**Міністерство освіти і науки України**

**Національний університет «Львівська політехніка»**



Звіт

до лабораторної роботи № 2

з дисципліни «Системне програмування»

на тему: «ЗМІШАНЕ ПРОГРАМУВАННЯ НА МОВАХ С ТА АСЕМБЛЕР»

Варіант №6

Виконав:

ст.гр. КІ-38

Карпюк Н.М.

Прийняв:

Козак Н.Б.

**Львів 2020**

**Мета роботи :** оволодіти навиками створення програм, частини яких написані різними мовами програмування. Засвоїти правила взаємодії між програними модулями різних.

ЗАВДАННЯ

1. Створити дві програми. Прша програма реалізує взаємовиклики С – ASM та здійснює обчислення, заданого виразу, згідно варіанту (Табл. 2.3). Програма повинна складатися з кількох модулів, передача параметрів між якими здійснюється через стек. Константа передається через спільну пам’ять. Основний модуль – створюється мовою С. Він повинен забезпечувати:

• ввід даних з клавіатури;

• виклик підпрограми обчислення виразу;

• вивід на екран результату обчислення виразу.

Модуль безпосередніх обчислень – здійснює всі обчислення виразу. Створюється мовою Assembler.

2. Друга програма реалізує взаємовиклики С – ASM – С та здійснює обчислення, заданого виразу, згідно варіанту. Програма повинна складатися з кількох модулів, передача параметрів між якими здійснюється через стек. Основний модуль – створюється мовою С. Він повинен забезпечувати:

• ввід даних з клавіатури;

• виклик підпрограми обчислення виразу;

Модуль безпосередніх обчислень – здійснює всі обчислення і вивід на екран результату обчислення виразу викликом стандартної функції printf() . Створюється мовою Assembler.

3. Відлагодити та протестувати програми. Результати роботи програм продемонструвати викладачу.

4. Скласти звіт про виконану роботу з приведенням тексту програми та коментарів до неї, а також результатів її роботи.

5. Дати відповідь на контрольні запитання.

Варіанти завдань:

|  |  |  |
| --- | --- | --- |
| 6. | X=K+B2/4-D2\*4-E1 | 6DD02316 |

Код програми:

1 Програма (c-asm)

c\_module.cpp

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <conio.h>

#include <stdio.h>

extern "C" int pruklad(int, int, char);

extern "C" {

short B = 0, D = 0;

char E = 0; }

int main() {

int K = 0x6DD02316;

printf("X = K + B2 / 4 - D2 \* 4 - E1 where K = 6DD02316");

printf("\n");

printf("Char(E) from 0 to 255\n");

printf("Enter numbers:\n");

printf("Enter B = ");

scanf("%d", &B);

printf("Enter D = ");

scanf("%d", &D);

printf("Enter E = ");

scanf("%d", &E);

printf("Result in C(from\_C): %d ", K + B / 4 - D \* 4 - E);

printf("\n");

int X = pruklad(B, D, E);

printf("Result in C from ASM(from\_C) %d", X);

printf("\n");

\_getch();

return 0;

}

asm\_module.asm

.386

.model flat, c

option casemap:none

PUBLIC pruklad

.data

\_T1 dword 0

\_T2 dword 0

\_T3 dword 0

\_T4 dword 0

\_T5 dword 0

\_K EQU 6DD02316h

.code

pruklad PROC

push ebp

mov ebp,esp

;T1 = D2\*4

mov eax, dword ptr[ebp+12]

mov ecx,4

imul ecx

mov \_T1,eax

;T2 = B2/4

mov eax, dword ptr [ebp+8]

mov ecx, 4

idiv ecx

mov \_T2,eax

;T3 = K+T2

mov eax,\_K

add eax,\_T2

mov \_T3,eax

;T4 = T3-T1

mov eax,\_T3

sub eax,\_T1

mov \_T4,eax

;T5 = T4-E1

mov eax,\_T4

sub ax, word ptr[ebp+16]

mov \_T5,eax

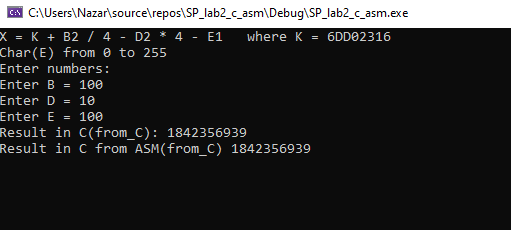
pop ebp

ret

pruklad ENDp

END

Результат виконання:



2 Програма (c-asm-c)

c\_module.cpp

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <conio.h>

#include <stdio.h>

extern "C" int pruklad(int, int, char);

extern "C" {

short B = 0, D = 0;

char E = 0; }

int main() {

int K = 0x6DD02316;

printf("X = K + B2 / 4 - D2 \* 4 - E1 where K = 6DD02316");

printf("\n");

printf("Char(E) from 0 to 255\n");

printf("Enter numbers:\n");

printf("Enter B = ");

scanf("%d", &B);

printf("Enter D = ");

scanf("%d", &D);

printf("Enter E = ");

scanf("%d", &E);

printf("Result in C(from\_C): %d ", K + B / 4 - D \* 4 - E);

printf("\n");

pruklad(B, D, E);

\_getch();

return 0;

}

asm\_module.asm

.386

.model flat, c

printf proto arg1:ptr byte, printlist:vararg

option casemap:none

PUBLIC pruklad

.data

\_T1 dword 0

\_T2 dword 0

\_T3 dword 0

\_T4 dword 0

\_T5 dword 0

\_K EQU 6DD02316h

Msg db 'Result in Asm(from\_Asm): %d', 0

.code

pruklad PROC

push ebp

mov ebp,esp

;T1 = D2\*4

mov eax, dword ptr[ebp+12]

mov ecx,4

imul ecx

mov \_T1,eax

;T2 = B2/4

mov eax, dword ptr [ebp+8]

mov ecx, 4

idiv ecx

mov \_T2,eax

;T3 = K+T2

mov eax,\_K

add eax,\_T2

mov \_T3,eax

;T4 = T3-T1

mov eax,\_T3

sub eax,\_T1

mov \_T4,eax

;T5 = T4-E1

mov eax,\_T4

sub ax, word ptr[ebp+16]

mov \_T5,eax

pop ebp

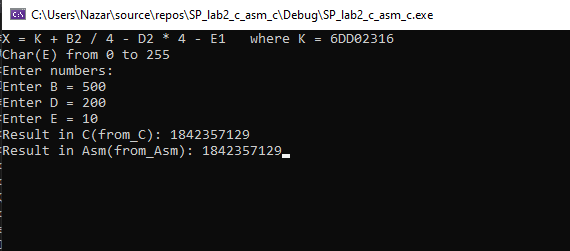
invoke printf, addr Msg, eax

ret

pruklad ENDp

END

Результат виконання:



Висновок: на даній лабораторній роботі я ознайомився з змішаним програмуванням на мовах C та Assembler.