

**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. 0383

\_\_\_\_\_

Сабанов П.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Написать программу, заменяющую обработчик прерывания от системного таймера на обработчик, выводящий звуковой сигнал. По завершении программа должна восстановить старый обработчик прерывания.

### **Ход работы.**

Была написана функция `interfunction`, являющаяся обработчиком прерывания. Она выводит звук частоты 100.

Была написана функция `disable_sound`, отключающая звук.

В главной функции происходит считывание текущего обработчика прерывания таймера (номер вектора 08h) и установка нового обработчика прерывания (функция `interfunction`). Затем программа ждёт нажатия пользователем любой кнопки клавиатуры. После этого она возвращает старый обработчик прерывания, вызывает функцию `disable_sound` и завершается.

### **Выводы.**

Был написан обработчик прерывания `interfunction`, издающий звук.

Была написана программа, издающая звук при прерывании от системного таймера. Перед завершением программа возвращает старый обработчик прерывания и отключает звук, который мог включить `interfunction`.

## ПРИЛОЖЕНИЕ А

### Исходный код программы

DOSSEG

.MODEL SMALL

.STACK 100h

.DATA

keep\_cs dw 0

keep\_ip dw 0

.CODE

; номер вектора прерывания

vector\_n db 08h

; функция-обработчик прерывания

; void interruption();

interfunction proc far

; сохраняем регистры

push ax

push bx

push cx

push dx

; выводим звук

mov cx, 100 ; частота

in al, 61h ; получаем значение из управляющего регистра порта В PPI (контроллера 8255)

or al, 3 ; устанавливаем биты 0 и 1 (включить спикер и использовать 2-й канал для генерации импульсов спикера)

out 61h, al ; выводим значение в управляющий регистр

mov al, 10110110b ; управляющее слово таймера

out 43h, al ; выводим значение в порт таймера

mov dx, 12h

mov ax, 34ddh ;  $DX:AX = 1193181$  - частота работы таймера

div cx ; значение счётчика таймера  $AX = DX:AX / CX$

out 42h, al ; выводим младший байт счётчика во 2-й канал таймера

mov al, ah

out 42h, al ; выводим старший байт

; разрешение обработки прерываний с более низкими уровнями, чем только что обработанное

mov al, 20h

```

out 20h, al

; восстанавливаем регистры
pop dx
pop cx
pop bx
pop ax

iret

interfunction endp

disable_sound proc near
    push ax
    in al, 61h
    and al, not 3
    out 61h, al
    pop ax
    ret
disable_sound endp

main proc far

    mov ax, @data
    mov ds, ax

; сохраняем функцию прерывания
    mov ah, 35h ; функция получения вектора
    mov al, vector_n ; номер вектора
    int 21h
    mov keep_ip, bx ; запоминание смещения
    mov keep_cs, es ; запоминание вектора прерывания

; устанавливаем нашу функцию прерывания
    push ds
    mov dx, offset interfunction
    mov ax, seg interfunction
    mov ds, ax
    mov ah, 25h ; функция установки вектора
    mov al, vector_n
    int 21h
    pop ds

; ждём нажатия клавиши
    mov ah, 0
    int 16h

```

```

; возвращаем сохранённую функцию прерывания
cli
push ds
mov dx, keep_ip
mov ax, keep_cs
mov ds, ax
mov ah, 25h ; функция установки вектора
mov al, vector_n
int 21h
pop ds
sti

; выключаем звук
call disable_sound

; выход из программы
mov ah, 4ch
xor al, al
int 21h

main endp

end main

```

## ПРИЛОЖЕНИЕ Б

## Листинг компиляции программы

12/26/21 12:55:2

1-1

DOSSEG

MODEL SMALL

```
.STACK 100h
```

.DATA

```
keep_cs dw 0
```

```
keep_ip dw 0
```

.CODE

$$; \mathbb{D} \alpha \mathbb{D} \ddot{Y} \mathbb{D} \mathbb{C} \mathbb{D} \mu \tilde{N} \square \mathbb{D}^2 \mathbb{D} \mu \mathbb{D}^{\circ} \tilde{N} \square \mathbb{D} \ddot{Y} \tilde{N} \square \mathbb{D}^{\circ} \mathbb{D} ; \tilde{N} \square \mathbb{D} \mu \tilde{N} \square \tilde{N} \square \mathbb{D}^2 \mathbb{D}^{\circ} \mathbb{D} \alpha \mathbb{D} \dot{Z} \tilde{N}$$

