисковод. Если после последнего обращения к НГМД прошло более 2 секунд, обработчик прерывания выключает двигатель. Ячейка с адресом `0040:0040h` содержит время, оставшееся до выключения двигателя. Это время постоянно уменьшается обработчиком прерывания таймера. Когда оно становится равно 0, двигатель НГМД отключается.
- Вызов пользовательского прерывания `1Ch`. Его стандартный обработчик состоит из одной команды `IRET`. Во время выполнения прерывания `1Ch` все аппаратные прерывания запрещены.