

**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. 1381		Демчук П. Д.
Преподаватель		Ефремов М. А.

Санкт-Петербург

2022

## **Цель работы.**

Написание собственного прерывания.

## **Задание.**

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.

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
SAVE_CS DW 0 ; для хранения сегмента
SAVE_IP DW 0 ; и смещения прерывания
; -- в начале программы
MOV AH, 35H ; функция получения вектора
MOV AL, 1CH ; номер вектора
INT 21H
MOV SAVE_IP, BX ; запоминание смещения
MOV SAVE_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, SAVE_IP
MOV AX, SAVE_CS
MOV DS, AX
MOV AH, 25H
MOV AL, 1CH
INT 21H ; восстанавливаем вектор
```

POP DS

STI

**Выводы.**

В ходе лабораторной работы была написана программа с собственным прерыванием на языке Ассемблер.

## ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.asm

```
ASSUME CS:CODE, DS:DATA SS:AStack
```

```
AStack    SEGMENT    STACK
```

```
            DW 512 DUP(?)
```

```
AStack    ENDS
```

```
DATA SEGMENT
```

```
    keep_cs dw 0
```

```
    keep_ip dw 0
```

```
    save_ss dw 0
```

```
    save_sp dw 0
```

```
    save_ax dw 0
```

```
DATA ENDS
```

```
CODE SEGMENT
```

```
.186
```

```
SUBR_INT  PROC  FAR
```

```
    start:
```

```
    ;сохранение регистров
```

```
    mov save_ss, ss
```

```
    mov save_sp, sp
```

```
    mov save_ax, ax
```

```
    mov sp, offset start
```

```
    mov ax, seg STACK
```

```
    mov ss, ax
```

```

mov ax, save_ax

push ax
push ds
mov ax, seg SUBR_INT
mov ds, ax
mov ax, save_ax

;обработка прерывания
mov ah, 29h
mov al, 0Bh
out 70h, al
in al, 71h
and al, 11111011b
out 71h, al
mov al, 4
call PRINT
mov al, 'h'
int 29h
mov al, ' '
int 29h
mov al, 2
call PRINT
mov al, 'm'
int 29h
mov al, ' '
int 29h
mov al, 0
call PRINT

```

```
mov al, 's'
int 29h
mov al, ' '
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
out 20h, al
iret
```

```
SUBR_INT ENDP
```

```
PRINT PROC NEAR
```

```
out 70h, al
in al, 71h
push ax
shr al, 4
add al, '0'
int 29h
pop ax
and al, 0Fh
add al, 30h
```

```

        int 29h

        ret

PRINT   ENDP

; Главная процедура
Main PROC   FAR

        push ds

        sub ax, ax

        push ax

        mov ax, DATA

        mov ds, ax

;текущий вектор прерывания

        mov ah, 35h

        mov al, 08h

        int 21h

        mov keep_ip, bx

        mov keep_cs, es

;установка нового вектора прерывания

        push ds

        mov dx, offset SUBR_INT

        mov ax, seg SUBR_INT

        mov ds, ax

        mov ah, 25h

        mov al, 08h

        int 21h

```



```

    pop ds

    int 08h

;восстановление изначального вектора прерывания
cli    ;if=0
push ds
mov dx, keep_ip
mov ax, keep_cs
mov ds, ax
mov ah, 25h
mov al, 08h
int 21h
pop ds
sti    ;if=1

mov ah, 4ch
int 21h

Main ENDP

CODE ENDS

END Main

```