**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

; номер вектора прерывания

vector\_n db 08h

.CODE

; функция-обработчик прерывания

; void interruption();

interfunction proc far

jmp interfunction\_start

interfunction\_keep\_ss dw 0

interfunction\_keep\_sp dw 0

interfunction\_stack db 40 dup('#')

interfunction\_stack\_end:

interfunction\_start:

; сохраняем стек

mov interfunction\_keep\_ss, ss

mov interfunction\_keep\_sp, sp

; устанавливаем новый стек

mov sp, seg interfunction\_stack\_end

mov ss, sp

mov sp, offset interfunction\_stack\_end

; сохраняем регистры

push ax

push bx

push cx

push dx

;<действия по обработке прерывания>

mov ax, 8000 ; частота звука

mov cx, ax

mov al, 10110110b

out 43h, al ; код для установления канала 2 таймера-счетчика на работу в качестве делителя частоты

mov ax, cx ; заносим в ax высоту звука

out 42h, al

mov al, ah

out 42h, al ; заносим поочередно 2 байта в порт 42h

in al, 61h

mov ah, al

or al, 3

out 61h, al ; установление битов 0 и 1 в единицу

xor cx, cx ; cx = 0

loop $ ; цикл, пока динамик работает

mov al, ah

out 61h, al ; выключение динамика (изначальное значение порта 61h)

;<конец действий по обработке прерывания>

; 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

; восстанавливаем стек

mov sp, interfunction\_keep\_sp

mov ss, interfunction\_keep\_ss

iret

interfunction 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

cli

mov bh, vector\_n ; в bh номер вектора

push ds

mov dx, offset interfunction

mov ax, seg interfunction

mov ds, ax

mov ah, 25h ; функция установки вектора

mov al, bh ; номер вектора

int 21h

pop ds

sti

; ждём нажатия клавиши

mov ah, 1

int 21h

; возвращаем сохранённую функцию прерывания

cli

mov bh, vector\_n ; в bh номер вектора

push ds

mov dx, keep\_ip

mov ax, keep\_cs

mov ds, ax

mov ah, 25h ; функция установки вектора

mov al, bh ; номер вектора

int 21h

pop ds

sti

; 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 12/27/21 16:52:0

Page 1-1

DOSSEG

.MODEL SMalL

.STaCK 100h

.DaTa

0000 0000 keep\_cs dw 0

0002 0000 keep\_ip dw 0

; ÐœÐŸÐŒÐµÑ Ð²ÐµÐºÑÐŸÑÐ° Ð¿ÑÐµÑÑÐ²Ð°ÐœÐžÑ



0004 08 vector\_n db 08h

.CODE

; ÑÑÐœÐºÑÐžÑ-ÐŸÐ±ÑÐ°Ð±ÐŸÑÑÐžÐº Ð¿ÑÐµÑÑ

Ð²Ð°ÐœÐžÑ

; void interruption();

0000 interfunction proc far

0000 EB 2D 90 jmp interfunction\_start

0003 0000 interfunction\_keep\_ss dw 0

0005 0000 interfunction\_keep\_sp dw 0

0007 0028[ interfunction\_stack db 40 dup('#')

23

]

002F interfunction\_stack\_end:

002F interfunction\_start:

