

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №5**

**по дисциплине «Организация ЭВМ и систем» Тема:**

**Написание собственного прерывания.**

Студентка гр. 0382

Здобнова К.Д.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Написать программу с использованием собственного прерывания.

### **Задание.**

1е

1 - 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек;

Е - Выполнить чтение и вывод на экран отсчета часов реального времени из памяти CMOS (в формате BCD).

### **Выполнение работы.**

Программа обработки прерывания - это отдельная процедура, имеющая структуру:

```
SUBR_INT PROC FAR
```

```
    PUSH AX    ; сохранение изменяемых регистров
```

```
    <действия по обработке прерывания>
```

```
    POP AX     ; восстановление регистров
```

```
    MOV AL, 20H
```

```
    OUT 20H,AL
```

```
    IRET
```

```
SUBR_INT ENDP
```

Две последние строки перед IRET необходимы для разрешения обработки прерываний с более низкими уровнями, чем только что обработанное.

Программа, использующая новые программы обработки прерываний, при своем завершении должна восстанавливать оригинальные векторы прерываний. Функция 35 прерывания 21H возвращает текущее значение вектора прерывания, помещая значение сегмента в ES, а смещение в BX. В этом случае программа

должна содержать следующие инструкции:

; -- в сегменте данных

KEEP\_CS DW 0 ; для хранения сегмента

KEEP\_IP DW 0 ; и смещения прерывания

; -- в начале программы

MOV AH, 35H ; функция получения вектора

MOV AL, 1CH ; номер вектора

INT 21H

MOV KEEP\_IP, BX ; запоминание смещения

MOV KEEP\_CS, ES ; и сегмента

Для задания адреса собственного прерывания с заданным номером в таблицу векторов прерываний используется функция 25H прерывания 21H, которая устанавливает вектор прерывания на указанный адрес нового обработчика.

PUSH DS

MOV DX, OFFSET ROUT ; смещение для процедуры в DX

MOV AX, SEG ROUT ; сегмент процедуры

MOV DS, AX ; помещаем в DS

MOV AH, 25H ; функция установки вектора

MOV AL, 60H ; номер вектора

INT 21H ; меняем прерывание

POP DS

В конце программы восстанавливается старый вектор прерывания

CLI

PUSH DS

MOV DX, KEEP\_IP

MOV AX, KEEP\_CS



## **Выводы.**

Реализована программа с использованием собственного прерывания.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: *lab5.asm*

```
AStack  SEGMENT STACK
        DB 1024 DUP(?)
AStack  ENDS

DATA     SEGMENT
        KEEP_CS DW 0      ;для хранения сегмента вектора прерывания
        KEEP_IP DW 0      ;для хранения смещения вектора прерывания
DATA     ENDS

CODE     SEGMENT
        ASSUME CS:CODE, DS:DATA, SS:AStack
.186
SUBR_INT PROC FAR

        jmp H_start

        SAVE_SS DW 0
        SAVE_SP DW 0
        SAVE_AX DW 0
        MY_STACK DW 1024 DUP(?)

H_start:
        mov SAVE_SP, sp
        mov SAVE_AX, ax
        mov SAVE_SS, ss
        mov sp, OFFSET H_start
        mov ax, SEG MY_STACK
        mov ss, ax
        mov ax, SAVE_AX
        ;сохранение изменяемых регистров
        push ax
        push ds
```

```

mov ax, SEG SUBR_INT
mov ds, ax
mov ax, SAVE_AX
mov ah,29h ;заносим функцию вывода строки

mov al,0Bh ; CMOS 0Bh - управляющий регистр B
out 70h,al ; порт 70h - индекс CMOS
in al,71h ; порт 71h - данные CMOS
and al,11111011b ; обнулить бит 2 (форма чисел - BCD)
out 71h,al ; и записать обратно
mov al,4 ; CMOS 04h - час
call Print_cmos
mov al,':' ;двоеточие
int 29h
mov al,2 ;CMOS 02h - минута
call Print_cmos
mov al,':' ;двоеточие
int 29h
mov al,0 ;CMOS 00h - секунда
call Print_cmos
mov al,' ' ;пробел
int 29h

;восстановление регистров
pop ds
pop ax

mov sp, SAVE_SP
mov ax, SAVE_SS
mov ss, ax
mov ax, SAVE_AX
mov al,20h ;разрешение обработки прерываний
out 20h,al ;с более низкими уровнями
iret ;выход из прерывания
SUBR_INT ENDP

```

```

Print_cmos proc near
    out 70h,al                ; послать AL в индексный порт CMOS
    in al,71h                ; прочитать данные
    push ax
    shr al,4                 ; выделить старшие четыре бита
    add al,'0'               ; добавить ASCII-код цифры 0
    int 29h                 ; вывести на экран
    pop ax
    and al,0Fh              ; выделить младшие четыре бита
    add al,30h              ; добавить ASCII-код цифры 0
    int 29h                 ; вывести на экран
    ret
Print_cmos endp

