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 Прянишников А. Н.

Группа ИУ7-55Б

Оценка (баллы) _____

Преподаватели Рязанова Н.Ю.

Москва — 2021 г.

Получение дизассемблированного кода обработчика прерывания int 8h

Для выполнения лабораторной работы на виртуальную машину была поставлена операционная система Windows XP.

Для определения адреса вектора из таблицы векторов прерываний нужно вычислить смещение. Так как номер прерывания – 8h, а длина far-адреса – 4, то нужно умножить номер вектора на 4 и перевести в 16-ричную систему. Получившееся значение – **20h**.

Для получения содержимого по адресу 0000:0020h, то есть адреса обработчика прерывания, используется программа-отладчик **debug**, доступная в командной строке Windows XP. После команды **D 0000:0020 L 4** на экране появляется значение четырёх байт: **46 07 0A 02**.

Так как у байтов обратный порядок следования, нужно поменять порядок местами. Итоговый начальный адрес обработчика прерывания int 8h – **020A:0746**.

Получение дизассемблированного кода производится через утилиту sourcer. Для получения листинга кода нужно задать начальный и конечный адреса. Конец обработчика прерывания можно найти, зная, что код обработчика заканчивается командой **iret**. По адресу 020A:07B0 находится команда jmp $-164h. По смещению -164h находится несколько команд, в числе которых iret по адресу **020A:06AC**. Поэтому листинг кода выполнялся в два этапа: сначала получения кода от смещения 0746 до смещения 07B0, а затем - от 064C до 06AC.

