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: Написание собственного прерывания.

Студент гр. 0382 _____ Злобин А.С.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Узнать как работает механизм прерываний в ассемблере. Научится писать собственные прерывания.

Задание.

Вариант 1а

Назначение заменяемого вектора прерывания:

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

Действие, реализуемые программой обработки прерываний:

B - Выдача звукового сигнала с заданной высотой звука.

Выполнение работы.

В главной процедуре main сначала вызывается функция 35h прерывания 21h для получения текущего вектора прерывания 60h. Значения CS этого вектора, хранящегося в результате в ES, и IP, хранящегося в BX, записываются в память для того, чтобы возвратить этот вектор в конце программы. Для задания нового адреса прерывания используется функция 25h прерывания 21h, перед которой в DX записывается смещение процедуры с созданным прерыванием, а сегмент записывается в DS, в AL записывается номер прерывания. Далее программа считывает ввод с клавиатуры с помощью функции 00h прерывания 16h: при нажатии на ‘a’ значения в BX (по умолчанию 4500) уменьшается на 100, если на ‘s’, то увеличивается на 100. Это значение используется для определения частоты звука. В конце программы с помощью той же функции 25h и сохранённых CS и IP восстанавливается изначальный вектор для прерывания. Само прерывание реализовано в процедуре inter. В начале и в конце происходит сохранения и восстановления регистров, которые используются в процессе. Для

подачи звука сначала подаём значение 10110110b на порт 43h, управляющий микросхемой 8253, для установки канала 2 таймера-счётчика, подключенного к динамику, в качестве делителя частоты (1.19 МГц делится на 16-битовое число, которое записывается в регистр канала 2 по адресу 42h). Порт канала 2 8-битовый, поэтому изначальное значение передаём по половине. Само значение читается из регистра BX, изменение которого 3 описано выше. Далее в значении порта вывода 61h устанавливает биты 0 и 1 в положение 1 для пропуска сигнала на динамик, исходное значение запоминаем. Далее идёт цикл в 16^4 итераций для продления звука, после чего восстанавливается исходное значение порта 61h, благодаря чему динамик выключается.

Тестирование.

При нажатии на ‘a’ частота звука уменьшается, при нажатии на ‘d’ частота увеличивается, при нажатии на другую клавишу, программа завершается.

Выводы.

Был изучен механизм прерываний в языке ассемблер.

В ходе данной лабораторной работы была разработана программа , которая создает свое собственное прерывание, которое выполнить вывод сообщения на экран заданное число раз, после чего вставить фиксированную задержку и вывести сообщение о завершении обработчика.

Приложение А

Исходный код программы

Название файла: lb5.asm

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

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

```
DATA SEGMENT
KEEP_CS DW 0
    KEEP_IP DW 0

DATA ENDS

CODE SEGMENT
inter PROC FAR
        push ax
      push cx

      mov al, 10110110b
      out 43h, al

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

      in al, 61h
      mov ah, al
      or al, 3
      out 61h, al

      sub cx, cx
      l: loop l

      mov al, ah
      out 61h, al

        pop ax
      pop cx
        mov al, 20h
        out 20h, al
        iret
inter  ENDP
main PROC FAR
      mov ah, 35h
      mov al, 60h
      int 21h
      mov KEEP_IP, bx
      mov KEEP_CS, es

      mov bx, 4500

      push ds
      mov dx, offset inter
      mov ax, seg inter
      mov ds, ax
      mov ah, 25h
      mov al, 60h
      int 21h
      pop ds
```

```
      jmp readKey
      incFrec:
            cmp bx, 100
            jle readKey
            sub bx, 100
            int 60h
            jmp readKey
      decFrec:
            cmp bx, 10000
            jge readKey
            add bx, 100
            int 60h
      readKey:
            mov ah, 0h
            int 16h
            cmp al, 'a'
            je decFrec
            cmp al, 's'
            je incFrec

      cli
      push ds
      mov dx, KEEP_IP
      mov ax, KEEP_CS
      mov ds, ax
      mov ah, 25h
      mov al, 60h
      int 21h
      pop  ds
      sti

      mov ah, 4ch
      int 21h
main ENDP
CODE ENDS
END main
```