*МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ*

*НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ*

*“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”*

*Лабораторна робота №4*

*з предмету “Системне програмування”*

*Виконав:*

*Студент 2-го курсу ФІОТ*

*групи ІО-22*

*Коломієць Олег*

*Київ 2014*

Лабораторна робота №4

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.

*Роздруківка тексту програми*

.586

.model flat, c

.code

Add\_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, 20 ; ECX = потрібна кількість повторень

mov edx,0

clc ; обнулює біт CF регістру EFLAGS

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

dec ecx ; лічильник зменшуємо на 1

jnz cycle

pop ebp

ret 12

Add\_LONGOP endp

Sub\_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, 120 ; ECX = потрібна кількість повторень

mov edx,0

clc ; обнулює біт CF регістру EFLAGS

cycle:

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

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

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

inc edx

dec ecx ; лічильник зменшуємо на 1

jnz cycle

pop ebp

ret 12

Sub\_LONGOP endp

end

.586

.model flat, stdcall

option casemap:none

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

include \masm32\include\windows.inc

include module.inc

include longop.inc

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\user32.lib

.data

Captionb1 db "A+B (1 ed)",0

Captionb2 db "A+B (2 ed)",0

Captionb3 db "A-B ",0

TextBuf db 20 dup(?)

TextBuf1 db 120 dup(?)

ValueA dd 20 dup(?)

ValueB dd 20 dup(?)

ValueA1 dd 120 dup(?)

ValueB1 dd 120 dup(?)

Result dd 20 dup(0)

Result1 dd 120 dup(0)

.code

main:

;А+B 1

mov eax , 80010001h

mov esi ,0

cycleAB1:

mov DWord ptr[ValueA+esi], eax

mov DWord ptr[ValueB+esi], 80000001h

add eax , 10000h

add esi,4

cmp esi,20

jb cycleAB1

push offset ValueA

push offset ValueB

push offset Result

call Add\_LONGOP

push offset TextBuf

push offset Result

push 192

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Captionb1,0

;А+B 2

mov eax ,9h

mov esi ,0

cycleAB2:

mov DWord ptr[ValueA+esi], eax

mov DWord ptr[ValueB+esi], 00000001h

add eax , 1h

add esi,4

cmp esi,11

jb cycleAB2

push offset ValueA

push offset ValueB

push offset Result

call Add\_LONGOP

push offset TextBuf

push offset Result

push 192

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Captionb2,0

;А-B

mov eax , 9h

mov esi ,0

cycleAB:

mov DWord ptr[ValueA1+esi], 0

mov DWord ptr[ValueB1+esi], eax

add eax , 1h

add esi,4

cmp esi,120

jb cycleAB

push offset ValueA1

push offset ValueB1

push offset Result1

call Sub\_LONGOP

push offset TextBuf1

push offset Result1

push 768

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf1, ADDR Captionb3,0

invoke ExitProcess,0

end main