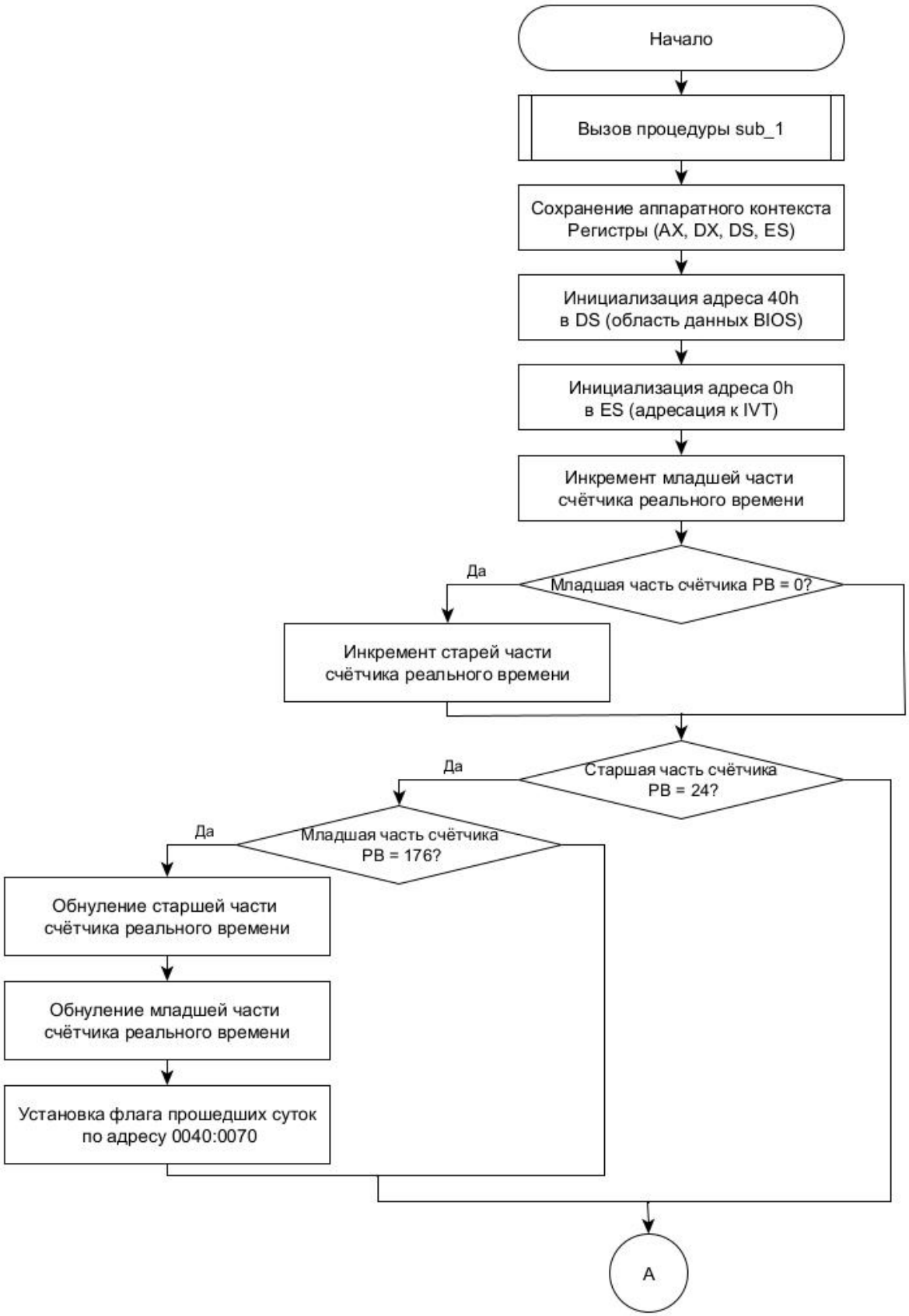
Листинг обработчика INT 8h

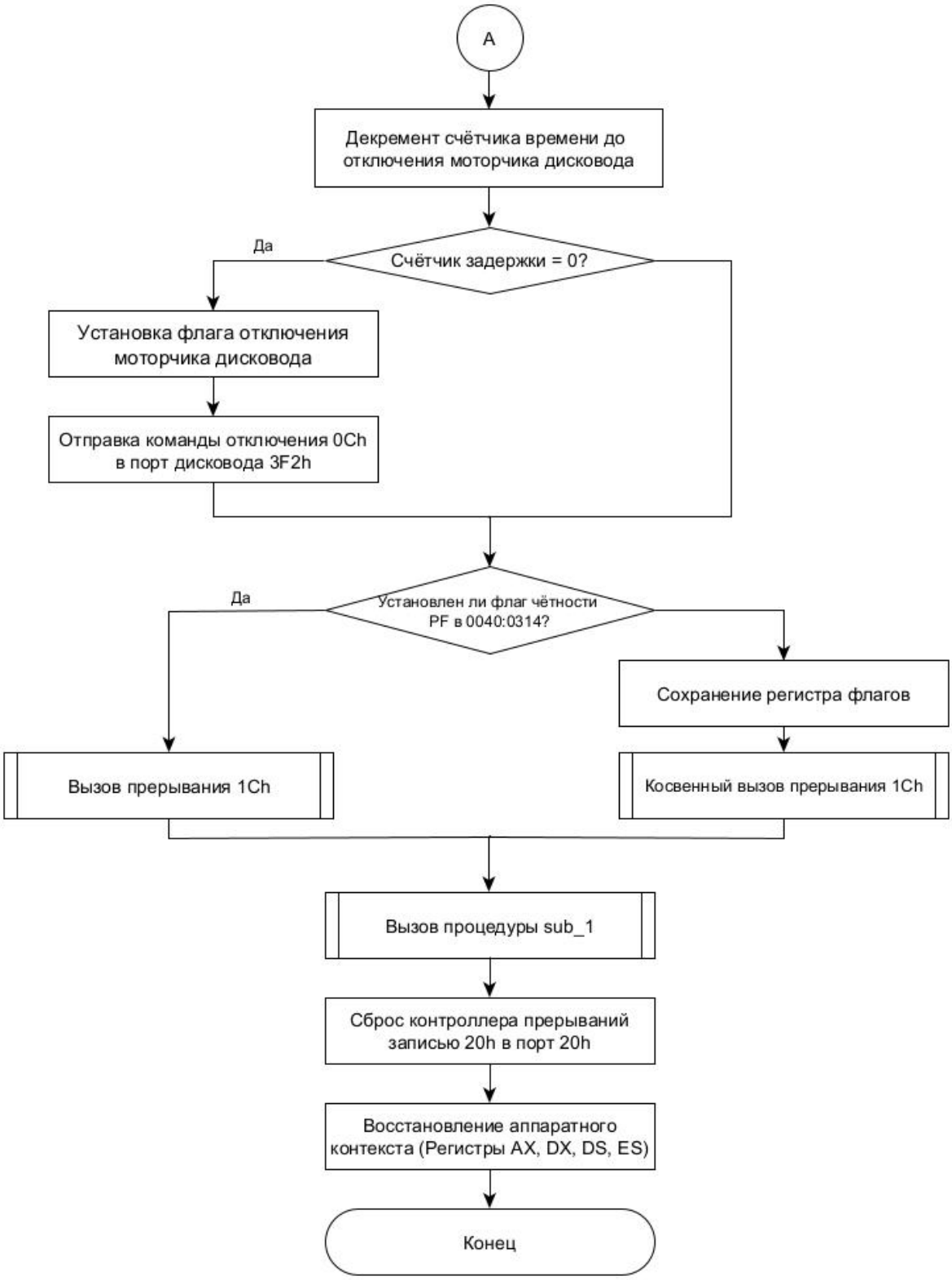
```
;; Вызов процедуры sub_1
020A:0746  E8 0070       ;*      call    sub_1           ;*(07B9)s

;; Сохранение аппаратного контекста
020A:0749  06                    push    es
020A:074A  1E                    push    ds
020A:074B  50                    push    ax
020A:074C  52                    push    dx

;; Установка 40h в DS, 0 в ES
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 двух младших байтов счётчика реального времени по адресу
   0040:006C
020A:0756  FF 06 006C            inc word ptr ds:[6Ch]   ; (0040:006C=1F3Ah)
020A:075A  75 04                 jnz loc_1           ; Jump if not zero

;; Инкремент двух старших байтов счётчика реального времени по адресу
   0040:006E
;; Если на счётчике 0, значит, прошёл час
020A:075C  FF 06 006E            inc word ptr ds:[6Eh]   ; (0040:006E=15h)

;; Сброс счётчика реального времени при наступлении новых суток
020A:0760           loc_1:
;; 18h = 24часа. Сравниваем, прошли ли сутки
020A:0760  83 3E 006E 18         cmp word ptr ds:[6Eh],18h   ; (0040:006E=15h
   )
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=1
   F3Ah)
020A:076D  75 0D                 jne loc_2           ; Jump if not equal

