

# ЗВІТ

Виконав:

Заночкин Є. Д.

КИТ-119а, Варіант 7

13 жовтня 2020 р.

## Лабораторна робота №2

Теми: Команди зсуву. Виконання арифметичних виразів альтернативними операціями. Тестування бітів

### Завдання

2 – 1. Знайти результат виразу  $ab + cd/e + f \text{ — } g$ .

2 – 2. Заданий масив А з  $N = 25$  елементів. Привести програму визначення кількості елементів масиву А, у яких біти 0 і 2 рівні 1.

### Тексти програм

1)

```
include \masm64\include64\masm64rt.inc ; бібліотеки
count PROTO arg_a:QWORD,arg_b:QWORD,arg_c:QWORD,arg_d:QWORD,arg_e:QWORD, arg_f:QWORD,
arg_g:QWORD
.data
_a dq 16 ; аргумент a
_b dq 4 ; аргумент b
_c dq 64 ; аргумент c
_d dq 32 ; аргумент d
_e dq 128 ; аргумент e
_f dq 256 ; аргумент f
_g dq 512 ; аргумент g
_res dq 0 ; переменная результата
_res1 dq 0 ; переменная результата
_res2 dq 0 ; переменная результата
_res3 dq 0 ; переменная результата
_title db "Лабораторная работа 2-1. Команды сдвига.",0
_text db "Уравнение  $ab + cd/e + f \text{ — } g$ ",0ah,"Результат выполнения арифм. команд: %d",0ah,"Число тактов: %d",0ah,0ah,
"Результат выполнения команд сдвига: %d",0ah,"Число тактов: %d",0ah,0ah,
"Автор: Заночкин Е. Д., КИТ-119а",0
buf1 dq 3 dup(0),0 ; буфер для вывода
.code
```

```

count proc arg_a:QWORD, arg_b:QWORD, arg_c:QWORD, arg_d:QWORD, arg_e:QWORD, arg_f:QWORD,
arg_g:QWORD
    mov r10,rdx ; заносим в r10 аргумент b
    rdtsc      ; rdx,rax — получение числа тактов
    xchg rdi,rax ; обмен значениями регистров rdi и rax
    mov rax,rcx ; заносим в rax аргумент a
    mul r10     ; умножаем a на b
    mov rsi,rax ; заносим в rsi значение a*b
    mov rax,r8  ; заносим в rax аргумент c
    mul r9      ; умножаем c на d
    xor rdx,rdx ; обнуляем rdx
    div arg_e   ; (c*d)/e
    add rax,rsi ; суммируем a*b и (c*d)/e
    add rax, arg_f ; добавляем аргумент f
    sub rax, arg_g ; отнимаем аргумент g
    mov _res,rax ; заносим результат в _res
    rdtsc      ; получение числа тактов
    sub rax,rdi ; вычитание из последнего числа тактов предыдущего числа
    mov _res1,rax ; заносим число тактов в _res1
    ret
count endp

count2 proc arg_a:QWORD, arg_b:QWORD, arg_c:QWORD, arg_d:QWORD, arg_e:QWORD, arg_f:QWORD,
arg_g:QWORD
    rdtsc      ; rdx,rax — получение числа тактов
    xchg rdi,rax ; обмен значениями регистров rdi и rax
    sal rcx,2   ; арифметический сдвиг влево (умножение) аргумента a на 2 (a*b)
    mov rsi,rcx ; заносим в rsi значение a*b
    sal r8,5    ; арифметический сдвиг влево (умножение) аргумента c на 5 (c*d)
    sar r8,7    ; арифметический сдвиг вправо (деление) значения c*d на 7 (c*d/e)
    add r8,rsi  ; суммируем a*b и (c*d)/e
    add r8,arg_f ; добавляем аргумент f
    sub r8,arg_g ; отнимаем аргумент g
    mov _res2,r8 ; заносим результат в _res2
    rdtsc      ; получение числа тактов
    sub rax, rdi ; вычитание из последнего числа тактов предыдущего числа
    mov _res3,rax ; заносим число тактов в _res3
    ret
count2 endp

entry_point proc
    invoke count,_a,_b,_c,_d,_e,_f,_g
    invoke count2,_a,_b,_c,_d,_e,_f,_g
    invoke wsprintf, ADDR buf1, ADDR _text, _res, _res1,_res2,_res3

