НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Системне програмування »

*Звiт з лабораторної роботи №1*

*Тема: «*Виконання арифметичних операцій та процедури з параметрами*»*

Виконав:

ст. гр. KIT-119D

Гряник Г.В.

Перевірив:

Харків – 2020

**Мета :**

- поглибити і закріпити знання з архітектури МП платформи х64 і навички по їх програмування;

- набути практичних навичок складання, налагодження та виконання програм, написаних мовою ассемблера для програмування арифметичних операцій під МП платформи 64 в середовищі masm64.

- набути практичних навичок складання, налагодження та виконання програм, написаних мовою ассемблера для програмування арифметичних операцій, представлених у вигляді процедур з параметрами під МП платформи 64

***Індивідуальне завдання***

Варіант 6.

* Згідно номеру студента в групі вибрати варіант завдання і написати на асемблері програму обчислення одного з виразів : **2d/c-cd;**
* Згідно номеру студента в групі вибрати варіант завдання і написати на асемблері програму обчислення одного з виразів: **a/b/c-d/e+f**
* Заданий масив А з числом елементів більше 7. Написати програму визначення кількості пар елементів, які задовольняють умову Аi ≤ Ві.

**Текст програми**

Програма 1: (LR1-1)

include \masm64\include64\masm64rt.inc

.data ;

a1 dq 2 ; операнд а1 размерностью 64 разряда константа 2

c1 dq 3 ; операнд c1 размерностью 64 разряда;змзмінна c

d1 dq 6 ; операнд d1 размерностью 64 разряда ;змзмінна d

res1 dq 0 ; операнд res1 размерностью 64 разряда; змінна резудьт

;Текст для MessageBox

title1 db "Лаб.1-1.Рішення рівяння. masm64",0

txt1 db "Рівняння 2d/с – сd за допомогою MessageBox: ",10,

"Змінні: c=%d", ",d=%d",10,

"Результат: %d",10,"Адрес змінної в памяти: %ph",10,10,

"Автор: Гряник Г.В., гр.КІТ-119Д",0

buf1 dq 3 dup(0),0

.code ;cекция кода

entry\_point proc

mov rax, d1 ; пересылка d1 в rax

mul a1 ; множення rax х 2

div c1 ; ділення rax х c1

mov res1,rax ; сохранение промежуточного результата

mov rax, d1 ;пересылка d1 в rax

mul c1 ;множення rax на с1

sub res1,rax;2d/с – сd

;Створення MessageBox

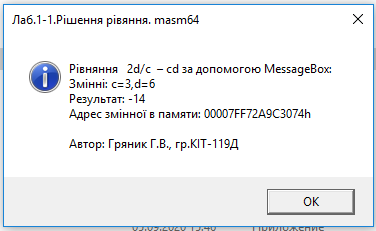
invoke wsprintf,ADDR buf1,ADDR txt1,c1,d1,res1,ADDR res1

invoke MessageBox,0,ADDR buf1,ADDR title1,MB\_ICONINFORMATION

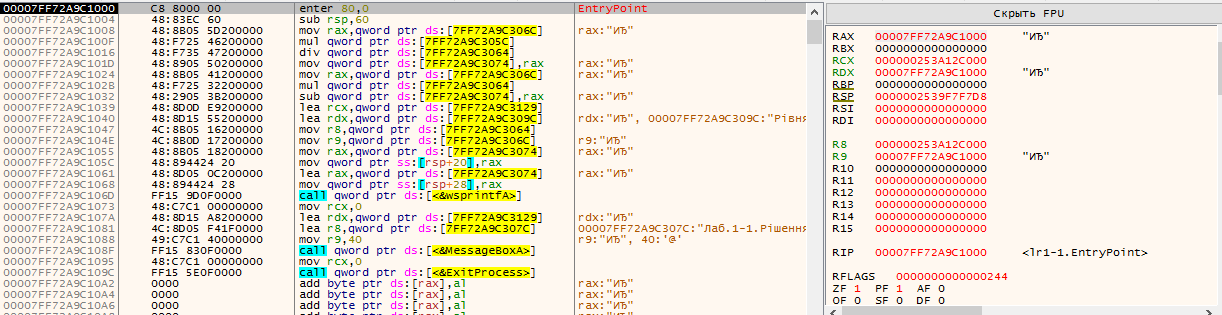
invoke ExitProcess,0

entry\_point endp; кінець роботи програми

end



**Рис 1 – результат виконання роботи програми** LR1-1



**Рис 2 – результат налагодження програми** LR1-1

Алгоритм: Прорама в з мінні заносить данні після чого в регістр rax заноситься данні d. За допомогою команд mul ta div проводяться операції множення та ділення відповідно. Ці занні зберігаються в змінні res1 після чого регістр rax заноситься данні d. Далі ця змінна множиться на с задопомогою команди mul та від збереженго нами результату раніше, віднімається оримане число.Таким чином ми оримали результат який вивидиться.

