МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Лабораторна робота №5

з предмету “Системне програмування”

Виконав:

студент 2-го курсу ФІОТ

групи ІО-31

Долинний О.В.

Київ 2015

Лабораторна робота №5

Програмування множення чисел підвищеної розрядності

Мета: Навчитися програмувати на асемблері множення чисел підвищеної

розрядності, а також закріпити навички програмування власних процедур у

модульному проекті.

Завдання:

1. Створити у середовищі MS Visual Studio проект з ім’ям Lab5.

2. Написати вихідний текст програми згідно варіанту завдання. У проекті мають бути три модуля на асемблері:

- головний модуль: файл main5.asm. Цей модуль створити та написати заново, частково використавши текст модуля main4.asm попередньої роботи №4;

- другий модуль: використати module попередніх робіт№3, 4;

- третій модуль: модуль longop попередньої роботи №4 доповнити новим кодом відповідно завданню.

3. У цьому проекті кожний модуль може окремо компілюватися.

4. Скомпілювати вихідний текст і отримати виконуємий файл програми.

5. Перевірити роботу програми. Налагодити програму.

6. Отримати результати – кодовані значення чисел згідно варіанту завдання.

7. Проаналізувати та прокоментувати результати, вихідний текст та дизасембльований машинний код програми.

**Варіант завдання:** 11

n = 30 + 11\*2 = 52

Роздруківка тексту програми

.586

.model flat, stdcall

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\user32.lib

include \work\Lab4\Lab4\module.inc

include \work\Lab4\Lab4\longop.inc

.const

caption1 db 'n!', 0

caption2 db '(n!)^2', 0

.data

ct dd 52

fact\_result db 1, 31 dup (0)

buf db 32 dup (0)

fact\_sqr\_result db 64 dup (0)

fact\_text db 64 dup (0)

fact\_sqr\_text db 128 dup (0)

.code

main:

@cycle: ;цикл обчислення факторіала

mov esi, offset fact\_result

mov edi, offset buf

mov ecx, 8

rep movsd

mov eax, 0

mov edi, offset fact\_result

mov ecx, 8

rep stosd

push offset buf

push ct

push offset fact\_result

push 32

call Mul\_N\_32\_LONGOP

dec ct

cmp ct, 1

jg @cycle

push offset fact\_text

push offset fact\_result

push 256

call StrHex\_MY

invoke MessageBoxA, 0, ADDR fact\_text, ADDR caption1, 0

mov eax, 0

mov edi, offset fact\_sqr\_result

mov ecx, 16

stosd

push offset fact\_result

push offset fact\_result

push offset fact\_sqr\_result

push 32

call Mul\_N\_N\_LONGOP

push offset fact\_sqr\_text

push offset fact\_sqr\_result

push 512

call StrHex\_MY

invoke MessageBoxA, 0, ADDR fact\_sqr\_text, ADDR caption2, 0

invoke ExitProcess, 0

end main

.586

.model flat, c

.data

cote dd ?

.code

Mul\_N\_32\_LONGOP proc

local ct:DWORD

push ebp

mov ebp,esp

mov esi, [ebp+28] ;ESI = адреса множеного

mov ebx, [ebp+24] ;EBX = множник

mov edi, [ebp+20] ;EDI = адреса результату

mov ecx, [ebp+16] ;кількість байтів

shr ecx, 2 ;кількість подвійних слів

mov ct, ecx

xor ecx, ecx

cycle\_p: ;для парних доданків

mov eax, [esi+4\*ecx]

mul ebx

mov [edi+4\*ecx], eax

mov [edi+4\*ecx+4], edx

inc ecx

inc ecx

cmp ecx, ct

jl cycle\_p

xor ecx, ecx

inc ecx

cycle\_np: ;для непарних доданків

mov eax, [esi+4\*ecx]

mul ebx

add [edi+4\*ecx], eax

adc [edi+4\*ecx+4], edx

adc byte ptr [edi+4\*ecx+8], 0

inc ecx

inc ecx

cmp ecx, ct

jl cycle\_np

pop ebp ;відновлення стеку

ret 16

Mul\_N\_32\_LONGOP endp

Mul\_N\_N\_LONGOP proc

local ct:DWORD

push ebp

mov ebp,esp

mov esi, [ebp+28] ;ESI = адреса множеного

mov ebx, [ebp+24] ;EBX = адреса множника

mov edi, [ebp+20] ;EDI = адреса результату

mov ecx, [ebp+16] ; кількість байтів

shr ecx, 2

mov cote, ecx

xor ecx, ecx

cycle:

push ebx

push ecx

push edi

shl ecx, 2

add ebx, ecx

add edi, ecx

shr ecx, 2

xor ecx, ecx

cycle\_p: ;для парних доданків

mov eax, [esi+4\*ecx]

mul dword ptr [ebx]

add [edi+4\*ecx], eax

adc [edi+4\*ecx+4], edx

adc byte ptr [edi+4\*ecx+8], 0

inc ecx

inc ecx

cmp ecx, cote

jl cycle\_p

xor ecx, ecx

inc ecx

cycle\_np: ;для непарних доданків

mov eax, [esi+4\*ecx]

mul dword ptr [ebx]

add [edi+4\*ecx], eax

adc [edi+4\*ecx+4], edx

adc byte ptr [edi+4\*ecx+8], 0

inc ecx

inc ecx

cmp ecx, cote

jl cycle\_np

pop edi

pop ecx

pop ebx

inc ecx

cmp ecx, cote

jl cycle

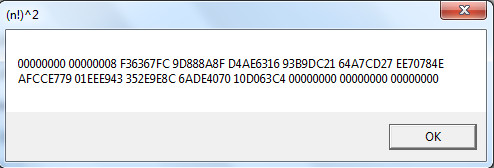
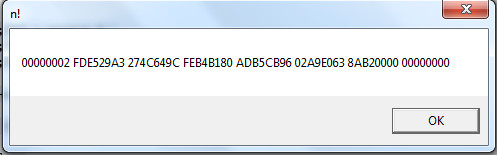
pop ebp ;відновлення стеку

ret 16

Mul\_N\_N\_LONGOP endp

end

**Результати:**



**Аналіз результатів:**

Програма виконує обчислення факторіалу та квадрату факторіалу за допомогою операцій множення на 32-бітове число(обчислення факторіалу), та множення на n-бітове число(возведення факторіалу у квадрат).

**Висновок:**

Під час виконання лабораторної роботи були покращені навички написання власних модулів, роботи з циклами, а також були закріпленні основні навички в операціях множення чисел з підвищеною розрядністю.