•

vector\_n db 08h

$$\begin{aligned} & \cdot \tilde{N} \square \tilde{N} \square \mathcal{D} \alpha \mathcal{D}^{\circ} \tilde{N} \square \mathcal{D} \tilde{z} \tilde{N} \square - \mathcal{D} \tilde{Y} \mathcal{D} \pm \tilde{N} \square \mathcal{D}^{\circ} \mathcal{D} \pm \mathcal{D} \tilde{Y} \tilde{N} \square \tilde{N} \square \mathcal{D} \tilde{z} \mathcal{D}^{\circ} \mathcal{D}_{\zeta} \tilde{N} \square \mathcal{D}_{\mu} \tilde{N} \square \tilde{N} \\ & \square \mathcal{D}^2 \mathcal{D}^{\circ} \mathcal{D} \alpha \mathcal{D} \tilde{z} \tilde{N} \square \end{aligned}$$

```
; void interruption();
```

interfunction proc far

$$; \tilde{N} \square \mathfrak{D} \breve{Y} \tilde{N} \square \tilde{N} \square \mathfrak{D}^{\circ} \mathfrak{D} \alpha \tilde{N} \square \mathfrak{D} \mu \mathfrak{D} \mathfrak{E} \tilde{N} \square \mathfrak{D} \mu \mathfrak{D}^3 \mathfrak{D} \breve{z} \tilde{N} \square \tilde{N} \square \tilde{N} \square \tilde{N} \square$$

push ax

push bx

push cx

push dx

$$; \mathbb{D}^2 \tilde{N} \square \mathbb{D}^2 \mathbb{D} \check{Y} \mathbb{D} \check{Z} \mathbb{D} \check{z} \mathbb{D} \mathbb{C} \mathbb{D} \cdot \mathbb{D}^2 \tilde{N} \square \mathbb{D}^0$$

```
mov cx, 100 ; ̃◻Đ°̃◻̃◻ĐŸ̃◻Đ°
```

in al, 6lh ; ĐđĐŸĐ»Ń□ŃĐ°ĐμĐĐЄ Đ·ĐœĐ°Ń□ĐμĐœĐ  
žĐμ ĐžĐ· Ń□ĐđŃ□Đ°Đ²Đ»Ń□Ń□Ń□ĐμĐ³ĐŸ Ń□ĐμĐ³ĐžŃ□Ń□Ń  
□Đ° ĐđĐŸŃ□Ń□Đ° B PPI (Đ°ĐŸĐœŃ□Ń□ĐŸĐ»Đ»ĐμŃ□Đ° 82  
55)

