**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний технічний університет України**

**«Київський Політехнічний Інститут»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

**Лабораторна робота №10**

*з дисципліни «Системне програмування – 1»*

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

**Виконав:**

студент 2-го курсу ФІОТ

групи ІО-44

*Барабаш Т.А.*

**Перевірив:**

Старший викладач

*Порєв В. М.*

**Київ – 2016**

**ЛАБОРАТОРНА РОБОТА №10**

*Використання у проекті C++ модулів на асемблері*

**Мета:** Навчитися використовувати у програмах на асемблері функції Windows динамічного виділення пам’яті та запису файлів

1. **Завдання**
2. Створити у середовищі MS Visual Studio проект C++ з ім’ям Lab10.
3. Написати вихідний текст програми згідно варіанту завдання. У проекті мають бути такі файли вихідного тексту:

- головний файл: lab10.cpp

- файли двох модулів на асемблері: module.asm та longop.asm.

1. У цьому проекті кожний модуль може окремо компілюватися.
2. Скомпілювати вихідний текст і отримати виконуємий файл програми.
3. Перевірити роботу програми. Налагодити програму.
4. Отримати результати – кодовані значення чисел згідно варіанту завдання.
5. Проаналізувати та прокоментувати результати, вихідний текст та машинний код програми

|  |  |  |
| --- | --- | --- |
| 3. | AB + C – D | 512 |

**ІІ. Код програми**

void myWork(HWND hWnd)

{

const long len = 16;

long oA[len] = { 0xab010001, 0x0001cbb1, 0x00d100a1, 0x00d10001,

0xab010001, 0x0001cbb1, 0x00d100a1, 0x00d10001,

0xab010001, 0x0001cbb1, 0x00d100a1, 0x00d10001,

0xab010001, 0x0001cbb1, 0x00d100a1, 0x00d10001};

long oB[len] = { 0x0a001001, 0x00010ee1, 0x00010001, 0x00010001,

0xab010001, 0x0001cbb1, 0x00d100a1, 0x00d10001,

0xab010001, 0x0001cbb1, 0x00d100a1, 0x00d10001,

0xab010001, 0x0001cbb1, 0x00d100a1, 0x00d10001};

long oC[len \* 2] = { 0x0ccb1301, 0x00015601, 0x78010001, 0x00910001, 0x00410001, 0x00010001, 0x00010001, 0x00010001,

0x0ccb1301, 0x00015601, 0x78010001, 0x00910001, 0x00410001, 0x00010001, 0x00010001, 0x00010001,

0x0ccb1301, 0x00015601, 0x78010001, 0x00910001, 0x00410001, 0x00010001, 0x00010001, 0x00010001,

0x0ccb1301, 0x00015601, 0x78010001, 0x00910001, 0x00410001, 0x00010001, 0x00010001, 0x00010001};

long oD[len \* 2] = { 0x00001001, 0x0ee10001, 0x00010d01, 0x0b010001, 0x0a010001, 0x00310001, 0x00010001, 0x00010001,

0x00001001, 0x0ee10001, 0x00010d01, 0x0b010001, 0x0a010001, 0x00310001, 0x00010001, 0x00010001,

0x00001001, 0x0ee10001, 0x00010d01, 0x0b010001, 0x0a010001, 0x00310001, 0x00010001, 0x00010001,

0x00001001, 0x0ee10001, 0x00010d01, 0x0b010001, 0x0a010001, 0x00310001, 0x00010001, 0x00010001};

long oAB[len \* 2] = { 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0,

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

long res1[len \* 2] = { 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0,

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

long res2[len \* 2] = { 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0,

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

char TextBuf[len \* 2 \* 32];

Mul\_NN\_LONGOP(len, oA, len, oB, oAB);

StrHex\_MY(len \* 2 \* 32, oAB, TextBuf);

MessageBox(hWnd, TextBuf, "Результат A \* B", MB\_OK);

Add\_LONGOP(len \* 2, oAB, oC, res1);

StrHex\_MY(len \* 2 \* 32, res1, TextBuf);

MessageBox(hWnd, TextBuf, "Результат A \* B + C", MB\_OK);

Sub\_LONGOP(len \* 2, res1, oD, res2);

StrHex\_MY(len \* 2 \* 32, res2, TextBuf);

MessageBox(hWnd, TextBuf, "Результат A \* B + C - D в 16", MB\_OK);

StrDec\_MY(len \* 2 \* 32, res2, TextBuf);

