Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №11

з дисципліни «Системне програмування»

на тему:

**«Програмування команд SSE у модулях на асемблері»**

Виконала:

студентка 2-го курсу

групи ІО-32

Руденко Т.А.

Номер залікової книжки: 3224

Київ - 2015 р.

Мета: Навчитися програмувати модулі на асемблері, у яких містяться команди

SSE, команди x87 FPU, а також використовувати такі модулі у проектах C++.

Лістинг:

Lab11.cpp

void myVectSSE(HWND hWnd)

{

float oA[8] = { 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0 };

float oB[8] = { 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 };

float result0 = 0;

VectSSE(&result0, oB, oA, 8);

std::string text = std::to\_string(result0);

MessageBox(hWnd, text.c\_str(), "Скалярний добуток SSE", MB\_OK);

SYSTEMTIME st;

long tst, ten;

GetLocalTime(&st);

tst = 60000 \* (long)st.wMinute + 1000 \* (long)st.wSecond + (long)st.wMilliseconds;

for (long i = 0; i<1000000; i++) //повторюємо мільйон разів

{

VectSSE(&result0, oB, oA, 8);

}

GetLocalTime(&st);

ten = 60000 \* (long)st.wMinute + 1000 \* (long)st.wSecond + (long)st.wMilliseconds - tst;

std::string text2 = std::to\_string(ten);

MessageBox(hWnd, text2.c\_str(), "Час виконання", MB\_OK);

}

void myVectFPU(HWND hWnd)

{

float oA[8] = { 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0 };

float oB[8] = { 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 };

float result0 = 0;

VectFPU(&result0, oB, oA, 8);

std::string text = std::to\_string(result0);

MessageBox(hWnd, text.c\_str(), "Скалярний добуток FPU", MB\_OK);

SYSTEMTIME st;

long tst, ten;

GetLocalTime(&st);

tst = 60000 \* (long)st.wMinute + 1000 \* (long)st.wSecond + (long)st.wMilliseconds;

for (long i = 0; i<1000000; i++) //повторюємо мільйон разів

{

VectFPU(&result0, oB, oA, 8);

}

GetLocalTime(&st);

ten = 60000 \* (long)st.wMinute + 1000 \* (long)st.wSecond + (long)st.wMilliseconds - tst;

std::string text2 = std::to\_string(ten);

MessageBox(hWnd, text2.c\_str(), "Час виконання", MB\_OK);

}

void myVectC(HWND hWnd)

{

double oA[8] = { 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0 };

double oB[8] = { 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 };

double result = 0;

SYSTEMTIME st;

long tst, ten;

GetLocalTime(&st);

tst = 60000 \* (long)st.wMinute + 1000 \* (long)st.wSecond + (long)st.wMilliseconds;

for (long i = 0; i<1000000; i++)

{

result = 0;

for (long i = 0; i < 8; i++)

{

result = result + oA[i] \* oB[i];

}

}

GetLocalTime(&st);

ten = 60000 \* (long)st.wMinute + 1000 \* (long)st.wSecond + (long)st.wMilliseconds - tst;

TCHAR buf[32];

\_itoa\_s(result, buf, 10);

MessageBox(hWnd, buf, "Скалярний добуток С++", MB\_OK);

\_itoa\_s(ten, buf, 10);

MessageBox(hWnd, buf, "Час виконання", MB\_OK);

}

Vectsse.asm

.data

temp dd 4 dup(0)

.code

VectSSE proc dest:DWORD, pB:DWORD, pA:DWORD, bits:DWORD

mov edx, bits ; n

mov esi, pA ; a

mov ebx, pB ; b

mov edi, dest ; res

cycle:

sub edx, 4

movups xmm0, [esi+edx\*4]

movups xmm1, [ebx+edx\*4]

mulps xmm0, xmm1

addps xmm2, xmm0

cmp edx, 0

jne cycle

haddps xmm2, xmm2

haddps xmm2, xmm2

movups temp, xmm2

mov eax, dword ptr[temp]

mov dword ptr[edi], eax

ret

VectSSE endp

end

Vectfpu.asm

.data

temp dd 4 dup(0)

.code

VectFPU proc dest:DWORD, pB:DWORD, pA:DWORD, bits:DWORD

mov edx, bits ; n

mov esi, pA ; a

mov ebx, pB ; b

mov edi, dest ; res

fld dword ptr[temp]

cycle:

dec edx

fld dword ptr[esi+edx\*4]

fmul dword ptr[ebx+edx\*4]

faddp st(1), st(0)

cmp edx, 0

jne cycle

fstp dword ptr[edi]

ret

VectFPU endp

end