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

ОТЧЕТ

по лабораторной работе №5

по дисциплине «ОргЭВМиС»

Тема: Написание собственного прерывания.

Студент гр. 0382 _____ Афанасьев Н. С.

Преподаватели _____ Ефремов М.А.

Санкт-Петербург
2021

Цель работы.

Изучить работу с прерываниями, научиться создавать собственное прерывание.

Задание.

Вариант 2: создать собственное прерывание с номером 08h, которое будет выдавать звуковой сигнал с заданной высотой звука.

Выполнение работы.

В главной процедуре main сначала вызывается функция 35h прерывания 21h для получения текущего вектора прерывания 08h, который генерируется системой примерно 18 раз в секунду. 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 ‘d’, 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, подключённого 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

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.

Разработанный программный код см. в приложении А.

Файл листинга см. в приложении В.

Тестирование.

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

Выводы.

Была изучена работа с прерываниями в Ассемблере, было изучено создание собственных прерываний.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЕ КОДЫ ПРОГРАММ

Название файла: lr5.cpp

```
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, 08h
           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, 08h
            int 21h
            pop ds

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

            cli
            push ds
            mov dx, KEEP_IP
            mov ax, KEEP_CS
            mov ds, ax
            mov ah, 25h
            mov al, 08h
            int 21h
            pop  ds
            sti

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

ПРИЛОЖЕНИЕ В

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lr5.lst

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

0000                    STACK SEGMENT STACK
0000  0400[                        DW 1024 DUP(?)
       ????
                ]

0800                    STACK ENDS

0000                    DATA SEGMENT
0000  0000                        KEEP_CS DW 0
0002  0000                      KEEP_IP DW 0

0004                    DATA ENDS

0000                    CODE SEGMENT
0000                         inter PROC FAR
0000  50                         push ax

0001  B0 B6                              mov al, 10110110b
0003  E6 43                              out 43h, al

0005  8B C3                              mov ax, bx
0007  E6 42                              out 42h, al
0009  8A C4                              mov al, ah
000B  E6 42                              out 42h, al

000D  E4 61                              in al, 61h
000F  8A E0                              mov ah, al
0011  0C 03                              or al, 3
0013  E6 61                              out 61h, al

0015  2B C9                              sub cx, cx
0017  E2 FE                              l: loop l

0019  8A C4                              mov al, ah
001B  E6 61                              out 61h, al

001D  58                            pop ax
001E  B0 20                              mov al, 20h
0020  E6 20                              out 20h, al
0022  CF                            iret
0023                           inter  ENDP
0023                           main PROC FAR
0023  B4 35                              mov ah, 35h
0025  B0 08                              mov al, 08h
0027  CD 21                              int 21h
0029  89 1E 0002 R                       mov KEEP_IP, bx
002D  8C 06 0000 R                       mov KEEP_CS, es

0031  BB 1194                            mov bx, 4500
```

```
0034  1E                        push ds
0035  BA 0000 R                 mov dx, offset inter

0038  B8 ---- R                 mov ax, seg inter
003B  8E D8                          mov ds, ax
003D  B4 25                          mov ah, 25h
003F  B0 08                          mov al, 08h
0041  CD 21                          int 21h
0043  1F                        pop ds

0044  EB 15 90                       jmp readKey
0047                            incFrec:
0047  83 FB 64                            cmp bx, 100
004A  7E 0F                                jle readKey
004C  83 EB 64                             sub bx, 100
004F  EB 0A 90                             jmp readKey
0052                            decFrec:
0052  81 FB 2710                           cmp bx, 10000
0056  7D 03                                jge readKey
0058  83 C3 64                             add bx, 100
005B                            readKey:
005B  B4 00                                mov ah, 0h
005D  CD 16                                int 16h
005F  3C 61                                cmp al, 'a'
0061  74 EF                                je decFrec
0063  3C 64                                cmp al, 'd'
0065  74 E0                                je incFrec

0067  FA                        cli
0068  1E                        push ds
0069  8B 16 0002 R                    mov dx, KEEP_IP
006D  A1 0000 R                 mov ax, KEEP_CS
0070  8E D8                          mov ds, ax
0072  B4 25                          mov ah, 25h
0074  B0 08                          mov al, 08h
0076  CD 21                          int 21h
0078  1F                        pop  ds
0079  FB                        sti

007A  B4 4C                          mov ah, 4ch
007C  CD 21                          int 21h
007E                        main ENDP
007E                CODE ENDS
                    END main
```

Segments and Groups:

N a m e	Length	Align	Combine	Class
CODE	007E	PARA	NONE	
DATA	0004	PARA	NONE	
STACK	0800	PARA	STACK	

Symbols:

N a m e	Type	Value	Attr
DECFREC	L NEAR	0052	CODE

```
INCFREC  . . . . . .     L NEAR     0047  CODE
INTER  . . . . . .       F PROC     0000  CODE  Length     =
0023

KEEP_CS  . . . . . .     L WORD     0000  DATA
KEEP_IP  . . . . . .     L WORD     0002  DATA

L  . . . . . .           L NEAR     0017  CODE

MAIN . . . . . .         F PROC     0023  CODE  Length     =
005B

READKEY  . . . . . .     L NEAR     005B  CODE

@CPU . . . . . .         TEXT  0101h
@FILENAME  . . . . . .   TEXT  lr5
@VERSION . . . . . .     TEXT  510


     92 Source  Lines
     92 Total   Lines
     16 Symbols

  48034 + 461273 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```