

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

**ОТЧЕТ**  
**по лабораторной работе №5**  
**по дисциплине «Организация ЭВМ и систем»**  
**ТЕМА: РАЗРАБОТКА СОБСТВЕННОГО ПРЕРЫВАНИЯ.**

Студент гр. 0382

\_\_\_\_\_

Андрющенко К.С.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Изучить работу с прерываниями на языке программирования Ассемблер. Создать собственное прерывание и обработать прерывание в соответствии с заданием.

### **Индивидуальное задание.**

Вариант 2. Шифр: 1В.

1 - 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек;

В - Выдача звукового сигнала;

### **Ход работы.**

В ходе работы была разработана программа на языке Assembler, которая обрабатывает прерывание и выдает звуковой сигнал в соответствие с индивидуальным заданием.

В процедуре main() происходит получение вектора прерывания по номеру 08h, при помощи прерывания 21h и функции 35h. Значение регистра CS данного вектора помещается в ES, IP – в BX, для того чтобы вернуть данный вектор в конце программы.

С помощью функции 25h прерывания 21h задаем новый адрес прерывания. Прерывание устанавливается на выполнение процедуры. В DX записывается смещение процедуры с созданным прерыванием, в DS (значение прерывания запоминается в стеке) – сегмент, в AL – номер прерывания.

Частота звука записывается в BX. Для того чтобы пользователь имел возможность изменять частоту звука при тестировании, программа считывает ввод с клавиатуры:

- W – увеличение частоты;
- S – уменьшение частоты;
- При нажатии другой клавиши выполнение программы завершается;

Для вывода звука сначала вызывается процедура включения звукового сигнала. Затем процедура задержки и уже потом процедура прекращения

звукового сигнала. В конце программы выполняется восстановление старого прерывания с помощью 25h и сохраненных CS и IP.

Установка канала 2 таймера – счетчика осуществляется с помощью подали значения 10110110b на порт 43h.

### **Тестирование программы.**

При запуске программы частота звука изменяется в зависимости от нажатых клавиш. Крайние случаи обработаны. При нажатии клавиши кроме s и w программа завершается.

### **Выводы.**

В ходе данной лабораторной работы было изучено устройство прерываний в программах на языке ассемблера, разработана программа, которая выполняет прерывание от системного таймера с выводом звукового сигнала.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb5.asm

```
; Стек программы
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 ; звуковое прерывание от таймера
    PUSH CX ; сохранение изменяемых регистров
    PUSH AX
    MOV AL, 10110110b
    out 43h, al
    MOV AX, BX
    OUT 42H, AL ; включение таймера, который будет выдавать импульсы на
динамик с заданной частотой
    MOV AH, AL
    OUT 42H, AL
    IN AL, 61H ; получаем состояние динамика
    OR AL, 00000011b
    OUT 61H, AL ; включить динамик
    sub cx, cx ;
    TIME: LOOP TIME ; уменьшает значение в регистре CX, если CX != 0, то
выполняется переход к метке
    ; флаги не меняются
    MOV AL, AH
    OUT 61H, AL ; выключить динамик
    ;mov save_ss, 0000h
    ;mov save_sp, 0000h
    POP AX ; восстановление регистров
    POP CX
    MOV AL, 20H
    OUT 20H, AL ; вывод на порт
    IRET
    SUBR_INT ENDP
; Головная процедура
Main PROC FAR
    MOV AH, 35H ; функция получения вектора
    MOV AL, 08h ; номер вектора прерывания в соответствии с заданием
```

```

INT 21H ; реализуется процедура прерывания
MOV KEEP_IP, BX ; запоминание смещения
MOV KEEP_CS, ES ; и сегмента вектора прерывания
MOV BX, 3000 ; высота звука
PUSH DS
MOV DX, OFFSET SUBR_INT ; смещение для процедуры в DX
MOV AX, SEG SUBR_INT ; сегмент процедуры
MOV DS, AX ; помещаем в DS
MOV AH, 25H ; функция установки вектора?
MOV AL, 08h ; номер вектора
INT 21H ; меняем прерывание
POP DS
; Команды для проверки работы программы
inputKeyBoard:
    mov ah, 0h
    int 16h
    cmp al, 'w'
    je UpVol
    cmp al, 's'
    je DownVol
    jmp endInput
UpVol:
    cmp bx, 1000
    jge inputKeyBoard
    add bx, 500
    jmp inputKeyBoard
DownVol:
    cmp bx, 500
    jle inputKeyBoard
    sub bx, 500
    jmp inputKeyBoard

endInput:
    CLI ; Сброс флага прерываний IF - 0
    PUSH DS
    MOV DX, KEEP_IP
    MOV AX, KEEP_CS
    MOV DS, AX
    MOV AH, 25H
    MOV AL, 08h ;
    INT 21H ; восстанавливаем старый вектор прерывания
    POP DS
    STI
    MOV AH, 4Ch
    INT 21h
Main ENDP
CODE ENDS
END Main

