**Звіт**

Автор: Салтиков Іван;

Група: КН-921Д;

# Лабораторна робота №2. Команди зсуву. Виконання арифметичних виразів альтернативними операціями. Тестування бітів

***Мета.*** набути практичних навичок складання, налагодження та виконання програм, написаних мовою ассемблера для обчислення математичного рівняння за допомогою арифметичних команд і команд зсуву, порівняння часу їх виконання під МП платформу х64.

набути практичних навичок складання, налагодження та виконання програм, написаних мовою ассемблера для аналізу бітів в масиві чисел при застосуванні команд зсуву, визначення часу виконання програми під МП платформи х64.

Хід роботи

include \masm64\include64\masm64rt.inc

.data

a1 dq 8 ; инициализация переменных

b1 dq 256

c1 dq 8

d1 dq 4

e1 dq 8

f1 dq 16

g1 dq 2

res1 dq ? ; переменная для результата 1

res2 dq ? ; переменная для результата 2

tick1 dq ? ; переменная количества тактов 1

tick2 dq ? ; переменная количества тактов 2

title1 db " Лабораторная работа №2. Команды сдвига",0

txt1 db "Уравнение a+b/c/d+efg",10,10,

"Результат:",10,

"Арифметические операции: %d",10,

"Арифметические сдвиги: %d",10,10,

"Количество тактов:",10,

"Для арифметических операций: %d",10,

"Для арифметических сдвигов: %d",10,10,

"Автор: Иван Салтыков",0

buf1 dq 3 dup(0),0

.code

entry\_point proc

mov r10, rdx ; запись R10 в RAX

rdtsc ; получение числа тактов

xchg rdi, rax ; обмен значениями регистров

xor rdx,rdx ; очистка RDX

mov rax,b1 ; запись b1 в RAX

div c1 ; b/c

div d1 ; b/c/d

mov rsi,rax ; запись RAX в регистр промежуточного результата

mov rax,e1 ; запись e1 в RAX

mul f1 ; e\*f

mul g1 ; e\*f\*g

add rax,rsi ; b/c/d+efg

add rax,a1 ; a+b/c/d+efg

mov res1,rax ; запись RAX в переменную результата 1

rdtsc ; получение числа тактов

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

mov tick1, rax ; пересылка RAX в tick1

mov r10, rdx ; запись RDX в R10

rdtsc ; получение числа тактов

xchg rdi, rax ; обмен значениями регистров

xor rdx,rdx ; инициализация регистра rdx

mov rax,b1 ; запись b1 в регистр RAX

sar rax,5 ; b/c/d

mov rax,rsi ; запись RAX в регистр промежуточного результата

mov rax,e1 ; запись e1 в RAX

sal rax,5 ; efg

add rax,rsi ; b/c/d+efg

add rax,a1 ; a+b/c/d+efg

mov res2,rax ; запись RAX в переменную результата 2

rdtsc ; получение числа тактов

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

mov tick2, rax ; пересылка rax в tick2

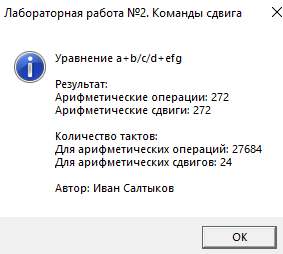
invoke wsprintf,ADDR buf1,ADDR txt1,res1,res2,tick1,tick2 ; преобразование данных в строку

invoke MessageBox,0,ADDR buf1,ADDR title1,MB\_ICONINFORMATION ; вызов окна с результатом

invoke ExitProcess,0

entry\_point endp

end



include \masm64\include64\masm64rt.inc

.data ; секция переменных

arrA dq 90, 32, 72, 22, 31, 11 ; массив А

len1 equ ($-arrA)/type arrA ; длинна массива А

arrB dq len1 dup(?), 0 ; массив В

a1 dq 1 ; переменные результата

a2 dq 1

a3 dq 1

title1 db "Лабораторная работа 2. Тестирование битов",0

txt1 db "Задание: Задан массив А из 40 элементов. Необходимо создать массив В из элементов массива А, где 0,2 и 5 бит имеют нули.",10,10,

"Результат: ",10,

"B[1]: %d",10,

"B[2]: %d",10,

"B[3]: %d",10,

10,"Автор: Салтыков Иван",0

buf1 dq 3 dup(0),0

.code ; директива сегмента кода

entry\_point proc

xor rcx,rcx ; очистка регистра RCX

xor rbp,rbp ; очистка регистра RBP

xor rdi,rdi ; очистка регистра RDI

xor rax,rax ; очистка регистра RAX

mov rcx, len1 ; запись длинны массива в RCX

lea rbp, arrA ; установка указателя на начало массива А

lea rdi, arrB ; установка указателя на начало массива В

mov rax, 0

m1:

xor rax,rax ; очистка регистра RAX

xor rbx,rbx ; очистка регистра RBX

xor rdx,rdx ; очистка регистра RDX

mov r10, [rbp] ; запись элемента массива А в регистр R10

bt r10, 0 ; проверка 0 бита

setc al ; запись результата в AL

cmp al, dl ; сравнение бита с 0

jne BitsNotZero ; если 0 бит имеет не нулевое значение

bt r10, 2 ; проверка 2 бита

setc al ; запись результата в AL

cmp al, dl ; сравнение бита с 0

jne BitsNotZero ; если 2 бит имеет не нулевое значение

bt r10, 5 ; проверка 5 бита

setc al ; запись результата в AL

cmp al, dl ; сравнение бита с 0

je BitsZero ; если все требуемые биты = 0

jne BitsNotZero ; если 5 бит имеет не нулевое значение

BitsZero:

mov [rdi], r10 ; запись в массив В элемент из А

add rdi, type arrB ; перемещение на следующий элемент массива В

BitsNotZero:

add rbp, type arrA ; перемещение на следующий элемент массива А

dec ecx ; уменьшение счётчика кол-ва циклов

jnz m1

jmp \_end ; переход в конец

\_end:

xor rax,rax ; очистка регистра RAX

xor rbx,rbx ; очистка регистра RBP

lea rbx,byte ptr arrB ; установка указателя в начало массива С

mov rax,[rbx] ; запись из массива С в регистр RAX

mov a1,rax ; запись из RAX в переменную res1

xor rax,rax ; очистка регистра RAX

add rbx,type arrB ; переместиться на следующий элемент массива

mov rax,[rbx] ; запись из массива С в регистр RAX

mov a2,rax ; запись из RAX в переменную res2

xor rax,rax ; очистка регистра RAX

add rbx,type arrB ; переместиться на следующий элемент массива

mov rax,[rbx] ; запись из массива С в регистр RAX

mov a3,rax ; запись из RAX в переменную res2

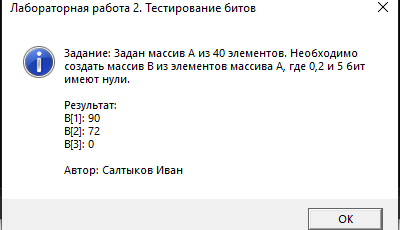
invoke wsprintf,ADDR buf1,ADDR txt1,a1,a2,a3 ; преобразование данных в строку

invoke MessageBox,0,ADDR buf1,ADDR title1,MB\_ICONINFORMATION ; вывод окна с результатом

invoke ExitProcess,0

entry\_point endp

end



Висновки

При виконанні даної лабораторної роботи було набуто навичок по виконанню команд зсуву, виконанню арифметичних виразів альтернативними операціями та тестуванням бітів

Програма протестована i виконується без помилок.