

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

**ОТЧЕТ**  
**по лабораторной работе №5**  
**по дисциплине «Архитектура ЭВМ и систем»**  
**Тема: Разработка собственного прерывания.**

Студент гр. 9383

Моисейченко К.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Изучить работу прерываний, написать свое в соответствии с вариантом

### **Задание.**

Написать прерывание в соответствии с вариантом.

Вариант 4В:

Цифра в шифре задает номер и назначение заменяемого вектора прерывания: 4 - 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек.

Буква определяет действия, реализуемые программой обработки прерываний: В - Выдача звукового сигнала.

Замечание: для исключения возможного взаимного влияния системных и пользовательских прерываний рекомендуется отвести в программе под стек не менее 1К байт.

### **Ход работы.**

В ходе работы была реализована программа на языке Ассемблер, которая сохраняет старый вектор прерывания, устанавливает новый, вызывает его обработку и восстанавливает старый вектор прерывания.

### **Выводы.**

Были изучены механизмы работы прерываний, а также реализован собственный в соответствии с заданием.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

#### Файл Ir5.asm:

```
EOF EQU '$'

AStack SEGMENT STACK
    DB 1024 DUP(?)
AStack ENDS

DATA    SEGMENT
    KEEP_CS DW 0      ;для хранения сегмента вектора прерывания
    KEEP_IP DW 0      ;для хранения смещения вектора прерывания
DATA    ENDS

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

MY_INT PROC FAR

    jmp procedure

    KEEP_SS DW 0
    KEEP_SP DW 0
    KEEP_AX DW 0
    MY_STACK DW 1024 DUP(?)

    procedure:

MOV KEEP_SP, SP
MOV KEEP_AX, AX
MOV AX, SS
MOV KEEP_SS, AX
MOV AX, KEEP_AX
MOV SP, OFFSET procedure
MOV AX, seg MY_STACK
MOV SS, AX
```

```

    push AX
    push DX
;----- Часть кода выводющая 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 KEEP_AX, AX
    MOV SP, KEEP_SP
    MOV AX, KEEP_SS
    MOV SS, AX
    MOV AX, KEEP_AX

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

MAIN PROC FAR

    mov ax, DATA

```

```

mov ds, ax
GETVECTOR:      ;сохраняем вектор прерывания
mov ah, 35h     ;функция получения вектора
mov al, 08h     ;номер вектора
int 21h
mov KEEP_CS, es
mov KEEP_IP, bx

SETVECTOR:      ;устанавливаем новый вектор прерывания
push ds
mov dx, OFFSET MY_INT
mov ax, SEG MY_INT
mov ds, ax
mov ah, 25h ;функция установки вектора
mov al, 08h ;номер вектора
int 21h
pop ds

BUTTON_LOOP:
mov ah, 00h
int 16h
cmp al, 27
je RESTOREVECTOR
loop BUTTON_LOOP

RESTOREVECTOR:  ;восстанавливаем старый вектор прерывания
cli
push ds
mov dx, KEEP_IP
mov ax, KEEP_CS
mov ds, ax
mov ah, 25h ;функция установки вектора
mov al, 08h ;номер вектора
int 21h
pop ds
sti

mov ah, 4ch ;завершение программы

```

```

        int 21h

MAIN ENDP

CODE ENDS

END MAIN

```

## Файл lr5.asm:

Microsoft (R) Macro Assembler Version 5.10 12/22/20  
 11:35:2  
 Page  
 1-1

```

= 0024                                EOF EQU '$'

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

0400                                AStack ENDS

0000                                DATA SEGMENT
0000 0000                            KEEP_CS DW 0 ;для хранения сегмента
вект
ора прерывания
0002 0000                            KEEP_IP DW 0 ;для хранения смещения
вект
ора прерывания
0004                                DATA ENDS

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

0000                                MY_INT PROC FAR

```

```

0000  E9 0809 R          jmp procedure

0003  0000                KEEP_SS DW 0
0005  0000                KEEP_SP DW 0
0007  0000                KEEP_AX DW 0
0009  0400[                MY_STACK DW 1024 DUP(?)
    ????
]

0809                                procedure:

0809  2E: 89 26 0005 R          MOV KEEP_SP, SP
080E  2E: A3 0007 R          MOV KEEP_AX, AX
0812  8C D0                MOV AX, SS
0814  2E: A3 0003 R          MOV KEEP_SS, AX
0818  2E: A1 0007 R          MOV AX, KEEP_AX
081C  BC 0809 R          MOV SP, OFFSET procedure
081F  B8 ---- R          MOV AX, seg MY_STACK
0822  8E D0                MOV SS, AX

0824  50                push AX
0825  52                push DX
    ;----- Часть кода выводющая звук
0826  B0 B6                MOV AL , 10110110b
0828  E6 43                OUT 43H, AL; Set mode for 2nd
channel
082A  B8 03E8            MOV AX , 1000; Pitch of sound
082D  E6 42                OUT 42H, AL
082F  8A C4                MOV AL , AH
0831  E6 42                OUT 42H, AL; Set it to speaker port
0833  E4 61                IN AL, 61H
0835  8A E0                MOV AH, AL

Microsoft (R) Macro Assembler Version 5.10          12/22/20
11:35:2

