

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.

Вариант 11.

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

keep_ss dw 0

keep_sp dw 0

.CODE

; номер вектора прерывания

vector_n db 08h

; функция-обработчик прерывания

; void interruption();

interfunction proc far

mov keep_ss, ss

mov keep_sp, sp

; 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 счетчика 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

mov sp, keep_sp
mov ss, keep_ss

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

```
#Microsoft (R) Macro Assembler Version 5.10
```

12/26/21 21:00:2

Page 1-1

DOSSEG

```
.MODEL SMALL
```

```
.STACK 100h
```

.DATA

```
0000 0000 keep_cs dw 0
```

```
0002  0000                keep_ip dw 0
```

```
0004  0000          keep_ss dw 0
```

```
0006  0000                keep_sp dw 0
```

```
. CODE
```

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

```
0000 08                                vector_n db 08h
```

; Ñ Õ Ñ † ÐœÐ°ÑŸÐžÑ -ÐŸÐ±Ñ † Ð°Ð±ŸÑ † Ñ ‡ ÐžÐ° ÐžÑ † ÐµÑ † Ñ
Ð²Ð°ÐœŽÑ

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

```
0001      interfunction proc far
```

```
0001  8C 16 0004 R      mov keep_ss, ss
```

```
0005  89 26 0006 R      mov keep_sp, sp
```

; Ñ Ꞓ ĐỠÑꝯÑ Ꞓ Ð°ÐæÑꜦÐµÐƐ Ñ Ꞓ ÐµÐ³ÐžÑ Ꞓ Ꞓ Ꞓ Ꞓ Ꞓꝳ

```
0009 50          push ax
```

```
000A 53          push bx
```

```
000B  51      push cx
```

```
000C 52          push dx
```

$$; \mathfrak{D}^2 \tilde{N} \mathfrak{D}^2 \mathfrak{D} \ddot{Y} \mathfrak{D} \check{Z} \mathfrak{D} \check{Z} \mathfrak{D} \mathfrak{C} \quad \mathfrak{D} \cdot \mathfrak{D}^2 \tilde{N} \mathfrak{D} \mathfrak{O}$$

```
000D B9 0064      mov cx, 100 ;
```

```
0010    E4 61                in al, 61h ; 0;0Y0»Ñ;Ñ;0°0μ0E 0·0æ0°Ñ;0μ0æ0
```

žĐµ ĐžĐ· Ñ†ĐžÑ†Đ°Đ²Đ»ÑÑÑÑ†ĐµĐ³ĐŸ Ñ†ĐµĐ³ĐžÑ†Ñ†Ñ
(Đ° ĐžĐŸÑ(ÑĐ° B PPI (Đ°ĐŸĐœÑ(ÑĐŸĐ»Đ»ĐµÑ(Đ° 82
55)

0012 0C 03 or al, 3 ; Ñ†Ñ†Ñ†Đ°ĐœĐ°Đ²Đ»ĐžĐ²Đ°ĐµĐœ Đ±ĐžÑ
(Ñ) 0 Đž 1 (Đ²Đ°Đ»ÑÑÑĐžÑ(Ñ†Ñ)ĐžĐžĐ°ĐµÑ(Đž Đž
Ñ†ĐžĐŸĐ»ÑœĐ·ĐŸĐ²Đ°Ñ†Ñœ 2-Đ¹ Đ°Đ°ĐœĐ°Đ» ĐžĐ»Ñœ Đ
³ĐµĐœĐµÑ(Đ°Ñ(ĐžĐž ĐžĐœĐžÑ)Đ»Ñ(Ñ)ĐŸĐ² Ñ)ĐžĐžĐ°Đµ
Ñ(Đ°)

