|  |  |
| --- | --- |
| 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 (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

**Отчет по лабораторной работе №1 часть 1**

**по курсу «Операционные системы»**

**по теме «Дизассемблирование INT 8h»**

Студент: Завойских Е. В.

Группа: ИУ7-53Б

Преподаватель: Рязанова Н. Ю.

*2022 г.*

**Листинг прерывания int 8h:**

Temp.lst Sourcer v5.10 13-Sep-22 3:53 pm Page 1

**; Вызов подпрограммы sub\_2**

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

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

**; Занесение значений регистров es, ds, ax, dx в стек**

020A:0749 06 push es

020A:074A 1E push ds

020A:074B 50 push ax

020A:074C 52 push dx

**; Занесение в сегментный регистр ds адреса области данных BIOS, в es – адреса таблицы векторов прерываний**

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 счетчика тиков. Если эта часть переполнилась — инкремент два старших байтов счетчика**

020A:0756 FF 06 006C inc word ptr ds:[6Ch] ; (0040:006C=0E550h)

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

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

020A:0760 loc\_1:

**; Проверка, что старшее слово счетчика тиков равна 24**

020A:0760 83 3E 006E 18 cmp word ptr ds:[6Eh],18h ; (0040:006E=0Fh)

020A:0765 75 15 jne loc\_2 ; Jump if not equal

**; Проверка, что младшее слово счетчика тиков равна 176**

020A:0767 81 3E 006C 00B0 cmp word ptr ds:[6Ch],0B0h ; (0040:006C=0E550h)

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

**; Зануление счетчика тиков**

020A:076F A3 006E mov word ptr ds:[6Eh],ax ; (0040:006E=0Fh)

020A:0772 A3 006C mov word ptr ds:[6Ch],ax ; (0040:006C=0E550h)

**; По адресу 0040:0070h заносится флаг окончания суток**

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

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

020A:077C loc\_2:

**; Сохранение значения ax**

020A:077C 50 push ax

**; По адресу 0040:0040h хранится время, оставшееся до выключения двигателя НГМД. Это время уменьшается обработчиком прерывания**

020A:077D FE 0E 0040 dec byte ptr ds:[40h] ; (0040:0040=3Bh)

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

**; Время до отключения моторчика равно 0, обработчик прерывания передает контроллеру дисковода команды 0Ch (0000 1100) в порт 3F2h на остановку моторчика:**

**2 бит равен 1 – разрешение работы контроллера**

**3 бит равен 1 – разрешение прерываний от контроллера**

**4-7 биты равны 0 – выключение моторчика**

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

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

020A:078E loc\_3:

**; Восстановление значения ax**

020A:078E 58 pop ax

**; По адресу 0040:0314h содержится копия регистра флагов. Проверяется, установлен ли флаг PF**

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

**; Младший байт FLAGS копируется в ah**

020A:0797 9F lahf ; Load ah from flags

020A:0798 86 E0 xchg ah,al

020A:079A 50 push ax

**; Косвенный вызов прерывания 1Ch**

020A:079B 26: FF 1E 0070 call dword ptr es:[70h] ; (0000:0070=6ADh)

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

020A:07A2 90 nop

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)

**; Сброс контроллера 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 регистров dx, ax, ds, es из стека**

020A:07AC 5A pop dx

020A:07AD 58 pop ax

020A:07AE 1F pop ds

020A:07AF 07 pop es

**; Выход из обработчика**

**; Переход по адресу 020A:064Ch, iret расположен по адресу 020A:06ACh**

020A:07B0 E9 FE99 jmp $-164h

Temp.lst Sourcer v5.10 18-Sep-22 5:55 pm Page 1

020A:064C loc\_1:

020A: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\_9 ; Jump if not zero

020A:065B 55 push bp

020A:065C 8B EC mov bp,sp

020A:065E 8B 46 0A mov ax,[bp+0Ah]

020A:0661 5D pop bp

020A:0662 A9 0100 test ax,100h

020A:0665 75 43 jnz loc\_9 ; Jump if not zero

020A:0667 A9 0200 test ax,200h

020A:066A 74 22 jz loc\_5 ; Jump if zero

020A:066C F0> 81 0E 0314 0200 lock or word ptr ds:[314h],200h ; (0040:0314=3200h)

020A:0673 F7 06 0314 0003 test word ptr ds:[314h],3 ; (0040:0314=3200h)

020A:0679 75 2F jnz loc\_9 ; Jump if not zero

020A:067B loc\_2:

020A:067B 86 E0 xchg ah,al

020A:067D FC cld ; Clear direction

020A:067E A8 04 test al,4

020A:0680 75 25 jnz loc\_8 ; Jump if not zero

020A:0682 loc\_3:

020A:0682 A8 08 test al,8

020A:0684 75 11 jnz loc\_6 ; Jump if not zero

020A:0686 70 19 jo loc\_7 ; Jump if overflow=1

020A:0688 loc\_4:

020A:0688 9E sahf ; Store ah into flags

020A:0689 58 pop ax

020A:068A 1F pop ds

020A:068B CA 0002 retf 2 ; Return far

020A:068E loc\_5:

020A:068E F0> 81 26 0314 FDFF lock and word ptr ds:[314h],0FDFFh ; (020A:0314=3231h)

