

**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. 9383

Хотяков Е.П.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Получить навыки создания собственных прерываний, изучить процесс вывода звука на динамик.

### **Задание.**

Разработать собственное прерывание. Шифр: ЗВ.

З - 23h - прерывание, генерируемое при нажатии клавиш Control+c.

В - Выдача звукового сигнала

### **Ход работы:**

В сегменте данных создаются переменные типа DW для хранения сегмента и смещения прерывания KEEP\_CS и KEEP\_IP.

В сегменте кода создана процедура обработки прерывания MY\_INT, которая выводит звук на динамик.

Прерывание меняется следующим образом: с помощью функции 35h прерывания 21h получаем вектор прерывания 23h. Затем запоминаем этот вектор в переменные KEEP\_CS и KEEP\_IP. Далее с помощью функции 25h прерывания 21h меняем прерывание 23h.

В конце программы восстанавливаем вектор прерывания.

Разработанный программный код см. в приложении А.

### **Выводы.**

Получены навыки создания собственных прерываний, изучен процесс вывода звука на динамик

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

#### Файл lb5.asm

*AStack SEGMENT STACK*

*DB 1024 DUP(?)*

*AStack ENDS*

*DATA SEGMENT*

*KEEP\_CS DW 0 ;для хранения сегмента*

*KEEP\_IP DW 0 ;и смещения вектора прерывания*

*DATA ENDS*

*CODE SEGMENT*

*ASSUME CS:CODE, DS:DATA, SS:AStack*

*MY\_INT PROC FAR*

*push ax*

*push dx*

*;----- Часть кода выводящая звук*

*MOV AL , 10110110b*

*OUT 43H, AL; Set mode for 2nd channel*

*MOV AX , 1000; Pitch of sound*

*OUT 42H, AL*

*MOV AL , AH*

*OUT 42H, AL; Set it to speaker port*

*IN AL, 61H*

*MOV AH, AL*

*OR AL,3*

*OUT 61H, AL*

*SUB CX, CX*

*KILL\_TIME:*

*LOOP KILL\_TIME*

*MOV AL, AH*

*OUT 61H, AL*

*;-----*

*pop dx*

*pop ax*

*mov al,20h*

*out 20h,al*

*iret*

*MY\_INT ENDP*

*MAIN PROC FAR*

*MOV AX, DATA*

*MOV DS, AX*

*;сохраняем вектор 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 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*

*enter\_ctrl\_c:*

*mov ah,0*

*int 16h*

*cmp al,3*

*jne enter\_ctrl\_c*

*INT 23H*

*CLI*

*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*

## Файл lb5.lst

#Microsoft (R) Macro Assembler Version 5.10

11/25/20 17:38:5

Page 1-1

```
0000          AStack SEGMENT STACK
0000 0400[          DB 1024 DUP(?)
          ??
          ]

0400          AStack ENDS

0000          DATA SEGMENT
0000 0000          KEEP_CS DW 0 ;для хранения с
          егмента
0002 0000          KEEP_IP DW 0 ;и смещения век
          тора прерывания
0004          DATA ENDS

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

0000          MY_INT PROC FAR
0000 50          push ax
0001 52          push dx
          ;----- Часть кода выводящая
          звук
0002 B0 B6          MOV AL , 10110110b
0004 E6 43          OUT 43H, AL; Set mode for 2nd channel
0006 B8 03E8          MOV AX , 1000; Pitch of sound
0009 E6 42          OUT 42H, AL
000B 8A C4          MOV AL , AH
000D E6 42          OUT 42H, AL; Set it to speaker port
```

000F E4 61	IN AL, 61H
0011 8A E0	MOV AH, AL
0013 0C 03	OR AL, 3
0015 E6 61	OUT 61H, AL
0017 2B C9	SUB CX, CX
0019	KILL_TIME:
0019 E2 FE	LOOP KILL_TIME
001B 8A C4	MOV AL, AH
001D E6 61	OUT 61H, AL
	;-----
001F 5A	pop dx
0020 58	pop ax
0021 B0 20	mov al, 20h
0023 E6 20	out 20h, al
0025 CF	iret
0026	MY_INT ENDP

0026	MAIN PROC FAR
0026 B8 ---- R	MOV AX, DATA
0029 8E D8	MOV DS, AX
	;сохраняем вектор прерывания

#Microsoft (R) Macro Assembler Version 5.10 11/25/20 17:38:5

Page 1-2

002B B4 35	MOV AH, 35H ; функция получения вектора
002D B0 23	MOV AL, 23H ; номер вектора
002F CD 21	INT 21H
0031 89 1E 0002 R	MOV KEEP_IP, BX ; запоминание смещения
0035 8C 06 0000 R	MOV KEEP_CS, ES ; и сегмента

## ♦ектора прерывания

0039 1E	PUSH DS
003A BA 0000 R	MOV DX, OFFSET MY_INT ; смещение д
	ля процедуры в DX
003D B8 ---- R	MOV AX, SEG MY_INT ; сегмент про♦
	♦еудуры
0040 8E D8	MOV DS, AX; помещаем в DS
0042 B4 25	MOV AH, 25H; функция установ♦
	♦и вектора
0044 B0 23	MOV AL, 23H; номер вектора
0046 CD 21	INT 21H; меняем прерывание
0048 1F	POP DS
0049	enter_ctrl_c:
0049 B4 00	mov ah,0
004B CD 16	int 16h
004D 3C 03	cmp al,3
004F 75 F8	jne enter_ctrl_c
0051 CD 23	INT 23H
0053 FA	CLI
0054 1E	PUSH DS
0055 8B 16 0002 R	MOV DX, KEEP_IP
0059 A1 0000 R	MOV AX, KEEP_CS
005C 8E D8	MOV DS, AX
005E B4 25	MOV AH, 25H
0060 B0 23	MOV AL, 23H
0062 CD 21	INT 21H
	; восстанавливаем стары
	й вектор прерывания
0064 1F	POP DS
0065 FB	STI



0066 B4 4C                    MOV AH, 4CH

0068 CD 21                    INT 21H

006A                         MAIN ENDP

006A                         CODE ENDS

END MAIN

#Microsoft (R) Macro Assembler Version 5.10

11/25/20 17:38:5

Symbols-1

### Segments and Groups:

<i>N a m e</i>	<i>Length</i>	<i>Align</i>	<i>Combine</i>	<i>Class</i>
ASTACK.....	0400	PARA		STACK
CODE.....	006A	PARA		NONE
DATA.....	0004	PARA		NONE

### Symbols:

<i>N a m e</i>	<i>Type</i>	<i>Value</i>	<i>Attr</i>
ENTER_CTRL_C.....	L NEAR	0049	CODE
KEEP_CS .....	L WORD	0000	DATA
KEEP_IP .....	L WORD	0002	DATA
KILL_TIME .....	L NEAR	0019	CODE
MAIN.....	F PROC	0026	CODE Length = 0044
MY_INT.....	F PROC	0000	CODE Length = 0026
@CPU.....	TEXT	0101h	
@FILENAME .....	TEXT	LAB5	

@VERSION..... TEXT 510

90 Source Lines

90 Total Lines

14 Symbols

48004 + 459256 Bytes symbol space free

0 Warning Errors

0 Severe Errors