MessageBox(hWnd, TextBuf, "Результат A \* B + C - D в 10", MB\_OK);

}

.586

.model flat, c

.code

Add\_LONGOP proc counter:DWORD, pA:DWORD, pB:DWORD, res:DWORD

mov edi, res

mov ebx, pB

mov esi, pA

mov ecx, counter

clc

mov edx, 0

@cycle:

mov eax, dword ptr [esi + edx]

adc eax, dword ptr [ebx + edx]

mov dword ptr [edi + edx], eax

inc edx

inc edx

inc edx

inc edx

dec ecx

jnz @cycle

@exitp:

ret

Add\_LONGOP endp

Sub\_LONGOP proc counter:DWORD, pA:DWORD, pB:DWORD, res:DWORD

mov edi, res

mov ebx, pB

mov esi, pA

mov ecx, counter

clc

mov edx, 0

@cycle:

mov eax, dword ptr [esi + edx]

sbb eax, dword ptr [ebx + edx]

mov dword ptr [edi + edx], eax

inc edx

inc edx

inc edx

inc edx

dec ecx

jnz @cycle

@exitp:

ret

Sub\_LONGOP endp

Mul\_N32\_LONGOP proc counter:DWORD, N:DWORD, pA:DWORD, res:DWORD

mov edi, res

mov ebx, N

mov esi, pA

mov ecx, counter

clc

mov ebp, ecx

dec ebp

@zero:

mov dword ptr [edi + ebp \* 4], 0

dec ebp

jge @zero

mov ebp, 0

@cycle:

add ebp, 4

mov eax, dword ptr [esi + ebp - 4]

mul ebx

add dword ptr [edi + ebp - 4], eax

adc dword ptr [edi + ebp], edx

dec ecx

jnz @cycle

ret

Mul\_N32\_LONGOP endp

Mul\_NN\_LONGOP proc aLen:DWORD, pA:DWORD, bLen:DWORD, pB:DWORD, res:DWORD

mov edi, res

mov ebx, pB

mov ecx, bLen

mov maxCounter2, ecx

mov esi, pA

mov ecx, aLen

mov maxCounter1, ecx

mov counter1, 0h

@outer:

mov eax, counter1

cmp eax, maxCounter1

jge @exitp

mov counter2, 0h

@inner:

mov ecx, counter1

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

mov ecx, counter2

mul dword ptr [ebx + 4 \* ecx]

add ecx, counter1

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

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

inc counter2

mov eax, counter2

cmp eax, maxCounter2

jl @inner

inc counter1

jmp @outer

@exitp:

ret

Mul\_NN\_LONGOP endp

Div\_N32\_LONGOP proc aLen:DWORD, pA:DWORD, nDivider:DWORD, res:DWORD

mov esi, res

mov ecx, nDivider

mov divider, ecx

mov ebx, pA

mov ecx, aLen

xor edx, edx

dec ecx

@cycle:

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

div divider

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

dec ecx

jge @cycle

@exitp:

ret

Div\_N32\_LONGOP endp

Copy\_LONGOP proc dest:DWORD, src:DWORD, len:DWORD

mov edi, dest

mov ebx, src

mov ecx, len

dec ecx

@copy\_my:

mov eax, [ebx + 4 \* ecx]

mov [edi + 4 \* ecx], eax

dec ecx

jge @copy\_my

ret

Copy\_LONGOP endp

.data

counter1 dd 0h

counter2 dd 0h

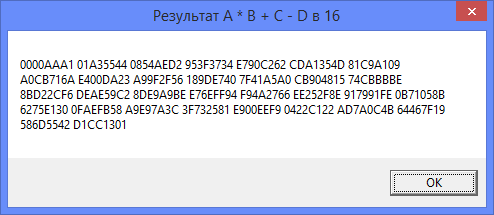
maxCounter1 dd 0h

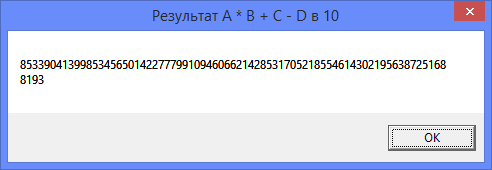
maxCounter2 dd 0h

divider dd 1

end

**III. Результат**





**ІV. Висновок**

У ході виконання лабораторної роботи було створено програму на С++ для обчислення заданої функції, використовуючи раніше написані на асемблері модулі.