****

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**Факультет прикладної математики**

**Кафедра системного програмування і спеціалізованих комп’ютерних систем**

**Лабораторна робота №5**

**з дисципліни “Системне програмування”**

**Тема: «Ознайомлення з методикою програмування зовнішнього пристрою ПЕОМ "мишка" і набуття навиків безпосередньої роботи з вiдеопам’яттю в текстовому режимі»**

**Варіант 5**

**Виконав:**

**студент II-го курсу**

**групи КВ-41**

**Горпинич-Радуженко Іван**

**Київ 2016**

**Завдання на виконання роботи**

Забезпечити виведення на екран в шестадцятирічному форматі вмісту регістрів ax, bx, cx і dx, одержуваних процедурою користувача для обробки подій від мишки. Умовою виклику процедури прийняти будь-яке переміщення мишки. Основна програма реалізується згідно з Прикладом 1.

Microsoft (R) Macro Assembler Version 5.00 3/30/16 00:30:08

Page 1-1

.386

0000 data segment use16

0000 0000 sax dw 0

0002 0000 scx dw 0

0004 data ends

assume ds:data, cs:code

0000 code segment use16

0000 prmaus proc far

0000 57 push di

0001 06 push es

0002 1E push ds

0003 60 pusha

0004 68 B800 push 0b800h

0007 07 pop es

0008 68 ---- R push data

000B 1F pop ds

000C A3 0000 R mov sax, ax

000F 89 0E 0002 R mov scx, cx

0013 B1 00 mov cl, 0

0015 BF 0010 mov di, 16

0018 @aloop:

0018 A1 0000 R mov ax, sax

001B D3 E8 shr ax, cl

001D 25 000F and ax, 1111b

0020 3D 0009 cmp ax, 9

0023 0F 8E 002D R jle @d

0027 05 0037 add ax, 55

002A EB 04 90 jmp @out

002D @d:

002D 05 0030 add ax, 48

0030 @out:

0030 26: 88 05 mov es:[di], al

0033 83 EF 02 sub di, 2

0036 80 C1 04 add cl, 4

0039 80 F9 10 cmp cl, 16

003C 75 DA jne @aloop

003E B1 00 mov cl, 0

0040 BF 00B0 mov di, (16+160)

0043 @bloop:

Microsoft (R) Macro Assembler Version 5.00 3/30/16 00:30:08

Page 1-2

0043 8B C3 mov ax, bx

0045 D3 E8 shr ax, cl

0047 25 000F and ax, 1111b

004A 3D 0009 cmp ax, 9

004D 0F 8E 0057 R jle @bd

0051 05 0037 add ax, 55

0054 EB 04 90 jmp @bout

0057 @bd:

0057 05 0030 add ax, 48

005A @bout:

005A 26: 88 05 mov es:[di], al

005D 83 EF 02 sub di, 2

0060 80 C1 04 add cl, 4

0063 80 F9 10 cmp cl, 16

0066 75 DB jne @bloop

0068 B1 00 mov cl, 0

006A BF 0150 mov di, (16+160\*2)

006D @cloop:

006D A1 0002 R mov ax, scx

0070 D3 E8 shr ax, cl

0072 25 000F and ax, 1111b

0075 3D 0009 cmp ax, 9

0078 0F 8E 0082 R jle @cd

007C 05 0037 add ax, 55

007F EB 04 90 jmp @cout

0082 @cd:

0082 05 0030 add ax, 48

0085 @cout:

0085 26: 88 05 mov es:[di], al

0088 83 EF 02 sub di, 2

008B 80 C1 04 add cl, 4

008E 80 F9 10 cmp cl, 16

0091 75 DA jne @cloop

0093 B1 00 mov cl, 0

0095 BF 01F0 mov di, (16+160\*3)

0098 @dloop:

0098 8B C2 mov ax, dx

009A D3 E8 shr ax, cl

009C 25 000F and ax, 1111b

009F 3D 0009 cmp ax, 9

00A2 0F 8E 00AC R jle @dd

00A6 05 0037 add ax, 55

Microsoft (R) Macro Assembler Version 5.00 3/30/16 00:30:08

Page 1-3

00A9 EB 04 90 jmp @dout

00AC @dd:

