Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №8

з дисципліни «Системне програмування»

на тему:

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

Виконала:

студентка 2-го курсу

групи ІО-32

Руденко Т.А.

Номер залікової книжки: 3224

Київ - 2015 р.

Мета: Навчитися програмувати операції з плаваючою точкою на асемблері.

Лістинг:

.586

.model flat, stdcall

option casemap :none

include d:\\masm32\include\windows.inc

include d:\\masm32\include\kernel32.inc

include d:\\masm32\include\user32.inc

include module.inc

include longop.inc

includelib d:\\masm32\lib\kernel32.lib

includelib d:\\masm32\lib\user32.lib

.data

Caption1 db "Результат",0

n dd 3.0

a dd 4 dup(0)

b dd 0

res dd 0

one dd 1.0

text dt ?

buf dd 0

.code

main:

mov dword ptr[a], 01000000111010000000000000000000b ; 7.25

mov dword ptr[a+4], 01000000101100000000000000000000b ; 5,5

mov dword ptr[a+8], 01000000001010000000000000000000b ; 2.625

mov dword ptr[a+12], 01000000001010000000000000000000b ; 2.625

mov dword ptr[b], 01000000010000000000000000000000b ; 3

mov ecx, 3

@function:

fld b

fmul n

fsin

fmul dword ptr[a+4\*ecx]

fld buf

faddp st(1), st(0)

fst buf

fld n

fsub one

fstp dword ptr[n]

dec ecx

cmp ecx, 0

jne @function

;fstp dword ptr[buf]

fld buf

fadd [a]

fstp dword ptr[buf]

;xor eax, eax

push offset buf

push offset text

push 10

call FloatToDec

invoke MessageBoxA, 0, ADDR text, ADDR Caption1, 0

invoke ExitProcess, 0

end main

.586

.model flat, c

.data

base dd 0

tmpEBX dd 0

resAddr dd 0

tmp dd 10 dup(0)

expon dt 0

mant dt 0

intMask dw 0000011110000000b

.code

FloatToDec proc

push ebp

mov ebp,esp

mov eax, [ebp+16]

mov ebx, [ebp+12]

mov edi, [ebp+8]

mov cx, word ptr[eax+2]

shr cx, 7

sub cl, 127

mov edx, dword ptr[eax]

mov dword ptr[expon], edx

and word ptr[expon+2], 0000000001111111b

or word ptr[expon+2], 0000000010000000b

mov edx, dword ptr[expon]

mov dword ptr[mant], edx

shl dword ptr[mant], cl

and dword ptr[mant+2], 0000000001111111b

mov ch, 23

sub ch, cl

mov cl, ch

shr dword ptr[expon], cl

mov byte ptr[ebx], 02Eh

mov esi, 0

@decimalWrite:

mov edx, dword ptr[mant]

shl dword ptr[mant], 1

shl edx, 3

add dword ptr[mant], edx

mov dx, intMask

and dx, word ptr[mant+2]

and word ptr[mant+2], 0000000001111111b

shr dx, 7

add dl, 48

mov byte ptr[ebx+esi+1], dl

inc esi

cmp esi, 6

jl @decimalWrite

push offset expon

push 8

push ebx

call Int\_to\_Dec

pop ebp

ret 12

FloatToDec endp

Int\_to\_Dec proc

push ebp

mov ebp,esp

mov esi, [ebp+16]

mov ebx, [ebp+12]

mov ecx, [ebp+8]

sub ebx, 4

mov base, 10

mov tmpEBX, ebx

mov edi, 10

@cycle:

mov ebx, tmpEBX

mov edx, 0

mov eax, dword ptr[esi]

div edi

mov dword ptr[esi], eax

add dl, 48

mov al, byte ptr[ecx+7]

mov byte ptr[ecx+8], al

shl dword ptr[ecx+4], 8

mov al, byte ptr[ecx+3]

mov byte ptr[ecx+4], al

shl dword ptr[ecx], 8

mov byte ptr[ecx], dl

mov edx, tmpEBX

sub edx, 4

cmp dword ptr[esi+edx], 0

jne @cycle

pop ebp

ret 12

Int\_to\_Dec endp

end