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 № 2**

по теме: Процедуры с параметрами

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 1-1 «Выполнение арифметических операций»

Цель занятия:

— углубить и закрепить знания по архитектуре МП платформы 64 и навыки по ее программированию;

— приобрести практические навыки составления, налаживания и выполнения программ, написанных языком ассемблера для программирования арифметических операций, представленных в виде процедур с параметрами под МП платформы 64.

Вариант 5.1: a/c + bd — efg;

Вариант 5.2: Задан массив А с числом элементов больше 8. Написать программу определения суммы и количества элементов массива А, которые удовлетворяют условие Аi ≤ Е при Е = – 13.

Текст программы:

Листинг 1 задания программы 64-разрядной системы.

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,arg\_g:QWORD

.data ; начало блока данных

\_a1 dq 4

\_b1 dq 2

\_c1 dq 2

\_d1 dq 3

\_e1 dq 2

\_f1 dq 3

\_g1 dq 4

\_res1 dq 0

\_title db "Лаб.1-2. Процедуры. masm64", 0

strbuf dq ?,0

\_text db "Уравнение a/c + bd — efg",0ah,"Результат: %d",10,"Адрес переменной в памяти: %p",0ah,0ah,

"Автор: Головачев Д.А. ,каф. ВТП, НТУ ХПИ",0;

.code ; начало блока кода

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

mov rax,rcx ; arg\_a

sub rax, r8 ; a/с

mov rcx,r8 ; сохранил

mov rax,rdx ; b

mul r9 ; b\*d

add rcx, rax ; сново сохранил

mov rax,arg\_e ; e

mov r11,arg\_f ; f

mul r11 ; e\*f

mov r12,arg\_g ; g

xor rdx,rdx

mul r12 ; e\*f\*g

sub rcx,rax ; последнее действие

mov \_res1, rcx ; результат

ret

count endp

entry\_point proc

invoke count,\_a1,\_b1,\_c1,\_d1,\_e1,\_f1,\_g1

invoke wsprintf,ADDR strbuf,ADDR \_text,\_res1,ADDR \_res1

invoke MessageBox,0,addr strbuf, addr \_title, MB\_ICONINFORMATION

invoke ExitProcess,0

entry\_point endp

end

Листинг 2 программы задания 1, 32-разрядной системы

.686

.model flat, stdcall

option casemap:none

include \masm32\include\windows.inc

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

includelib \masm32\lib\user32.lib

includelib \masm32\lib\kernel32.lib

count PROTO arg\_a:DWORD,arg\_b:DWORD,arg\_c:DWORD,

arg\_d:DWORD, arg\_e:DWORD, arg\_f:DWORD

.data ; блок данных

\_a dd 2

\_b dd 2

\_c dd 2

\_d dd 2

\_e dd 5

\_f dd 2

\_res dd ?,0

strbuf dd ?,0

\_text db "Уравнение a/c + bd — efg",0ah,"Результат: %d",0ah,

"Адрес переменной в памяти: %p",0ah,0ah, "Автор: каф. ВТП, НТУ ХПИ",0

\_title db "Лабораторная работа No2 (masm32)",0

.code ; блок кода

count proc arg\_a:DWORD,arg\_b:DWORD,arg\_c:DWORD,arg\_d:DWORD,

arg\_e:DWORD, arg\_f:DWORD

push 3 ; пушим недостающее значение

mov eax,arg\_b ; заносим 1 значение в регистр

mul arg\_d ; умножаем на другое значение

mov esi,eax ; сохранение промежуточного результата

mov eax,arg\_a ; заносим новое значение в регистр

div arg\_c ; делим на другое значение

add esi,eax ; складываем с промежуточным результатом

mov eax,arg\_e ; заносим новое значение в регистр

mul arg\_f ; умножаем на другое начение

pop arg\_f ; выпушиваем из стека последню перменную

mul arg\_f ; умножаем на эту переменую акумулятор

sub eax,esi ; производим итоговое вычитание

mov esi,eax ; сохраняем результат

sub eax,esi ; инвертируем

sub eax,esi ; результат

mov \_res,eax ; сохранение итогового результата

ret

count endp

start:

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

invoke wsprintf, ADDR strbuf, ADDR \_text, \_res, ADDR \_res

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

MB\_ICONINFORMATION

invoke ExitProcess, 0

END start

Листинг 2 программы задания 1, 32-разрядной системы

title передача массива; masm64 изменением в размерности mas2.