;; Обнуление счётчика реального времени (если прошёл один день)
020A:076F  A3 006E               mov word ptr ds:[6Eh],ax    ; (0040:006E=15h
   )
020A:0772  A3 006C               mov word ptr ds:[6Ch],ax    ; (0040:006C=1
   F3Ah)

;; Установка флага прошедших суток по адресу 0040:0070
020A:0775  C6 06 0070 01         mov byte ptr ds:[70h],1 ; (0040:0070=0)

;; Заносим в Al 8 = 00001000
```

```
020A:077A  0C 08                or  al,8

;; Декремент счётчика времени до отключения моторчика дисковода по известном
   у адресу в области данных BIOS
020A:077C           loc_2:
020A:077C  50                   push    ax
020A:077D  FE 0E 0040           dec byte ptr ds:[40h]   ; (0040:0040=0BBh)
020A:0781  75 0B                jnz loc_3           ; Jump if not zero

;; Установка флага отключения моторчика дисковода
020A:0783  80 26 003F F0        and byte ptr ds:[3Fh],0F0h  ; (0040:003F=0)


;; порт 3F2h - адрес порта цифрового управления
;; Отправляем команду 0Ch (00001100)
;; Отправление команды отключения моторчика 0Ch в порт дисковода 3F2h
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 - установлен ли флаг PF?
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
;; Косвенный вызов прерывания 1Ch (1Ch * 4 = 70h)
