**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний технічний університет України**

**«Київський Політехнічний Інститут»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

**Лабораторна робота №4**

*з дисципліни «Системне програмування – 1»*

*на тему: «Програмування арифметичних операцій підвищеної розрядності»*

**Виконав:**

студент 2-го курсу ФІОТ

групи ІО-44

*Барабаш Т.А.*

**Перевірив:**

Старший викладач

*Порєв В. М.*

**Київ – 2016**

**ЛАБОРАТОРНА РОБОТА №4**

*Програмування арифметичних операцій підвищеної розрядності*

**Мета:** Навчитися створювати модульні проекти на асемблері, а також закріпити знання основних форматів представлення чисел у комп’ютері.

**I. Завдання**

1. Створити у середовищі MS Visual Studio проект з ім’ям Lab4.
2. Написати вихідний текст програми згідно варіанту завдання. У проекті мають бути три модуля на асемблері:

* головний модуль: файл main4.asm. Цей модуль створити та написати
* заново, частково використавши текст модуля main3.asm попередньої
* роботи №3;
* другий модуль: використати module попередньої роботи №3;
* третій модуль: створити новий з ім'ям longop.

1. У цьому проекті кожний модуль може окремо компілюватися.
2. Скомпілювати вихідний текст і отримати виконуємий файл програми.
3. Перевірити роботу програми. Налагодити програму.
4. Отримати результати – кодовані значення чисел згідно варіанту завдання.
5. Проаналізувати та прокоментувати результати, вихідний текст та дизасембльований машинний код програми.

|  |  |  |
| --- | --- | --- |
| № варіанту | Розрядність додавання  (біт) | Розрядність віднімання  (біт) |
| 3 | 160 | 960 |

**ІІ. Код програми**

**ІІ.А. Lab4.asm**

.586

.model flat, stdcall

option casemap :none

include \masm32\include\windows.inc

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

include module.inc

include longop.inc

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\user32.lib

.data

Caption db "№4", 0

Mes db "Виконав: студент групи ІО-44", 13, 10, "Барабаш Т.А.", 0

CaptionAdd db "A + B", 0

CaptionSub db "A - B", 0

AddA db 160 dup(0)

AddB db 160 dup(0)

ResultAdd db 160 dup(0)

TextResultAdd db 160 dup(0)

ResultA db 160 dup(0)

SubA db 960 dup(0)

SubB db 960 dup(0)

ResultSub db 960 dup(0)

TextResultSub db 960 dup(0)

ResultS db 960 dup(0)

.code

main:

invoke MessageBoxA, 0, ADDR Mes, ADDR Caption, 0

mov eax, 80010001h

mov ecx, 0

SetValueA:

mov dword ptr[AddA + ecx], eax

add eax, 00010000h

add ecx, 4

cmp ecx, 160

jl SetValueA

mov eax, 80010001h

mov ecx, 0

SetValueB:

mov dword ptr[AddB + ecx], eax

add eax, 00010000h

add ecx, 4

cmp ecx, 160

jl SetValueB

push offset AddA

push offset AddB

push offset ResultA

call Add\_LONGOP

push offset TextResultAdd

push offset ResultA

push 160

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextResultAdd, ADDR CaptionAdd, 0

mov eax, 80080018h

mov ecx, 0

SetValueA\_Sub:

mov dword ptr[SubA + ecx], eax

add eax, 00010000h

add ecx, 4

cmp ecx, 960

jl SetValueA\_Sub

mov eax, 00000018h

mov ecx, 0

SetValueB\_Sub:

mov dword ptr[SubB + ecx], eax

add eax, 00010000h

add ecx, 4

cmp ecx, 960

jl SetValueB\_Sub

push offset SubA

push offset SubB

push offset ResultS

call Sub\_LONGOP

push offset TextResultSub

push offset ResultS

push 960

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextResultSub, ADDR CaptionSub, 0

invoke ExitProcess, 0

end main

**ІІ.B. Lab4.asm**

.586

.model flat, c

.code

Add\_LONGOP proc

push ebp

mov ebp, esp

mov esi, [ebp+16]

mov ebx, [ebp+12]

mov edi, [ebp+8]

mov ecx, 0

AddAB:

mov eax, dword ptr[esi+ecx]

adc eax, dword ptr[ebx+ecx]

mov dword ptr [edi+ecx], eax

add ecx, 4

cmp ecx, 80

jl AddAB

pop ebp

ret 12

Add\_LONGOP endp

Sub\_LONGOP proc

push ebp

mov ebp, esp

mov esi, [ebp+16]

mov ebx, [ebp+12]

mov edi, [ebp+8]

mov ecx, 0

SubAB:

mov eax, dword ptr[esi+ecx]

sbb eax, dword ptr[ebx+ecx]

mov dword ptr[edi+ecx], eax

add ecx, 4

cmp ecx, 480

jl SubAB

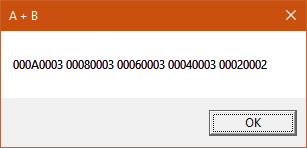
pop ebp

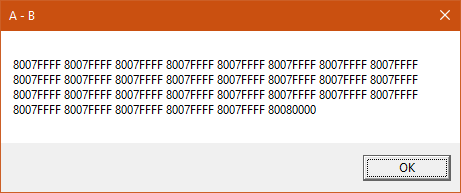
ret 12

Sub\_LONGOP endp

end

**III. Результат**





**ІV. Висновок**

У ході виконання лабораторної роботи було закріплено на практиці навички створення модульних проектів у середовищі Microsoft Visual Studio 2015 та створено програму для додавання та віднімання оперантів з підвищеною розрядністю. Операції ADC та SBB дозволяють нам додавати та віднімати з переносом (ADC) чи позичанням біту (SBB).