

**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. 1382

\_\_\_\_\_ Исайкин Г. И.

Преподаватель

\_\_\_\_\_ Ефремов М. А

Санкт-Петербург

2022

### **Цель работы.**

Изучить работу прерывания и написать собственное на языке Ассемблер.

### **Задание.**

Разработать на языке Ассемблера программу, которая переопределяет прерывание по определённому вектору, выполняет его и затем восстанавливает изначальное прерывания для этого вектора.

8 вариант. Переписать прерывание по вектору 23h, новое прерывание должно издавать звук.

### **Выполнение работы.**

Программа состоит из трёх сегментов:

- STACK – блок стека
- DATA – блок данных
- CODE — блок кода

В DATA хранятся переменные KEEP\_CS и KEEP\_IP. Они будут хранить изначальные значения по вектору прерывания. В блоке CODE, помимо MAIN есть SUBR\_INT — пользовательская функция прерывания. Она издаёт звук через динамик путём изменения второго бита в выводном порте 61h. В MAIN программа получает изначальные значения по вектору перемещения и записывает их в переменные KEEP\_CS и KEEP\_IP. Затем мы изменяем значения по вектору на соответствующие SUBR\_INT и исполняем процедуру прерывания по вектору. После мы возвращаем изначальные значения по вектору.

В результате теста программы проблем не возникло, звук выводился.

### **Выводы.**

Были изучены прерывания и написание собственных прерываний на языке Ассемблер.

# ПРИЛОЖЕНИЕ А

## ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lr5.asm

```
STACK SEGMENT STACK
    DW 512 DUP(0)
STACK ENDS

DATA SEGMENT
    KEEP_CS DW 0
    KEEP_IP DW 0
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:STACK
    SUBR_INT PROC FAR
        push ax
        push cx

        sub cx, cx

        in al, 61h
        mov ah, al
        and al, 0FEh
        sound:
        or al, 2
        out 61h, al
        and al, 0FDh
        out 61h, al
        loop sound
        mov al, ah
        out 61h, al

        pop cx
        pop ax

        mov al, 20h
        out 20h, al
        iret
    SUBR_INT ENDP

    MAIN PROC FAR
        push DS
        sub AX, AX
        push AX
        mov AX, DATA
        mov DS, AX

        mov AX, 3523h
        int 21h
        mov KEEP_CS, es
        mov KEEP_IP, bx
```

```
push ds
mov dx, offset SUBR_INT
mov ax, seg SUBR_INT
mov ds, ax
mov ax, 2523h
int 21h
pop ds
```

```
int 23h
```

```
CLI
mov dx, KEEP_IP
mov ax, KEEP_CS
mov ds, ax
mov ax, 2523h
int 21h
pop ds
STI
```

```
mov AH, 4ch
int 21h
```

```
MAIN ENDP
```

```
CODE ENDS
```

```
END MAIN
```

# ДИАГНОСТИЧЕСКИЕ СООБЩЕНИЯ

Название файла: lr5.lst

Microsoft (R) Macro Assembler Version 5.10  
01:39:5

11/27/22

Page

1-1

```
0000          STACK SEGMENT STACK
0000  0200[          DW 512 DUP(0)
          0000
          ]

0400          STACK ENDS

0000          DATA SEGMENT
0000  0000          KEEP_CS DW 0
0002  0000          KEEP_IP DW 0
0004          DATA ENDS

0000          CODE SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:STACK
0000          SUBR_INT PROC FAR
0000  50          push ax
0001  51          push cx

0002  2B C9          sub cx, cx

0004  E4 61          in al, 61h
0006  8A E0          mov ah, al
0008  24 FE          and al, 0FEh
000A          sound:
000A  0C 02          or al, 2
000C  E6 61          out 61h, al
000E  24 FD          and al, 0FDh
0010  E6 61          out 61h, al
0012  E2 F6          loop sound
0014  8A C4          mov al, ah
0016  E6 61          out 61h, al

0018  59          pop cx
0019  58          pop ax

001A  B0 20          mov al, 20h
001C  E6 20          out 20h, al
001E  CF          iret
001F          SUBR_INT ENDP

001F          MAIN PROC FAR
001F  1E          push DS
0020  2B C0          sub AX, AX
0022  50          push AX
0023  B8 ---- R      mov AX, DATA
0026  8E D8          mov DS, AX
```

```

0028 B8 3523          mov AX, 3523h
002B CD 21           int 21h
002D 8C 06 0000 R    mov KEEP_CS, es
0031 89 1E 0002 R    mov KEEP_IP, bx

0035 1E             push ds
0036 BA 0000 R      mov dx, offset SUBR_INT
Microsoft (R) Macro Assembler Version 5.10
01:39:5

```

11/27/22

Page

1-2

```

0039 B8 ---- R      mov ax, seg SUBR_INT
003C 8E D8          mov ds, ax
003E B8 2523        mov ax, 2523h
0041 CD 21          int 21h
0043 1F            pop ds

0044 CD 23          int 23h

0046 FA            CLI
0047 8B 16 0002 R    mov dx, KEEP_IP
004B A1 0000 R      mov ax, KEEP_CS
004E 8E D8          mov ds, ax
0050 B8 2523        mov ax, 2523h
0053 CD 21          int 21h
0055 1F            pop ds
0056 FB            STI

0057 B4 4C          mov AH, 4ch
0059 CD 21          int 21h
005B                MAIN ENDP

005B                CODE ENDS

```

END MAIN

```

Microsoft (R) Macro Assembler Version 5.10
01:39:5

```

11/27/22

Symbols-1

#### Segments and Groups:

	N a m e	Length	Align	Combine Class
CODE . . . . .		005B	PARA	NONE
DATA . . . . .		0004	PARA	NONE
STACK . . . . .		0400	PARA	STACK

#### Symbols:

	N a m e	Type	Value	Attr
KEEP_CS . . . . .		L WORD	0000	DATA

KEEP_IP . . . . .	L WORD	0002 DATA
MAIN . . . . . 003C	F PROC	001F CODE Length =
SOUND . . . . .	L NEAR	000A CODE
SUBR_INT . . . . . 001F	F PROC	0000 CODE Length =
@CPU . . . . .	TEXT	0101h
@FILENAME . . . . .	TEXT	LR5
@VERSION . . . . .	TEXT	510

75 Source Lines  
 75 Total Lines  
 13 Symbols

48034 + 461273 Bytes symbol space free

0 Warning Errors  
 0 Severe Errors