



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     020A:074D  B8 0040          mov     ax,40h
12     020A:0750  8E D8           mov     ds,ax
13
14     ; Обнуление регистров AX и ES
15     020A:0752  33 C0           xor     ax,ax             ; Zero register
16     020A:0754  8E C0           mov     es,ax
17
18     ; Инкремент младших двух байт счетчика реального времени
19     020A:0756  FF 06 006C       inc     word ptr ds:[6Ch] ;
20     (0040:006C=124Dh)
21
22     020A:075A  75 04           jnz     loc_16           ; Jump if not
23     zero
24
25     ; Инкремент старших двух байт счетчика реального времени
26     020A:075C  FF 06 006E       inc     word ptr ds:[6Eh] ;
27     (0040:006E=3)
28
29     ; Проверка на то, что прошло 24 часа
30     ; 24 * 60 * 60 * (1193180 / 65536) ~= 1573040 = 1800B0h
31     020A:0760          loc_16:
32     020A:0760  83 3E 006E 18     cmp     word ptr ds:[6Eh],18h ;
33     (0040:006E=3)
34
35     020A:0765  75 15           jne     loc_17           ; Jump if not
36     equal
37
38     020A:0767  81 3E 006C 00B0   cmp     word ptr ds:[6Ch],0B0h ;
39     (0040:006C=124Dh)
40
41     020A:076D  75 0D           jne     loc_17           ; Jump if not
42     equal
43
44     ; Если прошло 24 часа, то счетчик реального времени обнуляется, и ус
45     танавливается единица 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 значения 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      ; Декремент счетчика времени до выключения моторчика
47      020A:077D  FE 0E 0040            dec byte ptr ds:[40h]      ;
      (0040:0040=43h)
48      020A:0781  75 0B                  jnz loc_18                ; Jump if not
      zero
49
50      ; Если счетчик времени до выключения моторчика равен 0, устанавливаем
      флаги, отвечающие за отключение моторчика
51      020A:0783  80 26 003F F0          and byte ptr ds:[3Fh],0F0h  ;
      (0040:003F=0)
52
53      ; Отключение моторчика дисковод
54      020A:0788  B0 0C                  mov al,0Ch
55      020A:078A  BA 03F2                mov dx,3F2h
56      020A:078D  EE                      out dx,al                ; port 3F2h,
      dsk0 contrl output
57
58      020A:078E                loc_18:
59
60      ; Восстановление регистра AX
61      020A:078E  58                      pop ax
62
63      ; Проверяем флаг четности PF
64      020A:078F  F7 06 0314 0004        test word ptr ds:[314h],4
      ; (0040:0314=3200h)
65      020A:0795  75 0C                  jnz loc_19                ; Jump if not
      zero
66
67      ; Запись младшего байта регистра флагов в AH
68      020A:0797  9F                      lahf                    ; Load ah from
      flags

```

```

69      020A:0798  86 E0                      xchg     ah,al
70
71      ; Сохранение регистра AX
72      020A:079A  50                      push     ax
73
74      ; Косвенный вызов прерывания 1Ch.
75      ; Вызов осуществляется через call, чтобы при выполнении команды iret
76      ; в регистр флагов был записан регистр AX, который
77      ; был сохранен до вызова прерывания
78      020A:079B  26: FF 1E 0070          call     dword ptr es:[70h] ;
          (0000:0070=6ADh)
79      020A:07A0  EB 03                      jmp short loc_20             ; (07A5)
80      020A:07A2  90                      nop
81
82      020A:07A3                      loc_19:
83      020A:07A3  CD 1C                      int 1Ch                     ; Timer break (call
          each 18.2ms)
84
85      020A:07A5                      loc_20:
86      020A:07A5  E8 0011                    call     sub_2              ; (07B9)
87
88      ; Сброс контроллера прерываний
89      020A:07A8  B0 20                      mov al,20h                 ; ' '
90      020A:07AA  E6 20                      out 20h,al                 ; port 20h,
          8259-1 int command
91      ; al = 20h, end of interrupt
92
93      ; Восстановление регистров DX, AX, DS и ES
94      020A:07AC  5A                      pop dx
95      020A:07AD  58                      pop ax
96      020A:07AE  1F                      pop ds
97      020A:07AF  07                      pop es
98
99      020A:07B0  E9 FE99                    jmp loc_1                   ; (064C)
100
101      ; ...
102
103      020A:064C                      loc_1:
104      020A:064C  1E                      push     ds
105      020A:064D  50                      push     ax
106
107      ; ...
108
109      020A:06AA  58                      pop ax
110      020A:06AB  1F                      pop ds
111

```

112	020A:06AC	CF	iret	; 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 0040:0314)
19         ; Команда lock используется для того, чтобы следующая за ней команда
20         ; была неделимой
21         020A:07C9      F0> 81 26 0314 FDFH      lock and word ptr ds:[314
           h],0FDFHh ; (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

