

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

**отчет**  
**по лабораторной работе №4**  
**по дисциплине «Организация ЭВМ и систем»**  
**Тема: Разработка собственного прерывания.**

Студентка гр. 9382

\_\_\_\_\_

Круглова В. Д.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Научиться разрабатывать и применять прерывание.

### **Задание.**

Разработать, используя:

1Ch - прерывание от часов - генерируется автоматически операционной системой 18 раз в сек;

Печать сообщения на экране;

### **Ход работы:**

Нужно переопределить прерывание 1Ch на вывод строки. Для этого была создана процедура, в которой вызывается цикл для автоматического срабатывания прерывания. В процессе выполнения работы были использованы фрагменты кода, представленные в методических указаниях к лабораторной работе.

### **Исходный код программы.**

```
STACKSG SEGMENT PARA STACK 'Stack'
        DW      1024 DUP (?)
STACKSG ENDS

DATASG SEGMENT PARA 'Data'
        KEEP_CS DW 0 ; для хранения сегмента
        KEEP_IP DW 0 ; и смещения вектора прерывания
        GREETING DB 'Subbotin Maksim 9382 $'
        KKK DW 0
        crlf db 0ah, 0dh, '$'
DATASG ENDS
CODE SEGMENT
```

```
ASSUME DS:DataSG, CS:Code, SS:STACKSG
```

```
WRITE PROC FAR
```

```
    jmp aaa
```

```
    ST_SS DW 0000
```

```
    ST_SP DW 0000
```

```
    ST_AX DW 0000
```

```
    INT_STACK DW 20 DUP(0)
```

```
aaa:
```

```
    mov ST_SP, SP ; сохранение SP
```

```
    mov ST_AX, AX
```

```
    mov AX, SS
```

```
    mov ST_SS, AX ; сохранение SS
```

```
    mov AX, INT_STACK
```

```
    mov SS, AX ;новый стек
```

```
    mov AX, ST_AX
```

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

```
    PUSH DX
```

```
    mov AH, 09H ;вызов того,
```

```
    int 21h ;что хранится в dx
```

```
    dec KKK
```

```
    mov dx, KKK
```

```
    add dx, 48
```

```
    mov ah,2
```

```
    int 21h
```

```
    mov dx, OFFSET crlf
```

```
    mov ah, 09H
```

```
    int 21h
```

```
    POP DX ;восстановление регистров
```

```
    POP AX
```

```
    mov ST_AX, AX
```

```
    mov AX, ST_SS ; восстанавливаю SS
```

```
    mov SS, AX
```

```
    mov SP, ST_SP ; восстанавливаю SP
```

```

mov AX, ST_AX
MOV AL, 20H
OUT 20H, AL
IRET
WRITE ENDP

```

```

Main PROC FAR
mov ax, DATASG ;ds setup
mov ds, ax

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 вектора прерывания

CLI
PUSH DS
MOV DX, OFFSET WRITE
MOV AX, SEG WRITE ; сегмент процедуры
MOV DS, AX ; помещаем в DS
MOV AH, 25H ; функция установки вектора
MOV AL, 1CH ; номер вектора
INT 21H ; меняем прерывание
POP DS
STI

mov DX, OFFSET GREETING
mov KKK, 7

count_loop:
cmp KKK, 0
jnz count_loop
CLI
PUSH DS
MOV DX, KEEP_IP
MOV AX, KEEP_CS

```

```

        MOV     DS,  AX
        MOV     AH,  25H
        MOV     AL,  1CH
        INT     21H                ;  восстанавливаем  старый  вектор
прерывания
        POP     DS
        STI

        mov     ah, 4Ch;
        int     21h;

Main      ENDP
CODE      ENDS
END Main

```

### **Выводы.**

Получены навыки в области разработки и применения собственного прерывания.