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=6
   ADh)
020A:07A0  EB 03                jmp short loc_5     ; (07A5)
020A:07A2  90                   nop
;; Вызов прерывания 1Ch
020A:07A3           loc_4:
020A:07A3  CD 1C                int 1Ch         ; Timer break (call each 18
   .2ms)

;; Вызов процедуры sub_1
020A:07A5           loc_5:
020A:07A5  E8 0011              call    sub_1           ;*(07B9)

;; Сброс контроллера прерываний записью 20h в порт 20h
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

;; Восстановление аппаратного контекста
020A:07AC  5A                   pop dx
020A:07AD  58                   pop ax
020A:07AE  1F                   pop ds
020A:07AF  07                   pop es

;; Завершение прерывания
020A:07B0  E9 FE99              jmp $-164h
***
20A:064C  1E                    push    ds
020A:064D  50                   push    ax
020A:064E  B8 0040              mov ax,40h
020A:0651  8E D8                mov ds,ax
020A:0653  F7 06 0314 2400      test    word ptr ds:[314h],2400h    ;
   (0040:0314=3200h)
020A:0659  75 4F                jnz loc_8           ; Jump if not zero
***
020A:06AA           loc_8:
020A:06AA  58                   pop ax
020A:06AB  1F                   pop ds
020A:06AC  CF                   iret
```

