

**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. 9382

\_\_\_\_\_

Павлов Р.В.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Реализовать собственное прерывание, переопределив обработчик прерывания для одного из векторов прерываний.

### **Задание (вариант 2, 1В).**

1Ch - прерывание от часов - генерируется автоматически операционной системой 18 раз в секунду, выдача звукового сигнала.

### **Ход работы.**

- 1) Написана процедура, настраивающая таймер на канал вывода на динамик и подающая на выход частоту звука, которая рассчитывается, исходя из значения в регистре ВХ.
- 2) Главная процедура реализует замену обработчика вектора прерывания 1Ch, включение и выключение динамика, цикл с убывающим значением регистра ВХ, которое используется для расчёта частоты звука, выводимого динамиком (является делителем частоты таймера), а также восстановление исходного обработчика.

### **Выводы.**

В результате выполнения лабораторной работы был написан собственный обработчик прерывания, выполняющий вывод звука на динамик.

## ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД

- имя файла : lab5.asm

```
stack segment stack
    dw 6 dup(?)                ; вполне достаточно
stack ends

data segment
    keep_seg dw 0              ; здесь сохраним дефолтный
    вектор
    keep_offset dw 0
data ends

code segment
    assume ds:data, cs:code, ss:stack

    interrupt proc far

        push ax                ; AX будем менять,
    сохраняем на стеке

        mov al, 182            ; настраиваем таймер
    out 43h, al

        mov ax, bx             ; передаём делитель частоты
    out 42h, al
    mov al, ah
    out 42h, al

        mov al, 20h            ; возвращаем возможность
    управления другим прерываниям (с более низким приоритетом)
    out 20h, al

        pop ax                 ; восстанавливаем AX
    iret
    interrupt endp
```

```

main proc far
    push ds
    sub ax, ax                ; инициализируем сегмент дан-
ных
    push ax

    mov ax, data
    mov ds, ax

    mov ax, 351ch            ; получаем вектор прерывания,
сохраняем в памяти
    int 21h

    mov keep_offset, bx
    mov keep_seg, es

    ;-----
    cli
    push ds
    mov dx, offset interrupt ; устанавливаем новый обработчик
прерывания
    mov ax, seg interrupt
    mov ds, ax

    mov ax, 251ch
    int 21h

    pop ds
    sti

    ;-----
    in al, 61h              ; включаем звук
    mov ah, al
    or al, 3
    mov bx, 12000           ; начальный делитель частоты
    out 61h, al

    za:

```

```

        mov cx, 0ffffh                ; ставим задержку на очень
МНОГО ТАКТОВ                          ;

        loopер:
        loop loopер

        sub bx, 100
        cmp bx, 1500                  ; уменьшаем делитель частоты
на 100 (частота звука тем временем растёт)
        jg за

        xor bx, bx

        mov al, ah                    ; теперь надо выключить, об-
щара спит
        out 61h, al

        ;-----

        cli
        push ds

        mov dx, keep_offset           ; восстанавливаем исходный
вектор прерывания
        mov ax, keep_seg
        mov ds, ax

        mov ah, 25h
        mov al, 1ch
        int 21h

        pop ds
        sti

        ;-----

        ret

    main endp
code ends
end main

