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 №8**

З дисципліни «Системне програмування»

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

Виконав: Перевірив:

Студент групи ІО-41 ст. викладач

Смішний Д.М. Порєв В.М.

Київ 2016

**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.

Текст програми:

FloatDec\_MY proc

push ebp

mov ebp, esp

mov esi, [ebp + 8]

mov edi, [ebp + 12]

mov eax, esi

and eax, 80000000h

cmp eax, 0

je @end\_sign

mov byte ptr[edi], 45

inc edi

@end\_sign:

mov ecx, edi

mov eax, esi

and eax, 7F800000h

shr eax, 23

cmp eax, 0

jne @next

mov byte ptr[edi], 48

jmp @endproc

@next:

cmp eax, 0FFh

jne @next2

mov byte ptr[edi], 78

mov byte ptr[edi + 1], 65

mov byte ptr[edi + 2], 78

jmp @endproc

@next2:

sub eax, 7Fh

cmp eax, 0

jge @next3

mov byte ptr[edi], 48

inc ecx

mov ebx, esi

and ebx, 7FFFFFh

add ebx, 800000h

mov edx, 0FFFFFFFFh ;-1

imul edx

mov edx, ecx

mov ecx, eax

shr ebx, cl

mov ecx, edx

jmp @fraction

@next3:

jg @next4

mov byte ptr[edi], 49

inc ecx

mov ebx, esi

and ebx, 7FFFFFh

jmp @fraction

@next4:

push ecx

mov ecx, 23

sub ecx, eax

push ecx

mov eax, esi

and eax, 7FFFFFh

add eax, 800000h

xor ebx, ebx

mov ebx, 1

shl ebx, cl

mov edx, ebx

@mask:

inc cl

shl ebx, 1

add ebx, edx

cmp cl, 24

jne @mask

mov edx, eax

and edx, ebx

mov ebx, eax

sub ebx, edx

pop ecx

shr edx, cl

mov eax, 23

sub eax, ecx

mov ecx, eax

shl ebx, cl

mov eax, edx

pop ecx

push ebx

mov ebx, 10

@full\_part:

xor edx, edx

div ebx

add edx, 48

mov byte ptr[ecx], dl

inc ecx

cmp eax, 0

jne @full\_part

mov eax, ecx

dec eax

@reverse:

xor edx, edx

mov dh, byte ptr[eax]

mov dl, byte ptr[edi]

mov byte ptr[eax], dl

mov byte ptr[edi], dh

inc edi

dec eax

cmp edi, eax

jl @reverse

pop ebx

@fraction:

mov byte ptr[ecx], 44

inc ecx

mov ax, 6

@cycle:

shl ebx, 1

mov edx, ebx

shl edx, 2

add ebx, edx

mov edx, ebx

and edx, 0FF800000h

shr edx, 23

add dl, 48

mov [ecx], dl

and ebx, 7FFFFFh

inc ecx

dec ax

cmp ax, 0

jne @cycle

@endproc:

pop ebp

ret 8

FloatDec\_MY endp

main:

fld valueA11

fmul valueA22

fld valueA21

fmul valueA12

fsub

fst buffOperand1

fld valueB1

fmul valueA22

fld valueB2

fmul valueA12

fsub

fst buffOperand2

fld valueB2

fmul valueA11

fld valueB1

fmul valueA21

fsub

fst buffOperand3

fld buffOperand2

fdiv buffOperand1

fstp resultX1

fld buffOperand3

fdiv buffOperand1

fstp resultX2

push offset X1Text

push resultX1

call FloatDec\_MY

invoke MessageBoxA, 0, ADDR X1Text, ADDR Caption1, 0

push offset X2Text

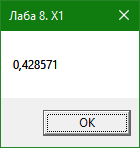
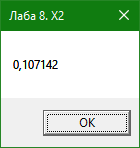
push resultX2

call FloatDec\_MY

invoke MessageBoxA, 0, ADDR X2Text, ADDR Caption2, 0

invoke ExitProcess, 0

end main



**Висновки:**

Під час виконання лабораторної роботи було вдосконалено знання роботи з модулями і здобуто навичку ділити числа з плаваючою точкою за допомогою FPU x87, розв’язуючи завдання лабораторної роботи.