```

```

invoke MessageBox,0, addr buf1, addr _title, MB_ICONINFORMATION
invoke ExitProcess,0
entry_point endp
end

```

2)

```

include \masm64\include64\masm64rt.inc ; библиотеки для подключения
.data
mas1 dw 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25 ; массив чисел
len1 equ ($-mas1)/type mas1 ; определение количества байтов mas1 (к-во
элементов)
title1 db "Лабораторная работа 2-2. Тестирование битов",0 ; заголовок окна вывода
txt1 db "Задан массив A из N = 25 элементов. Привести программу определения количества элементов
массива A, у которых биты 0 и 2 равны 1.",10,10,
"Результат: %d",10,10,
"Автор: Заночкин Е.Д., КИТ-119а",0
buf1 dq 3 dup(0),0
.code ; директива сегмента кода
entry_point proc
mov rcx,len1 ; заносим в rcx длину массива mas1
mov r15, 0 ; счетчик количества элементов mas1[0] = 1 && mas1[2] = 1
lea rsi, mas1 ; занесение адреса массива mas1 в rsi
@1: movzx rax, word ptr [rsi] ; занесение элемента mas1 в rax
bt rax, 0 ; проверка 0 бита
jc m1 ; перейти на m1, если CF = 1 (rax = 1)
jmp m3 ; безусловный переход на метку m3
m1: bt rax, 2 ; проверка 2 бита
jc m2 ; перейти на m2, если CF = 1 (rax = 1)
jmp m3 ; безусловный переход на метку m3
m2: inc r15 ; увеличение счетчика элементов mas1[0] = 0
jmp m3 ; безусловный переход на метку m3
m3: inc rsi ; подготовка адреса для считывания нового элемента
inc rsi ; переход на следующий элемент массива
dec rcx ; проверка счетчика количества байтов в массиве mas1
jnz @1 ; если в rcx не 0, то переход в начало цикла
invoke wsprintf,ADDR buf1,ADDR txt1, r15
invoke MessageBox,0,ADDR buf1,ADDR title1,MB_ICONINFORMATION
invoke ExitProcess, 0
entry_point endp
end

