Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №5

з дисципліни «Системне програмування»

на тему:

**«Програмування множення чисел підвищеної розрядності»**

Виконала:

студентка 2-го курсу

групи ІО-32

Руденко Т.А.

Номер залікової книжки: 3224

Київ - 2015 р.

**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: n = 30 + 2×24 = 78.

Лістинг:

**Main.asm**

.586

.model flat, stdcall

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

option casemap :none

.data

Caption db "n!" ,0

Caption1 db "n! x n!" ,0

textBuf dd 30 dup(?)

textBuf1 dd 30 dup(?)

v dd 16 dup(?)

v2 dd 1h,0h,0h,0h,0h,0h,0h,0h,0h,0h,0h,0h,0h,0h,0h,0h

x dd 1h

n dd 78

result dd 18 dup(?)

.code

main:

@cycle:

push offset v

push offset v2

push x

call M\_Nx32

inc x

mov ecx, 0

@cycleIn:

mov eax, dword ptr[v + 4 \* ecx]

mov dword ptr[v2 + 4 \* ecx], eax

inc ecx

cmp ecx, 10

jl @cycleIn

dec n

jnz @cycle

push offset textBuf

push offset v

push 395

call StrHex\_MY

invoke MessageBoxA, 0, ADDR textBuf, ADDR Caption, 40h

push offset v

push offset v

push offset result

call M\_NxN

push offset textBuf1

push offset result

push 800

call StrHex\_MY

invoke MessageBoxA, 0, ADDR textBuf1, ADDR Caption1, 40h

invoke ExitProcess,0

end main

**longop.asm**

.586

.model flat, c

.data

x dd 1h

bitNumber dd ?

a dd 0

b dd 0

r dd 0

.code

M\_Nx32 proc

push ebp

mov ebp, esp

mov esi, [ebp + 16]

mov edi, [ebp + 12]

mov ebx, [ebp + 8]

mov x, ebx

mov ecx, 8

xor ebx, ebx

@cycle1:

mov eax, dword ptr[edi + 8 \* ebx]

mul x

mov dword ptr[esi + 8 \* ebx], eax

mov dword ptr[esi + 8 \* ebx + 4], edx

inc ebx

dec ecx

jnz @cycle1

mov ecx, 8

xor ebx, ebx

@cycle2:

mov eax, dword ptr[edi + 8 \* ebx + 4]

mul x

clc

adc eax, dword ptr[esi + 8 \* ebx + 4]

mov dword ptr[esi + 8 \* ebx + 4], eax

clc

adc edx, dword ptr[esi + 8 \* ebx + 8]

mov dword ptr[esi + 8 \* ebx + 8], edx

inc ebx

dec ecx

jnz @cycle2

pop ebp

ret 12

M\_Nx32 endp

M\_NxN proc

push ebp

mov ebp, esp

mov esi, dword ptr[ebp + 16]

mov edi, dword ptr[ebp + 12]

mov ebx, dword ptr[ebp + 8]

mov ecx, 16

@cycle:

push ecx

mov ecx, 16

@cycleInner:

push ecx

mov ecx, a

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

mov ecx, b

mul dword ptr[edi + 4 \* ecx]

mov ecx, r

add eax, dword ptr[ebx + 4 \* ecx]

mov dword ptr[ebx + 4 \* ecx], eax

mov eax, dword ptr[ebx + 4 \* ecx]

add edx, dword ptr[ebx + 4 \* ecx + 4]

mov dword ptr[ebx + 4 \* ecx + 4], edx

mov eax, dword ptr[ebx + 4 \* ecx + 4]

inc a

inc r

pop ecx

dec ecx

jnz @cycleInner

inc b

xor eax, eax

mov a, eax

mov eax, b

mov r, eax

pop ecx

dec ecx

jnz @cycle

pop ebp

ret 8

M\_NxN endp

end

**Висновок:** Під час виконання лабораторної роботи були покращені навички написання власних процедур у модульному проекті, а також були закріпленні основні навички в операціях множення чисел підвищеної розрядності.