**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 №4**з дисципліни

**«Системне програмування»**Тема: **«Організація взаємозв'язку програм мовою Асемблера з С++ програмами»**

Виконав: Сидоренко Владислав Олегович

Студент групи КВ-32

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

**м. Київ**

**2014**

**Завдання:**

*Варіант 18:*

Розробити функцію **void Big2sSub(byte\* M1, byte\* M2, short len)**, де M1, M2 –

надвеликі цілі додатні числа (байтові масиви довжиною len). Операція – М1=М1-М2.

Повинні використовуватись команди для 32-розрядних даних. Якщо значення len не

кратно 4, то для додавання останніх байт використовувати команди для 8-розрядних

даних.

**Лістинг:**

\_Big2sSub:

00DC18E8 55 push ebp

00DC18E9 8B EC mov ebp,esp

00DC18EB BE 00 00 00 00 mov esi,0

00DC18F0 66 8B 75 10 mov si,word ptr [ebp+10h]

00DC18F4 66 83 E6 03 and si,3

00DC18F8 8B DE mov ebx,esi

00DC18FA 8B 55 08 mov edx,dword ptr [ebp+8]

00DC18FD 8B 4D 0C mov ecx,dword ptr [ebp+0Ch]

00DC1900 F8 clc

start\_32\_loop:

00DC1901 3B DE cmp ebx,esi

00DC1903 74 02 je start\_32\_loop+6h (0DC1907h)

00DC1905 66 9D popf

NEW\_FIRST\_ITER:

00DC1907 8B 04 13 mov eax,dword ptr [ebx+edx]

00DC190A 1B 04 0B sbb eax,dword ptr [ebx+ecx]

00DC190D 66 9C pushf

00DC190F 89 04 13 mov dword ptr [ebx+edx],eax

00DC1912 83 C3 04 add ebx,4

00DC1915 3B 5D 10 cmp ebx,dword ptr [ebp+10h]

00DC1918 7C E7 jl start\_32\_loop (0DC1901h)

00DC191A 66 83 FE 00 cmp si,0

00DC191E 74 1E je start\_32\_loop+3Dh (0DC193Eh)

00DC1920 66 BB 00 00 mov bx,0

start\_8\_loop:

00DC1924 66 83 FB 00 cmp bx,0

00DC1928 74 02 je start\_32\_loop+2Bh (0DC192Ch)

00DC192A 66 9D popf

FIRST\_ITER:

00DC192C 8A 24 13 mov ah,byte ptr [ebx+edx]

00DC192F 1A 24 0B sbb ah,byte ptr [ebx+ecx]

00DC1932 66 9C pushf

00DC1934 88 24 13 mov byte ptr [ebx+edx],ah

00DC1937 66 43 inc bx

00DC1939 66 3B DE cmp bx,si

00DC193C 7C E6 jl start\_32\_loop+23h (0DC1924h)

END\_LABEL:

00DC193E 66 9D popf

00DC1940 5D pop ebp

00DC1941 C3 ret

**Код програми:**

*main.c:*

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <time.h>

#define n 35 // кількість байтів у надвеликому числі

**typedef** **unsigned** **char** byte; // для роботи з байтами використовується тип char

**void** Big2sSub(byte\* p1, byte\* p2, **int** p3);

**void** PrintBinary(byte \*number, **short** length)

{

**for** (**short** i = length; i != 0; --i)

{

byte currentPart = \*(number + i - 1);

currentPart & 0x80 ? printf("1") : printf("0");

currentPart & 0x40 ? printf("1") : printf("0");

currentPart & 0x20 ? printf("1") : printf("0");

currentPart & 0x10 ? printf("1") : printf("0");

currentPart & 0x08 ? printf("1") : printf("0");

currentPart & 0x04 ? printf("1") : printf("0");

currentPart & 0x02 ? printf("1") : printf("0");

currentPart & 0x01 ? printf("1") : printf("0");

printf(" ");

}

printf("\n\n");

}

**int** main()

{

srand(time(NULL));

byte x[n], y[n]; //надвеликі числа

**for** (**int** i = 0; i<n; i++)

{

x[i] = 86;

y[i] = 85;

}

printf("Before: \n\n");

printf("M1 = \n");

PrintBinary(x,n);

printf("M2 = \n");

PrintBinary(y,n);

Big2sSub(x, y, n);

printf("New M1 = \n");

PrintBinary(x, n);

**return** 0;

}

*Big2sSub.asm:*

title "KV32\_Sydorenko"

.386

.model flat, C

.code

Big2sSub proc

push ebp

mov ebp, esp;// базова адреса фактичних параметрів

mov esi,0

mov si, [ebp + 16] ;length

and si, 00000011b

mov ebx, esi

mov edx, [ebp + 8] ;m1

mov ecx, [ebp + 12] ;m2

clc

start\_32\_loop:

cmp ebx,esi

je NEW\_FIRST\_ITER

popf

NEW\_FIRST\_ITER:

mov eax, dword ptr [edx + ebx]

sbb eax, dword ptr [ecx + ebx]

pushf

mov [edx + ebx], eax

add ebx,4

cmp ebx,[ebp + 16]

jl start\_32\_loop

cmp si,0

je END\_LABEL

mov bx,0

start\_8\_loop:

cmp bx,0

je FIRST\_ITER

popf

FIRST\_ITER:

mov ah,byte ptr [edx + ebx]

sbb ah,byte ptr [ecx + ebx]

pushf

mov byte ptr [edx + ebx], ah

inc bx

cmp bx, si

jl start\_8\_loop

END\_LABEL:

popf

pop ebp

ret

Big2sSub endp

end