or al, 3 ;  $\tilde{N} \square \tilde{N} \square \tilde{N} \square \mathbb{D}^{\circ} \mathbb{D} \mathbb{C} \mathbb{E} \mathbb{D}^{\circ} \mathbb{D}^2 \mathbb{D} \gg \mathbb{D} \mathbb{Z} \mathbb{D}^2 \mathbb{D}^{\circ} \mathbb{D} \mu \mathbb{D} \mathbb{C} \mathbb{E} \mathbb{D} \pm \mathbb{D} \mathbb{Z} \tilde{N}$

```

                                □Ń□ 0 Đž 1 (Đ²Đ°Đ»Ń□Ń□ĐžŃ□Ń□ Ń□ĐġĐžĐ°ĐμŃ□ Đž Đž
                                Ń□ĐġĐŸĐ»Ń□Đ·ĐŸĐ²Đ°Ń□Ń□ 2-Đ¹ Đ°Đ°ĐœĐ°Đ» ĐžĐ»Ń• Đ
                                ³ĐμĐœĐμŃ□Đ°Ń□ĐžĐž ĐžĐœĐġŃ□Đ»Ń□Ń□ĐŸĐ² Ń□ĐġĐžĐ°Đμ
                                ŃœĐ°)
000C E6 61                    out 61h, al ; Đ²Ń□Đ²ĐŸĐžĐžĐœ Đ·ĐœĐ°Ń□ĐμĐœĐž
                                Đμ Đ² Ń□ĐġŃ□Đ°Đ²Đ»Ń□Ń□Ń□ĐžĐ¹ Ń□ĐμĐ³ĐžŃ□Ń□Ń□
000E B0 B6                    mov al, 10110110b ; Ńf ĐġŃœĐ°Đ²Đ»Ń• ŃžŃ%ĐμĐμ
                                Ń□Đ»ĐŸĐ²ĐŸ Ń□Đ°Đ¹ĐœĐμŃ□Đ°
0010 E6 43                    out 43h, al ; Đ²Ń□Đ²ĐŸĐžĐžĐœ Đ·ĐœĐ°Ń□ĐμĐœĐž
                                Đμ Đ² ĐġĐŸŃ□Ń□ Ń□Đ°Đ¹ĐœĐμŃ□Đ°
0012 BA 0012                  mov dx, 12h
0015 B8 34DD                  mov ax, 34ddh ; DX:AX = 1193181 - Ń‡ Đ°Ń• Ń, Đ
                                ŸŃ□Đ° Ń□Đ°Đ±ĐŸŃ□Ń□ Ń□Đ°Đ¹ĐœĐμŃ□Đ°
0018 F7 F1                    div cx ; Đ·ĐœĐ°Ń□ĐμĐœĐžĐμ Ń□Ń□Ń□Ń□Ń□ĐžĐ°Đ°
                                Ń□Đ°Đ¹ĐœĐμŃ□Đ° AX = DX:AX / CX
001A E6 42                    out 42h, al ; Đ²Ń□Đ²ĐŸĐžĐžĐœ ĐœĐ»Đ°ĐžŃ□ĐžĐ¹
                                Đ±Đ°Đ¹Ń□ Ń□Ń□ĐμŃ□Ń□ĐžĐ°Đ° Đ²ĐŸ 2-Đ¹ Đ°Đ°ĐœĐ°Đ»
                                Ń□Đ°Đ¹ĐœĐμŃ□Đ°
001C 8A C4                    mov al, ah
#Microsoft (R) Macro Assembler Version 5.10      12/26/21 12:55:2
                                Page 1-2

001E E6 42                    out 42h, al ; Đ²Ń□Đ²ĐŸĐžĐžĐœ Ń□Ń□Đ°Ń□Ń□ĐžĐ¹
                                Đ±Đ°Đ¹Ń,

                                ; Ń□Đ°Đ·Ń□ĐμŃ□ĐμĐœĐžĐμ ĐŸĐ±Ń□Đ°Đ±ĐŸŃ□Đ°Đž Đ
                                ġŃ□ĐμŃ□Ń□Đ²Đ°ĐœĐžĐ¹ Ń□ Đ±ĐŸĐ»ĐμĐμ ĐœĐžĐ·Đ°ĐžĐœĐ
                                ž Ń□Ń□ĐŸĐ²ĐœŃ□ĐœĐž, Ń□ĐμĐœ Ń□ĐŸĐ»Ń□Đ°ĐŸ Ń□Ń□ĐŸ
                                ĐŸĐ±Ń□Đ°Đ±ĐŸŃ□Đ°ĐœĐœĐŸĐμ

0020 B0 20                    mov al, 20h
0022 E6 20                    out 20h, al

                                ; Đ²ĐŸŃ□Ń□Ń□Đ°ĐœĐ°Đ²Đ»ĐžĐ²Đ°ĐμĐœ Ń□ĐμĐ³ĐžŃ□
                                Ń, ŃœŃ«

0024 5A                        pop dx
0025 59                        pop cx
0026 5B                        pop bx
0027 58                        pop ax

0028 CF                        ired

0029                            interfunction endp

0029                            disable_sound proc near

