НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №4

з дисципліни «Системне програмування»

Виконала:

студентка 2 курсу

ФІОТ гр. ІО-32

Руденко Т.А.

Перевірив:

Порєв В.М.

Київ 2015

**Програмування арифметичних операцій підвищеної розрядності**

**Мета:** Навчитися програмувати на асемблері основні арифметичні операції

підвищеної розрядності, а також отримати перші навички програмування

власних процедур у модульному проекті.

Лістинг:

Main4.asm

.586

.model flat, stdcall

option casemap :none

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

include \masm32\include\windows.inc

include module.inc

include longop.inc

includelib \masm32\lib\user32.lib

includelib \masm32\lib\kernel32.lib

.const

CaptionSub db 'Sub', 0

CaptionAdd db 'Add', 0

CaptionAdd2 db 'Add(2)', 0

.data

TextAdd db 108 dup(?)

Text1Add db 108 dup(?)

TextSub db 32 dup(?)

ValueAAdd dq 80010001h, 80020001h, 80030001h, 80040001h, 80050001h, 80060001h, 80070001h, 80080001h, 80090001h, 800100001h, 800110001h, 800120001h, 800130001h, 800140001h, 800150001h, 800160001h, 800170001h, 800180001h, 800190001h, 800200001h, 800210001h, 800220001h, 800230001h, 800240001h, 800250001h, 800260001h, 800270001h

ValueBAdd dq 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h,80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h, 80000001h

ValueA1Add dq 00250000h, 00260000h, 00270000h, 00280000h, 00290000h, 00300000h, 00310000h, 00320000h, 00330000h, 00340000h, 00350000h, 00360000h, 00370000h, 00380000h, 00390000h, 00400000h, 00410000h, 00420000h, 00430000h, 00440000h, 00450000h, 00460000h, 00470000h, 00480000h, 00490000h, 00500000h, 00510000h, 00520000h

ValueB1Add dq 00000001h, 00000001h, 00000001h, 00000001h, 00000001h, 00000001h, 00000001h, 00000001h, 00000001h, 00000001h,00000001h, 00000001h, 00000001h, 00000001h, 00000001h, 00000001h, 00000001h, 00000001h, 00000001h, 00000001h,00000001h, 00000001h, 00000001h, 00000001h, 00000001h, 00000001h, 00000001h

ValueASub dd 0, 0, 0, 0, 0

ValueBSub dd 00250000h, 00260000h, 00270000h, 00280000h, 00290000h, 00300000h, 00310000h, 00320000h

ResultAdd dd 27 dup(0)

Result1Add dd 27 dup(0)

ResultSub dd 8 dup(0)

.code

main:

mov eax,0

push offset ValueAAdd

push offset ValueBAdd

push offset ResultAdd

call Add\_864\_LONGOP

push offset ValueA1Add

push offset ValueB1Add

push offset Result1Add

call Add\_864\_LONGOP

push offset ValueASub

push offset ValueBSub

push offset ResultSub

call Sub\_256\_LONGOP

push offset TextAdd

push offset ResultAdd

push offset 864

call StrHex\_MY

push offset Text1Add

push offset Result1Add

push offset 864

call StrHex\_MY

push offset TextSub

push offset ResultSub

push offset 256

call StrHex\_MY

invoke MessageBox, 0, ADDR TextAdd, ADDR CaptionAdd, MB\_ICONINFORMATION

invoke MessageBox, 0, ADDR Text1Add, ADDR CaptionAdd2, MB\_ICONINFORMATION

invoke MessageBox, 0, ADDR TextSub, ADDR CaptionSub, MB\_ICONINFORMATION

invoke ExitProcess,0

end main

longop.asm

mov ebp,esp

mov esi, [ebp+16] ;ESI = адреса A

mov ebx, [ebp+12] ;EBX = адреса B

mov edi, [ebp+8] ;EDI = адреса результату

mov ecx, 27; ECX = потрібна кількість повторень

clc ; обнулює біт CF регістру EFLAGS

mov edx, 0

cycle:

mov eax, dword ptr[esi+4\*edx]

adc eax, dword ptr[ebx+4\*edx] ; додавання групи з 32 бітів

mov dword ptr[edi+4\*edx], eax

inc edx; модифікація зсуву

dec ecx ; лічильник зменшуємо на 1

jnz cycle

pop ebp ;відновлення стеку

ret 12

Add\_864\_LONGOP endp

Sub\_256\_LONGOP proc

push ebp

mov ebp,esp

mov esi, [ebp+16] ;ESI = адреса A

mov ebx, [ebp+12] ;EBX = адреса B

mov edi, [ebp+8] ;EDI = адреса результату

mov ecx, 8; ECX = потрібна кількість повторень

clc ; обнулює біт CF регістру EFLAGS

mov edx, 0

cycle:

mov eax, dword ptr[esi+4\*edx]

sbb eax, dword ptr[ebx+4\*edx] ; додавання групи з 32 бітів

mov dword ptr[edi+4\*edx], eax

inc edx; модифікація зсуву

dec ecx ; лічильник зменшуємо на 1

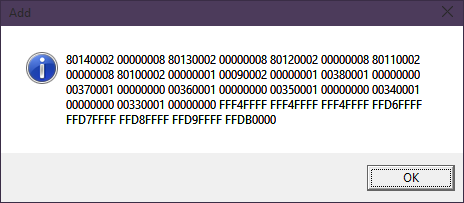
jnz cycle

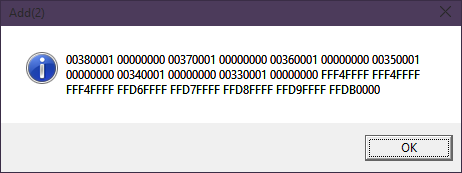
pop ebp ;відновлення стеку

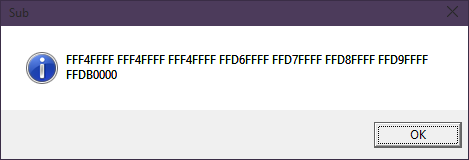
ret 12

Sub\_256\_LONGOP endp

end







Висновок:

Під час виконання даної лабораторної роботи ми навчилися додавати і віднімати числа з підвищеною точністю, а також закріпили навички створення власних модулів.