**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: Написание собственного прерывания**

|  |  |  |
| --- | --- | --- |
| Студент гр. 8382 |  | Мирончик П.Д. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2019

**Цель работы.**

Изучение прерываний BIOS и DOC Различных видов, реализация собственного прерывания и перехват прерываний системы.

**Формулировка задачи.**

Разработать на языке Ассемблер программу, реализующую перехват прерывания, выполнение определенного действия при получении прерывания и, после выполнения основной части программы, восстановить стандартные прерывания системы.

Требуется реализовать перехват прерывания таймера (08h) и приостановку вывода при получении прерывания.

**Ход работы**

В ходе работы была реализована программа, перехватывающая прерывание системы и выполняющая требуемую задачу.

Поскольку прерывание таймера вызывается 18 раз в секунду, было решено сделать 2-х секундное замедление. В переменную TACTS\_REMAIN изначально было занесено значение 36 – количество тактов, в течение которых мы должны “удерживать” программу.

При получении прерывания мы проверяем, первый ли это запуск (или мы уже раньше получали это прерывание). Если первый, то выводим строку “wait two seconds, please”. После этого мы уменьшаем значение переменной на 1. В основном теле программы (функция Main) крутится цикл, проверяющий значение этой переменной – если значение больше нуля, значит еще не прошло достаточно тактов и нужно подождать.

После завершения цикла мы выводим строку “ok” и восстанавливаем вектор прерывания по умолчанию.

**Выводы**

В ходе выполнения данной работы были изучены прерывания Ассемблера: виды, способы перехвата и способы обращения к вектору перехвата по умолчанию, циклы, реализующие замедление на определенное время и реализация собственного вектора прерывания.

**Приложение А. Код программы**

AStack SEGMENT STACK

DW 1024 DUP(?)

AStack ENDS

; Данные программы

DATA SEGMENT

KEEP\_V DD 0

TACTS\_REMAIN DW 36 ; waiting for 2 seconds

MESSAGE DB 'wait two seconds, please',10,13,'$'

THANK DB 'ok',10,13,'$'

DATA ENDS

; Код программы

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:AStack

Main PROC FAR

mov al, 1

mov dl, 1

mov ah, 33h

int 21h

push DS

sub AX,AX

push AX

mov AX,DATA

mov DS,AX

mov ah,35h ; get vector

mov al,08h

int 21h

mov word ptr KEEP\_V, bx

mov word ptr KEEP\_V+2, es

push ds ; set custom vector

mov dx, offset IWorker

mov ax, seg IWorker

mov ds, ax

mov ah, 25h

mov al, 08h

int 21h

pop ds

Delay: ; wait until TACTS\_REMAIN more than 0

mov ax,[TACTS\_REMAIN]

cmp ax, 0

jg Delay

mov ah, 09h

mov dx, offset THANK

int 21h

cli ; restore vector

push ds

mov dx, word ptr KEEP\_V

mov ax, word ptr KEEP\_V+2

mov ds, ax

mov ah, 25h

mov al, 08h

int 21h

pop ds

sti

ret

Main ENDP

IWorker PROC FAR

pushf ; call "super method"

call dword ptr [KEEP\_V]

push ax

push dx

mov ax, [TACTS\_REMAIN]

cmp ax,36

jl not\_first\_launch

push ax ; print message if it is first tact we interrupted

mov ah, 09h

mov dx, offset MESSAGE

int 21h

pop ax

not\_first\_launch:

dec ax

mov TACTS\_REMAIN, ax

pop dx

pop ax

mov al, 20h

out 20h, al

iret

IWorker ENDP

CODE ENDS

END Main

**Приложение Б. Файл листинга**

Microsoft (R) Macro Assembler Version 5.10 11/6/19 14:43:18

Page 1-1

0000 AStack SEGMENT STACK

0000 0400[ DW 1024 DUP(?)

????

