

**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. 0382

Сергеев Д.А,

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Изучение работы с прерываниями на языке Ассемблера. Создать собственное прерывание и заменить им существующие.

### **Задание.**

Вариант 16 – 3b

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

В - Выдача звукового сигнала с заданной высотой звука.

### **Выполнение работы.**

Первым делом в процедуре MAIN с помощью функции 35 прерывания 21h мы получаем текущее значение вектора прерываний, в ES помещается значение сегмента, а в BX – смещение. Далее сохраняются исходные значения вектора. После этого в BX заносится значение частоты звука. После этого с помощью функции 25 прерывания 21h меняется оригинальное 23h прерывание на реализованное по заданию. Далее происходит считывание клавиши и в случае, когда скан-код клавиши соответствует Ctrl+C вызывается реализованное прерывание.

В прерывании SUBR\_INT имеется свой стек subr\_int\_stack для корректной работы в случае заполнения стека оригинальной программы. Далее в соответствии с алгоритмом реализуется вывод звука на динамик.

В конце программы восстанавливается оригинальный вектор прерываний.

Исходный программный код смотрите в приложении А.

Файл листинга смотрите в приложении Б.

### **Тестирование.**

Результаты представлены в таблице 1.

Таблица 1 – Результаты тестирования

№	Входные данные	Комментарий
1	Mov bx,5000	Выводится звук

2	Mov bx,1000	Выводится звук, более высокий чем в №1
3	Mov bx,10000	Выводится звук, более низкий чем в №1 и №2

### **Выводы.**

В ходе работы была изучена работа с прерываниями, написано собственное прерывание, это прерывание было интегрировано в оригинальный вектор прерываний, а затем вызвано из программы.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.asm

```
AStack SEGMENT STACK
    DW 512 DUP(?) ; выделим 1 Кбайт памяти
AStack ENDS
; Данные программы
DATA SEGMENT
    KEEP_CS DW 0 ; для хранения сегмента
    KEEP_IP DW 0 ; и смещения вектора прерывания
DATA ENDS
; Код программы
CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack

SUBR_INT PROC FAR
    jmp subr_int_start
    old_SP dw 0
    old_SS dw 0
    subr_int_stack dw 64 dup(0)
subr_int_start:
    mov old_SS,SS
    mov old_SP,SP
    mov ax,seg subr_int_stack
    mov ss,ax
    mov sp, offset subr_int_start

    push cx
    push ax
    mov al,10110110b
    out 43h,al
    mov ax, bx
    out 42h,al

    mov ah,al
    out 42h,al
    in al,61h
    or al, 00000011b
    out 61h, al
    sub cx,cx
Kill_time:
    loop Kill_time
    mov al,ah
    out 61h,al
    pop ax
    pop cx
    mov al,20h
    out 20h,al
    mov ss,old_SS
    mov sp,old_SP
    IRET
SUBR_INT ENDP

; Головная процедура
Main PROC FAR
```

```

mov ah,35h ;функция получения вектора
mov al,23h
int 21h
mov KEEP_CS,ES
mov KEEP_IP,BX
mov bx,8000

```

```

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

```

Input:

```

mov ah, 0
int 16h
cmp al, 3 ;код символа после нажатия
jne Input

```

```

INT 23H

```

```

CLI ; Сброс флага прерываний IF - 0
PUSH DS
MOV DX, KEEP_IP
MOV AX, KEEP_CS
MOV DS, AX
MOV AH, 25H
MOV AL, 23h ;
INT 21H ; восстанавливаем старый вектор прерывания
POP DS
STI