```

## ПРИЛОЖЕНИЕ Б

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb5.lst

Microsoft (R) Macro Assembler Version 5.10

12/2/21 03:59:08

Page

1-1

```

; Стек программы
0000 AStack SEGMENT STACK
0000 0200[ DW 512 DUP(?) ; выделим 1 Кбай
т
        памяти
        ????
        ]

0400 AStack ENDS
; Данные программы
0000 DATA SEGMENT
0000 0000 KEEP_CS DW 0 ; для хранения се
г
        мента
0002 0000 KEEP_IP DW 0 ; и смещения вект
о
        ра прерывания
        ;save_ss dw 0000h
        ;save_sp dw 0000h
        ;ind_stack dw 512 DUP(?)
0004 DATA ENDS
; Код программы
0000 CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack
0000 SUBR_INT PROC FAR ; звуковое преры
        вание от таймера
        ;jmp h_start
        ;h_start:
        ;mov save_ss, SS
        ;mov save_sp, sp
        ;mov sp, seg ind_stack
        ;mov ss, sp
        ;mov sp, OFFSET h_start
0000 51 PUSH CX ; сохранение изменяе
        мых регистров
0001 50 PUSH AX

0002 B0 B6 MOV AL, 10110110b
0004 E6 43 out 43h, al
0006 8B C3 MOV AX, BX
0008 E6 42 OUT 42H, AL ; включение тайме
р

```

а, который будет выдавать  
импульсы на динамик с зад❖

## ◆нной частотой

```
000A 8A E0      MOV AH, AL
000C E6 42      OUT 42H, AL
000E E4 61      IN AL, 61H ; получаем состоян
```

◆

## ◆е динамика

```
0010 0C 03      OR AL, 00000011b
0012 E6 61      OUT 61H, AL ; включить динами
```

к

```
0014 2B C9      sub cx, cx ;
0016 E2 FE      TIME: LOOP TIME ; уменьшает зна
```

ч

ение в регистре CX, если C◆

◆ != 0, то выполняется перех◆

◆д к метке

Microsoft (R) Macro Assembler Version 5.10

12/2/21 03:59:08

Page

1-2

; флаги не меняются

```
0018 8A C4      MOV AL, AH
001A E6 61      OUT 61H, AL ; выключить динам
```

и

к

```
      ;mov save_ss, 0000h
      ;mov save_sp, 0000h
001C 58      POP AX ; восстановление рег◆
```

◆с тр о в

```
POP CX
001D 59      MOV AL, 20H
001E B0 20      OUT 20H, AL ; вывод на порт
0020 E6 20
0022 CF      IRET
0023      SUBR_INT ENDP
```

; Головная процедура

```
0023      Main PROC FAR
```

```
0023 B4 35      MOV AH, 35H ; функция получен
```

