Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №8**

з дисципліни «Системне програмування»

Виконав:

студент групи ІО-32

Попенко Р.Л.

Перевірив:

Порєв М.В.

Київ, 2015 р.

**Тема:** Виконання операцій з плаваючою точкою та вивчення команд x87 FPU

**Мета:** Навчитися програмувати операції з плаваючою точкою на асемблері.

**Завдання:**

1. Створити у середовищі MS Visual Studio проект з ім’ям Lab8.

2. Написати вихідний текст програми згідно варіанту завдання. У проекті

мають бути головний файл main8.asm та інші модулі (за необхідності).

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 – файл числових значень згідно варіанту завдання.

7. Проаналізувати та прокоментувати результати, вихідний текст та

дизасембльований машинний код програми.

**Завдання**



**Код програми**

***MAIN8.ASM***

.686

.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

Text db 45 dup(0), 0

Caption db "Лабораторна робота №8", 0

sourse0 dd 0.5655345345

sourse1 dd 1.2134563454

sourse2 dd -78.0033454212

iterations dd 5

AArray dd 0.5, 1.0, 1.0, 0.6, 0.7

BValue dd 0.7854

BValue2 dd 2.35619

result dd 0

.code

start:

push iterations

push offset result

push offset AArray

push offset BValue

call TAN\_Function\_longop

push offset Text

push sourse2

call FloatToDec\_module

invoke MessageBoxA, 0, ADDR Text, ADDR Caption, MB\_ICONINFORMATION

invoke ExitProcess, 0

end start

***longop.ASM***

.586

.model flat, c

.xmm

.data

x dd 0

y dd 0

ten dd 10

.code

TAN\_Function\_longop proc

LOCAL temp : DWORD

;push ebp

;mov ebp, esp

mov edi, [ebp + 8] ;B

mov esi, [ebp + 12] ;A array

mov eax, [ebp + 16] ;Result

mov ecx, [ebp + 20] ;iterations

dec ecx

fld dword ptr[esi]

@cycle:

fld dword ptr [edi]

fptan

fstp temp ;1

mov edx, temp

fst temp ;tan

mov edx, temp

mov ebx, ecx

@tan\_pow:

fmul temp

dec ebx

cmp ebx, 0

jne @tan\_pow

fmul dword ptr[esi + ecx\*4]

faddp ST(1), ST(0)

dec ecx

cmp ecx, 0

jne @cycle

fstp dword ptr[eax]

;pop ebp

ret 12

TAN\_Function\_longop endp

Div10\_longop proc

push ebp

mov ebp, esp

mov edi, [ebp + 8] ; Source многоразрядное (адрес) и частное

;mov esi, [ebp + 12] ; часткове (адрес)

;mov ebx, [ebp + 16] ; залишок

mov ecx, [ebp + 12] ; розрядність в группах по 4 байта

xor edx, edx

dec ecx

@cycle:

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

div ten

mov dword ptr[edi + ecx\*4], eax ;для удобства частное записывается в source

dec ecx

cmp ecx, -1

jne @cycle

pop ebp

ret 8

Div10\_longop endp

Shift\_longop proc

push ebp

mov ebp, esp

mov edi, [ebp + 8] ; Operand

mov esi, [ebp + 12] ; Result

mov ecx, [ebp + 16] ; N

;mov edx, [ebp + 20] ; N в байтах

@cycle:

mov ebx, [ebp + 20]

dec ebx

xor dh, dh

@shift\_oper:

mov al, byte ptr[edi + ebx]

mov ah, al

and al, 0001h

shr ah, 1

mov byte ptr[edi + ebx], ah

add byte ptr[edi + ebx], dh

shl al, 7

mov dh, al

dec ebx

cmp ebx, -1

jne @shift\_oper

mov ebx, [ebp + 20]

dec ebx

@shift\_res:

mov al, byte ptr[esi + ebx]

