**ЛАБОРАТОРНА РОБОТА № 1**

з дисципліни

«Архітектура комп`ютерів»

Шейко Р. О.

Групи КІ-21-2

**Завдання на лабораторну роботу**

1. Для FASM створити програму, яка виконує додавання або віднімання двох чисел у знаковому форматі, у результаті встановити прапори переповнення, переносу та нульового результату у регістрі прапорів (по одному). Перевірити встановлення прапорів при збільшенні розрядності.

2. Створити програму за своїм варіантом арифметичного виразу (відповідно за номером студента в групі, табл. 4.5) в знакових форматах довжиною 32 та 64 біти. Проаналізувати хід виконання програми у налагоджувачі x64dbg.

y = (b/2-2/k)/(b-a\*k+1)

1. Завдання знаходиться вище.

2. Код до програми першого завдання знаходиться у А. Коди для другого завдання знаходяться в Б і В(32dbg i 64dbg відповідно).

3. В моїх програмі було віднято 5 від 3. Результати трасування зображено нижче на рис 1. Прикладаю також фото дебаггеру рис 2.

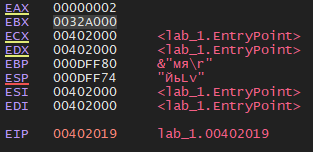


Рис. 1. - Результати трасування

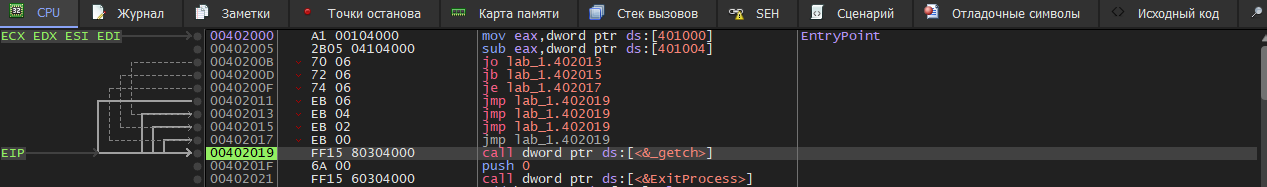


Рис. 2. - Результати дебаггер

4. Значення вихідних змінних та отримані результати для індивідуального завдання 2.

a dq 4 ; Перше число

b dq 2 ; Друге число

k dq 1 ; Третє число

result dq ?