```

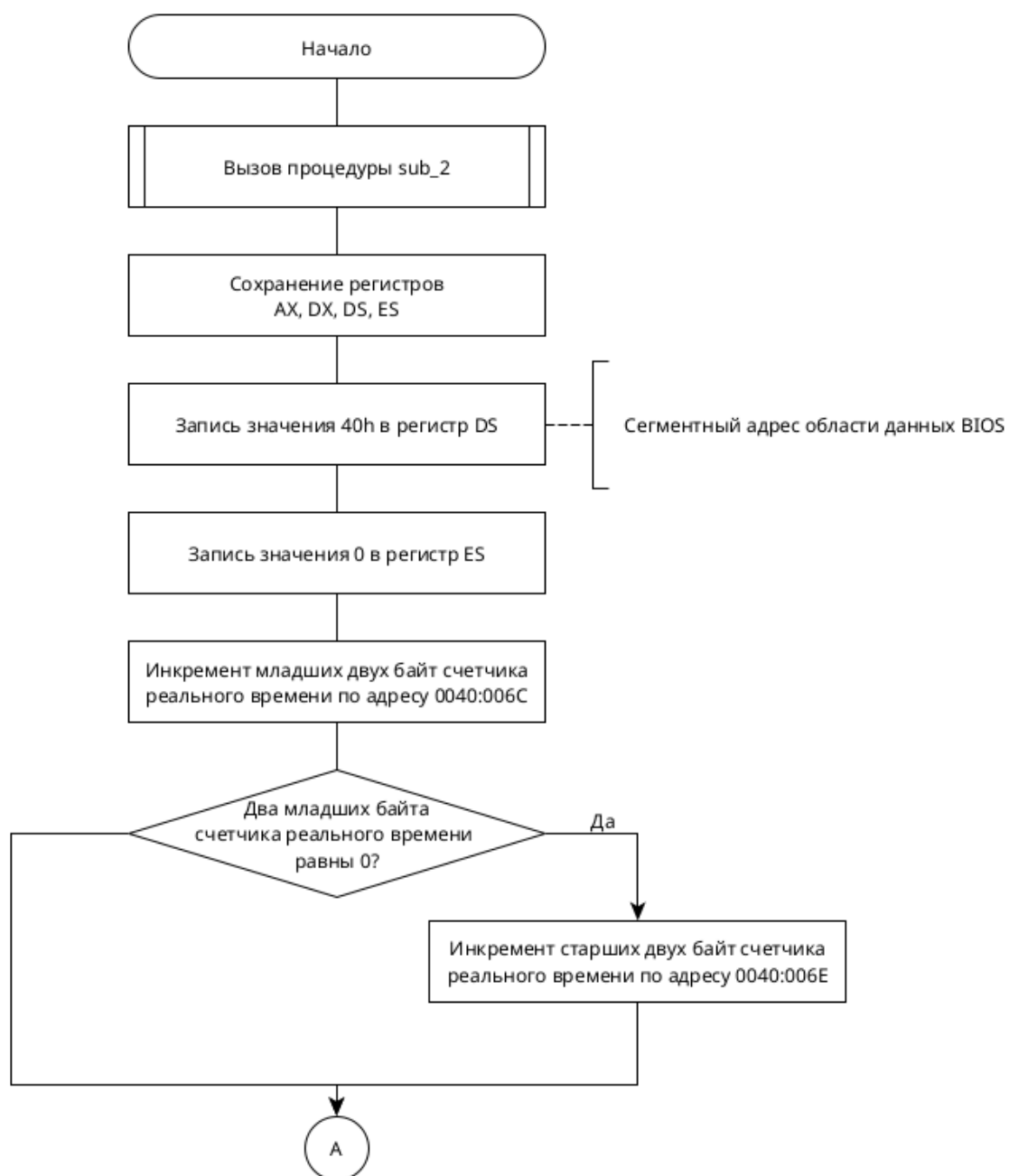
```

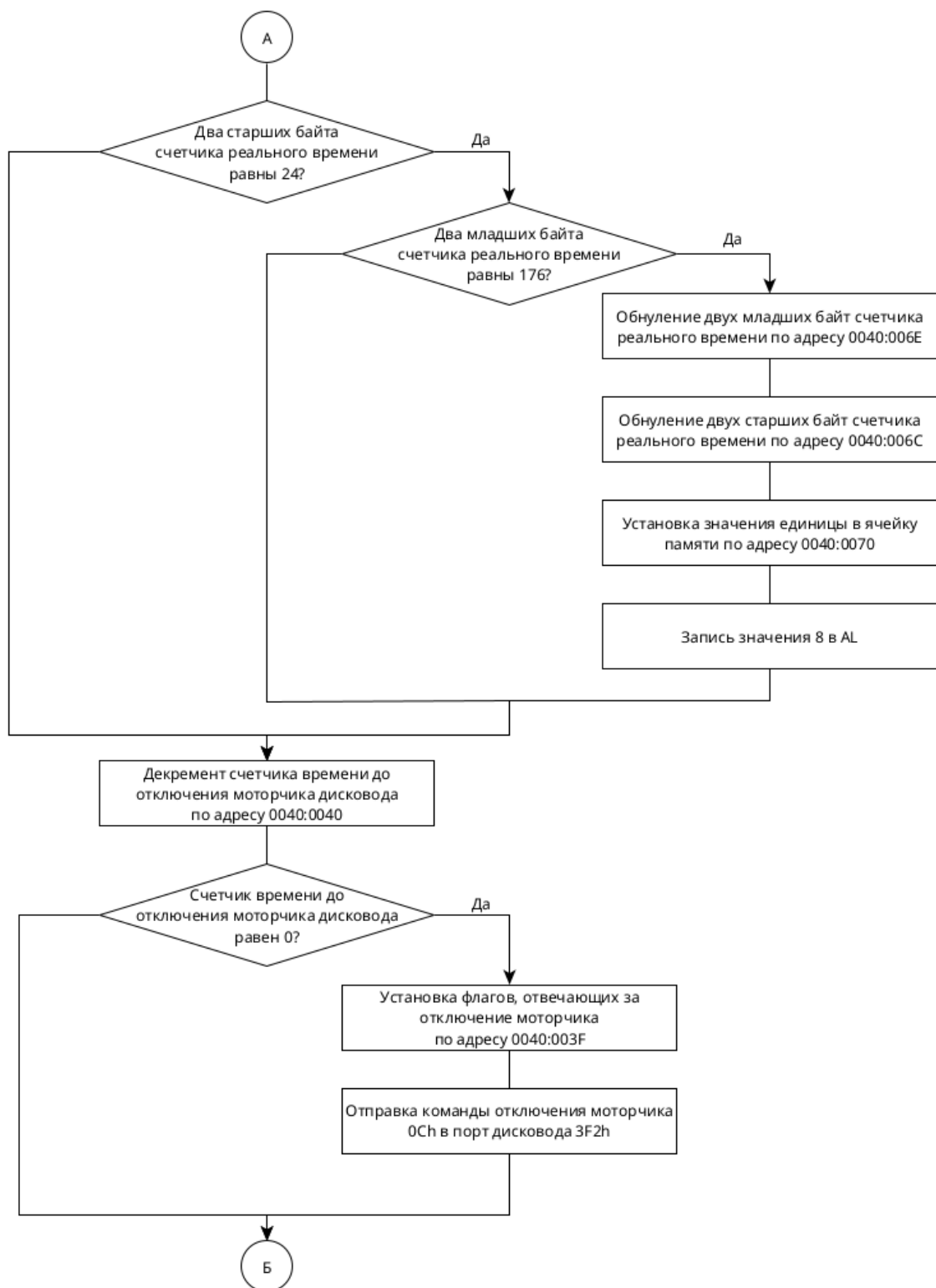
35      ; Если установлен хотя бы DF или старший бит IOPL, то происходит сброс
      ;   флага IF
36      020A:07D5  FA                      cli                      ; Disable interrupts
37      020A:07D6  EB F8                  jmp short loc_21          ; (07D0)
38      020A:07D8                   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

