Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет среднего профессионального образования

**ОТЧЕТ**

**о Лабораторной работе № 3**

по теме: Команды сдвига. Выполнение арифметических выражений альтернативными операциями

по дисциплине: Системное программирование

Специальность:

09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
| Проверил:  Тропченко А.А. \_\_\_\_\_\_\_\_\_\_  Дата: «\_\_» \_\_\_\_\_\_\_ 20\_\_г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  Студент группы Y2431  Головачев Д.А. |

Санкт-Петербург 2020/2021

Лабораторная работа 2-1 «Команды сдвига. Выполнение арифметических выражений альтернативными операциями»

Цель занятия:

— приобрести практические навыки составления, отладки и выполнения программ, написанных языком ассемблера для вычисления математического уравнения при помощи арифметических команд и команд сдвига, сравнение времени их выполнения под МП платформу х64.

Вариант 6: a/c — e/f – adb

Текст программы:

Листинг программы 64-разрядной системы.

include \masm64\include64\masm64rt.inc ; присоединение библиотеки

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

.data ; секция данных

\_a dq 16 ; операнд \_a размерностью 64 разряда

\_b dq 16 ; операнд \_b размерностью 64 разряда

\_c dq 64 ; операнд \_c размерностью 64 разряда

\_d dq 32 ; операнд \_d размерностью 64 разряда

\_e dq 128 ; операнд \_e размерностью 64 разряда

\_f dq 256 ; операнд \_f размерностью 64 разряда

\_res dq ?,0 ; ячейка результата программы без сдвигов

\_res1 dq ?,0 ; ячейка тактов программы без сдивгов

\_res2 dq ?,0 ; ячейка результата программы со сдивгами

\_res3 dq ?,0 ; ячейка тактов программы со сдивгами

\_title db "Лабораторная работа №3. Команды сдвига. (masm64)",0

\_text db "Уравнение a/c — e/f – adb",0ah,"Результат выполнения арифм. команд: %d",0ah,"Число тактов: %d",0ah,0ah, ; текст для MessageBox

"Результат выполнения команд cдвига: %d",0ah,"Число тактов: %d",0ah,0ah, ; текст для MessageBox

"Автор: Головачев Д.А., ФСПО ИТМО",0 ; текст для MessageBox

strbuf dq ?,0 ; буфер

.code ; секция кода

count proc arg\_a:QWORD,arg\_b:QWORD,arg\_c:QWORD,arg\_d:QWORD,arg\_e:QWORD,arg\_f:QWORD ; запуск процедуры с параметрами

rdtsc ; получение числа тактов

xchg rdi,rax ; обмен значениями регистров

mov rax,arg\_a ; сохрание a в аккумуляторе

xor rdx,rdx ; обнуление rdx

div arg\_c ; деление аккумулятора на c

mov rsi,rax ; сохранение промежуточного значения

mov rax,arg\_e ; сохранение e в аккумуляторе

xor rdx,rdx ; обнуление rdx

div arg\_f ; деление аккумулятора на f

sub rsi,rax ; разность промежуточного значения

mov rax,arg\_a ; сохранение a в аккумуляторе

mul arg\_d ; умножение аккумулятора на d

mul arg\_b ; умножение аккумулятора на b

sub rsi,rax ; разность rsi и rax

mov \_res,rsi ; сохранение rsi в ячейке результата

rdtsc ; получение числа тактов

sub rax,rdi ; вычитание из последнего числа тактов предыдущего числа

mov \_res1,rax ; сохранение значения тактов в ячейке для тактов

ret ; завершение программы

count endp ; завершение процедуры с параметрами

count2 proc arg\_a:QWORD,arg\_b:QWORD,arg\_c:QWORD,arg\_d:QWORD,arg\_e:QWORD,arg\_f:QWORD ; запуск процедуры с параметрами

rdtsc

xchg rdi,rax ; обмен значениями регистров

mov rax,arg\_a ; сохрание a в аккумуляторе

sar rax,6 ; деление на 2 в 6 степени с помощью сдвига

mov rsi,rax ; сохранение промежуточного результата

mov rax,arg\_e ; сохрание e в аккумуляторе

sar rax,8 ; деление на 2 в 8 степени с помощью сдвига

sub rsi,rax ; разность промежуточного значения

mov rax,arg\_a ; сохрание a в аккумуляторе

sal rax,5 ; умножение на 2 в 5 степени с помощью сдвига

sal rax,4 ; умножение на 2 в 4 степени с помощью сдвига

sub rsi,rax ; разность rsi и rax

mov \_res2,rsi ; сохранение rsi в ячейке результата

rdtsc ; получение числа тактов

sub rax, rdi ; вычитание из последнего числа тактов предыдущего числа

mov \_res3,rax ; сохранение значения тактов в ячейке для тактов

ret ; завершение программы

count2 endp ; завершение процедуры с параметрами

entry\_point proc

invoke count,\_a,\_b,\_c,\_d,\_e,\_f

invoke count2,\_a,\_b,\_c,\_d,\_e,\_f

invoke wsprintf, ADDR strbuf, ADDR \_text, \_res, \_res1,\_res2,\_res3

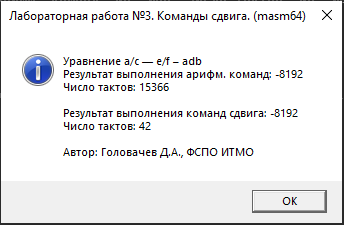
invoke MessageBox, NULL, addr strbuf, addr \_title, MB\_ICONINFORMATION

invoke ExitProcess, 0

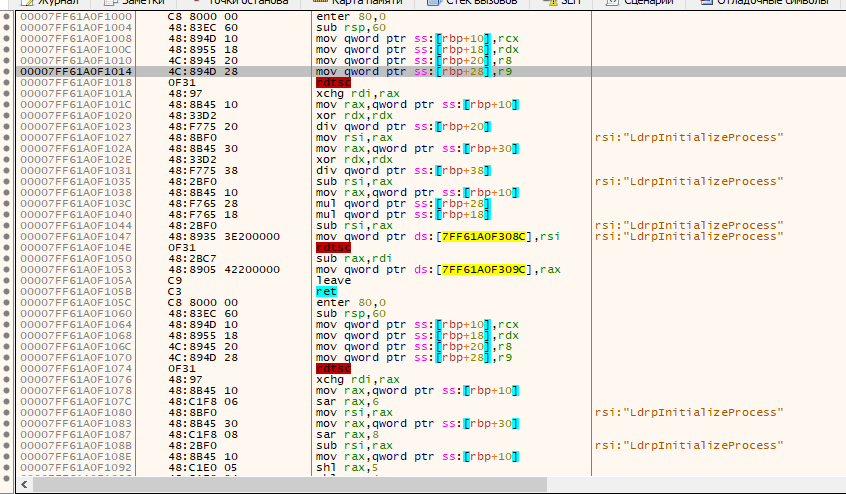
entry\_point endp

end

Результат выполнения программы и отладчиков:



*Рисунок 1. Результат выполнения программы 64-разрядной системы*





*Рисунок 2. Алгоритм выполнения 64-разрядной программы*

Особенности программы.

Программа проводит вычисления функции 2 раза, 1 раз – аналогично с 1 лабораторной работой, а, второй раз, операции умножения и деления заменяются операциями сдвига. Также, в каждой функции, замеряется количество использованных тактов.