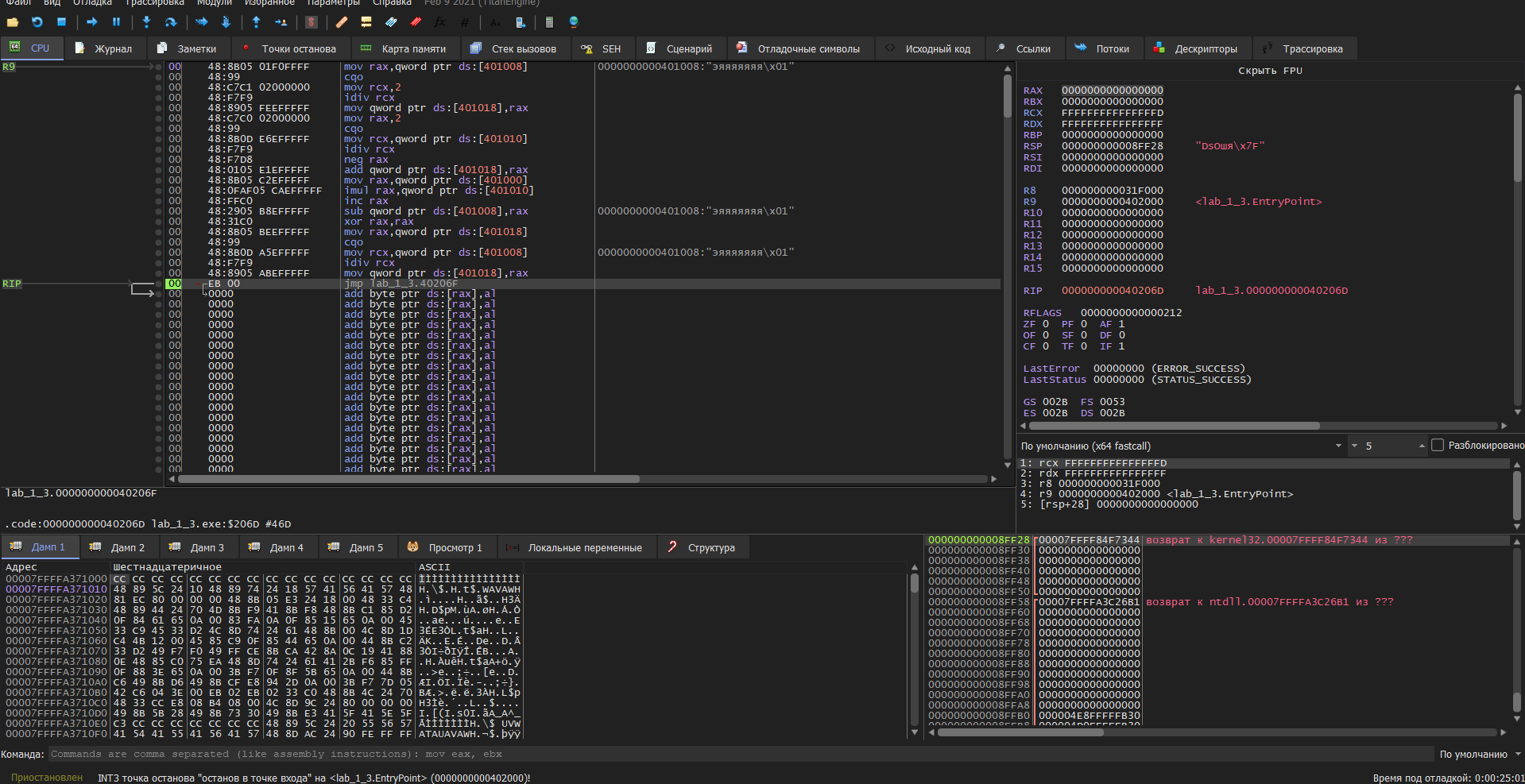


Рис 1. Значення співдають з обчисленням формул

Додаток А

*Лістинг завдання 1*

*format PE console*

*entry main*

*include 'win32a.inc'*

*section '.data' data readable writeable*

*num1 dd 5 ; Ïåðøå ÷èñëî*

*num2 dd 3 ; Äðóãå ÷èñëî*

*strOf db "Overflow flag (OF) is set.", 0*

*strCr db "Carry flag (CF) is set.", 0*

*strZr db "Zero flag (ZF) is set." , 0*

*strRs db "Result: ", 0*

*section '.code' code readable executable*

*main:*

*mov eax, [num1] ; Çàâàíòàæèòè ïåðøå ÷èñëî â eax*

*sub eax, [num2] ; Â³äíÿòè äðóãå ÷èñëî â³ä ïåðøîãî*

*; Âñòàíîâèòè ïðàïîðè ïåðåïîâíåííÿ, ïåðåíîñó òà íóëüîâîãî ðåçóëüòàòó*

*jo overflow ; Ïåðåâ³ðêà OF*

*jc carry ; Ïåðåâ³ðêà CF*

*jz zero ; Ïåðåâ³ðêà ZF*

*; ßêùî æîäåí ç ïðàïîð³â íå âñòàíîâëåíî, âèâåñòè ðåçóëüòàò*

*; push strRs*

*; push eax*

*; call [printf]*

*jmp exit*

*overflow:*

*; push strOf*

*; call [printf]*

*jmp exit*

*carry:*

*;push strCr*

*; call [printf]*

*jmp exit*

*zero:*

*;push strZr*

*;call [printf]*

*jmp exit*

*exit:*

*call [getch]*

*invoke ExitProcess, 0*

*section '.idata' import data readable*

*library kernel, 'kernel32.dll',\*

*msvcrt, 'msvcrt.dll'*

*import kernel,\*

*ExitProcess, 'ExitProcess'*

*import msvcrt,\*

*printf, 'printf',\*

*scanf, 'scanf',\*

*getch, '\_getch'*

Додаток Б

*Лістинг завдання 2*

*format PE console*

*entry main*

*include 'win32a.inc'*

*section '.data' data readable writeable*

*;resStr db 'Result: %d', 0*

*a dd 4 ; Перше число*

*b dd 2 ; Друге число*

*k dd 1 ; Третє число*

*result dd ?*

*section '.code' code readable executable*

*main:*

*;y = (b/2-2/k)/(b-a\*k+1) ;поменяли 53 на 2*

*;9/15*

*;b/2*

*mov eax, [b]*

*cdq*

*mov ecx, 2*

*idiv ecx*

*mov [result], eax ;result записано от b/2*

*;53/k*

*mov eax, 2*

*cdq*

*mov ecx, [k]*