mov ah, al

and al, 0001h

shr ah, 1

mov byte ptr[esi + ebx], ah

add byte ptr[esi + ebx], dh

shl al, 7

mov dh, al

dec ebx

cmp ebx, -1

jne @shift\_res

dec ecx

cmp ecx, 0

jne @cycle

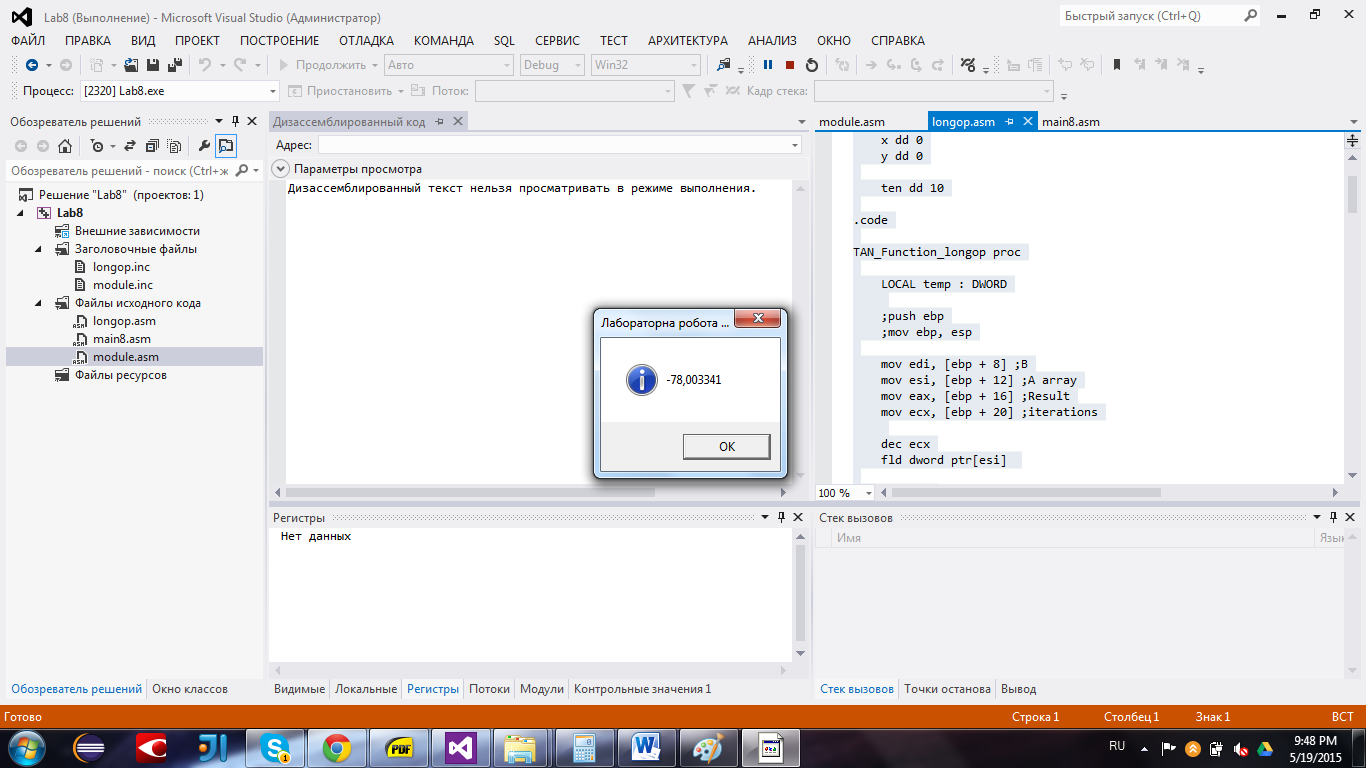
pop ebp

ret 16

Shift\_longop endp

end

**Результати виконання програми**



**Висновок:** Під час виконання лабораторної роботи я закріпив навички програмування на Асемблері, а саме: навчився програмувати операції з плаваючою точкою на асемблері.