и

я вектора

```
0025 B0 08      MOV AL, 08h ; номер вектора пр
```

◆

◆рывания в соответствии с заданием

```
0027 CD 21      INT 21H ; реализуется проце
```

д

ура прерывания

```
0029 89 1E 0002 R      MOV KEEP_IP, BX ; запоминание с
```

м

ещения

```

002D 8C 06 0000 R      MOV KEEP_CS, ES ; и сегмента ве
к ?
0031 BB 0BB8          MOV BX, 3000 ; высота звука
0034 1E              PUSH DS
0035 BA 0000 R      MOV DX, OFFSET SUBR_INT ; смещение д
ля процедуры в DX
0038 B8 ---- R      MOV AX, SEG SUBR_INT ; сегмент про?
? е д у р ы
003B 8E D8          MOV DS, AX ; помещаем в DS
003D B4 25          MOV AH, 25H ; функция установ
к
и вектора?
003F B0 08          MOV AL, 08h ; номер вектора
0041 CD 21          INT 21H ; меняем прерывание
0043 1F              POP DS
; Команды для проверки раб
оты программы
0044          inputKeyBoard:
0044 B4 00          mov ah, 0h
0046 CD 16          int 16h
0048 3C 77          cmp al, 'w'
004A 74 07          je UpVol
004C 3C 73          cmp al, 's'
004E 74 0F          je DownVol
0050 EB 19 90      jmp endInput
0053          UpVol:
0053 81 FB 03E8      cmp bx, 1000
0057 7D EB          jge inputKeyBoard
0059 81 C3 01F4      add bx, 500
005D EB E5          jmp inputKeyBoard

```

Microsoft (R) Macro Assembler Version 5.10

12/2/21 03:59:08

Page

1-3

```

005F          DownVol:
005F 81 FB 01F4      cmp bx, 500
0063 7E DF          jle inputKeyBoard
0065 81 EB 01F4      sub bx, 500
0069 EB D9          jmp inputKeyBoard

006B          endInput:
006B FA          CLI ; Сброс флага прерыван?
? й IF - 0
006C 1E              PUSH DS
006D 8B 16 0002 R    MOV DX, KEEP_IP
0071 A1 0000 R      MOV AX, KEEP_CS
0074 8E D8          MOV DS, AX
0076 B4 25          MOV AH, 25H
0078 B0 08          MOV AL, 08h ;
007A CD 21          INT 21H ; восстанавливаем с

```

т

арый вектор прерывания



```

007C 1F          POP DS
007D FB          STI
007E B4 4C      MOV AH, 4Ch
0080 CD 21      INT 21h
0082           Main ENDP
0082           CODE ENDS
           END Main

```

Microsoft (R) Macro Assembler Version 5.10

12/2/21 03:59:08

Symbols-1

# Segments and Groups:

	N a m e	Length	Align	Combine Class
ASTACK	. . . . .	0400	PARA	STACK
CODE	. . . . .	0082	PARA	NONE
DATA	. . . . .	0004	PARA	NONE

# Symbols:

	N a m e	Type	Value	Attr
	DOWNVOL . . . . .	L NEAR	005F	CODE
	ENDINPUT . . . . .	L NEAR	006B	CODE
	INPUTKEYBOARD . . . . .	L NEAR	0044	CODE
	KEEP_CS . . . . .	L WORD	0000	DATA
	KEEP_IP . . . . .	L WORD	0002	DATA
005F	MAIN . . . . .	F PROC	0023	CODE Length =
0023	SUBR_INT . . . . .	F PROC	0000	CODE Length =
	TIME . . . . .	L NEAR	0016	CODE
	UPVOL . . . . .	L NEAR	0053	CODE
	@CPU . . . . .	TEXT	0101h	
	@FILENAME . . . . .	TEXT	1b5	
	@VERSION . . . . .	TEXT	510	

```

103 Source Lines
103 Total Lines
17 Symbols

```

48034 + 457176 Bytes symbol space free

```

0 Warning Errors
0 Severe Errors

```