020A:0695 EB E4 jmp short loc\_2 ; (067B)

020A:0697 loc\_6:

020A:0697 70 EF jo loc\_4 ; Jump if overflow=1

020A:0699 50 push ax

020A:069A B0 7F mov al,7Fh

020A:069C 04 02 add al,2

020A:069E 58 pop ax

020A:069F EB E7 jmp short loc\_4 ; (0688)

020A:06A1 loc\_7:

020A:06A1 50 push ax

020A:06A2 32 C0 xor al,al ; Zero register

020A:06A4 58 pop ax

020A:06A5 EB E1 jmp short loc\_4 ; (0688)

020A:06A7 loc\_8:

020A:06A7 FD std ; Set direction flag

020A:06A8 EB D8 jmp short loc\_3 ; (0682)

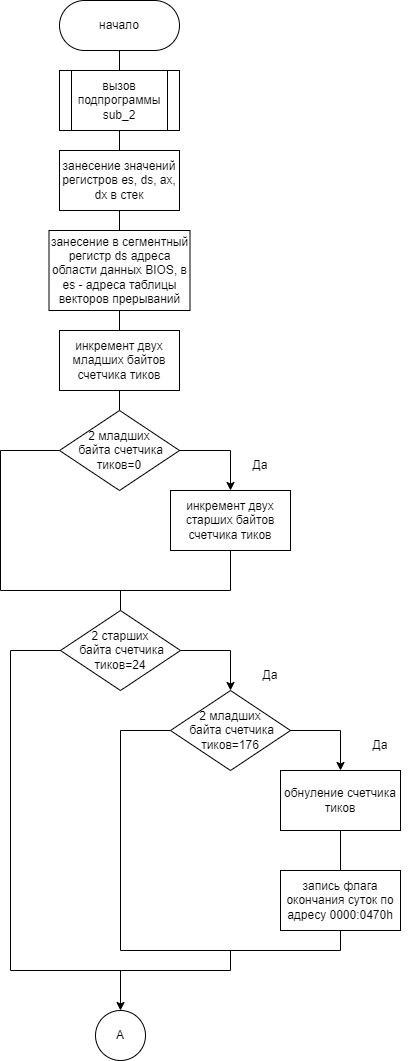
020A:06AA loc\_9:

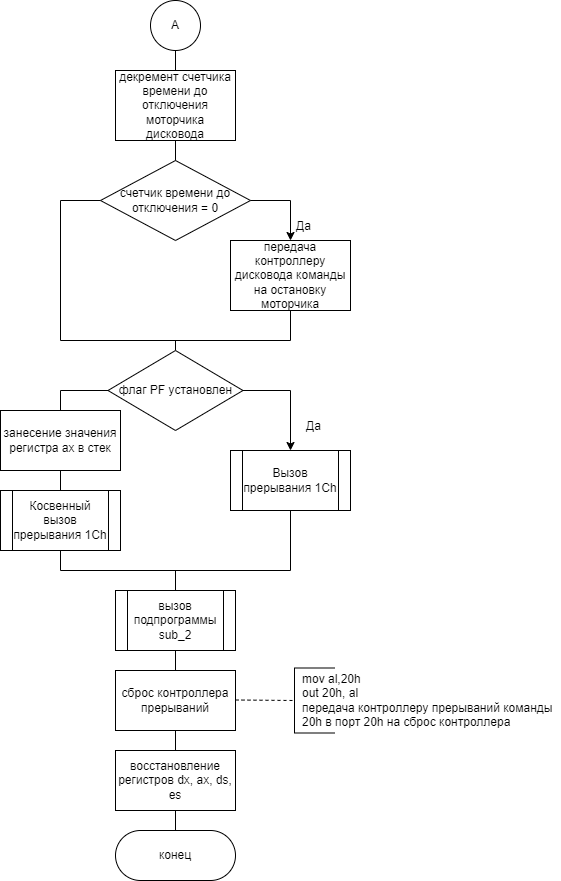
020A:06AA 58 pop ax

020A:06AB 1F pop ds

020A:06AC CF **iret** ; Interrupt return

**Схема алгоритма прерывания int 8h:**





**Листинг подпрограммы sub\_2:**

sub\_2 proc near

**; Занесение значений регистров ds, ax в стек**

020A:07B9 1E push ds

020A:07BA 50 push ax

**; Занесение в сегментный регистр ds адреса области данных BIOS**

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

020A:07BE 8E D8 mov ds,ax

**; Младший байт регистра флагов копируется в ah**

020A:07C0 9F lahf ; Load ah from flags

**; Проверка DF и старшего бита IOPL регистра флагов**

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

**; Сброс флага IF (блокировка локальной шины памяти)**

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

020A:07D0 loc\_6:

**; Запись ah в младший байт регистра флагов**

020A:07D0 9E sahf ; Store ah into flags

**; Восстановление регистров ax, ds из стека**

020A:07D1 58 pop ax

020A:07D2 1F pop ds

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

**; cli**

020A:07D5 loc\_7:

020A:07D5 FA cli ; Disable interrupts

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

020A:07D8 loc\_8:

020A:07D8 C3 retn

sub\_2 endp

**Схема алгоритма подпрограммы sub\_2:**