include \masm64\include64\masm64rt.inc

.data

titl db "Вывод через функцию MessageBox",0; название упрощенного окна

buf1 dq 3 dup(0) ; буфер вывода сообщения

ifmt1 db "Результаты:", 10,"Сумма элементов <r14 = %d",10,"r14 = %d",10,

"Количество элементов <r14 = %d",10,0ah,"Автор: Головачев Данила ФСПО",0

mas1 db 1,4,-7,3,-10,7,-15,6

mas2 dq 8 dup(?),0 ; изменение размера элементов массива для удобства преобр. wsprintf

len1 equ ($-mas2)/type mas2

.code ; директива сегмента кода

entry\_point proc

mov rcx,len1 ; len1=8

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

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

mov r15, 0 ; тут будет количество элеметов

mov r14, -13 ; с этм мы сравниваем положительные числа

mov r12, 243 ; с этим мы сравниваем отрицательные числ

mov r13, 0 ; тут будет сумма

@1: mov al,[rsi] ;начало цикла

mov [rdi],al ;подменяем массив, чтобы легко было по нему идти

cmp [rdi], r14 ;проверка для положительных чисел

JG ab ;переход, если больше

inc r15

add r13, [rdi]

ab:

cmp [rdi], r12 ; сравнение отрицательных

JBE av ; переход, если меньше или равно

inc r15

sub r13, 256

add r13, [rdi]

av:

inc rsi ;переключение элемента массва

add rdi,8 ; переключениеэлемента другого массива

dec rcx ;счетчик цикла

jnz @1

invoke wsprintf,ADDR buf1,ADDR ifmt1,r13,r14,r15;

invoke MessageBox,0,addr buf1,addr titl,MB\_ICONINFORMATION;

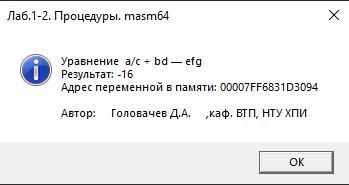
invoke ExitProcess, 0 ; возвращение управления ОС и освобождение ресурсов

entry\_point endp

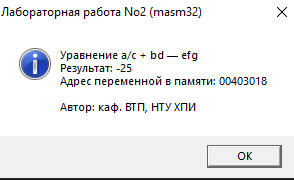
end

Листинг 3 программы задания 2, массив

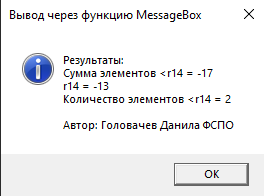
Результат выполнения программы и отладчиков:



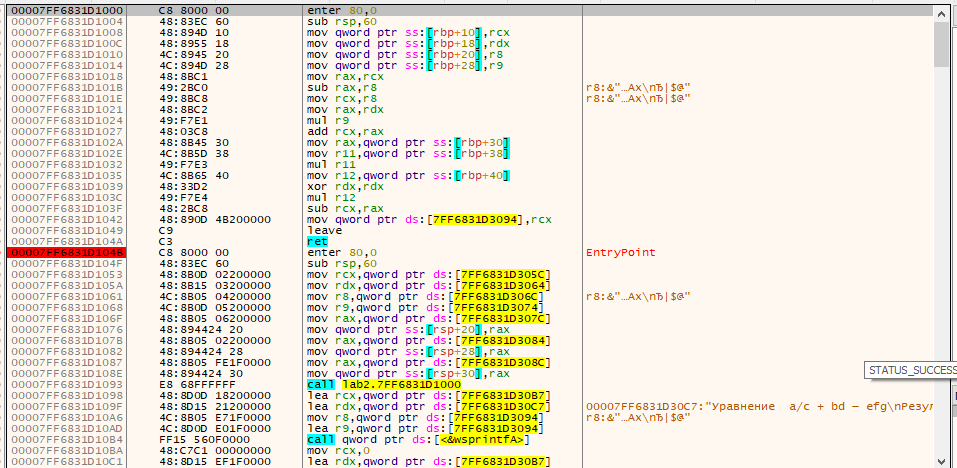
*Рисунок 1. Результат выполнения задания 1 программы 64-разрядной системы*