**Програма 2:**

include \masm64\include64\masm64rt.inc ; библиотеки

count PROTO arg\_a:QWORD,arg\_b:QWORD,arg\_c:QWORD,arg\_d:QWORD, arg\_e:QWORD,arg\_f:QWORD

.data ;

a1 dq 2 ; операнд а1 размерностью 64 разряда константа 2

b1 dq 5 ; операнд b1 размерностью 64 разряда ;змзмінна d

c1 dq 3 ; операнд c1 размерностью 64 разряда;змзмінна c

d1 dq 16 ; операнд d1 размерностью 64 разряда ;змзмінна d

e1 dq 8 ; операнд e1 размерностью 64 разряда ;змзмінна e

f1 dq 13 ; операнд f1 размерностью 64 разряда ;змзмінна f

tmp1 dq 0;

res1 dq 0 ; операнд res1 размерностью 64 разряда; змінна результ

;Текст для MessageBox

title1 db "Лаб.1-2.1 Рішення рівяння. masm64",0

txt1 db "Рівняння a+b + c\*d - e-f",10,

"Змінні: a=%d, b=%d, c=%d, d=%d, e=%d, f=%d",10,

"Результат: %d",10,"Адрес змінної в памяти: %ph",10,10,

"Автор: Гряник Г.В., гр.КІТ-119Д",0

buf1 dq 3 dup(0),0

.code;cекция кода

count proc arga:QWORD, argb:QWORD, argc:QWORD,argd:QWORD, arge:QWORD,argf:QWORD

mov rax,[rbp] ; исследование ; смотрим в x64dbg содержимое [rbp]

mov rax,[rbp+10h] ; a1;

mov rax,[rbp+18h] ; b1

mov rax,[rbp+20h] ; c1

mov rax,[rbp+28h] ; d1

mov rax,[rbp+30h] ; e1

mov rax,[rbp+38h] ; f1

mov rax,rcx ; arg a

add rax ,[rbp+18h] ; arg b ; rdx:rax

mov rsi,rax

mov rax, [rbp+20h] ; arg c

mov r11, [rbp+28h]

mul r11 ; arg d

add rax,rsi

add rax,[rbp+30h] ; arg e

sub rax,[rbp+38h] ; arg f

mov res1,rax

ret

count endp

entry\_point proc

invoke count,a1,b1,c1,d1,e1,f1

;Створення MessageBox

invoke wsprintf,ADDR buf1,ADDR txt1,a1,b1,c1,d1,e1,f1,res1,ADDR res1

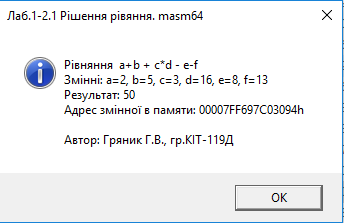
invoke MessageBox,0,ADDR buf1,ADDR title1,MB\_ICONINFORMATION

invoke ExitProcess,0

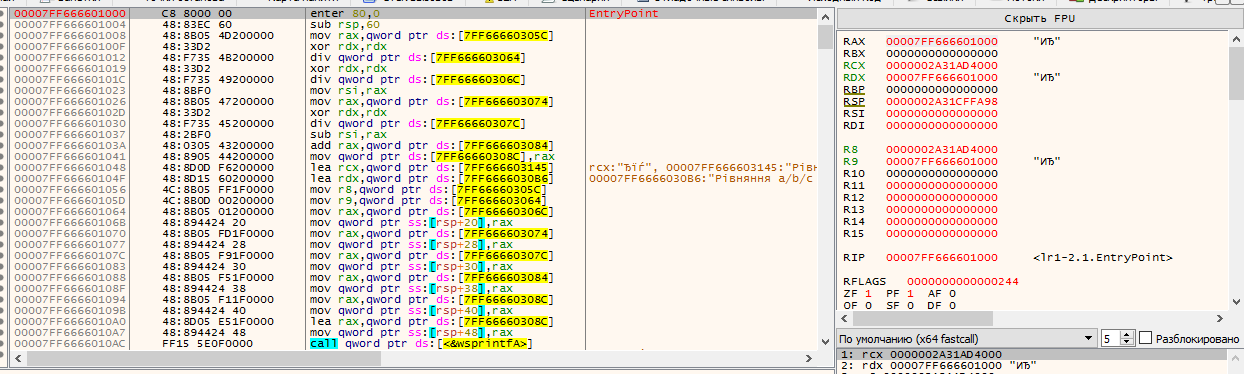
entry\_point endp; кінець роботи програми

