

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: Создание собственных прерываний

Студент гр. 0383

Трофимов К. М.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2021

Цель работы.

Создать собственное прерывание, которое будет давать звуковой сигнал определённой частоты.

Формулировка задания.

Вариант №17, 3С

3 - 23h - прерывание, генерируемое при нажатии клавиш Control+C ;
С - Выдача звукового сигнала;

Ход работы.

Прерывание реализуется в процедуре `interrupt_func`. Прерывание работает с положительным числом, хранящимся в `ax`, что представляет из себя высоту звука. Тональность генерируется при помощи генератора тона в микросхеме 8253, чей второй канал даёт выход на динамик. Чтобы установить канал таймера, необходимо передать код `0B6h` в порт `43h`, который является управляющим для той же микросхемы. Таким образом канал 2 настраивается на работу делителя частоты, то есть на разбиение частоты на 16 битовое число, которое будет загружено в `42h`, то есть в регистр второго канала микросхемы.

Далее происходит вывод звука, путём изменения битов управляющего порта `61h`, бит 1 выводного порта `61h` подключен к динамику. Всякий раз, когда программа меняет значение этого бита, диффузор динамика двигается либо наружу, либо внутрь. Быстро меняя значение этого бита, программа генерирует звук. После динамика выключается и восстанавливает значения регистров `ax` и `sx`, которые были занесены в стек в начале процедуры.

Далее в `Main` при помощи функции прерывания `int 21h`, в которую передаётся `35h`, запоминается наш вектор прерывания `23h` в переменные `keep_ip` и `keep_cs`. При помощи `int 21h 25h` мы устанавливаем новый вектор прерывания, который реализовали ранее, и позже вызываем его в программе, заранее положив в `ax` высоту звука. После выполнения

восстанавливаем вектор прерывания 23h при помощи переменных keep_cs и keep_ip, заведённых ранее.

Результаты тестирования программы см. в таблице 1.

Код программы см. в приложении А.

Тестирование.

Таблица 1 — тестирование.

№	Входные данные	Выходные данные	Комментарии
1	В ах лежит 1500	Выводится высокий звук	-
2	В ах лежит 5000	Выводится низкий звук	-

Вывод.

В ходе выполнения лабораторной работы было реализовано собственное прерывание на языке ассемблера.

Приложение А.

Исходный код программы.

lab5.asm

```
Sstack SEGMENT stack
    dw 1024 DUP (?)
stack ENDS

data    SEGMENT
    keep_cs dw 0 ; сегмент
    keep_ip dw 0 ; прерывание
data    ENDS

code    SEGMENT
    assume cs:code, ds:data, ss:stack

interrupt_func PROC far
    jmp start
    spec_sp dw 0000h
    spec_ss dw 0000h
    SPEC_STACK db 40 DUP(0)

start:
    mov spec_sp, sp
    mov spec_ss, ss
    mov sp, SEG SPEC_STACK
    mov ss, sp
    mov sp, offset start
    push ax
    push cx

;обработка прерывания
    mov cx, ax
    mov al, 10110110b ; 0B6H
    out 43h, al
    mov ax, cx ; высота звука
    out 42h, al
    mov al, ah
    out 42h, al
    in al, 61h ; генерация звука
    mov ah, al
    or al, 3
    out 61h, al
    sub cx, cx
    sound:
        pop
        loop sound ; пока динамик работает
    mov al, ah
    out 61h, al ; выключение динамика

; восстановление регистров
    pop cx
    pop ax
    mov ss, spec_ss
    mov sp, spec_sp
    mov al, 20h
    out 20h, al
    iret
```

```
interrupt_func ENDP
```

```
Main PROC far
; текущий вектор прерывания
mov ah, 35h; функция получения вектора
mov al, 23h; номер вектора
int 21h
mov keep_ip, bx; запоминание смещения
mov keep_cs, es; запоминание сегмента

; установка вектора прерывания
push ds
mov dx, offset interrupt_func
mov ax, seg interrupt_func
mov ds, ax
mov ah, 25h ; функция установки вектора
mov al, 23h ; номер вектора
int 21h
pop ds

; ожидание нажатия ctrl+c
ctrl_c:
mov ah, 0
int 16h
cmp al, 3
jne ctrl_c

mov ax, 3000 ; частота звука
int 23h

; восстановление изначального прерывания
cli ; сброс if
push ds
mov dx, keep_ip
mov ax, keep_cs
mov ds, ax
mov ah, 25h
mov al, 23h
int 21h ; восстанавливаем вектор
pop ds
sti ; разрешаем аппаратные прерывания

mov ah, 4ch
int 21h
Main ENDP
code ENDS
END Main
```