00AC 05 0030 add ax, 48

00AF @dout:

00AF 26: 88 05 mov es:[di], al

00B2 83 EF 02 sub di, 2

00B5 80 C1 04 add cl, 4

00B8 80 F9 10 cmp cl, 16

00BB 75 DB jne @dloop

00BD 61 popa

00BE 1F pop ds

00BF 07 pop es

00C0 5F pop di

00C1 CB ret

00C2 prmaus endp

00C2 begin:

00C2 B8 ---- R mov ax, data

00C5 8E D8 mov ds, ax

00C7 33 C0 xor ax, ax

00C9 CD 33 int 33h

00CB B8 000C mov ax, 0ch

00CE B9 001F mov cx, 11111b

00D1 06 push es

00D2 0E push cs

00D3 07 pop es

00D4 8D 16 0000 R lea dx, prmaus

00D8 CD 33 int 33h

00DA 07 pop es

00DB B8 B800 mov ax, 0b800h

00DE 8E C0 mov es, ax

00E0 BA 000F mov dx, 15

00E3 33 C0 xor ax, ax

00E5 8B F8 mov di,ax

00E7 B8 0003 mov ax, 3

00EA CD 10 int 10h

Microsoft (R) Macro Assembler Version 5.00 3/30/16 00:30:08

Page 1-4

00EC B2 61 mov dl, 'a'

00EE B9 0004 mov cx, 4

00F1 @regloop:

00F1 26: 88 15 mov es:[di], dl

00F4 FE C2 inc dl

00F6 81 C7 00A0 add di, (80\*2)

00FA 49 dec cx

00FB 75 F4 jnz @regloop

00FD 8B F8 mov di,ax

00FF B9 0004 mov cx, 4

0102 @xloop:

0102 26: C6 45 02 78 mov byte ptr es:[di+2], 'x'

0107 26: C6 45 04 3A mov byte ptr es:[di+4], ':'

010C 81 C7 00A0 add di, (80\*2)

0110 49 dec cx

0111 75 EF jnz @xloop

0113 B4 01 mov ah, 01h

0115 CD 21 int 21h

0117 33 C9 xor cx,cx

0119 B8 000C mov ax,0ch

011C CD 33 int 33h

011E code ends

end begin

Microsoft (R) Macro Assembler Version 5.00 3/30/16 00:30:08

Symbols-1

N a m e Size Length Align Combine Class

CODE . . . . . . . . . . . . . . 16 Bit 011E PARA NONE

DATA . . . . . . . . . . . . . . 16 Bit 0004 PARA NONE

Symbols:

N a m e Type Value Attr

BEGIN . . . . . . . . . . . . . L NEAR 00C2 CODE

PRMAUS . . . . . . . . . . . . . F PROC 0000 CODE Length = 00C2

SAX . . . . . . . . . . . . . . L WORD 0000 DATA

SCX . . . . . . . . . . . . . . L WORD 0002 DATA

@ALOOP . . . . . . . . . . . . . L NEAR 0018 CODE

@BD . . . . . . . . . . . . . . L NEAR 0057 CODE

@BLOOP . . . . . . . . . . . . . L NEAR 0043 CODE

@BOUT . . . . . . . . . . . . . L NEAR 005A CODE

@CD . . . . . . . . . . . . . . L NEAR 0082 CODE

@CLOOP . . . . . . . . . . . . . L NEAR 006D CODE

@COUT . . . . . . . . . . . . . L NEAR 0085 CODE

@D . . . . . . . . . . . . . . . L NEAR 002D CODE

@DD . . . . . . . . . . . . . . L NEAR 00AC CODE

@DLOOP . . . . . . . . . . . . . L NEAR 0098 CODE

@DOUT . . . . . . . . . . . . . L NEAR 00AF CODE

@FILENAME . . . . . . . . . . . TEXT l1

@OUT . . . . . . . . . . . . . . L NEAR 0030 CODE

@REGLOOP . . . . . . . . . . . . L NEAR 00F1 CODE

@XLOOP . . . . . . . . . . . . . L NEAR 0102 CODE

194 Source Lines

194 Total Lines

22 Symbols

50962 + 465582 Bytes symbol space free

0 Warning Errors

0 Severe Errors