```

MAIN PROC FAR

```

    mov ax, DATA
    mov ds, ax
    ;сохраняем вектор прерывания
    mov ah, 35h             ;функция получения вектора
    mov al, 08h             ;номер вектора
    int 21h
    mov KEEP_CS, es
    mov KEEP_IP, bx

    ;устанавливаем новый вектор прерывания
    push ds
    mov dx, OFFSET SUBR_INT
    mov ax, SEG SUBR_INT
    mov ds, ax
    mov ah, 25h             ;функция установки вектора
    mov al, 08h             ;номер вектора
    int 21h
    pop ds

```



```

B_loop:
mov ah, 00h
int 16h
cmp al, 27
je Restoring_the_vector
loop B_loop

```

```

Restoring_the_vector: ;восстанавливаем старый вектор прерывания
cli
push ds
mov dx, KEEP_IP
mov ax, KEEP_CS
mov ds, ax
mov ah, 25h ;функция установки вектора
mov al, 08h ;номер вектора
int 21h
pop ds
sti

mov ah, 4ch ;завершение программы
int 21h

```

```

MAIN ENDP
CODE ENDS
END MAIN

```

**Название файла: lab5.LST**

Microsoft (R) Macro Assembler Version 5.10  
00:40:11

12/6/21

Page 1-1

```

0000          AStack  SEGMENT STACK
0000  0400[          DB 1024 DUP(?)
          ??
          ]

0400          AStack  ENDS

```

```

0000          DATA      SEGMENT
0000 0000          KEEP_CS DW 0      ;для хранения с
          егмента вектора прерываний
          ,я
0002 0000          KEEP_IP DW 0      ;для хранения с
          мещения вектора прерываний
          ,я
0004          DATA      ENDS

0000          CODE      SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:AStack
          .186
0000          SUBR_INT PROC FAR

0000 E9 0809 R          jmp H_start

0003 0000          SAVE_SS DW 0
0005 0000          SAVE_SP DW 0
0007 0000          SAVE_AX DW 0
0009 0400[          MY_STACK DW 1024 DUP(?)
          ?????
          ]

0809          H_start:
0809 2E: 89 26 0005 R          mov SAVE_SP, sp
080E 2E: A3 0007 R          mov SAVE_AX, ax
0812 2E: 8C 16 0003 R          mov SAVE_SS, ss
0817 BC 0809 R          mov sp, OFFSET H_start
081A B8 ---- R          mov ax, SEG MY_STACK
081D 8E D0          mov ss, ax
081F 2E: A1 0007 R          mov ax, SAVE_AX
          ;сохранение изменяемых
          регистров
0823 50          push ax
0824 1E          push ds

0825 B8 ---- R          mov ax, SEG SUBR_INT
0828 8E D8          mov ds, ax
082A 2E: A1 0007 R          mov ax, SAVE_AX
082E B4 29          mov ah,29h ;заносим функции
          ю вывода строки

0830 B0 0B          mov al,0Bh          ; CMOS 0Bh - 9

```

```

                                □правляющий регистр В
0832  E6 70                                out 70h,al                ; порт 70h - И
                                ,ндекс CMOS
Microsoft (R) Macro Assembler Version 5.10                12/6/21
00:40:11
                                Page                1-2

0834  E4 71                                in al,71h                ; порт 71h - д
                                анные CMOS
0836  24 FB                                and al,11111011b        ; обнулить
                                бит 2 (форма чисел - BCD)
0838  E6 71                                out 71h,al                ; и записаэ
                                □ь обратно
083A  B0 04                                mov al,4                ; CMOS 04h - чИ
                                °с
083C  E8 086B R                            call Print_cmos
083F  B0 3A                                mov al,':'                ;двоеточиИ
                                μ
0841  CD 29                                int 29h
0843  B0 02                                mov al,2                ;CMOS 02h - мин
                                ута
0845  E8 086B R                            call Print_cmos
0848  B0 3A                                mov al,':'                ;двоеточиИ
                                μ
084A  CD 29                                int 29h
084C  B0 00                                mov al,0                ;CMOS 00h - секэ
                                □нда
084E  E8 086B R                            call Print_cmos
0851  B0 20                                mov al,' '                ;пробел
0853  CD 29                                int 29h

                                ;восстановление региэ
                                □тров
0855  1F                                pop ds
0856  58                                pop ax

0857  2E: 8B 26 0005 R                            mov sp, SAVE_SP
085C  2E: A1 0003 R                            mov ax, SAVE_SS
0860  8E D0                                mov ss, ax
0862  2E: A1 0007 R                            mov ax, SAVE_AX
0866  B0 20                                mov al,20h                ;разрешение обэ
                                □аботки прерываний

