Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №10

з дисципліни «Системне програмування»

на тему:

**«Використання у проекті C++ модулів на асемблері»**

Виконала:

студентка 2-го курсу

групи ІО-32

Руденко Т.А.

Номер залікової книжки: 3224

Київ - 2015 р.

Мета: Навчитися створювати програми на С++ з використанням модулів на

асемблері.

Лістинг:

Lab10.cpp

void MyWork1(HWND hWnd)

{

long oA[11] = { 0x00000004, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000 };

long oB[11] = { 0x00000005, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000 };

long oC[11] = { 0x00000002, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000 };

long oO[11] = { 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000 };

long result1[22] = { 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000 };

long result2[22] = { 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000 };

long result3[22] = { 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000 };

long result4[22] = { 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000 };

long result5[22] = { 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000 };

long result6[22] = { 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000 };

char TextBuf[352];

char TextBuf10[44];

Mul\_NxN\_LONGOP(result1, oB, oB);

Add\_LONGOP(11, result2, result1, oA);

Mul\_NxN\_LONGOP(result3, oC, oC);

Mul\_NxN\_LONGOP(result4, result3, oC);

Sub\_LONGOP(22, result5, result4, result2);

Add\_LONGOP(22, result6, result5, result6);

StrToDec(44, 11, TextBuf10, result6);

MessageBox(hWnd, TextBuf10, "Результат A+B^2-C^3", MB\_OK);

StrHex\_MY(352, result6, TextBuf);

MessageBox(hWnd, TextBuf, "Результат A+B^2-C^3", MB\_OK);

}

longop.asm

.data

x dd 1h

bitNumber dd ?

a dd 0

b dd 0

r dd 0

.code

Add\_LONGOP proc bits:DWORD, dest:DWORD, pB:DWORD, pA:DWORD

mov esi, pA ;ESI = адреса A

mov ebx, pB ;EBX = адреса B

mov edi, dest ;EDI = адреса результату

mov ecx, bits; ECX = потрібна кількість повторень

mov edx, 0

clc ; обнулює біт CF регістру EFLAGS

mov edx, 0

cycle:

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

adc eax, dword ptr[ebx+4\*edx] ; додавання групи з 32 бітів

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

inc edx; модифікація зсуву

dec ecx ; лічильник зменшуємо на 1

jnz cycle

ret

Add\_LONGOP endp

Sub\_LONGOP proc bits:DWORD, dest:DWORD, pB:DWORD, pA:DWORD

mov esi, pA ;ESI = адреса A

mov ebx, pB ;EBX = адреса B

mov edi, dest ;EDI = адреса результату

mov ecx, bits; ECX = потрібна кількість повторень

clc ; обнулює біт CF регістру EFLAGS

mov edx, 0

cycle:

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

sbb eax, dword ptr[ebx+4\*edx] ; додавання групи з 32 бітів

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

inc edx; модифікація зсуву

dec ecx ; лічильник зменшуємо на 1

jnz cycle

ret

Sub\_LONGOP endp

Mul\_NxN\_LONGOP proc dest:DWORD, p2:DWORD, p1:DWORD

mov esi, p1

mov edi, p2

mov ebx, dest

mov dword ptr[a],0

mov dword ptr[b],0

mov dword ptr[r],0

mov ecx, 18

@cycle:

push ecx

mov ecx, 8

@cycleInner:

push ecx

mov ecx, a

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

mov ecx, b

mul dword ptr[edi + 4 \* ecx]

mov ecx, r

clc

adc eax, dword ptr[ebx + 4 \* ecx]

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

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

adc edx, dword ptr[ebx + 4 \* ecx + 4]

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

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

inc a

inc r

pop ecx

dec ecx

jnz @cycleInner

inc b

xor eax, eax

mov a, eax

mov eax, b

mov r, eax

pop ecx

dec ecx

jnz @cycle

ret

Mul\_NxN\_LONGOP endp

End