

**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: Разработка собственного прерывания.**

Студент гр. 9382

Савельев И.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Разработать собственное прерывание на языке ассемблер.

### **Задание.**

Вариант 2С

Цифра в шифре задает номер и назначение заменяемого вектора прерывания:

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

2 -60h -прерывание пользователя -должно генерироваться в программе;

3 -23h -прерывание, генерируемое при нажатии клавиш Control+C ;

4 -08h-прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек.

Буква определяет действия, реализуемые программой обработки прерываний:

A -Печать сообщения на экране;

B -Выдача звукового сигнала;

C -Приостановить вывод на экран (вставить цикл задержки).

Замечание: для исключения возможного взаимного влияния системных и пользовательских прерываний рекомендуется отвести в программе под стек не менее 1К байт.

### **Ход выполнения.**

Описано прерывание отдельной функцией реализующие приостановку вывода на экран. Изменено прерывание 60h на собственную функцию.

### **Тестирование.**

Начало выполнения пр	Через 3 сек
D:\>lab5.exe	D:\>lab5.exe
	D:\>

### **Вывод.**

В ходе выполнения лабораторной работы была разработано собственное прерывание на языке ассемблер.

## Приложение А. Исходный код программы.

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

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

CODE SEGMENT ;SEG CODE
ASSUME CS:CODE, DS:DATA, SS:STACKSG

MAKE_DELAY PROC FAR

    PUSH AX ;сохраняем все
    изменяемые регистры
    PUSH DX ;сохраняем все
    изменяемые регистры
    PUSH CX ;сохраняем все
    изменяемые регистры

    MOV CX, 0FH
    MOV DX, 5240H
    MOV AH, 86H
    INT 15H

    POP CX ;восстанавливаем
    регистры
    POP DX ;восстанавливаем
    регистры
```

```

        POP AX                                ;восстанавливаем
регистры
        MOV AL, 20H
        OUT 20H, AL

        IRET                                ;конец прерывания
MAKE_DELAY ENDP                             ;конец процедуры

Main PROC FAR

        MOV AH, 35H                        ; функция получения
текущего значения вектора прерывания
        MOV AL, 60H                        ; номер вектора
        INT 21H
        MOV KEEP_IP, BX                    ;      запоминание
смещения
        MOV KEEP_CS, ES                    ; и сегмента вектора
прерывания

        PUSH DS
        MOV DX, OFFSET MAKE_DELAY
        MOV AX, SEG MAKE_DELAY             ; 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

        int 60h;

```

```

        CLI                                ; сбросили флаг
прерывания
        PUSH DS
        MOV DX, KEEP_IP
        MOV AX, KEEP_CS
        MOV DS, AX
        MOV AH, 25H                      ; установки адреса
нового обработчика прерывания в поле
векторов прерываний
        MOV AL, 60H
        INT 21H                          ; восстанавливаем
старый вектор прерывания
        POP DS
        STI

        mov ah, 4Ch;
        int 21h;

        RET
Main      ENDP
CODE      ENDS
END Main

```