**Міністерство освіти та науки України  
Національний технічний університет України  
«Київський політехнічний інститут»  
Факультет прикладної математики  
Кафедра системного програмування і спеціалізованих  
комп’ютерних систем**

**Лабораторна робота №2**з дисципліни

**«Системне програмування»  
Тема: «Ознайомлення з методами адресації даних»**

Виконав: Сидоренко Владислав Олегович

Студент групи КВ-32

Перевірив(ла)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**м. Київ**

**2014**

**Мета роботи**

Вивчення типових структур даних та методів адресації їх елементів у мікропроцесорах 80х86 (Pentium) в реальному режимі.

Варіант 18:

Заданий логічний сегмент

**Data1 segment**

**Sumс4 dw 0**

**Sumnс4 dw 0**

**A1 dd 3 dup (5 dup (9 dup (1020201h)))**

**Data1 ends**

1. Створити сегмент кодів (наприклад, з ім‘ям Code1), в якому реалізувати наступне завдання: накопичити в слові за адресою sumc4 суму байтів масиву А1, в яких значення зміщення в сегменті кратне 4 (ділиться на 4), а за адресою sumnc4 – не кратне 4 (не ділиться на 4).

2. Створити ще один логічний сегмент даних (наприклад, з ім‘ям Data2), в якому задати директиви:

**A2 dw 2 dup (8 dup (4 dup (0)))**

**A3 dw 4 dup (4 dup (4 dup (0)))**

3. Створити ще один логічний сегмент кодів (наприклад, з ім‘ям Code2), в якому за допомогою ланцюгових команд заповнити масив А2 значенням суми із Sumc4, а в масив А3 - із Sumnс4. Забезпечити безумовну передачу управління з сегменту Code1 в сегмент Code2 за допомогою непрямої міжсегментної передачі управління.

**Лістинг програми:**

Microsoft (R) Macro Assembler Version 6.14.8444 11/16/14 23:56:09

lab3.asm Page 1 - 1

.486

.model flat

00000000 Data1 segment

00000000 0000 Sumc4 dw 0

00000002 0000 Sumnc4 dw 0

00000004 00000003 [ A1 dd 3 dup (5 dup (9 dup (1020201h)))

00000005 [

00000009 [

01020201

]

]

]

00000220 Data1 ends

00000000 Data2 segment

00000000 00000002 [ A2 dw 2 dup (8 dup (4 dup (0)))

00000008 [

00000004 [

0000

]

]

]

00000080 00000004 [ A3 dw 4 dup (4 dup (4 dup (0)))

00000004 [

00000004 [

0000

]

]

]

00000100 Data2 ends

00000000 Code1 segment

00000000 \_start:

00000000 66| BE 0000 mov si, 0

00000004 START\_LOOP:

00000004 66| 8B C6 mov ax,si ;check if current offset divide by 4

00000007 B2 04 mov dl,4

00000009 F6 F2 div dl

0000000B 80 FC 00 cmp ah,0

0000000E 74 16 je IS\_DIV\_BY\_4

00000010 66| A1 mov ax, Sumnc4 ;if it isn't

00000002 R

00000016 66| 03 04 35 add ax, word ptr[A1+si]

00000004 R

0000001E 66| A3 mov Sumnc4, ax

00000002 R

00000024 EB 14 jmp CHECK\_LOOP

00000026 IS\_DIV\_BY\_4: ;if it is

00000026 66| A1 mov ax,Sumc4

00000000 R

0000002C 66| 03 04 35 add ax,word ptr[A1+si]

00000004 R

00000034 66| A3 mov Sumc4,ax

00000000 R

0000003A CHECK\_LOOP:

0000003A 66| 81 FE 021C cmp si,540

0000003F 66| 46 inc si

00000041 72 C1 jb START\_LOOP

00000043 E8 00000000 R call GO\_TO\_CODE2

00000048 Code1 ends

00000000 Code2 segment

00000000 GO\_TO\_CODE2:

00000000 FC cld

00000001 66| A1 mov ax,Sumc4

00000000 R

00000007 8D 3D 00000000 R lea edi,A2

0000000D 66| B9 0040 mov cx,64

00000011 F3/ 66| AB rep stosw

00000014 FC cld

00000015 66| A1 mov ax,Sumnc4

00000002 R

0000001B 8D 3D 00000080 R lea edi,A3

00000021 66| B9 0040 mov cx,64

00000025 F3/ 66| AB rep stosw

00000028 C3 ret

00000029 Code2 ends

end \_start

Microsoft (R) Macro Assembler Version 6.14.8444 11/16/14 23:56:09

lab3.asm Symbols 2 - 1

Segments and Groups:

N a m e Size Length Align Combine Class

Code1 . . . . . . . . . . . . . 32 Bit 00000048 Para Private

Code2 . . . . . . . . . . . . . 32 Bit 00000029 Para Private

Data1 . . . . . . . . . . . . . 32 Bit 00000220 Para Private

Data2 . . . . . . . . . . . . . 32 Bit 00000100 Para Private

FLAT . . . . . . . . . . . . . . GROUP

\_DATA . . . . . . . . . . . . . 32 Bit 00000000 Para Public 'DATA'

\_TEXT . . . . . . . . . . . . . 32 Bit 00000000 Para Public 'CODE'

Symbols:

N a m e Type Value Attr

@CodeSize . . . . . . . . . . . Number 00000000h

@DataSize . . . . . . . . . . . Number 00000000h

@Interface . . . . . . . . . . . Number 00000000h

@Model . . . . . . . . . . . . . Number 00000007h

@code . . . . . . . . . . . . . Text \_TEXT

@data . . . . . . . . . . . . . Text FLAT

@fardata? . . . . . . . . . . . Text FLAT

@fardata . . . . . . . . . . . . Text FLAT

@stack . . . . . . . . . . . . . Text FLAT

A1 . . . . . . . . . . . . . . . DWord 00000004 Data1

A2 . . . . . . . . . . . . . . . Word 00000000 Data2

A3 . . . . . . . . . . . . . . . Word 00000080 Data2

CHECK\_LOOP . . . . . . . . . . . L Near 0000003A Code1

GO\_TO\_CODE2 . . . . . . . . . . L Near 00000000 Code2

IS\_DIV\_BY\_4 . . . . . . . . . . L Near 00000026 Code1

START\_LOOP . . . . . . . . . . . L Near 00000004 Code1

Sumc4 . . . . . . . . . . . . . Word 00000000 Data1

Sumnc4 . . . . . . . . . . . . . Word 00000002 Data1

\_start . . . . . . . . . . . . . L Near 00000000 Code1 Public

0 Warnings

0 Errors

**Код програми:**

.486

.model flat

Data1 segment

Sumc4 dw 0

Sumnc4 dw 0

A1 dd 3 dup (5 dup (9 dup (1020201h)))

Data1 ends

Data2 segment

A2 dw 2 dup (8 dup (4 dup (0)))

A3 dw 4 dup (4 dup (4 dup (0)))

Data2 ends

Code1 segment

\_start:

mov si, 0

START\_LOOP:

mov ax,si ;check if current offset divide by 4

mov dl,4

div dl

cmp ah,0

je IS\_DIV\_BY\_4

mov ax, Sumnc4 ;if it isn't

add ax, word ptr[A1+si]

mov Sumnc4, ax

jmp CHECK\_LOOP

IS\_DIV\_BY\_4: ;if it is

mov ax,Sumc4

add ax,word ptr[A1+si]

mov Sumc4,ax

CHECK\_LOOP:

cmp si,540

inc si

jb START\_LOOP

call GO\_TO\_CODE2

Code1 ends

Code2 segment

GO\_TO\_CODE2:

cld

mov ax,Sumc4

lea edi,A2

mov cx,64

rep stosw

cld

mov ax,Sumnc4

lea edi,A3

mov cx,64

rep stosw

ret

Code2 ends

end \_start