```

```

0029 50                push ax
002A E4 61             in al, 61h
002C 24 FC             and al, not 3
002E E6 61             out 61h, al
0030 58                pop ax
0031 C3                ret
0032                  disable_sound endp

0032                  main proc far

0032 B8 ---- R         mov ax, @data
0035 8E D8             mov ds, ax

; Ĥ□ĐŸŨ□Ũ□Đ°ĐœŨ□ĐμĐœ Ũ□Ũ□ĐœĐ°Ũ□ĐžŨ□ ĐĭŨ□ĐμŨ
□Ũ□Đ²Đ°ĐœĐžŨ□
0037 B4 35             mov ah, 35h ; Ũ□Ũ□ĐœĐ°Ũ□ĐžŨ□ ĐĭĐŸĐ»Ũ□Ũ□ĐμĐœ
ĐžŨ□ Đ²ĐμĐ°Ũ□ĐŸŨ□Đ°
0039 2E: A0 0000 R     mov al, vector_n ; ĐœĐŸĐœĐμŨ□ Đ²ĐμĐ°Ũ□ĐŸŨ□Đ
°
003D CD 21             int 21h
003F 89 1E 0002 R     mov keep_ip, bx ; Đ·Đ°ĐĭĐŸĐœĐžĐœĐ°ĐœĐžĐμ Ũ•
ĐœĐμŨ□ĐμĐœĐžŨ□
0043 8C 06 0000 R     mov keep_cs, es ; Đ·Đ°ĐĭĐŸĐœĐžĐœĐ°ĐœĐžĐμ Đ²
ĐμĐ°Ũ□ĐŸŨ□Đ° ĐĭŨ□ĐμŨ□Ũ□Đ²Đ°ĐœĐžŨ□

; Ũ□Ũ□Ũ□Đ°ĐœĐ°Đ²Đ»ĐžĐ²Đ°ĐμĐœ ĐœĐ°Ũ□Ũf Ũ,, Ũf Đ
œĐ°Ũ□ĐžŨ□ ĐĭŨ□ĐμŨ□Ũ□Đ²Đ°ĐœĐžŨ□
0047 1E                push ds
0048 BA 0001 R         mov dx, offset interfunction
004B B8 ---- R         mov ax, seg interfunction
004E 8E D8             mov ds, ax
0050 B4 25             mov ah, 25h ; Ũ□Ũ□ĐœĐ°Ũ□ĐžŨ□ Ũ□Ũ□Ũ□Đ°ĐœĐŸĐ²
#Microsoft (R) Macro Assembler Version 5.10      12/26/21 12:55:2
Page 1-3

Đ°Đž Đ²ĐμĐ°Ũ□ĐŸŨ□Đ°
0052 2E: A0 0000 R     mov al, vector_n
0056 CD 21             int 21h
0058 1F                pop ds

; Đ¶ĐžŨ□Đœ ĐœĐ°Đ¶Đ°Ũ□ĐžŨ□ Đ°Đ»Đ°Đ²ĐžŨ□Đž
0059 B4 00             mov ah, 0
005B CD 16             int 16h

```



	; D²ĐŸĐ·D²Ñ□Đ°Ñ□Đ°ĐμĐЄ Ñ□ĐŸÑ□Ñ□Đ°ĐœÑ□ĐœĐœÑ□
	Ñ□ Ñ□Ñ□ĐœĐ°Ñ□ĐžÑ□ ĐġÑ□ĐμÑ□Ñ□Đ²Đ°ĐœĐžÑ□
005D FA	cli
005E 1E	push ds
005F 8B 16 0002 R	mov dx, keep_ip
0063 A1 0000 R	mov ax, keep_cs
0066 8E D8	mov ds, ax
0068 B4 25	mov ah, 25h ; Ñ□Ñ□ĐœĐ°Ñ□ĐžÑ□ Ñ□Ñ□Ñ□Đ°ĐœĐŸĐ² Đ°Đž Đ²ĐμĐ°Ñ□ĐŸÑ□Đ°
006A 2E: A0 0000 R	mov al, vector_n
006E CD 21	int 21h
0070 1F	pop ds
0071 FB	sti

INTERFUNCTION ..... F PROC 0001 \_TEXT Length = 0028

KEEP\_CS ..... L WORD0000 \_DATA

KEEP\_IP ..... L WORD0002 \_DATA

MAIN ..... F PROC 0032 \_TEXT Length = 0049

VECTOR\_N ..... L BYTE 0000 \_TEXT

@CODE ..... TEXT \_TEXT

@CODESIZE ..... TEXT 0

@CPU ..... TEXT 0101h

@DATASIZE ..... TEXT 0

@FILENAME ..... TEXT lab5

@VERSION ..... TEXT 510

112 Source Lines

112 Total Lines

23 Symbols

47952 + 457258 Bytes symbol space free

0 Warning Errors

0 Severe Errors