]

0800 AStack ENDS

; Данные программы

0000 DATA SEGMENT

0000 00000000 KEEP\_V DD 0

0004 0024 TACTS\_REMAIN DW 36 ; waiting for 2 seco

nds

0006 77 61 69 74 20 74 MESSAGE DB 'wait two seconds, please',10,13,'$'

77 6F 20 73 65 63

6F 6E 64 73 2C 20

70 6C 65 61 73 65

0A 0D 24

0021 6F 6B 0A 0D 24 THANK DB 'ok',10,13,'$'

0026 DATA ENDS

; Код программы

0000 CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:AStack

0000 Main PROC FAR

0000 B0 01 mov al, 1

0002 B2 01 mov dl, 1

0004 B4 33 mov ah, 33h

0006 CD 21 int 21h

0008 1E push DS

0009 2B C0 sub AX,AX

000B 50 push AX

000C B8 ---- R mov AX,DATA

000F 8E D8 mov DS,AX

0011 B4 35 mov ah,35h ; get vector

0013 B0 08 mov al,08h

0015 CD 21 int 21h

0017 89 1E 0000 R mov word ptr KEEP\_V, bx

001B 8C 06 0002 R mov word ptr KEEP\_V+2, es

001F 1E push ds ; set custom vector

0020 BA 0052 R mov dx, offset IWorker

0023 B8 ---- R mov ax, seg IWorker

0026 8E D8 mov ds, ax

0028 B4 25 mov ah, 25h

Microsoft (R) Macro Assembler Version 5.10 11/6/19 14:43:18

Page 1-2

002A B0 08 mov al, 08h

002C CD 21 int 21h

002E 1F pop ds

002F Delay: ; wait until TACTS\_REMAIN more than 0

002F A1 0004 R mov ax,[TACTS\_REMAIN]

0032 3D 0000 cmp ax, 0

0035 7F F8 jg Delay

0037 B4 09 mov ah, 09h

0039 BA 0021 R mov dx, offset THANK

003C CD 21 int 21h

003E FA cli ; restore vector

003F 1E push ds

0040 8B 16 0000 R mov dx, word ptr KEEP\_V

0044 A1 0002 R mov ax, word ptr KEEP\_V+2

0047 8E D8 mov ds, ax

0049 B4 25 mov ah, 25h

004B B0 08 mov al, 08h

004D CD 21 int 21h

004F 1F pop ds

0050 FB sti

0051 CB ret

0052 Main ENDP

0052 IWorker PROC FAR

0052 9C pushf ; call "super method"

0053 FF 1E 0000 R call dword ptr [KEEP\_V]

0057 50 push ax

0058 52 push dx

0059 A1 0004 R mov ax, [TACTS\_REMAIN]

005C 3D 0024 cmp ax,36

005F 7C 09 jl not\_first\_launch

0061 50 push ax ; print message if it is first

tact we interrupted

0062 B4 09 mov ah, 09h

0064 BA 0006 R mov dx, offset MESSAGE

0067 CD 21 int 21h

0069 58 pop ax

006A not\_first\_launch:

006A 48 dec ax

006B A3 0004 R mov TACTS\_REMAIN, ax

Microsoft (R) Macro Assembler Version 5.10 11/6/19 14:43:18

Page 1-3

006E 5A pop dx

006F 58 pop ax

0070 B0 20 mov al, 20h

0072 E6 20 out 20h, al

0074 CF iret

0075 IWorker ENDP

0075 CODE ENDS

END Main

Microsoft (R) Macro Assembler Version 5.10 11/6/19 14:43:18

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

ASTACK . . . . . . . . . . . . . 0800 PARA STACK

CODE . . . . . . . . . . . . . . 0075 PARA NONE

DATA . . . . . . . . . . . . . . 0026 PARA NONE

Symbols:

N a m e Type Value Attr

DELAY . . . . . . . . . . . . . L NEAR 002F CODE

IWORKER . . . . . . . . . . . . F PROC 0052 CODE Length = 0023

KEEP\_V . . . . . . . . . . . . . L DWORD 0000 DATA

MAIN . . . . . . . . . . . . . . F PROC 0000 CODE Length = 0052

MESSAGE . . . . . . . . . . . . L BYTE 0006 DATA

NOT\_FIRST\_LAUNCH . . . . . . . . L NEAR 006A CODE

TACTS\_REMAIN . . . . . . . . . . L WORD 0004 DATA

THANK . . . . . . . . . . . . . L BYTE 0021 DATA

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

@FILENAME . . . . . . . . . . . TEXT main

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

109 Source Lines

109 Total Lines

16 Symbols

47940 + 461367 Bytes symbol space free

0 Warning Errors

0 Severe Errors