Схема алгоритма обработчика INT 8h

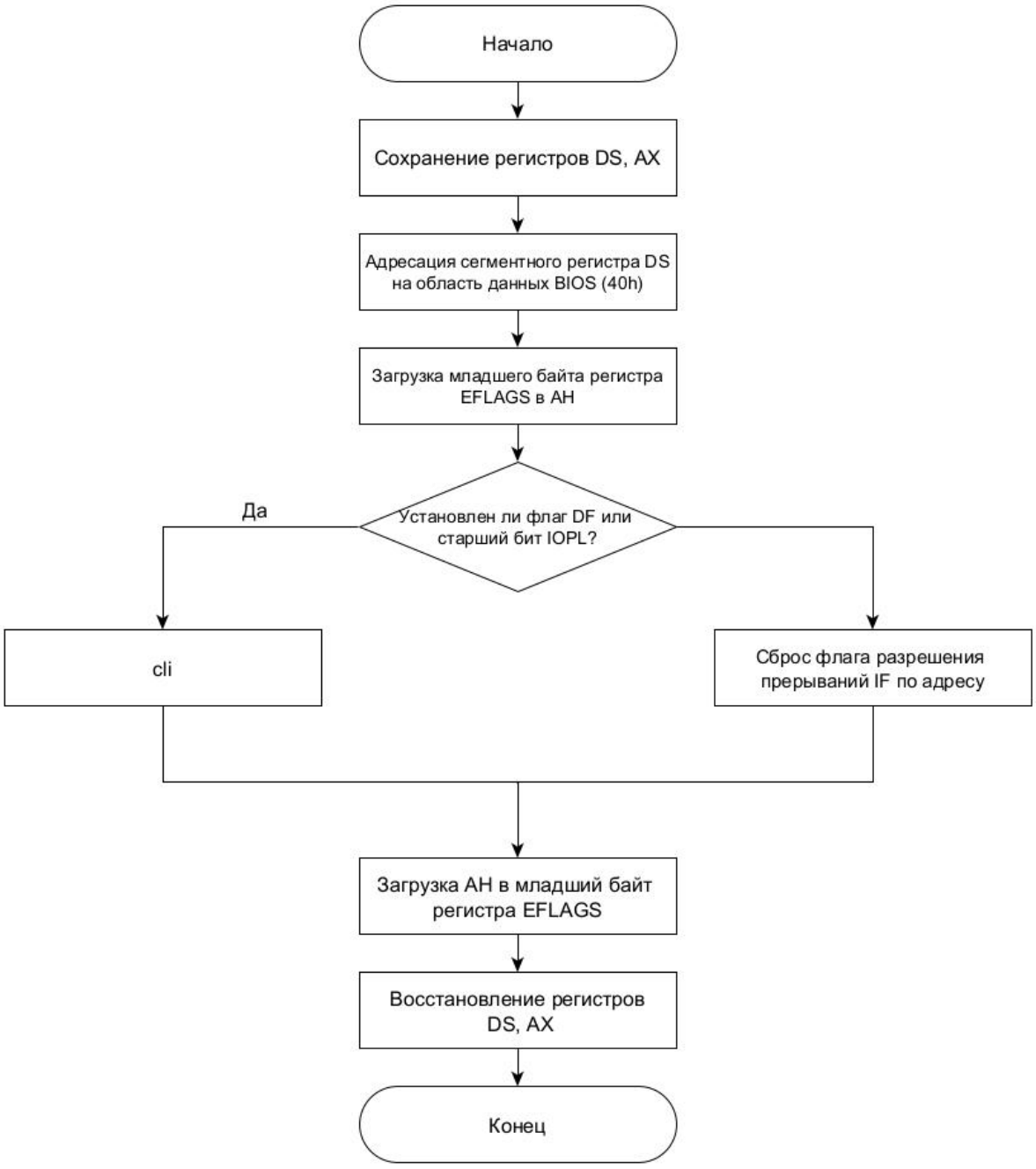

Начало
Вызов процедуры sub_1
Сохранение аппаратного контекста
Регистры (AX, DX, DS, ES)
Инициализация адреса 40h
в DS (область данных BIOS)
Инициализация адреса 0h
в ES (адресация к IVT)
Инкремент младшей части
счётчика реального времени
Младшая часть счётчика PB = 0?
Да
Инкремент старей части
счётчика реального времени
Старшая часть счётчика
PB = 24?
Да
Младшая часть счётчика
PB = 176?
Да
Обнуление старшей части
счётчика реального времени
Обнуление младшей части
счётчика реального времени
Установка флага прошедших суток
по адресу 0040:0070
A


A
Декремент счётчика времени до отключения моторчика дисковода
Да
Счётчик задержки = 0?
Установка флага отключения моторчика дисковода
Отправка команды отключения 0Ch в порт дисковода 3F2h
Да
Установлен ли флаг чётности PF в 0040:0314?
Сохранение регистра флагов
Вызов прерывания 1Ch
Косвенный вызов прерывания 1Ch
Вызов процедуры sub_1
Сброс контроллера прерываний записью 20h в порт 20h
Восстановление аппаратного контекста (Регистры AX, DX, DS, ES)
Конец

Листинг процедуры subroutine

```
sub_1       proc    near

;; Сохранение регистров DS, AX
020A:07B9  1E                   push    ds
020A:07BA  50                   push    ax

;; Установка сегментного регистра DS на область данных BIOS
020A:07BB  B8 0040              mov ax,40h
020A:07BE  8E D8                mov ds,ax

;;Загрузка EFLAGS в AH
020A:07C0  9F                   lahf                    ; Load ah from flags

;; Проверяем, поднят ли 10 или 13 флаг? (10 флаг - DF, 13 флаг - IOPL)
020A:07C1  F7 06 0314 2400      test    word ptr ds:[314h],2400h        ;
   (0040:0314=3200h)
020A:07C7  75 0C                jnz loc_7               ; Jump if not zero

;; На время выполнения команды будет заблокирована шина данных
;; Префикс lock делает команду неделимой
;; and - обращается два раза к памяти. Сначала для чтения значения, потом дл
   я записи
;; Чтобы ничего не изменилось, мы блокируем через lock

020A:07C9  F0> 81 26 0314 FDFF  lock    and word ptr ds:[314h],0FDFFh   ;
   (0040:0314=3200h)

;;
020A:07D0           loc_6:
;; Установка флаг SF, ZF, AF, PF и CF
020A:07D0  9E                   sahf                    ; Store ah into flags
020A:07D1  58                   pop ax
020A:07D2  1F                   pop ds
020A:07D3  EB 03                jmp short loc_8         ; (07D8)
020A:07D5           loc_7:

;; Сброс флага прерываний, то есть маскируемые прерывания запрещаются
020A:07D5  FA                   cli                     ; Disable interrupts
020A:07D6  EB F8                jmp short loc_6         ; (07D0)
020A:07D8           loc_8:
020A:07D8  C3                   retn
sub_1       endp
```

Схема алгоритма процедуры subroutine



Функции прерывания int 8h

- Инкремент счётчика реального времени по известному адресу в области данных BIOS
- Вызов пользовательского прерывания 1Ch.
- Декремент счётчика времени до отключения моторчика дисковода. Посылка команды в порт на отключение дисковода по истечении двух секунд.

Вывод

В ходе работы были вычислены адреса в памяти и дизассемблированы коды обработчика прерывания int 8h и подпрограммы subroutine, которая вызывается из кода обработчика. Были построены схемы алгоритмов обработчика прерывания int8h и подпрограммы subroutine.