**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний технічний університет України**

**«Київський Політехнічний Інститут»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

**Лабораторна робота №7**

*з дисципліни «Системне програмування – 1»*

*на тему: «Програмування операцій ділення чисел»*

**Виконав:**

студент 2-го курсу ФІОТ

групи ІО-44

*Барабаш Т.А.*

**Перевірив:**

Старший викладач

*Порєв В. М.*

**Київ – 2016**

**ЛАБОРАТОРНА РОБОТА №7**

*Програмування операцій ділення чисел*

**Мета:** Навчитися програмувати на асемблері ділення чисел, вивчити перетворення з двійкової у десяткову систему числення.

**I. Завдання**

1. Створити у середовищі MS Visual Studio проект з ім’ям Lab7.
2. Написати вихідний текст програми згідно варіанту завдання. У проекті мають бути три модуля на асемблері:

* головний модуль: файл main7.asm. Цей модуль створити та написати заново, частково використавши текст модуля main5.asm попередньої роботи №5;
* другий модуль: модуль module попередньої роботи №6;
* третій модуль: модуль longop попередньої роботи №6.

1. Додати у модулі процедури, які потрібні для виконання завдання. Обгрунтувати розподіл процедур по модулям.
2. У цьому проекті кожний модуль може окремо компілюватися.
3. Скомпілювати вихідний текст і отримати виконуємий файл програми.
4. Перевірити роботу програми. Налагодити програму.
5. Отримати результати – кодовані значення чисел згідно варіанту завдання.
6. Проаналізувати та прокоментувати результати, вихідний текст та дизасембльований машинний код програми.

N = 30 + 2 × 3 = 36;

y = × 2-m

**ІІ. Код програми**

**А. Lab7.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 "№7", 0

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

Caption1 db "n!", 0

Caption2 db "Function Result"

res\_fact db 608 dup(0)

text\_fact db 608 dup(0)

result db 608 dup (0)

var dd ?

eax\_buf db 4 dup(0)

.code

main:

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

push 96 ; X

push 2 ; M

call MyFunction; X / 3 \* 2 ^ m

mov [var] , eax

push offset res\_fact

push 36

call Factorial

push offset res\_fact

push offset text\_fact

push 19

push 76

call StrToDec\_MODULE

invoke MessageBoxA, 0, ADDR text\_fact, ADDR Caption, 0

push offset eax\_buf

push offset var

push 32

call StrHex\_MY

invoke MessageBoxA, 0, ADDR eax\_buf, ADDR Caption2, 0

invoke ExitProcess, 0

end main

**B. Longop.asm**

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

Factorial proc

push ebp

mov ebp,esp

mov ebx, [ebp+12] ;адреса результату

mov edi, [ebp+8] ;факторіал числа

add dword ptr[ebx], 00000001h

mov dword ptr [counter], 00000001h

factcycle:

mov ecx,0

mov esi,4

level1cycle:

mov eax, 0

mov edx, 0

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

mul dword ptr [counter]

mov dword ptr[level1 + 8\*ecx], eax

mov dword ptr[level1 + 4 + 8\*ecx], edx

inc ecx

dec esi

jnz level1cycle

mov ecx,0

mov esi,4

level2cycle:

mov eax, 0

mov edx, 0

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

mul dword ptr[counter]

mov dword ptr[level2 + 4+8\*ecx], eax

mov dword ptr[level2 + 8 + 8\*ecx], edx

inc ecx

dec esi

jnz level2cycle

mov ecx, 7

mov esi, 0

clc

sumlevels:

mov eax,0

mov eax, dword ptr[level1 + 4\*esi]

adc eax, dword ptr[level2 + 4\*esi]

mov [ebx + 4\*esi], eax

inc esi

dec ecx

jnz sumlevels

mov eax, 0

mov eax, dword ptr[counter]

add eax, 00000001h

mov dword ptr[counter], eax

cmp eax, [ebp+8]

jle factcycle

pop ebp

ret 12

Factorial endp

DIV10\_GR\_LONGOP proc

push ebp

mov ebp, esp

mov esi, [ebp + 20] ; Число

mov edi, [ebp + 16] ;Целая часть

mov ebx, [ebp + 12] ;Остаток

mov eax, [ebp + 8] ; Количество байтов

div two

mov byteNumber, eax; Количество байтов/2

push ebx

xor edx, edx ; обнуление edx

mov ecx, byteNumber; Счетчик

dec byteNumber

mov ebx, byteNumber

mov fractionalPart , 0

@cycle :

push ecx; Сохраняем значение счетчика

mov cx, 10

mov ax, word ptr[esi + 2 \* ebx]

div cx

mov fractionalPart, dx

mov word ptr[edi + 2 \* ebx], ax

dec ebx

pop ecx

dec ecx

jnz @cycle

pop ebx

mov ax, fractionalPart

mov word ptr[ebx], ax

pop ebp

ret 16

DIV10\_GR\_LONGOP endp

MyFunction proc

push ebp

mov ebp,esp

mov eax , [ebp + 12]; X

cdq

mov ecx , [ebp + 8] ; M

mov ebx , 3

div ebx

shr eax , cl

pop ebp

ret 8

MyFunction endp

**C. Module.asm**

StrToDec\_MODULE proc

push ebp

mov ebp, esp

mov esi, [ebp + 20] ;number

mov edi, [ebp + 16] ;text buf

mov eax, [ebp + 12]

mov ddNumber, eax ; number of dd

mov eax, [ebp + 8]

mov byteNumber, eax ; bytes on screan

xor ecx, ecx

xor ebx, ebx

@cycle:

push ecx

push edi

push esi; Число для перевода

push offset buf ; Целая часть

push offset decCode ; Остаток

push byteNumber ; Колическтво байтов

call DIV10\_GR\_LONGOP

pop edi

mov ebx, b

mov al, byte ptr[decCode]

add al, 48

mov byte ptr[edi + ebx], al ; запись символа в буфер текста

xor ecx, ecx

@cycleInner:

mov eax, dword ptr[buf + 4 \* ecx] ; целую часть в еах

mov dword ptr[esi + 4 \* ecx], eax ; еах в делимое

mov dword ptr[buf + 4 \* ecx], 0 ; обнуление

inc ecx

cmp ecx, ddNumber

jl @cycleInner

pop ecx

inc ecx

inc b

cmp ecx, byteNumber

jl @cycle

mov ebx, byteNumber

mov eax, byteNumber

xor edx, edx

div two

mov byteNumber, eax

dec ebx

xor ecx, ecx

@cycle1:

mov al, byte ptr[edi + ecx]; в аl первые байт буфера текста

mov ah, byte ptr[edi + ebx];в аh последние

mov byte ptr[edi + ecx], ah;менем их местами

mov byte ptr[edi + ebx], al

dec ebx

inc ecx

cmp ecx, byteNumber

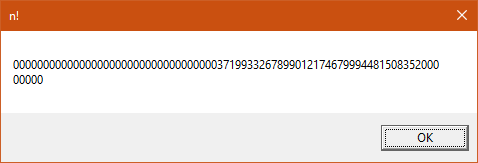
jl @cycle1

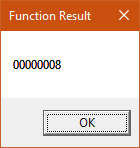
pop ebp

ret 16

StrToDec\_MODULE endp

**III. Результат**





**ІV. Висновок**

У ході виконання лабораторної роботи було реалізовано операції ділення та переводу у десяткову систему числення для чисел підвищеною розрядністю, а також створена процедура для обчислення заданої функції.