*idiv ecx ;eax от 53/k*

*neg eax ;eax от -(53/k)*

*add [result], eax ; result = (b/2-53/k)*

*mov eax, [a]*

*imul eax, [k]*

*inc eax*

*sub [b], eax*

*xor eax, eax*

*mov eax, [result];9*

*cdq*

*mov ecx, [b] ;15*

*idiv ecx*

*mov [result], eax*

*; Якщо жоден з прапорів не встановлено, вивести результат*

*; push resStr*

*;push eax*

*;call [printf]*

*jmp exit*

*exit:*

*invoke ExitProcess, 0*

*section '.idata' import data readable*

*library kernel, 'kernel32.dll',\*

*msvcrt, 'msvcrt.dll'*

*import kernel,\*

*ExitProcess, 'ExitProcess'*

*import msvcrt,\*

*printf, 'printf',\*

*scanf, 'scanf',\*

*getch, '\_getch'*

Додаток В

*Лістинг завдання 2(64 dbg)*

*format PE64 console*

*entry main*

*;include 'win32a.inc'*

*section '.data' data readable writeable*

*;resStr db 'Result: %d', 0*

*a dq 4 ; Ïåðøå ÷èñëî*

*b dq 2 ; Äðóãå ÷èñëî*

*k dq 1 ; Òðåòº ÷èñëî*

*result dq ?*

*section '.code' code readable executable*

*main:*

*;y = (b/2-2/k)/(b-a\*k+1) ;ïîìåíÿëè 53 íà 2*

*;9/15*

*;b/2*

*mov rax, [b]*

*cqo*

*mov rcx, 2*

*idiv rcx*

*mov [result], rax ;result çàïèñàíî îò b/2*

*;53/k*

*mov rax, 2*

*cqo*

*mov rcx, [k]*

*idiv rcx ;eax îò 53/k*

*neg rax ;eax îò -(53/k)*

*add [result], rax ; result = (b/2-53/k)*

*mov rax, [a]*

*imul rax, [k]*

*inc rax*

*sub [b], rax*

*xor rax, rax*

*mov rax, [result];9*

*cqo*

*mov rcx, [b] ;15*

*idiv rcx*

*mov [result], rax*

*; ßêùî æîäåí ç ïðàïîð³â íå âñòàíîâëåíî, âèâåñòè ðåçóëüòàò*

*; push resStr*

*;push eax*

*;call [printf]*

*jmp exit*

*exit:*

*;invoke ExitProcess, 0*

*;section '.idata' import data readable*

*; library kernel, 'kernel32.dll',\*

*; msvcrt, 'msvcrt.dll'*

*;*

*; import kernel,\*

*; ExitProcess, 'ExitProcess'*

*;*

*; import msvcrt,\*

*;; printf, 'printf',\*

*; scanf, 'scanf',\*

*; getch, '\_getch'*