```

Схемы алгоритмов

Схема алгоритма INT 8h





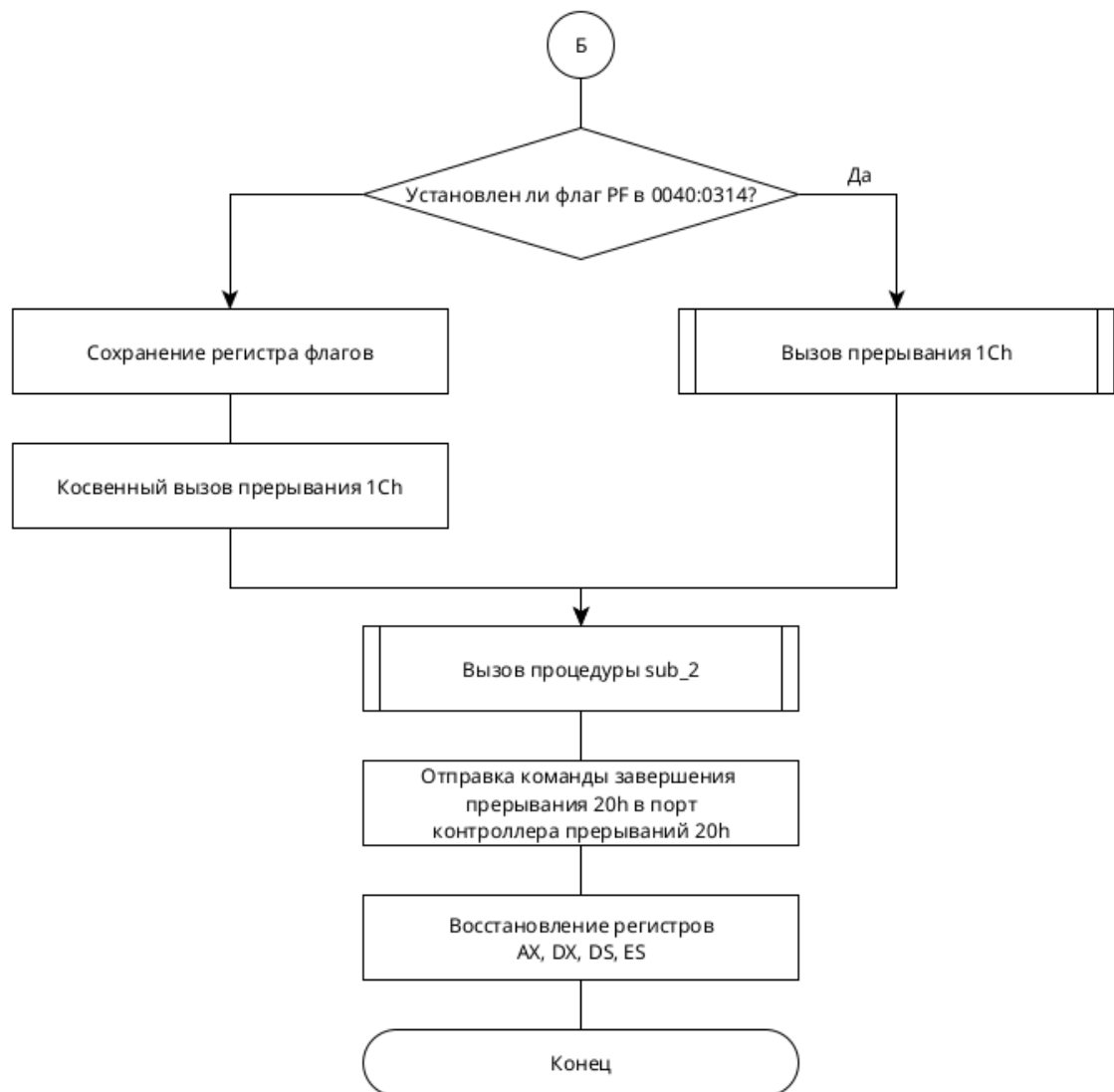