; ÑÐŸÑÑÐ°ÐœÑÐµÐŒ ÑÑÐµÐº

002F 2E: 8C 16 0003 R mov interfunction\_keep\_ss, ss

0034 2E: 89 26 0005 R mov interfunction\_keep\_sp, sp

; ÑÑÑÐ°ÐœÐ°Ð²Ð»ÐžÐ²Ð°ÐµÐŒ ÐœÐŸÐ²ÑÐ¹ ÑÑ

ÐµÐº

0039 BC ---- R mov sp, seg interfunction\_stack\_end

003C 8E D4 mov ss, sp

003E BC 002F R mov sp, offset interfunction\_stack\_end

; ÑÐŸÑÑÐ°ÐœÑÐµÐŒ ÑÐµÐ³ÐžÑÑÑÑ

0041 50 push ax

0042 53 push bx

0043 51 push cx

0044 52 push dx

;<ÐŽÐµÐ¹ÑÑÐ²ÐžÑ Ð¿ÐŸ ÐŸÐ±ÑÐ°Ð±Ð

ŸÑÐºÐµ Ð¿ÑÐµÑÑÐ²Ð°ÐœÐžÑ>

0045 B8 1F40 mov ax, 8000 ; ÑÐ°ÑÑÐŸÑÐ° Ð·Ð²ÑÐºÐ°

0048 8B C8 mov cx, ax

#Microsoft (R) Macro Assembler Version 5.10 12/27/21 16:52:0

Page 1-2

004A B0 B6 mov al, 10110110b

004C E6 43 out 43h, al ; ÐºÐŸÐŽ ÐŽÐ»Ñ ÑÑÑÐ°ÐœÐŸÐ²Ð

»ÐµÐœÐžÑ ÐºÐ°ÐœÐ°Ð»Ð° 2 ÑÐ°Ð¹ÐŒÐµÑÐ°-ÑÑÐµÑ

ÑÐžÐºÐ° ÐœÐ° ÑÐ°Ð±ÐŸÑÑ Ð² ÐºÐ°ÑÐµÑÑÐ²Ðµ

ÐŽÐµÐ»ÐžÑÐµÐ»Ñ ÑÐ°ÑÑÐŸÑÑ

004E 8B C1 mov ax, cx ; Ð·Ð°ÐœÐŸÑÐžÐŒ Ð² ax Ð²ÑÑÐŸÑ

Ñ Ð·Ð²ÑÐºÐ°

0050 E6 42 out 42h, al

0052 8A C4 mov al, ah

0054 E6 42 out 42h, al ; Ð·Ð°ÐœÐŸÑÐžÐŒ Ð¿ÐŸÐŸÑÐµÑÐµ

ÐŽÐœÐŸ 2 Ð±Ð°Ð¹ÑÐ° Ð² Ð¿ÐŸÑÑ 42h

0056 E4 61 in al, 61h

0058 8A E0 mov ah, al

005A 0C 03 or al, 3

005C E6 61 out 61h, al ; ÑÑÑÐ°ÐœÐŸÐ²Ð»ÐµÐœÐžÐµ Ð±Ðž

ÑÐŸÐ² 0 Ðž 1 Ð² ÐµÐŽÐžÐœÐžÑÑ

005E 33 C9 xor cx, cx ; cx = 0

0060 E2 FE loop $ ; ÑÐžÐºÐ», Ð¿ÐŸÐºÐ° ÐŽÐžÐœÐ°ÐŒÐžÐº

ÑÐ°Ð±ÐŸÑÐ°ÐµÑ

0062 8A C4 mov al, ah

0064 E6 61 out 61h, al ; Ð²ÑÐºÐ»ÑÑÐµÐœÐžÐµ ÐŽÐžÐœÐ°

ÐŒÐžÐºÐ° (ÐžÐ·ÐœÐ°ÑÐ°Ð»ÑÐœÐŸÐµ Ð·ÐœÐ°ÑÐµÐœÐž

Ðµ Ð¿ÐŸÑÑÐ° 61h)

