Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №7**

З дисципліни «Системне програмування»

Тема: «Програмування операцій ділення чисел»

Виконав: Перевірив:

Студент групи ІО-41 ст. викладач

Смішний Д.М. Порєв В.М.

Київ 2016

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

**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.

3. Додати у модулі процедури, які потрібні для виконання завдання. Обгрунтувати розподіл процедур по модулям.

4. У цьому проекті кожний модуль може окремо компілюватися.

5. Скомпілювати вихідний текст і отримати виконуємий файл програми.

6. Перевірити роботу програми. Налагодити програму.

7. Отримати результати – кодовані значення чисел згідно варіанту завдання. 8. Проаналізувати та прокоментувати результати, вихідний текст та дизасембльований машинний код програми.

Текст програми:

Mul\_N32\_LONGOP proc

push ebp

mov ebp, esp

mov esi, [ebp+16]

mov ebx, [ebp+12]

mov edi, [ebp+8]

mov ecx, 0

@cycle:

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

mul ebx

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

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

inc ecx

cmp ecx, 14

jb @cycle

pop ebp

ret

Mul\_N32\_LONGOP endp

StrDec proc

push ebp

mov ebp, esp

mov esi, [ebp+16] ; буфер тексту

mov edi, [ebp+12] ; число

mov ebx, [ebp+8] ; розрядність числа

mov Nbit, ebx

shr ebx, 5 ; кількість 32-бітових елементів

dec ebx

mov ecx, ebx

@copy:

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

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

dec ecx

cmp ecx, 0

jge @copy

@cycle:

push ebx

push esi

push edi

push offset number

push 10

push Nbit

push offset quotient

push offset remainder

call Div10\_LONGOP

pop edi

pop esi

pop ebx

mov al, byte ptr[remainder]

add al, 48

mov byte ptr[esi], al

inc counter

mov ecx, counter

@lshift:

mov dl, byte ptr[esi + ecx - 1]

mov byte ptr[esi + ecx - 1], 48

mov byte ptr[esi + ecx], dl

dec ecx

cmp ecx, 0

jne @lshift

mov ecx, ebx

@swap:

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

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

mov dword ptr[quotient + 4\*ecx], 0

dec ecx

cmp ecx, 0

jge @swap

mov dword ptr[remainder], 0

mov ecx, ebx

@check:

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

cmp eax, 0

jne @cycle

dec ecx

cmp ecx, 0

jge @check

mov esp, ebp

pop ebp

ret 12

StrDec endp

Div10\_LONGOP proc

push ebp

mov ebp, esp

mov esi, [ebp+24] ; ділене

mov eax, [ebp+20] ; дільник

mov ebx, [ebp+16] ; розрядність діленого

mov edi, [ebp+12] ; частка

mov ecx, [ebp+8] ; остача

mov buf, eax

shr ebx, 3

dec ebx

xor eax, eax

@cycle:

mov al, byte ptr[esi + ebx]

div byte ptr[buf]

mov byte ptr[edi + ebx], al

dec ebx

cmp ebx, 0

jge @cycle

mov byte ptr[ecx], ah

mov esp, ebp

pop ebp

ret 20

Div10\_LONGOP endp

Function\_LONGOP proc

push ebp

mov ebp, esp

mov esi, [ebp+16] ; y

mov eax, [ebp+12] ; x

mov cl, [ebp+8] ; m

cdq

mov ebx, eax

mov ebx, 11

idiv ebx

shr eax, cl

mov dword ptr[esi], eax

mov esp, ebp

pop ebp

ret 12

Function\_LONGOP endp

main:

; Факторіал в шістнадцятковій

@cycle:

push offset varA

push factorial

push offset temp

call Mul\_N32\_LONGOP

inc factorial

inc count

cmp count, 80

mov ecx, 14

@swap:

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

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

mov dword ptr[temp+4\*ecx-4], 0

dec ecx

jnz @swap

jb @cycle

push offset TextBuff1

push offset varA

push 448

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuff1, ADDR Caption1, MB\_ICONINFORMATION

; Факторіал в десятковій формі

push offset TextBuff1

push offset varA

push 448

call StrDec

invoke MessageBoxA, 0, ADDR TextBuff1, ADDR Caption2, 0

;Обчислення функції

push offset result

push 2200 ; x

push 5 ; m

call Function\_LONGOP

push offset TextBuff3

push offset result

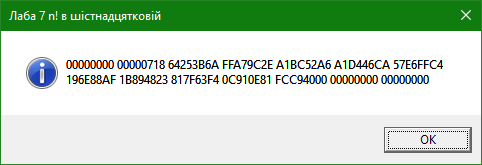
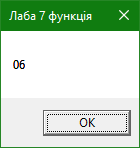
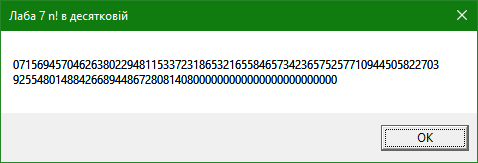
push 32

call StrDec

invoke MessageBoxA, 0, ADDR TextBuff3, ADDR Caption3, 0

invoke ExitProcess, 0

end main



**Висновки:**

Під час виконання лабораторної роботи було вдосконалено знання роботи з модулями і здобуто навичку ділити великі числа.