**ЛАБОРАТОРНА РОБОТА № 4**

**ОБЧИСЛЕННЯ ПРОСТИХ ЦІЛОЧИСЛЕНИХ**

**ВИРАЗІВ НА МОВІ ASSEMBLER**

***Мета:*** ознайомитися з типами цілочисельних даних платформ Win16 та Win32; ознайомитися з основними командами мови Assembler; набути практичних навичок в написанні програм для обчислення простих цілочисельних виразів на мові Assembler.

**Хід роботи:**

**Завдання 1**: Написати програму для обчислення заданого цілочисленого виразу для початкових даних в знаковому форматі довжиною 8 біт, використо-вуючи арифметичні операції ADD, INC, SUB, DEC, NEG. Виконати покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні.

**24.**  –(d+2)+a–e–b+c+(f–1)

Лістинг програми:

#include "stdafx.h"

#include <stdio.h>

#include <cstdlib>

int main(int argc, \_TCHAR\* argv[])

{

signed char a, b, c, d, e, f, res\_c, res\_asm;

printf("a = "); scanf\_s("%d", &a);

printf("b = "); scanf\_s("%d", &b);

printf("c = "); scanf\_s("%d", &c);

printf("d = "); scanf\_s("%d", &d);

printf("e = "); scanf\_s("%d", &e);

printf("f = "); scanf\_s("%d", &f);

res\_c = -(d + 2) + a - e - b + c + (f - 1);

printf("Result C = %d\n", res\_c);

\_\_asm {

mov al, d; // <al> = d

add al, 2; // <al> = d + 2

neg al; // <al> = -(d + 2)

add al, a; // <al> = -(d + 2) + a

sub al, e; // <al> = -(d + 2) + a - e

sub al, b; // <al> = -(d + 2) + a - e - b

add al, c; // <al> = -(d + 2) + a - e - b + c

mov bl, f; // <bl> = f

dec bl; // <bl> = f - 1

add al, bl; // <al> = -(d + 2) + a - e - b + c + (f - 1)

mov res\_asm, al; // res\_asm = al

}

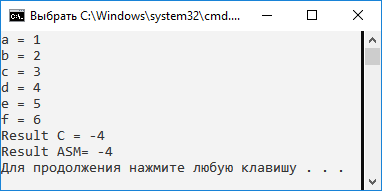
printf("Result ASM= %d\n", res\_asm);

system("Pause");

return 0;

}

Результат виконання програми:



**Значення регістрів при покроковому виконанні**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Крок** | **Команда** | **Значення регістра** | | | | **EFLAGS/FLAGS (CF, OF)** |
| **al** | **ah** | **bl** | **bh** |
| **1** | mov al, d | 4 | н/в | н/в | н/в | **-** |
| **2** | add al, 2 | 6 | н/в | н/в | н/в | **-** |
| **3** | neg al | -6 | н/в | н/в | н/в | **-** |
| **4** | add al, a | -5 | н/в | н/в | н/в | **-** |
| **5** | sub al, e | -10 | н/в | н/в | н/в | **-** |
| **6** | sub al, b | -12 | н/в | н/в | н/в | **-** |
| **7** | add al, c | -9 | н/в | н/в | н/в | **-** |
| **8** | mov bl, f | н/в | н/в | 6 | н/в | **-** |
| **9** | dec bl | н/в | н/в | 5 | н/в | **-** |
| **10** | add al, bl | -4 | н/в | н/в | н/в | **-** |
| **11** | mov res\_asm, al | -4 | н/в | н/в | н/в | **-** |

**Завдання 2**: Написати програму для обчислення заданого цілочисленого виразу для початкових даних в знаковому форматі довжиною 16 біт, використовуючи арифметичні операції ADD, INC, SUB, DEC, NEG. Виконати покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні.

Лістинг програми:

#include "stdafx.h"

#include <stdio.h>

#include <cstdlib>

int main(int argc, \_TCHAR\* argv[])

{

short int a, b, c, d, e, f, res\_c, res\_asm;

printf("a = "); scanf\_s("%hi", &a);

printf("b = "); scanf\_s("%hi", &b);

printf("c = "); scanf\_s("%hi", &c);

printf("d = "); scanf\_s("%hi", &d);

printf("e = "); scanf\_s("%hi", &e);

printf("f = "); scanf\_s("%hi", &f);

res\_c = -(d + 2) + a - e - b + c + (f - 1);

printf("Result C = %hi\n", res\_c);