;<ÐºÐŸÐœÐµÑ ÐŽÐµÐ¹ÑÑÐ²ÐžÐ¹ Ð¿ÐŸ

ÐŸÐ±ÑÐ°Ð±ÐŸÑÐºÐµ Ð¿ÑÐµÑÑÐ²Ð°ÐœÐžÑ>

; ÑÐ°Ð·ÑÐµÑÐµÐœÐžÐµ ÐŸÐ±ÑÐ°Ð±ÐŸÑÐºÐž Ð

¿ÑÐµÑÑÐ²Ð°ÐœÐžÐ¹ Ñ Ð±ÐŸÐ»ÐµÐµ ÐœÐžÐ·ÐºÐžÐŒÐ

ž ÑÑÐŸÐ²ÐœÑÐŒÐž, ÑÐµÐŒ ÑÐŸÐ»ÑÐºÐŸ ÑÑÐŸ

ÐŸÐ±ÑÐ°Ð±ÐŸÑÐ°ÐœÐœÐŸÐµ

0066 B0 20 mov al, 20h

0068 E6 20 out 20h, al

; Ð²ÐŸÑÑÑÐ°ÐœÐ°Ð²Ð»ÐžÐ²Ð°ÐµÐŒ ÑÐµÐ³ÐžÑ

ÑÑÑ

006A 5A pop dx

006B 59 pop cx

006C 5B pop bx

006D 58 pop ax

; Ð²ÐŸÑÑÑÐ°ÐœÐ°Ð²Ð»ÐžÐ²Ð°ÐµÐŒ ÑÑÐµÐº

006E 2E: 8B 26 0005 R mov sp, interfunction\_keep\_sp

0073 2E: 8E 16 0003 R mov ss, interfunction\_keep\_ss

0078 CF iret

0079 interfunction endp

0079 main proc far

0079 B8 ---- R mov ax, @data

007C 8E D8 mov ds, ax

; ÑÐŸÑÑÐ°ÐœÑÐµÐŒ ÑÑÐœÐºÑÐžÑ Ð¿ÑÐµÑ

#Microsoft (R) Macro Assembler Version 5.10 12/27/21 16:52:0

Page 1-3

ÑÐ²Ð°ÐœÐžÑ

007E B4 35 mov ah, 35h ; ÑÑÐœÐºÑÐžÑ Ð¿ÐŸÐ»ÑÑÐµÐœ

ÐžÑ Ð²ÐµÐºÑÐŸÑÐ°

0080 A0 0004 R mov al, vector\_n ; ÐœÐŸÐŒÐµÑ Ð²ÐµÐºÑÐŸÑÐ

°

0083 CD 21 int 21h

0085 89 1E 0002 R mov keep\_ip, bx ; Ð·Ð°Ð¿ÐŸÐŒÐžÐœÐ°ÐœÐžÐµ Ñ

ÐŒÐµÑÐµÐœÐžÑ

0089 8C 06 0000 R mov keep\_cs, es ; Ð·Ð°Ð¿ÐŸÐŒÐžÐœÐ°ÐœÐžÐµ Ð²

ÐµÐºÑÐŸÑÐ° Ð¿ÑÐµÑÑÐ²Ð°ÐœÐžÑ

; ÑÑÑÐ°ÐœÐ°Ð²Ð»ÐžÐ²Ð°ÐµÐŒ ÐœÐ°ÑÑ ÑÑÐ

œÐºÑÐžÑ Ð¿ÑÐµÑÑÐ²Ð°ÐœÐžÑ

008D FA cli

008E 8A 3E 0004 R mov bh, vector\_n ; Ð² bh ÐœÐŸÐŒÐµÑ Ð²ÐµÐºÑ

ÐŸÑÐ°

0092 1E push ds

0093 BA 0000 R mov dx, offset interfunction

0096 B8 ---- R mov ax, seg interfunction

0099 8E D8 mov ds, ax

009B B4 25 mov ah, 25h ; ÑÑÐœÐºÑÐžÑ ÑÑÑÐ°ÐœÐŸÐ²

ÐºÐž Ð²ÐµÐºÑÐŸÑÐ°

009D 8A C7 mov al, bh ; ÐœÐŸÐŒÐµÑ Ð²ÐµÐºÑÐŸÑÐ°

009F CD 21 int 21h

00A1 1F pop ds

00A2 FB sti

; Ð¶ÐŽÑÐŒ ÐœÐ°Ð¶Ð°ÑÐžÑ ÐºÐ»Ð°Ð²ÐžÑÐž

00A3 B4 01 mov ah, 1

00A5 CD 21 int 21h

; Ð²ÐŸÐ·Ð²ÑÐ°ÑÐ°ÐµÐŒ ÑÐŸÑÑÐ°ÐœÑÐœÐœÑ

Ñ ÑÑÐœÐºÑÐžÑ Ð¿ÑÐµÑÑÐ²Ð°ÐœÐžÑ

00A7 FA cli

00A8 8A 3E 0004 R mov bh, vector\_n ; Ð² bh ÐœÐŸÐŒÐµÑ Ð²ÐµÐºÑ

ÐŸÑÐ°

00AC 1E push ds

00AD 8B 16 0002 R mov dx, keep\_ip

00B1 A1 0000 R mov ax, keep\_cs

00B4 8E D8 mov ds, ax

00B6 B4 25 mov ah, 25h ; ÑÑÐœÐºÑÐžÑ ÑÑÑÐ°ÐœÐŸÐ²

ÐºÐž Ð²ÐµÐºÑÐŸÑÐ°

00B8 8A C7 mov al, bh ; ÐœÐŸÐŒÐµÑ Ð²ÐµÐºÑÐŸÑÐ°

00BA CD 21 int 21h

00BC 1F pop ds

00BD FB sti

; Ð²ÑÑÐŸÐŽ ÐžÐ· Ð¿ÑÐŸÐ³ÑÐ°ÐŒÐŒÑ

00BE B4 4C mov ah, 4ch

00C0 32 C0 xor al, al

00C2 CD 21 int 21h

00C4 main endp

#Microsoft (R) Macro Assembler Version 5.10 12/27/21 16:52:0

Page 1-4

end main

#Microsoft (R) Macro Assembler Version 5.10 12/27/21 16:52:0

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

DGROUP . . . . . . . . . . . . . GROUP

\_DATA . . . . . . . . . . . . 0005 WORD PUBLIC 'DATA'

STACK . . . . . . . . . . . . 0100 PARA STACK 'STACK'

\_TEXT . . . . . . . . . . . . . 00C4 WORD PUBLIC 'CODE'

Symbols:

N a m e Type Value Attr

INTERFUNCTION . . . . . . . . . F PROC 0000 \_TEXT Length = 0079

INTERFUNCTION\_KEEP\_SP . . . . . L WORD 0005 \_TEXT

INTERFUNCTION\_KEEP\_SS . . . . . L WORD 0003 \_TEXT

INTERFUNCTION\_STACK . . . . . . L BYTE 0007 \_TEXT Length = 0028

INTERFUNCTION\_STACK\_END . . . . L NEAR 002F \_TEXT

INTERFUNCTION\_START . . . . . . L NEAR 002F \_TEXT

KEEP\_CS . . . . . . . . . . . . L WORD 0000 \_DATA

KEEP\_IP . . . . . . . . . . . . L WORD 0002 \_DATA

MAIN . . . . . . . . . . . . . . F PROC 0079 \_TEXT Length = 004B

VECTOR\_N . . . . . . . . . . . . L BYTE 0004 \_DATA

@CODE . . . . . . . . . . . . . TEXT \_TEXT

@CODESIZE . . . . . . . . . . . TEXT 0

@CPU . . . . . . . . . . . . . . TEXT 0101h

@DATASIZE . . . . . . . . . . . TEXT 0

@FILENAME . . . . . . . . . . . TEXT lab5

@VERSION . . . . . . . . . . . . TEXT 510

131 Source Lines

131 Total Lines

27 Symbols

47950 + 453165 Bytes symbol space free

0 Warning Errors

0 Severe Errors