```

```

0837 0C 03                                OR AL, 3
0839 E6 61                                OUT 61H, AL
083B 2B C9                                SUB CX, CX
083D                                     KILL_TIME:
083D E2 FE                                LOOP KILL_TIME
083F 8A C4                                MOV AL, AH
0841 E6 61                                OUT 61H, AL
;-----
0843 5A                                pop DX
0844 58                                pop AX

0845 2E: A3 0007 R                        MOV KEEP_AX, AX
0849 2E: 8B 26 0005 R                        MOV SP, KEEP_SP
084E 2E: A1 0003 R                        MOV AX, KEEP_SS
0852 8E D0                                MOV SS, AX
0854 2E: A1 0007 R                        MOV AX, KEEP_AX

0858 B0 20                                mov al, 20h
085A E6 20                                out 20h, al
085C CF                                iret
085D                                     MY_INT ENDP

085D                                     MAIN PROC FAR

085D B8 ---- R                        mov ax, DATA
0860 8E D8                                mov ds, ax
0862                                     GETVECTOR:                                ;сохраняем вектор
прерывани
я
0862 B4 35                                mov ah, 35h                                ;функция получения
вектора
0864 B0 08                                mov al, 08h                                ;номер вектора
0866 CD 21                                int 21h
0868 8C 06 0000 R                        mov KEEP_CS, es
086C 89 1E 0002 R                        mov KEEP_IP, bx

```



```

0870          SETVECTOR:          ;устанавливаем  новый
вектор
    прерывания
0870  1E          push ds
0871  BA 0000 R    mov dx, OFFSET MY_INT
0874  B8 ---- R    mov ax, SEG MY_INT
0877  8E D8          mov ds, ax
0879  B4 25          mov  ah,  25h ;функция  установки
вектора
087B  B0 08          mov al, 08h ;номер вектора
087D  CD 21          int 21h
087F  1F          pop ds

0880          BUTTON_LOOP:
0880  B4 00          mov ah, 00h
0882  CD 16          int 16h
0884  3C 1B          cmp al, 27
0886  74 02          je  RESTOREVECTOR
0888  E2 F6          loop BUTTON_LOOP

```

```

088A          RESTOREVECTOR:      ;восстанавливаем  старый
век

```

```

Microsoft (R) Macro Assembler Version 5.10          12/22/20
11:35:2

```

Page

1-3

```

    топ прерывания
088A  FA          cli
088B  1E          push ds
088C  8B 16 0002 R    mov dx, KEEP_IP
0890  A1 0000 R    mov ax, KEEP_CS
0893  8E D8          mov ds, ax
0895  B4 25          mov  ah,  25h ;функция  установки
вектора
0897  B0 08          mov al, 08h ;номер вектора

```

```

0899  CD 21                      int 21h
089B  1F                      pop ds
089C  FB                      sti

089D  B4 4C                  mov ah, 4ch ;завершение программы
089F  CD 21                  int 21h

```

```

08A1                      MAIN ENDP

```

```

08A1                      CODE ENDS

```

```

END MAIN

```

```

Microsoft (R) Macro Assembler Version 5.10

```

```

12/22/20

```

```

11:35:2

```

```

Symbols-1

```

```

Segments and Groups:

```

Class	N a m e	Length	Align	Combine
	ASTACK . . . . .	0400	PARA	STACK
	CODE . . . . .	08A1	PARA	NONE
	DATA . . . . .	0004	PARA	NONE

```

Symbols:

```

N a m e	Type	Value	Attr
BUTTON_LOOP . . . . .	L NEAR	0880	CODE
EOF . . . . .	NUMBER	0024	
GETVECTOR . . . . .	L NEAR	0862	CODE
KEEP_AX . . . . .	L WORD	0007	CODE

KEEP_CS . . . . .	L WORD	0000	DATA
KEEP_IP . . . . .	L WORD	0002	DATA
KEEP_SP . . . . .	L WORD	0005	CODE
KEEP_SS . . . . .	L WORD	0003	CODE
KILL_TIME . . . . .	L NEAR	083D	CODE
MAIN . . . . .	F PROC	085D	CODE Length
= 0044			
MY_INT . . . . .	F PROC	0000	CODE Length
= 085D			
MY_STACK . . . . .	L WORD	0009	CODE Length
= 0400			
PROCEDURE . . . . .	L NEAR	0809	CODE
RESTOREVECTOR . . . . .	L NEAR	088A	CODE
SETVECTOR . . . . .	L NEAR	0870	CODE
@CPU . . . . .	TEXT	0101h	
@FILENAME . . . . .	TEXT	1r5	
@VERSION . . . . .	TEXT	510	

117 Source Lines

117 Total Lines

23 Symbols

48034 + 459226 Bytes symbol space free

0 Warning Errors

0 Severe Errors