**Звіт**

Автор: Салтиков Іван;

Група: КН-921д;

# Лабораторна робота №4. Структури. Макроси

***Мета.*** набути практичних навичок складання, налагодження та виконання програм з використанням структур, написаних мовою ассемблера для платформи х64 в середовищі masm64.

набути практичних навичок складання, налагодження та виконання програм з використанням макросів, написаних мовою ассемблера для співпроцесора платформи х64 в середовищі masm64.

Хід роботи

include \masm64\include64\masm64rt.inc ; библиотеки

IDI\_ICON EQU 1001

MSGBOXPARAMSA STRUCT ; объявление системной структуры

cbSize DWORD ?,?

hwndOwner QWORD ?

hInstance QWORD ?

lpszText QWORD ?

lpszCaption QWORD ?

dwStyle DWORD ?,?

lpszIcon QWORD ?

dwContextHelpId QWORD ?

lpfnMsgBoxCallback QWORD ?

dwLanguageId DWORD ?,?

MSGBOXPARAMSA ENDS

matrix STRUCT ; начало объявления пользовательской структуры

el1 dq ?

el2 dq ?

el3 dq ?

el4 dq ?

matrix ENDS ; конец объявления пользовательской структуры

.data

params MSGBOXPARAMSA <> ; инициализация системной структуры

matrixA1 matrix <1,2,3,4> ; инициализация 1 строки массива А

matrixA2 matrix <5,4,3,2> ; инициализация 2 строки массива А

matrixA3 matrix <1,3,5,4> ; инициализация 3 строки массива А

matrixA4 matrix <3,5,2,4> ; инициализация 4 строки массива А

matrixB1 matrix <2,1,4,5> ; инициализация 1 строки массива В

matrixB2 matrix <3,5,2,4> ; инициализация 2 строки массива В

matrixB3 matrix <1,3,5,4> ; инициализация 3 строки массива В

matrixB4 matrix <5,4,3,2> ; инициализация 4 строки массива В

matrixC1 matrix <?,?,?,?> ; инициализация 1 строки массива С

matrixC2 matrix <?,?,?,?> ; инициализация 2 строки массива С

matrixC3 matrix <?,?,?,?> ; инициализация 3 строки массива С

matrixC4 matrix <?,?,?,?> ; инициализация 4 строки массива С

title1 db "Лабораторная работа 4-1-1. Структуры",0

txt2 db "Задание: Перемножить две матрицы размерами 4х4 и привести результирующую.",10,10,

"Матрица А ",10,

"1 2 3 4",10,

"5 4 3 2",10,

"1 3 5 4",10,

"3 5 2 4",10,10,

"Матрица В ",10,

"2 3 1 5",10,

"1 5 3 4",10,

"4 2 5 3",10,

"5 4 4 2",10,0

buf2 dq 3 dup(0),0

txt1 db "Результат:",10,10,

"Матрица С",10,

"%d %d %d %d",10,

"%d %d %d %d",10,

"%d %d %d %d",10,

"%d %d %d %d",10,10,

"Автор: Салтыков Иван",10,

"Группа: КИТ-921д",0

buf1 dq 3 dup(0),0

.code

entry\_point proc

xor rax,rax ; очистка регистра RAX

xor rdx,rdx ; очистка регистра RDX

xor rbx,rbx ; очистка регистра RBX

xor rcx,rcx ; очистка регистра RCX

mov rcx,4 ; установка количества циклов

lea rsi,matrixA1; установка указателя в начало массива А

lea r10,matrixB1; установка указателя в начало массива В

lea rdi,matrixC1; установка указателя в начало массива С

str11: ; цикл умножения 1 строки на 1 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str11

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка количества циклов

lea rsi,matrixA1; установка указателя в начало массива А

lea r10,matrixB2; установка указателя в начало массива В

str12: ; цикл умножения 1 строки на 2 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str12

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA1; установка указателя в начало массива А

lea r10,matrixB3; установка указателя в начало массива В

str13: ; цикл умножения 1 строки на 3 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str13

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA1; установка указателя в начало массива А

lea r10,matrixB4; установка указателя в начало массива В

str14: ; цикл умножения 1 строки на 4 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str14

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA2; установка указателя в начало массива А

lea r10,matrixB1; установка указателя в начало массива В

lea rdi,matrixC2; установка указателя в начало массива С

str21: ; цикл умножения 2 строки на 1 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str21

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA2; установка указателя в начало массива А

lea r10,matrixB2; установка указателя в начало массива В

str22: ; цикл умножения 2 строки на 2 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str22

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA2; установка указателя в начало массива А

lea r10,matrixB3; установка указателя в начало массива В

str23: ; цикл умножения 2 строки на 3 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str23

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA2; установка указателя в начало массива А

lea r10,matrixB4; установка указателя в начало массива В

str24: ; цикл умножения 2 строки на 4 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str24

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA3; установка указателя в начало массива А

lea r10,matrixB1; установка указателя в начало массива В

lea rdi,matrixC3; установка указателя в начало массива С

str31: ; цикл умножения 3 строки на 1 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str31

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA3; установка указателя в начало массива А

lea r10,matrixB2; установка указателя в начало массива В

str32: ; цикл умножения 3 строки на 2 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str32

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA3; установка указателя в начало массива А

lea r10,matrixB3; установка указателя в начало массива В

str33: ; цикл умножения 3 строки на 3 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str33