```

## Результати виконання програм

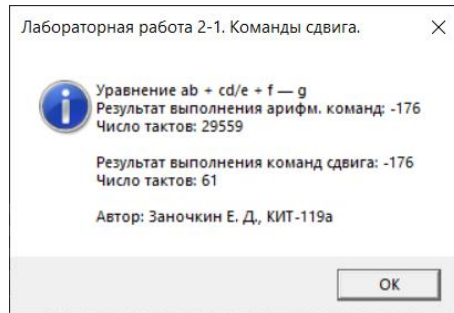


Рисунок 2.1а – Результат работы 2-1 в MessageBox

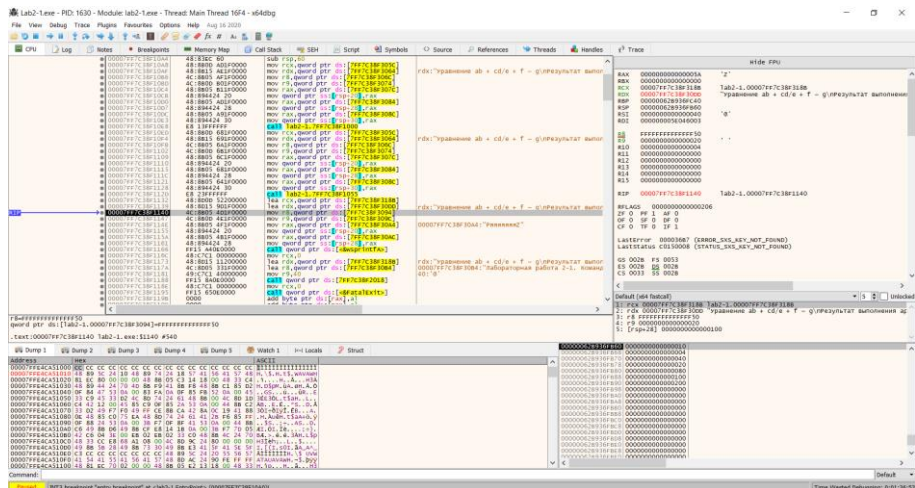


Рисунок 2.1б – Результат работы 2-1 в x64dbg

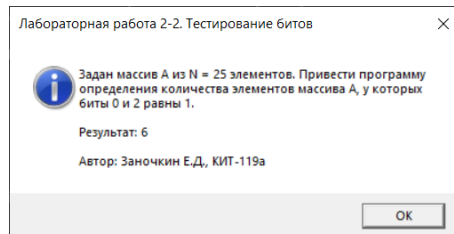


Рисунок 2.2а – Результат работы 2-2 в MessageBox

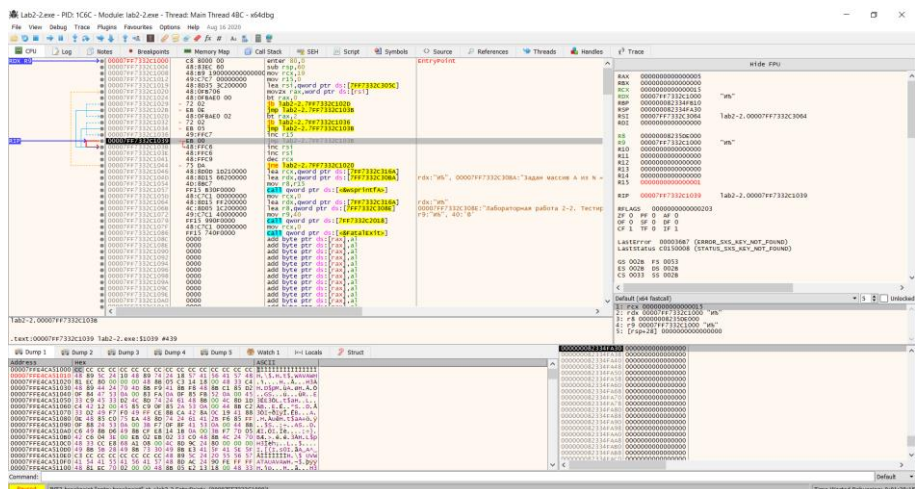


Рисунок 2.2б – Результат работы 2-2 в x64dbg

## Алгоритми виконання



Рисунок 2.3 – Алгоритм виконання програми 2-1

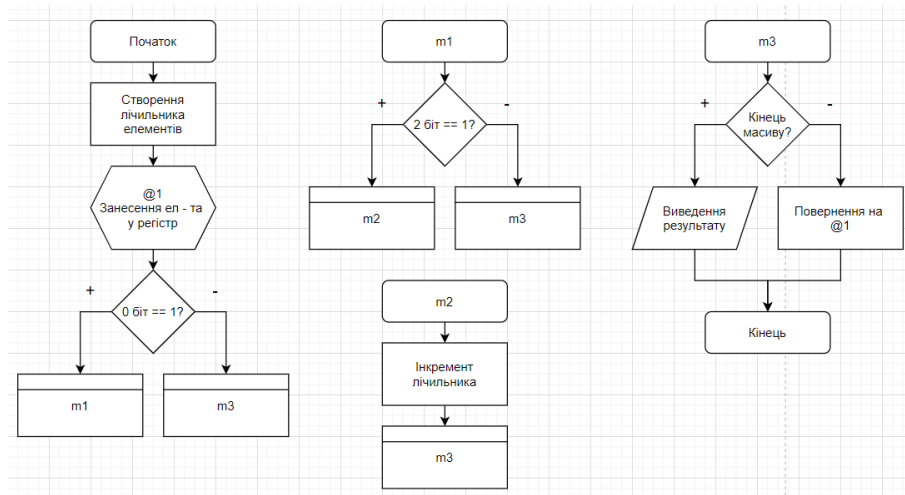


Рисунок 2.4 – Алгоритм виконання програми 2-2

## Висновок

Під час лабораторної роботи було створено 2 програми, які виконуються згідно з індивідуальним завданням, було набуто навички роботи з командами зсуву та тестування бітів. Програми протестовані, працюють без помилок.