

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

**ОТЧЁТ**  
**по лабораторной работе №5**  
**по дисциплине «Организация ЭВМ и систем»**  
**Тема: Разработка собственного прерывания.**

Студентка гр. 1381 \_\_\_\_\_

Туркова Д.Н.

Преподаватель \_\_\_\_\_

Ефремов М. А.

Санкт-Петербург  
2022

## Цель работы

Разработать программу обработки прерываний.

## Задание

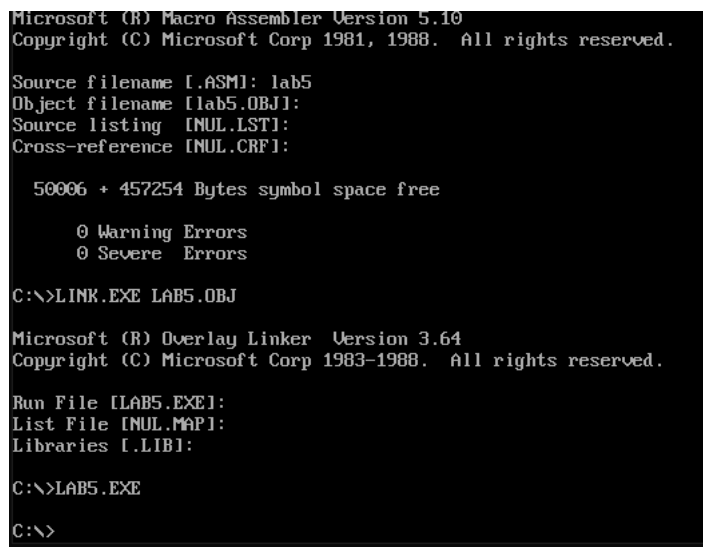
Цифра в шифре задает номер и назначение заменяемого вектора прерывания:

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

Буква определяет действия, реализуемые программой обработки прерываний:

В - Выдача звукового сигнала с заданной высотой звука.

## Выполнение работы



```
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Source filename [ASM]: lab5
Object filename [lab5.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

50006 + 457254 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>LINK.EXE LAB5.OBJ

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LAB5.EXE]:
List File [NUL.MAP]:
Libraries [LIB]:

C:\>LAB5.EXE

C:\>
```

Рис. 1: Сборка и запуск программы в эмуляторе dosbox

Задаются константы *interruption EQU 08h* - прерывание от системного таймера *TONE EQU 20* – хранит в себе частоту звука, *DURATION EQU 500* – как долго будет звучать звук. В сегменте данных создаются *KEEP\_CS DW 0* - для хранения сегмента *KEEP\_IP DW 0* - смещения прерывания.

В процедуре SUBR\_INT PROC FAR включается таймер, устанавливается состояние динамика и его включение, а также частота и

длительность звука. Цикл содержит вложенный цикл. Так как современные процессоры очень быстрые, если использовать один цикл, то звук включится и выключится настолько быстро, что вы не успеете ничего услышать. Поэтому вложенный цикл используется для задержки времени, в течении которого динамик включён.

*RESTORE PROC NEAR* - Т.к программа, использует новые программы обработки прерываний, при своем завершении она восстанавливать оригинальные векторы прерываний.

*REPLACEMENT PROC NEAR* - Функция 35 прерывания 21H возвращает текущее значение вектора прерывания, помещая значение сегмента в ES, а смещение в BX. 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.

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

## **Вывод**

Успешно реализована программа на языке Ассемблера, обрабатывающая прерывания.

# Приложение А

## Исходный код программы

Название файла: lab5.asm

```
interruption EQU 08h
```

```
TONE EQU 20
```

```
DURATION EQU 500
```

```
ASTACK SEGMENT STACK
```

```
    DB 2048 DUP(?)
```

```
ASTACK ENDS
```

```
DATA SEGMENT
```

```
keep_cs DW 0      ;для хранения сегмента
```

```
keep_ip DW 0      ;для смещения прерывания
```

```
check DW 300
```

```
DATA ENDS
```

```
CODE SEGMENT
```

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

```
MAIN PROC FAR
```

```
    push ds
```

```
    xor ax, ax
```

```
    push ax
```

```
    mov ax, DATA
```

```
    mov ds, ax
```

```
    call REPLACEMENT
```

```
repeat:
```

```
    cmp BYTE PTR [check], 0
```

```
    jne repeat
```

```
    call RESTORE
```

```
    ret
```

```
MAIN ENDP
```

```
SUBR_INT PROC FAR
```

```
    push ax
```

```
    push cx
```

```
    mov al, 20h
```

```
    out 20h, al
```

```
    dec WORD PTR [check]
```

```
    mov al, TONE ;Высота звука (частота)
```

```
    out 42h, al ;Включить таймер, который будет выдавать
```

```
импульсы на динамик с заданной частотой
```

```
    in al, 61h ;Получить состояние динамика
```

```
    or al, 00000011b ;Установить два младших бита
```

```
    out 61h, al ;Включить динамик
```

```
    mov cx, DURATION ;Установить длительность звука
```

```
sound:
```

```
    loop sound ;переход, пока cx!=0
```

```
    and al, 11111100b ;Сбросить два младших бита
```

```
    out 61h, al ;Выключить динамик
```

```
    pop cx
```

```

        pop ax
        iret
SUBR_INT ENDP

RESTORE PROC NEAR
        push dx
        push ax

        cli                                ; Команда CLI (Clear Interrupt flag)
сбрасывает флаг IF в значение 0, что запрещает прерывания
        push ds                            ;
        mov dx, keep_ip                    ;
        mov ax, keep_cs                    ;
        mov ds, ax                        ;
        mov ah, 25h                       ;
        mov al, interruption              ;
        int 21h                           ;
        pop ds                            ;
        sti                               ;

        pop ax
        pop dx
        ret
RESTORE ENDP

REPLACEMENT PROC NEAR                    ; возвращает текущее значение вектора прерывания
        push ax
        push dx

        mov ah, 35h                       ; функция получения вектора, возвращает
значение вектора прерывания для INT (AL)
        mov al, 1ch                       ; номер прерывания
        int 21h                           ;
        mov keep_ip, bx                   ; запоминание смещения
        mov keep_cs, es                   ; и сегмента

        push ds                            ;
        mov dx, OFFSET subr_int           ; смещение для процедуры в DX
        mov ax, SEG subr_int              ; сегмент процедуры
        mov ds, ax                        ; помещаем в DS
        mov ah, 25h                       ; функция установки вектора
        mov al, interruption              ; номер вектора
        int 21h                           ; меняем прерывание
        pop ds                            ; Извлечение операнда из стека

        pop dx
        pop ax
        ret
REPLACEMENT ENDP
CODE ENDS
END MAIN

```