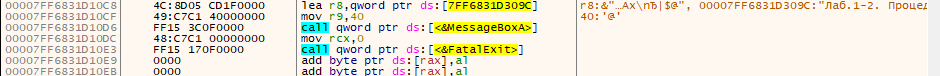


*Рисунок 2. Результат выполнения задания 1, программы 32-разрядной системы*

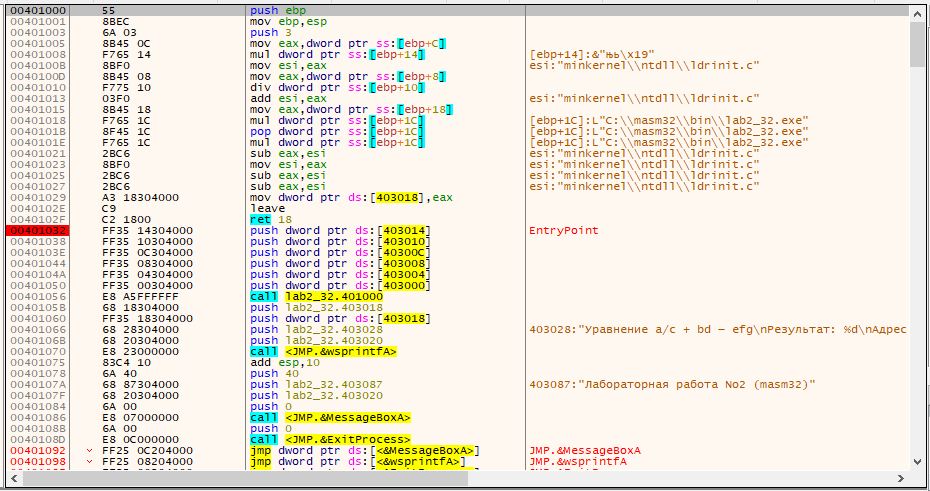


*Рисунок 3. Результат выполнения задания 2, массив*

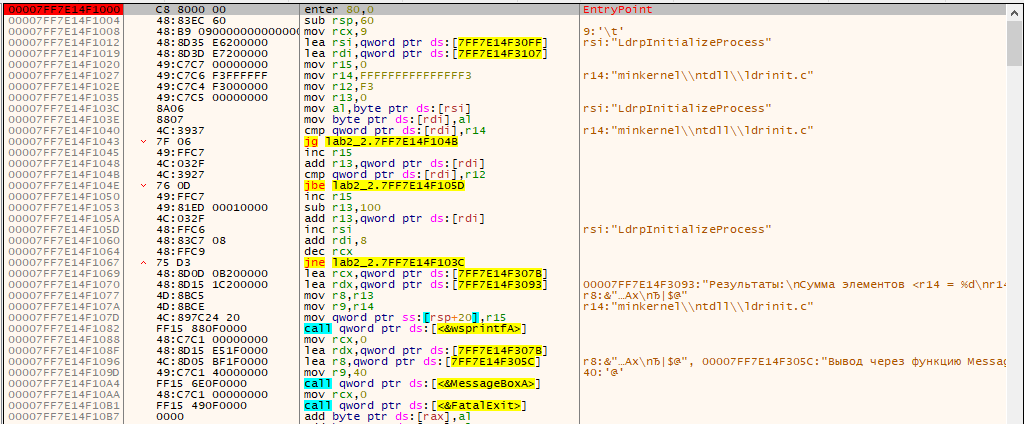




*Рисунок 4. Алгоритм выполнения задания 1, 64-разрядной программы*



*Рисунок 5. Алгоритм выполнения задания 1, 32-разрядной программы*



*Рисунок 6. Алгоритм выполнения задания 2*

Особенности программы задания 1 для 64-разрядной системы.

Вводятся 7 чисел. После выполнения деления, промежуточное значение сохраняется где-то в памяти, в регистре RCX, потом выполняется умножение и его результат прибавляется к тому значению в памяти. После этого выполняется умножение последних 3 элементов через регистр RAX, а результат прибавляется к значению предыдущих действий, сохраненному RCX, выводится результат.

Особенности программы задания 1 для 32-разрядной системы.

Тоже самое, что для 64, но, не хватило регистров на 7-ое значение, поэтому нужно было использовать стек для его ранения.

Особенности программы задания 2, массива

2 массива, 1 с числами, второй для определения длинны и красивого перехода, цикл по счётчику длинны. 2 ситуации, число больше нуля и число меньше нуля, рассматриваются друг за другом, при отрицательном числе, значение переворачивается.