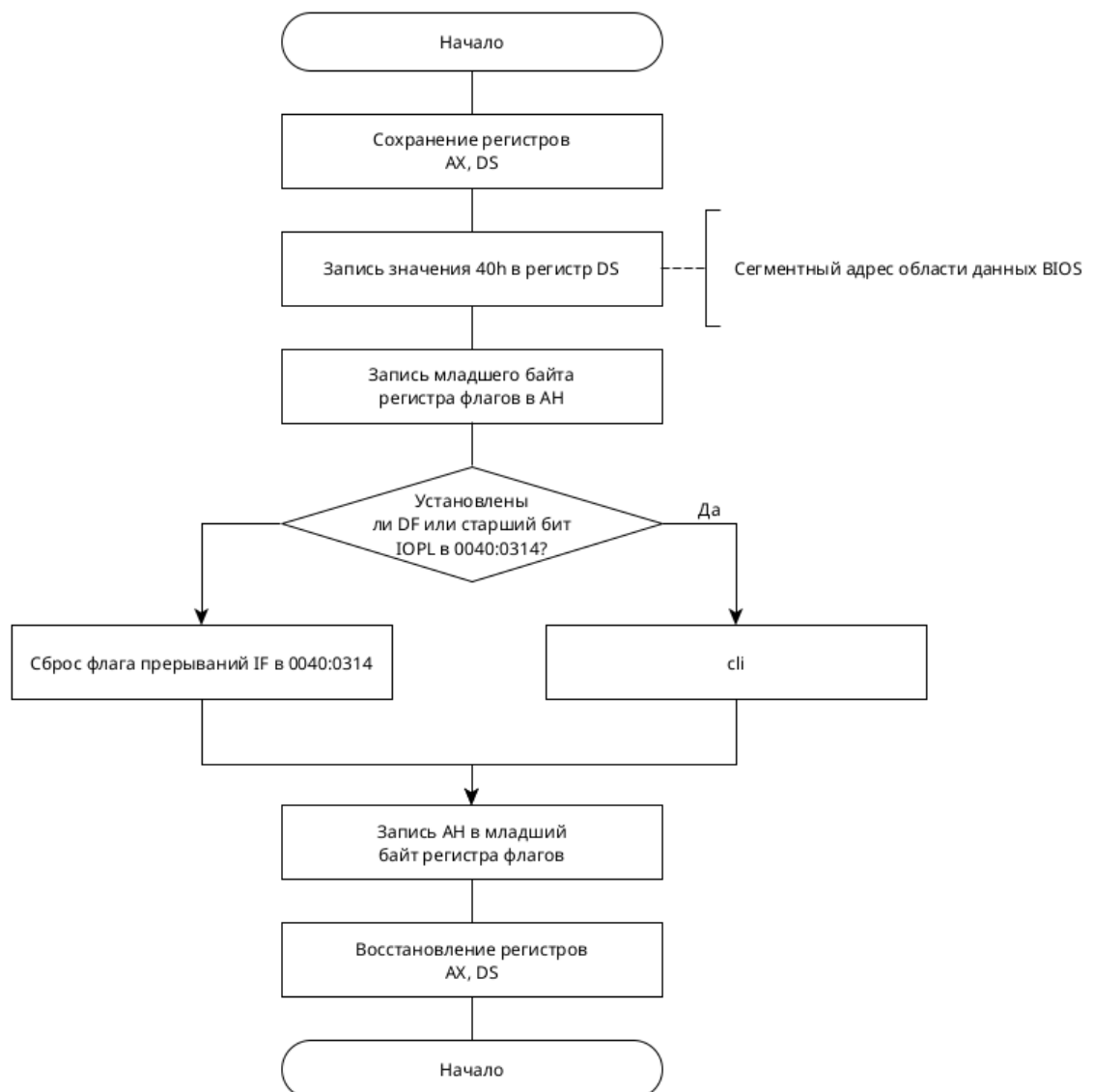


Схема алгоритма процедуры sub_2



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.