\_\_asm {

mov ax, d; // <ax> = d

add ax, 2; // <ax> = d + 2

neg ax; // <ax> = -(d + 2)

add ax, a; // <ax> = -(d + 2) + a

sub ax, e; // <ax> = -(d + 2) + a - e

sub ax, b; // <ax = -(d + 2) + a - e - b

add ax, c; // <ax> = -(d + 2) + a - e - b + c

mov bx, f; // <bx> = f

dec bx; // <bx> = f - 1

add ax, bx; // <ax> = -(d + 2) + a - e - b + c + (f - 1)

mov res\_asm, ax; // res\_asm = ax

}

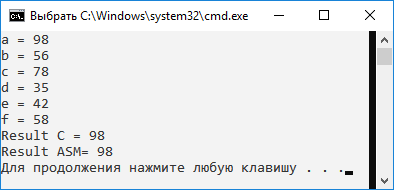
printf("Result ASM= %hi\n", res\_asm);

system("Pause");

return 0;

}

Результат виконання програми:



**Значення регістрів при покроковому виконанні**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Крок** | **Команда** | **Значення регістра** | | | **EFLAGS/FLAGS (CF, OF)** |
| **ax** | **bx** |  | | |
| **1** | mov ax, d | 35 | н/в | **-** | | |
| **2** | add ax, 2 | 37 | н/в | **-** | | |
| **3** | neg ax | -37 | н/в | **-** | | |
| **4** | add ax, a | 61 | н/в | **-** | | |
| **5** | sub ax, e | 19 | н/в | **-** | | |
| **6** | sub ax, b | -37 | н/в | **-** | | |
| **7** | add ax, c | 41 | н/в | **-** | | |
| **8** | mov bx, f | н/в | 58 | **-** | | |
| **9** | dec bx | н/в | 57 | **-** | | |
| **10** | add ax, bx | 98 | н/в | **-** | | |
| **11** | mov res\_asm, ax | 98 | н/в | **-** | | |

**Завдання 3**: Написати програму для обчислення заданого цілочисленого виразу для початкових даних в знаковому форматі довжиною 32 біт, використовуючи арифметичні операції ADD, INC, SUB, DEC, NEG. Виконати покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні.

Лістинг програми:

#include "stdafx.h"

#include <stdio.h>

#include <cstdlib>

int main(int argc, \_TCHAR\* argv[])

{

long a, b, c, d, e, f, res\_c, res\_asm;

printf("a = "); scanf\_s("%li", &a);

printf("b = "); scanf\_s("%li", &b);

printf("c = "); scanf\_s("%li", &c);

printf("d = "); scanf\_s("%li", &d);

printf("e = "); scanf\_s("%li", &e);

printf("f = "); scanf\_s("%li", &f);

res\_c = -(d + 2) + a - e - b + c + (f - 1);

printf("Result C = %li\n", res\_c);

\_\_asm {

mov eax, d; // <eax> = d

add eax, 2; // <eax> = d + 2

neg eax; // <eax> = -(d + 2)

add eax, a; // <eax> = -(d + 2) + a

sub eax, e; // <eax> = -(d + 2) + a - e

sub eax, b; // <eax = -(d + 2) + a - e - b

add eax, c; // <eax> = -(d + 2) + a - e - b + c

mov ebx, f; // <ebx> = f

dec ebx; // <ebx> = f - 1

add eax, ebx; // <eax> = -(d + 2) + a - e - b + c + (f - 1)

mov res\_asm, eax; // res\_asm = ax

}

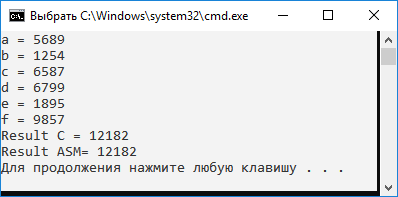
printf("Result ASM= %li\n", res\_asm);

system("Pause");

return 0;

}

Результат виконання програми:



**Значення регістрів при покроковому виконанні**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Крок** | **Команда** | **Значення регістра** | | | **EFLAGS/FLAGS (CF, OF)** |
| **eax** | **ebx** |  | | |
| **1** | mov eax, d | 6799 | н/в | **-** | | |
| **2** | add eax, 2 | 6801 | н/в | **-** | | |
| **3** | neg eax | -6801 | н/в | **-** | | |
| **4** | add eax, a | -1112 | н/в | **-** | | |
| **5** | sub eax, e | -3007 | н/в | **-** | | |
| **6** | sub eax, b | -4261 | н/в | **-** | | |
| **7** | add eax, c | 2326 | н/в | **-** | | |
| **8** | mov ebx, f | н/в | 9857 | **-** | | |
| **9** | dec ebx | н/в | 9856 | **-** | | |
| **10** | add eax, ebx | 12182 | н/в | **-** | | |
| **11** | mov res\_asm, eax | 12182 | н/в | **-** | | |

***Висновок:*** в ході виконання лабораторної роботи ознайомлено з типами цілочисельних даних платформ Win16 та Win32; ознайомлено з основними командами мови Assembler; набуто практичних навичок в написанні програм для обчислення простих цілочисельних виразів на мові Assembler.