0014 E6 61 out 61h, al ; Đ²ÑĐ²ĐŸĐžĐžĐœ Đ·ĐœĐ°Ñ»ĐµĐœĐž
Đµ Đ² Ñ†ĐžÑ†Đ°Đ²Đ»ÑÑÑÑ†ĐžĐ¹ Ñ†ĐµĐ³ĐžÑ†Ñ†Ñ†

0016 B0 B6 mov al, 10110110b ; Ñ†ĐžÑ†Đ°Đ²Đ»ÑÑÑÑ†ĐµĐµ
Ñ†Đ»ĐŸĐ²ĐŸ Ñ†Đ°Đ¹ĐœĐµÑ†Đ°

0018 E6 43 out 43h, al ; Đ²ÑĐ²ĐŸĐžĐžĐœ Đ·ĐœĐ°Ñ»ĐµĐœĐž
Đµ Đ² ĐžĐŸÑ(Ñ†ÑĐ°Đ¹ĐœĐµÑ(Đ°

001A BA 0012 mov dx, 12h

001D B8 34DD mov ax, 34ddh ; DX:AX = 1193181 - Ñ»Đ°Ñ(ÑĐ
ŸÑ(Đ° Ñ(Đ°Đ±ĐŸÑ(Ñ) Ñ(Đ°Đ¹ĐœĐµÑ(Đ°

#Microsoft (R) Macro Assembler Version 5.10

12/26/21 21:00:2

Page 1-2

0020 F7 F1 div cx ; Đ·ĐœĐ°Ñ»ĐµĐœĐžĐµ Ñ)Ñ»ÑœÑ(Ñ»ĐžĐ°Đ°
Ñ(Đ°Đ¹ĐœĐµÑ(Đ° AX = DX:AX / CX

0022 E6 42 out 42h, al ; Đ²ÑĐ²ĐŸĐžĐžĐœ ĐœĐ»Đ°ĐžÑ(ĐžĐ¹
Đ±Đ°Đ¹Ñ(Ñ)Ñ»ĐµÑ(Ñ»ĐžĐ°Đ° Đ²ĐŸ 2-Đ¹ Đ°Đ°ĐœĐ°Đ»
Ñ(Đ°Đ¹ĐœĐµÑ(Đ°

0024 8A C4 mov al, ah

0026 E6 42 out 42h, al ; Đ²ÑĐ²ĐŸĐžĐžĐœ ÑÑ(Đ°Ñ(ÑœĐžĐ¹
Đ±Đ°Đ¹Ñ(

; Ñ(Đ°Đ·Ñ(ĐµÑœĐµĐœĐžĐµ ĐŸĐ±Ñ(Đ°Đ±ĐŸÑ(Đ°Đž Đ
žÑ(ĐµÑ(ÑœĐ²Đ°ĐœĐžĐ¹ Ñ) Đ±ĐŸĐ»ĐµĐµ ĐœĐžĐ·Đ°ĐžĐœĐ
ž Ñ†Ñ†ĐŸĐ²ĐœÑœĐœž, Ñ†ĐµĐœ Ñ†ĐŸĐ»ÑœĐ°ĐŸ Ñ†Ñ†ĐŸ
ĐŸĐ±Ñ(Đ°Đ±ĐŸÑ(Đ°ĐœĐœĐŸĐµ

0028 B0 20 mov al, 20h

002A E6 20 out 20h, al

; Đ²ĐŸÑ†Ñ†Ñ†Đ°ĐœĐ°Đ²Đ»ĐžĐ²Đ°ĐµĐœ Ñ†ĐµĐ³ĐžÑ†
Ñ(Ñ(Ñ)