```

```

MOV AH, 4Ch
INT 21h

```

```

Main ENDP
CODE ENDS
END Main

```

## ПРИЛОЖЕНИЕ Б

### ФАЙЛ ЛИСТИНГА

Название файла: lb5.lst

Microsoft (R) Macro Assembler Version 5.10

12/9/21

02:51:18

Page

1-1

```
0000          AStack SEGMENT STACK
0000 0200[                DW 512 DUP(?) ; выделим 1 Кбайт
                        памяти
                        ???
                        ]

0400          AStack ENDS
                        ; Данные программы

0000          DATA SEGMENT
0000 0000                KEEP_CS DW 0 ; для хранения сег-
                        мента
0002 0000                KEEP_IP DW 0 ; и смещения векто-
                        ра прерывания
0004          DATA ENDS
                        ; Код программы

0000          CODE SEGMENT
                        ASSUME CS:CODE, DS:DATA, SS:AStack

0000          SUBR_INT PROC FAR
0000 E9 0087 R          jmp subr_int_start
0003 0000                old_SP dw 0
0005 0000                old_SS dw 0
0007 0040[                subr_int_stack dw 64 dup(0)
                        0000
                        ]

0087          subr_int_start:
0087 2E: 8C 16 0005 R          mov old_SS,SS
008C 2E: 89 26 0003 R          mov old_SP,SP
```

```

0091 B8 ---- R      mov ax,seg subr_int_stack
0094 8E D0          mov ss,ax
0096 BC 0087 R      mov sp, offset subr_int_start

0099 51            push cx
009A 50            push ax
009B B0 B6          mov al,10110110b
009D E6 43          out 43h,al
009F 8B C3          mov ax, bx
00A1 E6 42          out 42h,al

00A3 8A E0          mov ah,al
00A5 E6 42          out 42h,al
00A7 E4 61          in al,61h
00A9 0C 03          or al, 00000011b
00AB E6 61          out 61h, al
00AD 2B C9          sub cx,cx
00AF              Kill_time:
00AF E2 FE          loop Kill_time
00B1 8A C4          mov al,ah
00B3 E6 61          out 61h,al
00B5 58            pop ax
00B6 59            pop cx
00B7 B0 20          mov al,20h
00B9 E6 20          out 20h,al

```

Microsoft (R) Macro Assembler Version 5.10

12/9/21

02:51:18

Page

1-2

```

00BB 2E: 8E 16 0005 R      mov ss,old_SS
00C0 2E: 8B 26 0003 R      mov sp,old_SP
00C5 CF                  IRET
00C6                    SUBR_INT ENDP

```

; Головная процедура

```

00C6                    Main PROC FAR

```

```

00C6 B4 35              mov ah,35h ;функция получен

```

♦ я вектора

```
00C8 B0 23          mov al,23h
00CA CD 21          int 21h
00CC 8C 06 0000 R    mov KEEP_CS,ES
00D0 89 1E 0002 R    mov KEEP_IP,BX
00D4 BB 1F40         mov bx,8000
```

```
00D7 1E            push DS
00D8 BA 0000 R      mov dx,offset SUBR_INT
00DB B8 ---- R      mov ax,seg SUBR_INT
00DE 8E D8          mov ds,ax
00E0 B4 25          mov ah,25h
00E2 B0 23          mov al,23h
00E4 CD 21          int 21h
00E6 1F            pop ds
```

00E7 Input:

```
00E7 B4 00          mov ah, 0
00E9 CD 16          int 16h
00EB 3C 03          cmp al, 3 ;код символа посл♦
```

♦ нажатия

```
00ED 75 F8          jne Input
```

```
00EF CD 23          INT 23H
```

```
00F1 FA            CLI ; Сброс флага прерыва♦
```

♦ ий IF - 0

```
00F2 1E            PUSH DS
00F3 8B 16 0002 R    MOV DX, KEEP_IP
00F7 A1 0000 R      MOV AX, KEEP_CS
00FA 8E D8          MOV DS, AX
00FC B4 25          MOV AH, 25H
00FE B0 23          MOV AL, 23h ;
0100 CD 21          INT 21H ; восстанавливаем с
тарый вектор прерывания
0102 1F            POP DS
0103 FB            STI
```



```

0104  B4 4C                      MOV AH, 4Ch
0106  CD 21                      INT 21h
0108                      Main ENDP
0108                      CODE ENDS
                      END Main

```

Microsoft (R) Macro Assembler Version 5.10

12/9/21

02:51:18

Symbols-1

# Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK . . . . .	0400	PARA	STACK
CODE . . . . .	0108	PARA	NONE
DATA . . . . .	0004	PARA	NONE

# Symbols:

N a m e	Type	Value	Attr
INPUT . . . . .	L NEAR	00E7	CODE
KEEP_CS . . . . .	L WORD	0000	DATA
KEEP_IP . . . . .	L WORD	0002	DATA
KILL_TIME . . . . .	L NEAR	00AF	CODE
MAIN . . . . .	F PROC	00C6	CODE Length =
0042			
OLD_SP . . . . .	L WORD	0003	CODE
OLD_SS . . . . .	L WORD	0005	CODE
SUBR_INT . . . . .	F PROC	0000	CODE Length =
00C6			
SUBR_INT_STACK . . . . .	L WORD	0007	CODE Length =
0040			

```
SUBR_INT_START . . . . . L NEAR      0087 CODE

@CPU . . . . . TEXT  0101h
@FILENAME . . . . . TEXT  1b5
@VERSION . . . . . TEXT  510
```

```
92 Source  Lines
92 Total   Lines
18 Symbols
```

```
48034 + 461273 Bytes symbol space free
```

```
0 Warning Errors
0 Severe  Errors
```