end

**Результат**



**Рис 3 – результат виконання роботи програми** LR1-2.1



**Рис 4 – результат налагодження програми** LR1-2.1

Алгоритм: Прорама в з мінні заносить данні після чого в регістр rax заноситься данні a. Після чого проводиться операція ділення на b тa на c.Результат зерігаємо. Далі в регістр rax заноситься данні d. Ця змінна ділиться на с задопомогою команди div та від збереженго нами результату раніше, віднімається оримане число. І до отриманого результату дожаємо змінну f. Таким чином ми оримали результат який вивидиться.

**Програма 3:**

include \masm64\include64\masm64rt.inc ; библиотеки

.data ;

mas1 dq 88, 3, 21, 5, 19, 4, 5,4

; a, b, c, d, e, f ,g

mas2 dq 88, 4, 1, 5, 18, 7, 1,8

; a1,b1,c1, d1,e1, f1,g1

len1 equ $-mas1

res1 dq 0 ; операнд res1 размерностью 64 разряда; змінна результ

;Текст для MessageBox

title1 db "Лаб.1-2.2 Рішення рівяння. masm64",0

txt1 db "Завдання: Написать программу определения количества пар элементов, которые удовлетворяют условие Аi <= Ві.",10,

"Результат: %d",10,"Адрес змінної в памяти: %ph",10,10,

"Автор: Гряник Г.В., гр.КІТ-119Д",0

buf1 dq 3 dup(0),0

.code;cекция кода

entry\_point proc

mov rcx ,8;; загрузка счетчика количеством чисел массива

lea rsi,byte ptr mas1 ; занесение адреса начала элементов массива mas1

lea rdi,byte ptr mas2 ; занесение адреса массива результата массива mas2

jnz @1 ;визов цикла

minus: inc res1;+1 к результату

add rsi,8;звиг указателя

add rdi,8;звиг указателя

dec rcx;счётчик -1

cmp rcx,0; проверка не конец

je en

@1:

movzx rax, word ptr [rsi] ;занесениеелемента в rax

movzx rdx, word ptr [rdi] ;занесениеелемента в rdx

sub rax,rdx; доп. дейсвия для проверки

cmp rax,0; проверка rax <0;

JLE minus; перейти, если a <= b

add rsi,8;звиг указателя

add rdi,8;звиг указателя

dec rcx;счётчик -1

cmp rcx,0; проверка не конец

jnz @1

en: mov rax, 0;;конец

;Створення MessageBox

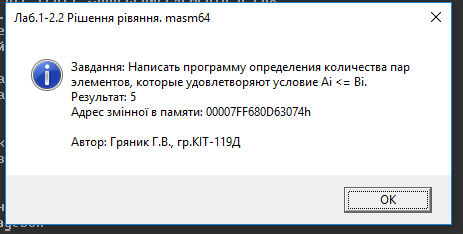
invoke wsprintf,ADDR buf1,ADDR txt1,res1,ADDR res1

invoke MessageBox,0,ADDR buf1,ADDR title1,MB\_ICONINFORMATION

invoke ExitProcess,0

entry\_point endp; кінець роботи програми

end



**Рис 5 – результат виконання роботи програми** LR1-2.2

****

**Рис 6 – результат налагодження програми** LR1-2.2

Алгоритм: Програма заносить дані в масив. Робить вказівник на перший елемент після чого в циклі порівнює елементи масивів. Якщо елементи відповідають умові то до результату додається одиниця. Виводиться результат роботи програми.

**Висновок:** Під час лабораторної роботи було набуто практичних навичок з роботи mas64, а саме, роботі з арифметичними операціями та роботи з масивами і циклами.