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 _____ **Желнин М.Ю.**

Преподаватель _____ **Ефремов М.А.**

Санкт-Петербург

2021

Цель работы.

Создать собственное прерывание с номером 23h, вызываемое в программе, которое будет давать звуковой сигнал определённой частоты.

Формулировка задания.

Вариант №8, 3B

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

B - Выдача звукового сигнала;

Ход работы.

Прерывание реализуется в процедуре MY_INT. 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 разбитие 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 динамик выключается и восстанавливает значения регистров AX и CX, 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 AX лежит 2000	Выводится звук	**Верно.**
2	В AX лежит 5000	Выводится более низкий звук.	**Верно.**

Вывод.

В ходе выполнения лабораторной работы было реализовано собственное прерывание на языке Ассемблер. Так же были изучены способы выводы звука на динамики.

Приложение А.

Исходный код программы.

lab5.asm

```
STACK SEGMENT STACK
        DW 1024 DUP (?)
STACK ENDS

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

CODE    SEGMENT
  ASSUME CS:CODE, DS:DATA, SS:STACK

MY_INT  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    ; сохранение изменяемых регистров
        PUSH CX

       ;<действия по обработке прерывания>
        MOV CX, AX
        MOV AL, 10110110b ; 0B6H
        OUT 43H, AL ; Код для установления канала 2 таймера-счетчика на работу в качестве делителя
частоты см. методу
        MOV AX, CX ; Заносим в AX высоту звука
        OUT 42H, AL
        MOV AL, AH
        OUT 42H, AL ; Заносим поочередно 2 байта в порт 42h(регистр канала 2)
        IN AL, 61H ; генерация звука путём сдвига диффузора туда-обратно
```

```
        MOV AH, AL
        OR AL, 3
        OUT 61H, AL
        SUB CX, CX
        WHILE_TIME:
        NOP
        LOOP WHILE_TIME ; Цикл, пока динамик работает
        MOV AL, AH
        OUT 61H, AL ; Выключение динамика (изначальное значение порта 61h)
       ;<конец действий по обработке прерывания>
        POP  CX
        POP  AX   ; восстановление регистров
        MOV  SS, spec_SS
        MOV  SP, spec_SP
        MOV  AL, 20H
        OUT  20H,AL
        IRET
MY_INT  ENDP

Main    PROC  FAR
        ; <Запоминание текущего вектора прерывания>
        MOV  AH, 35H   ; функция получения вектора
        MOV  AL, 23H   ; номер вектора
        INT  21H
        MOV  KEEP_IP, BX  ; запоминание смещения
        MOV  KEEP_CS, ES  ; и сегмента

        ; <Установка вектора прерывания>
        PUSH DS
        MOV  DX, OFFSET MY_INT ; смещение для процедуры в DX
        MOV  AX, SEG MY_INT    ; сегмент процедуры
        MOV  DS, AX         ; помещаем в DS
        MOV  AH, 25H        ; функция установки вектора
        MOV  AL, 23H        ; номер вектора
        INT  21H            ; меняем прерывание
        POP  DS

        ctrl_c:
   mov ah, 0
   int 16h
```

```
    cmp al, 3  ;код символа после нажатия
    jne ctrl_c

            mov ax, 1000
            int 23h



            ; <Восстановление изначального вектора прерывания>
            CLI
            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

            MOV AH, 4Ch
            INT 21h
Main        ENDP
CODE ENDS
            END Main
```