|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № 1**

|  |  |
| --- | --- |
| **Дисциплина**  Операционные системы  **Тема**  Дизассемблирование INT 8h  **Студент**  **Группа**  **Оценка (баллы) \_\_\_\_\_\_\_\_**  **Преподаватель** Рязанова Н.Ю |  |

Москва.

2020 г.

**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 (32-разрядный режим работы), который эмулирует реальный режим работы вычислительной системы на базе процессоров Intel.

**Задание:**

Используя sourser **( sr.exe**) получить дизассемблированный 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 комментариями;
* Графический алгоритм работы обработчика прерывания Int 8h, структурированный и выполненный в соответствии с ГОСТ19.701-90 ЕСПД – «Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения».

**Схема алгоритма**

Sourcer v5.10 5-Sep-20 3:13 pm Page 1

**; Вход в прерывание, вызов sub\_1**

020A:0746 E8 0070 ;\* call sub\_2 ; (07B9)

020A:0746 E8 70 00 db 0E8h, 70h, 00h

**; Сохранение регистров ES, DS, AX, DX**

020A:0749 06 push es

020A:074A 1E push ds

020A:074B 50 push ax

020A:074C 52 push dx

**; Установка регистров DS, ES и обнуление регистра AX**

020A:074D B8 0040 mov ax,40h ; Область памяти

020A:0750 8E D8 mov ds,ax; ds = 40h

020A:0752 33 C0 xor ax,ax ; Zero register

020A:0754 8EC0 mov es,ax; es = 0h

;**Инкремент счетчика времени (0040:006C)**

020A:0756 FF 06 006C inc word ptr ds:[6Ch] ;(0040:006C=7982h)

020A:075A 75 04 jnz loc\_1 ; Jump if not zero

**;Если младшая часть переполнилась, инкрементировать старшую часть счётчика**

020A:075CFF 06 006E inc word ptr ds:[6Eh] ; (0040:006E=0Ch)

**; Проверка наступления новых суток(прошло 24 часа)**

020A:0760 loc\_1: ; сейчас полночь (1800B0h)?

020A:0760 83 3E 006E 18 cmp word ptr ds:[6Eh],18h ; (0040:006E)

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)

020A:076D 75 0D jne loc\_2 ; Jump if not equal

**;Обнуление счетчика времени, если наступили новые сутки**

020A:076F A3 006E mov word ptr ds:[6Eh],ax; (0040:006E)

020A:0772 A3 006C mov word ptr ds:[6Ch],ax; (0040:006C)

**;Установка единицы в ячейку 0040:0070h** **(флаг переполнения)**

020A:0775 C6 06 0070 01 mov byte ptr ds:[70h],1; (0040:0070)

020A:077A 0C 08 or al,8 ; al = 8

**; Декремент счетчика до отключения моторчика дисковода (0040h:0040h)**

020A:077C loc\_2:

020A:077C 50 push ax

020A:077D FE 0E 0040 dec byte ptr ds:[40h]; (0040:0040)

020A:0781 75 0B jnz loc\_3 ; Jump if not zero

**;Установка флага отключения дисковода**

020A:0783 80 26 003FF0 and byte ptr ds:[3Fh],0F0h ; (0040:003F)

**;Посылка сигнала на отключение моторчика дисковода**

020A:0788 B0 0C mov al,0Ch

020A:078ABA 03F2 mov dx,3F2h

020A:078D EE out dx,al ; port 3F2h, dsk0 contrl output

**; Проверка возможности 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)

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 ; поменять местами ah, al

020A:079A 50 push ax

020A:079B 26: FF 1E 0070 call dword ptr es:[70h]

020A:07A0 EB 03 jmp short loc\_5 ; (07A5)

020A:07A2 90 nop

**; Вызов пользовательского прерывания 1Ch**

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\_2 ; (07B9)

**;Посылка сигнала "endofinterrupt" контроллеру 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

**; Восстановление значений регистров DX, AX, DS, ES**

020A:07AC 5A pop dx

020A:07AD 58 pop ax

020A:07AE 1F pop ds

020A:07AF 07 pop es

**; Переход**

020А:07B0 E9 FE99 jmp $-164h ; 020А:064C

020А:064C loc\_5:

020А:064C 1E push ds

020А:064D 50 push ax

…

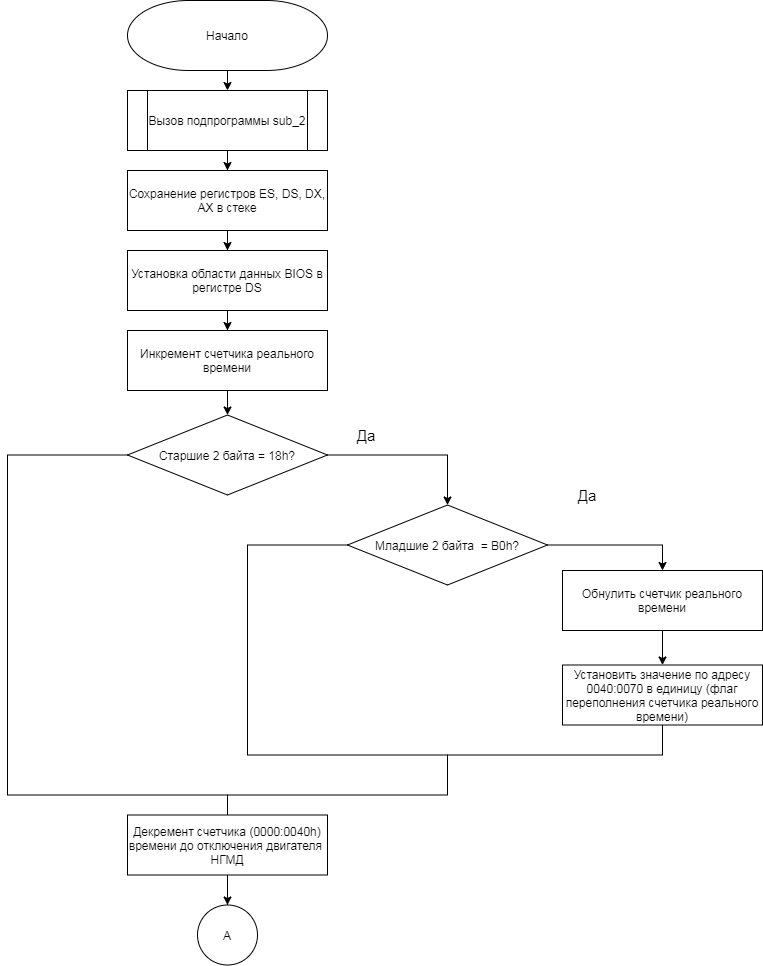
020А:06AA loc\_13:

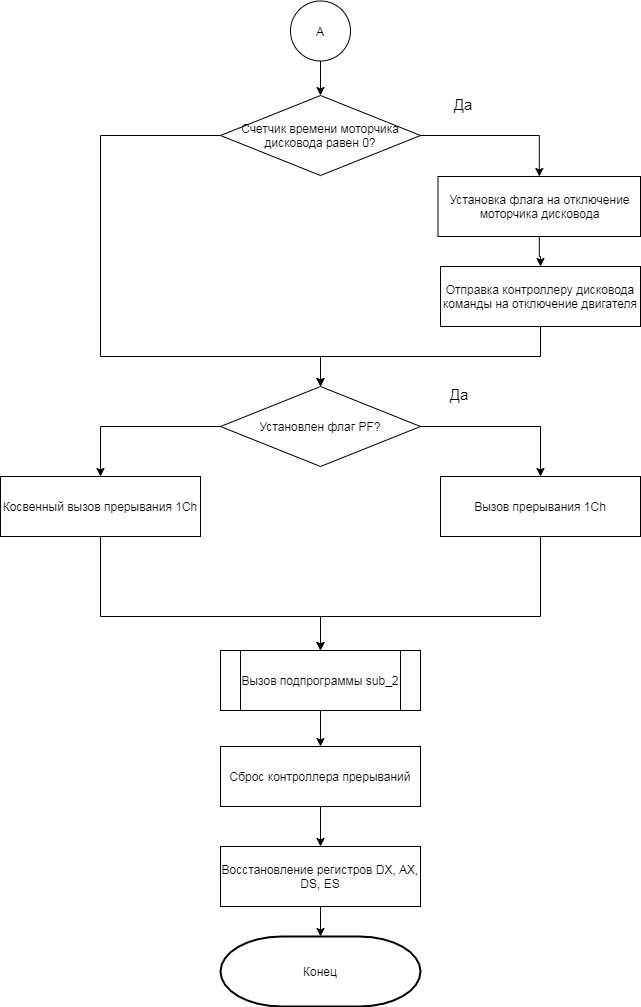
020А:06AA 58 pop ax

020А:06AB 1F pop ds

**; Возврат из обработчика прерываний**

020А:06AC CF iret





**Схема алгоритма sub\_2**

sub\_2 proc near

**; Сохранение регистров DS, AX**

020A:07B9 1E push ds

020A:07BA 50 push ax

**; Загрузка сегмента данных, загрузка ah**

020A:07BB B8 0040 mov ax,40h

020A:07BE 8E D8 mov ds, ax; ds = 40h

020A:07C0 9F lahf ; Загрузить флаги состояния в АН

**; Проверка равен ли старший бит IOPL единице**

020A:07C1 F7 06 0314 2400 test word ptr ds:[314h],2400h; (0040:0314)

020A:07C7 75 0C jnz loc\_7 ; Jump if not zero

; lock - Префикс блокировки шины данных

; 0FDFFh = 1111 1101 1111 1111b

**; Сброс флага разрешения прерывания (сброс IF = 0)**

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

020A:07D0 loc\_6:

020A:07D0 9E sahf ; Загрузить флаги состояния из АН

**; Восстановление регистров**

020A:07D1 58 pop ax

020A:07D2 1F pop ds

020A:07D3 EB 03 jmp short loc\_8 ; (07D8)

**; Запрет маскируемых прерываний командой cli**

020A:07D5 loc\_7:

020A:07D5 FA cli ; Disable interrupts

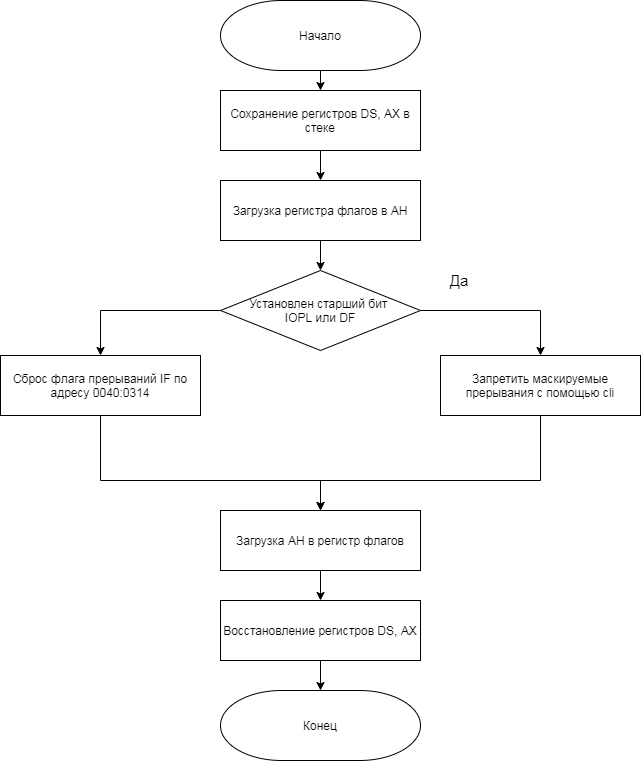
020A:07D6 EB F8 jmp short loc\_6 ; (07D0)

**; Возврат из процедуры**

020A:07D8 loc\_8:

020A:07D8 C3 retn

sub\_2 endp



Функции обработчика INT 8h:

1. Инкремент счётчика реального времени и контроль переполнения счётчика (наступление новых суток);
2. Декремент счётчика времени, оставшегося до выключения моторчика дисковода и при достижении нуля, запись в порт контроллера по шине данных команды на отключение моторчика дисковода;
3. Вызов пользовательского прерывания int 1Ch;