002C 5A pop dx

002D 59 pop cx

002E 5B pop bx

002F 58 pop ax

0030 8B 26 0006 R mov sp, keep_sp

0034 8E 16 0004 R mov ss, keep_ss

```
0038 CF                                iret

0039                                    interfunction endp

0039                                    disable_sound proc near
0039    50                                push ax
003A   E4 61                            in al, 61h
003C   24 FC                            and al, not 3
003E   E6 61                            out 61h, al
0040   5B                                pop ax
0041   C3                                ret
0042                                    disable_sound endp

0042                                    main proc far

0042   B8 ---- R                        mov ax, @data
0045   8E D8                            mov ds, ax

; ĊĤŸŃĠŢĐ°ĐœŃ ĐµĐƐ ņġŢĐœĐŲŦŶĐžŅ ð¿ŵĐµŴ
(Ň)Đ²Đ°ĐœĐžŷų
0047   B4 35                            mov ah, 35h ; ņġŢĐœĐŲŦŶĐžŷų Đ¿ĐŶĐ»ŜŢĐµĐœ
Đžŷų Đ²ĐµĐŲŢĐŸŖŢĐ°
0049   2E: A0 0000 R                    mov al, vector_n ; ĐœĐŶĐƐĐµń(Đ²ĐµĐŲŢĐŸŖŢĐ°
◌
004D   CD 21                            int 21h
004F   89 1E 0002 R                    mov keep_ip, bx ; Đ·Đ°Đ¿ĐŶĐƐĐžĐœĐ°ĐœĐžĐµ ņŢ
ĐƐĐµŴŧĐµĐœĐžŷų
0053   8C 06 0000 R                    mov keep_cs, es ; Đ·Đ°Đ¿ĐŶĐƐĐžĐœĐ°ĐœĐžĐµ Δ²
#Microsoft (R) Macro Assembler Version 5.10                      12/26/21 21:00:2
Page      1-3

ĐµĐŲŢĐŸŖŢĐ° Đ¿ŵĐµŴ(Ň)Đ²Đ°ĐœĐžŷųų
; ņŢŝŢŢĐ°ĐœĐ°Δ²Đ»ĐžĐ²Đ°ĐµĐƐ ĐœĐŲţſŢŢŢĐŶŢĐœĐŲŦŶĐžŷų ð¿ŵĐµŴŝŒĐ²Đ°ĐœĐžŷųų
0057   1E                                push ds
0058   BA 0001 R                        mov dx, offset interfunction
005B   B8 ---- R                        mov ax, seg interfunction
005E   8E D8                            mov ds, ax
0060   B4 25                            mov ah, 25h ; ņġŢĐœĐŲŦŶĐžŷųų ņŢŝŢŢĐ°ĐœĐŶĐ²
Đ°Đž Đ²ĐµĐŲŢĐŸŖŢĐ°
0062   2E: A0 0000 R                    mov al, vector_n
0066   CD 21                            int 21h
0068   1F                                pop ds
```



```

; 0069 B4 00 mov ah, 0
006B CD 16 int 16h

; 006D FA cli
006E 1E push ds
006F 8B 16 0002 R mov dx, keep_ip
0073 A1 0000 R mov ax, keep_cs
0076 8E D8 mov ds, ax
0078 B4 25 mov ah, 25h ;
007A 2E: A0 0000 R mov al, vector_n
007E CD 21 int 21h
0080 1F pop ds
0081 FB sti

; 0082 E8 0039 R call disable_sound

; 0085 B4 4C mov ah, 4ch
0087 32 C0 xor al, al
0089 CD 21 int 21h

008B main endp

end main

```

#Microsoft (R) Macro Assembler Version 5.10

12/26/21 21:00:2

Symbols-1

Segments and Groups:

| N a m e | Length | Align | Combine | Class |
|------------------|--------|-------|---------|---------|
| DGROUP | | | GROUP | |
| _DATA | 0008 | WORD | PUBLIC | 'DATA' |
| STACK | 0100 | PARA | STACK | 'STACK' |
| _TEXT | 008B | WORD | PUBLIC | 'CODE' |

Symbols:

| N a m e | Type | Value | Attr |
|---------|------|-------|------|
|---------|------|-------|------|

| | | |
|-------------------------|-------------|---------------------|
| DISABLE_SOUND | N PROC 0039 | _TEXT Length = 0009 |
| INTERFUNCTION | F PROC 0001 | _TEXT Length = 0038 |
| KEEP_CS | L WORD 0000 | _DATA |
| KEEP_IP | L WORD 0002 | _DATA |
| KEEP_SP | L WORD 0006 | _DATA |
| KEEP_SS | L WORD 0004 | _DATA |
| MAIN | F PROC 0042 | _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 |

121 Source Lines

121 Total Lines

25 Symbols

47952 + 455210 Bytes symbol space free

0 Warning Errors

0 Severe Errors