

**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. 1383

\_\_\_\_\_

Петров А.С.

Преподаватель

\_\_\_\_\_

Ефремов М. А.

Санкт-Петербург

2022

### **Цель работы.**

Разработать прерывание генерируемое при нажатии клавиш ctrl + c одновременно и написать программу, обрабатывающую это прерывание.

### **Задание на лабораторную работу.**

Разработать на языке Ассемблера прерывание:

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

Программа обработки прерываний должна выдавать звуковой сигнал с заданной высотой звука.

### **Выполнение работы.**

В начале выполнения главной процедуры с помощью функции 35h прерывания 21 h сохраняем адрес текущего прерывания и записываем сегмент в KEER\_CS, а смещение в KEER\_IP. С помощью функции 25h прерывания 21h устанавливаем вектор прерывания на указанный адрес обработчика Interruption.

Обработчик Interruption сохраняет изменяемые регистры ax, dx в стек, в al записывает информацию о динамике с порта 61h помощью команды in. Далее включается динамик, устанавливается высота звука и включается системный таймер через порт 42h. После выключения динамика восстанавливаются регистры ax, dx из стека командой pop. С помощью команды ret происходит возврат из прерывания.

Прерывание вызывается, когда с клавиатуры будет считано ctrl + c. Считывание происходит с порта 60h и первый полученный символ сравнивается с кодом 1d (ctrl). Если была нажата любая другая клавиша, то программа будет ожидать новую клавишу. Если была нажата клавиша ctrl, то проверяется следующий символ и сравнивается с кодом 2e (c). Если второй символ не c, то придется нажимать ctrl заново. Если нажата нужная

комбинация клавиш, то устанавливается высота звука в регистр al и происходит вызов прерывания.

С помощью команды cli сбрасываем флаг прерывания и восстанавливаем старый вектор прерывания.

### **Выводы.**

В ходе выполнения работы было разработано прерывание, генерируемое при одновременном нажатии клавиш ctrl + c, разработана программа обработки прерываний, которая при срабатывании прерывания выдает звуковой сигнал.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lr5.asm

```
; Стек программы
AStack      SEGMENT  STACK
              DW 1024 DUP(?)
AStack      ENDS

; Данные программы
DATA        SEGMENT
KEEP_CS dw 0
KEEP_IP dw 0
duration dw 1500; длительность звучания звукового сигнала
DATA        ENDS

; Код программы
CODE        SEGMENT
              ASSUME CS:CODE, DS:DATA, SS:AStack

Interruption PROC FAR; Прерывание
    push ax
    push dx
    mov dx, ax
    in  al, 61h
    push ax
    or  al, 00000011b
    out 61h, al
    mov al, dl
    out 42h, al
    mov cx, duration

; Задержка
Audio_Stop:
    push cx
    mov cx, duration
Audio_Stop_temp:
    nop
    loop Audio_Stop_temp
    pop cx
    loop Audio_Stop

    pop ax
    and al, 11111100b
    out 61h, al

    pop dx
    pop ax
    mov al, 20h
```

```

        out    20h, al

        iret
Interruption ENDP

; Головная процедура
Main      PROC   FAR
    push ds
    xor    ax, ax
    push ax
    mov    ax, DATA
    mov    ds, ax
    mov    ah, 35h
    mov    al, 23h
    int    21h
    mov    KEEP_CS, es
    mov    KEEP_IP, bx

    push ds
    mov    dx, offset Interruption
    mov    ax, seg Interruption
    mov    ds, ax
    mov    ah, 25h
    mov    al, 23h
    int    21h
    pop    ds

INPUT_CTRL:
    in     al, 60h
    cmp    al, 1dh                ; 1d - клавиша ctrl
    jne    INPUT_CTRL
INPUT_C:
    in     al, 60h
    cmp    al, 9dh                ; отпущение клавиши ctrl
    je     INPUT_CTRL
    cmp    al, 2eh                ; 2e - клавиша c
    jne    INPUT_C
    mov    al, 66
    int    23h

    cli

    push ds
    mov    dx, KEEP_IP
    mov    ax, KEEP_CS
    mov    ds, ax
    mov    ah, 25h
    mov    al, 23h
    int    21h
    pop    ds

    sti

```

```
        ret

Main      ENDP
CODE      ENDS
          END Main
```