```

## ПРИЛОЖЕНИЕ Б. ТЕКСТ ЛИСТИНГОВ

- имя файла: lab5.lst

#Microsoft (R) Macro Assembler Version 5.10

11/26/20 01:05:3

Page 1-1

```
0000          stack segment stack
0000 0006[          dw 6 dup(?)
                        ; вполне достаточно
        ????
        ]

000C          stack ends

0000          data segment
0000 0000          keep_seg dw 0
                        ; здесь сохраним дефолтный вект
ор
0002 0000          keep_offset dw 0
0004          data ends

0000          code segment
          assume ds:data, cs:code, ss:stack

0000          interrupt proc far

0000 50          push ax
                        ; AX будем менять, сохр
аняем на стеке

0001 B0 B6          mov al, 182
                        ; настраиваем таймер
0003 E6 43          out 43h, al

0005 8B C3          mov ax, bx
                        ; передаём делитель час
тоты
0007 E6 42          out 42h, al
0009 8A C4          mov al, ah
000B E6 42          out 42h, al

000D B0 20          mov al, 20h
                        ; возвращаем возможност
ь управления другим прерываниям (с более низким
приоритетом)
000F E6 20          out 20h, al

0011 58          pop ax
                        ; восстанавливаем AX
0012 CF          iret
0013          interrupt endp

0013          main proc far
0013 1E          push ds
0014 2B C0          sub ax, ax
                        ; инициализируем сегмен
```

```

                                Т данных
0016 50                                push ax

#Microsoft (R) Macro Assembler Version 5.10                                11/26/20 01:05:3
                                           Page      1-2

0017 B8 ---- R                                mov ax, data
001A 8E D8                                mov ds, ax

001C B8 351C                                mov ax, 351ch
                                           ; получаем вектор прерывания, с
                                           охраняем в памяти
001F CD 21                                int 21h

0021 89 1E 0002 R                                mov keep_offset, bx
0025 8C 06 0000 R                                mov keep_seg, es
                                           ;-----
0029 FA                                cli
002A 1E                                push ds
002B BA 0000 R                                mov dx, offset interrupt
                                           ; устанавливаем новый обработчик прерыв
                                           ания
002E B8 ---- R                                mov ax, seg interrupt
0031 8E D8                                mov ds, ax

0033 B8 251C                                mov ax, 251ch
0036 CD 21                                int 21h

0038 1F                                pop ds
0039 FB                                sti
                                           ;-----

003A E4 61                                in al, 61h
                                           ; включаем звук
003C 8A E0                                mov ah, al
003E 0C 03                                or al, 3
0040 BB 2EE0                                mov bx, 12000
                                           ; начальный делитель частоты
0043 E6 61                                out 61h, al

0045                                za:
0045 B9 FFFF                                mov cx, 0ffffh
                                           ; ставим задержку на очень мног
                                           о тактов

0048                                loopер:
0048 E2 FE                                loop loopер

004A 83 EB 64                                sub bx, 100
004D 81 FB 05DC                                cmp bx, 1500
                                           ; уменьшаем делитель частоты на
                                           100 (частота звука тем временем растёт)
0051 7F F2                                jg za

0053 33 DB                                xor bx, bx

0055 8A C4                                mov al, ah
                                           ; теперь надо выключить

```

```

                                , общага спит
0057  E6 61                                out 61h, al
                                ;-----
0059  FA                                cli
005A  1E                                push ds
005B  8B 16 0002 R                        mov dx, keep_offset
                                ; восстанавливаем исходный вект
                                ор прерывания
005F  A1 0000 R                        mov ax, keep_seg
0062  8E D8                        mov ds, ax
0064  B4 25                        mov ah, 25h
0066  B0 1C                        mov al, 1ch
0068  CD 21                        int 21h
006A  1F                        pop ds
006B  FB                        sti
                                ;-----
006C  CB                                ret
006D                                main endp
006D                                code ends
                                end main

```

## Segments and Groups:

N a m e	Length	Align	Combine Class
CODE . . . . .	006D	PARA	NONE
DATA . . . . .	0004	PARA	NONE
STACK . . . . .	000C	PARA	STACK

## Symbols:

N a m e	Type	Value	Attr
INTERRUPT . . . . .	F PROC	0000	CODE Length = 0013
KEEP_OFFSET . . . . .	L WORD	0002	DATA
KEEP_SEG . . . . .	L WORD	0000	DATA
LOOPER . . . . .	L NEAR	0048	CODE
MAIN . . . . .	F PROC	0013	CODE Length = 005A
ZA . . . . .	L NEAR	0045	CODE
@CPU . . . . .	TEXT	0101h	



```
@FILENAME . . . . . TEXT lab5
@VERSION . . . . . TEXT 510
```

```
103 Source Lines
103 Total Lines
14 Symbols
```

```
48020 + 461287 Bytes symbol space free
```

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