

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

**ОТЧЕТ**  
**по лабораторной работе №5**  
**по дисциплине «Организация ЭВМ и систем»**  
**Тема: Написание собственного прерывания**  
**Вариант 9**

Студент гр.1381

\_\_\_\_\_

Смирнов Д. Ю.

Преподаватель

\_\_\_\_\_

Ефремов М. А.

Санкт-Петербург

2022

### **Цель работы.**

Изучить, как работают прерывания. Написать собственное прерывание.

### **Задание.**

В соответствии с 9 вариантом шифр задания – 2b, где

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

b – Выдача звукового сигнала с заданной высотой звука.

### **Выполнение работы.**

В сегменте стека *STACK*, как и требуется по заданию, выделяется 1 Кбайт памяти, то есть *DW 512*.

В сегменте данных *DATA* содержится две переменных для хранения старого прерывания *60h* (его смещения и сегмента), – *KEEP\_CS*, *KEEP\_IP*.

В сегменте кода сначала определяем процедуру пользовательского прерывания *MY\_SUBR\_INT*. Нынешнее состояние регистров сохраняется в стек, и восстанавливаются в конце процедуры. Подается значение на порт *43h* (в двух командах, т.к. канал таймера двухбитовый). Значение высоты звука хранится записывается в регистр *AX*, после чего поочередно подается на порт *42h*. Считывается состояние динамика (порт *61h*) в регистр *AL*. Передав на порт вывода (*61h*) значение для управления динамиком, он включается. Обнуляется регистр *CX*. Цикл, отвечающий за продолжительность звучания сигнала (*sound*), работает пока динамик включен. После цикла в порт вывода *61h* записывается первоначальное значение и динамик отключается.

Процедура *Main*. Записывается в *DS* смещение до *DATA*. Вызывается прерывание *21h* (при *AH = 35h* и *AL = 60h*) получается прерывание, хранящееся по смещению *60h*. Значения регистров сохраняются в переменные (*BX* - в *KEEP\_IP*, *ES* - в *KEEP\_CS*). Новое прерывание *MY\_SUBR\_INT* записывается в прерывание *60h*, с помощью прерывания *21h* (при *AH = 25h*, *AL = 60h*). Вызов прерывания *60h*. После вызова нового прерывания происходит восстановление старого прерывания и выход из программы. Исходный код см. в приложении А.

**Выводы.**

В ходе выполнения лабораторной работы были изучены виды прерываний и работа с ними. Создано собственное прерывание. Написана выдача звукового сигнала с заданной высотой звука.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММ

Название файла: *lab5.asm*

```
SOUND_FREQ EQU 10000d

STACK SEGMENT STACK
    DW 512 DUP(?)
STACK ENDS

DATA SEGMENT
    KEEP_CS DW 0
    KEEP_IP DW 0
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:STACK
    MY_SUBR_INT PROC FAR
        push ax
        push cx

        mov al, 20h
        out 20h, al

        mov ax, SOUND_FREQ
        out 42h, al
        mov al, ah
        out 42h, al

        in al, 61h
        or al, 11b
        out 61h, al

        sub cx, cx
    sound:
        loop sound

        in al, 61h
        xor al, 11b
        out 61h, al

        pop cx
        pop ax
        iret
    MY_SUBR_INT ENDP

    MAIN PROC FAR
        push DS
        sub AX, AX
        push AX
        mov AX, DATA
        mov DS, AX

        mov AX, 3560h
        int 21h
```

```

mov KEEP_CS, es ;save regs
mov KEEP_IP, bx

push ds      ;custom interruption setting
mov dx, offset MY_SUBR_INT
mov ax, seg MY_SUBR_INT
mov ds, ax
mov ax, 2560h
int 21h
pop ds

int 60h

CLI      ;previous interruption restoring
push ds
mov dx, KEEP_IP
mov ax, KEEP_CS
mov ds, ax
mov ax, 2560h
int 21h
pop      DS
STI

mov AH, 4ch
int 21h
MAIN ENDP
CODE ENDS
END MAIN

```