```

```

0868 E6 20                                out 20h,al ;с более низким
и уровнями
086A CF                                iret ;выход из прерываний
,я
086B                                SUBR_INT ENDP

086B                                Print_cmos proc near
086B E6 70                                out 70h,al ; послать А
L в индексный порт CMOS
086D E4 71                                in al,71h ; прочитатъ
□ данные
086F 50                                push ax
0870 C0 E8 04                            shr al,4 ; выделить э
□такие четыре бита
0873 04 30                                add al,'0' ; добавить
ASCII-код цифры 0
0875 CD 29                                int 29h ; вывести И
½а экран
0877 58                                pop ax
Microsoft (R) Macro Assembler Version 5.10 12/6/21
00:40:11

```

Page 1-3

```

0878 24 0F                                and al,0Fh ; выделить
младшие четыре бита
087A 04 30                                add al,30h ; добавить
ASCII-код цифры 0
087C CD 29                                int 29h ; вывести И
½а экран
087E C3                                ret
087F                                Print_cmos endp

087F                                MAIN PROC FAR

087F B8 ---- R                            mov ax, DATA
0882 8E D8                                mov ds, ax
;сохраняем вектор преры
вания
0884 B4 35                                mov ah, 35h ;функция получиИ
ния вектора
0886 B0 08                                mov al, 08h ;номер вектора
0888 CD 21                                int 21h

```

```

088A 8C 06 0000 R      mov KEEP_CS, es
088E 89 1E 0002 R      mov KEEP_IP, bx

                        ;устанавливаем новый
                        вектор прерывания
0892 1E                push ds
0893 BA 0000 R          mov dx, OFFSET SUBR_INT
0896 B8 ---- R          mov ax, SEG SUBR_INT
0899 8E D8              mov ds, ax
089B B4 25              mov ah, 25h ;функция установИ
                        И вектора
089D B0 08              mov al, 08h ;номер вектора
089F CD 21              int 21h
08A1 1F                pop ds

08A2                    B_loop:
08A2 B4 00              mov ah, 00h
08A4 CD 16              int 16h
08A6 3C 1B              cmp al, 27
08A8 74 02              je Restoring_the_vector
08AA E2 F6              loop B_loop

08AC                    Restoring_the_vector: ;восстановИ
                        Иаем старый вектор прерэ
                        Ивания
08AC FA                cli
08AD 1E                push ds
08AE 8B 16 0002 R        mov dx, KEEP_IP
08B2 A1 0000 R          mov ax, KEEP_CS
08B5 8E D8              mov ds, ax
08B7 B4 25              mov ah, 25h ;функция установИ
                        И вектора
08B9 B0 08              mov al, 08h ;номер вектора
08BB CD 21              int 21h
08BD 1F                pop ds
Microsoft (R) Macro Assembler Version 5.10      12/6/21
00:40:11
                        Page      1-4

08BE FB                sti

08BF B4 4C              mov ah, 4ch ;завершение прогэ
                        €аммы

```

08C1 CD 21 int 21h

08C3 MAIN ENDP

08C3 CODE ENDS

END MAIN

Microsoft (R) Macro Assembler Version 5.10

12/6/21

00:40:11

Symbols-1

### Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK . . . . .	0400	PARA	STACK
CODE . . . . .	08C3	PARA	NONE
DATA . . . . .	0004	PARA	NONE

### Symbols:

N a m e	Type	Value	Attr
B_LOOP . . . . .	L NEAR	08A2	CODE
H_START . . . . .	L NEAR	0809	CODE
KEEP_CS . . . . .	L WORD	0000	DATA
KEEP_IP . . . . .	L WORD	0002	DATA
MAIN . . . . .	F PROC	087F	CODE Length = 0044
MY_STACK . . . . .	L WORD	0009	CODE Length = 0400
PRINT_CMOS . . . . .	N PROC	086B	CODE Length = 0014
RESTORING_THE_VECTOR . . . . .	L NEAR	08AC	CODE
SAVE_AX . . . . .	L WORD	0007	CODE
SAVE_SP . . . . .	L WORD	0005	CODE
SAVE_SS . . . . .	L WORD	0003	CODE
SUBR_INT . . . . .	F PROC	0000	CODE Length = 086B
@CPU . . . . .	TEXT	259	
@FILENAME . . . . .	TEXT	lab5	
@VERSION . . . . .	TEXT	510	

129 Source Lines

129 Total Lines

20 Symbols

48018 + 457192 Bytes symbol space free

0 Warning Errors

0 Severe Errors