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA3; установка указателя в начало массива А

lea r10,matrixB4; установка указателя в начало массива В

str34: ; цикл умножения 3 строки на 4 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str34

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA4; установка указателя в начало массива А

lea r10,matrixB1; установка указателя в начало массива В

lea rdi,matrixC4; установка указателя в начало массива С

str41: ; цикл умножения 4 строки на 1 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str41

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA4; установка указателя в начало массива А

lea r10,matrixB2; установка указателя в начало массива В

str42: ; цикл умножения 4 строки на 2 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str42

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA4; установка указателя в начало массива А

lea r10,matrixB3; установка указателя в начало массива В

str43: ; цикл умножения 4 строки на 3 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str43

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

mov rcx,4 ; установка кол-ва циклов в 4

lea rsi,matrixA4; установка указателя в начало массива А

lea r10,matrixB4; установка указателя в начало массива В

str44: ; цикл умножения 4 строки на 4 ряд

mov rax,[rsi] ; получение элемента массива А

mov rdx,[r10] ; получение элемента массива В

mul rdx ; перемножение элементов

add rbx,rax ; сложение элементов

add rsi,8 ; переход на следующий элемент массива А

add r10,8 ; переход на следующий элемент массива В

loop str44

mov [rdi],rbx ; запись результата в массив С

add rdi,8 ; перемещение на следующий элемент массива С

xor rbx,rbx ; очистка регистра RBX

invoke wsprintf,ADDR buf1,ADDR txt2 ; преобразование данных в строку

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

invoke wsprintf,ADDR buf2,ADDR txt1,matrixC1.el1,matrixC1.el2,matrixC1.el3,matrixC1.el4,matrixC2.el1,matrixC2.el2,matrixC2.el3,matrixC2.el4,matrixC3.el1,matrixC3.el2,matrixC3.el3,matrixC3.el4,matrixC4.el1,matrixC4.el2,matrixC4.el3,matrixC4.el4

mov params.cbSize,SIZEOF MSGBOXPARAMSA ; размер структуры

mov params.hwndOwner,0 ; дескриптор окна владельца

invoke GetModuleHandle,0 ; получение дескриптора программы

mov params.hInstance,rax ; сохранение дескриптора программы

lea rax, buf2 ; адрес сообщения

mov params.lpszText,rax

lea rax,title1 ; адрес заглавия окна

mov params.lpszCaption,rax

mov params.dwStyle,MB\_USERICON ; стиль окна

mov params.lpszIcon,IDI\_ICON ; ресурс значка

mov params.dwContextHelpId,0 ; контекст справки

mov params.lpfnMsgBoxCallback,0

mov params.dwLanguageId,LANG\_NEUTRAL ; язык сообщения

lea rcx,params

invoke MessageBoxIndirect ; вызов окна с результатом работы и иконкой

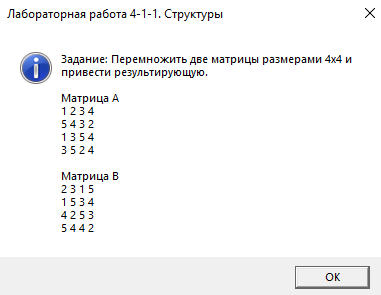
invoke ExitProcess,0

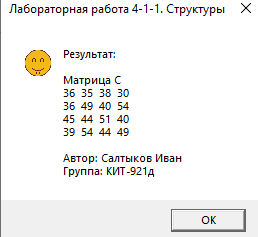
entry\_point endp

end

#define IDI\_ICON 1001

IDI\_ICON ICON DISCARDABLE "1.ico"





include \masm64\include64\masm64rt.inc

Computer STRUCT

serialNum1 dq ?

price1 dq ?

name1 dd ?

ownerSurname1 dd ?

size1 dq ?

Computer ENDS

IDI\_ICON EQU 1001

MSGBOXPARAMSA STRUCT ; объявление системной структуры

cbSize DWORD ?,?

hwndOwner QWORD ?

hInstance QWORD ?

lpszText QWORD ?

lpszCaption QWORD ?

dwStyle DWORD ?,?

lpszIcon QWORD ?

dwContextHelpId QWORD ?

lpfnMsgBoxCallback QWORD ?

dwLanguageId DWORD ?,?

MSGBOXPARAMSA ENDS

.data ; секция переменных

params MSGBOXPARAMSA <> ; инициализация системной структуры

PC1 Computer <1,2,"Best PC","BLABLA",5>

PC2 Computer <5,4,"123","IBM",4>

PC3 Computer <3,3,"142","Apple",1>

num1 dq 3 ; переменная константа

res1 dq ? ; переменная результата

title1 db "Лабораторная работа 4\_1\_2. Структуры",0 ; заголовок окна вывода

txt1 db "Задана последовательность структур. Структура содержит поля данные о компьютере: серийный номер, цена, название, фамилия владельца, размер монитора в дюймах. Вычислить среднюю цену компьютера.",10,10, ; вывод выражения

"Result: %d",10, ; вывод результата и адреса переменной

"Author: Ivan Saltikov",0

buf1 dq 3 dup(0),0

.code ; секция кода

entry\_point proc ; точка входа

xor rax,rax ; очистка регистра RAX

mov rax,PC1.price1 ; запись суммы первого ПК

add rax,PC2.price1 ; прибавление цены второго ПК

add rax,PC3.price1 ; прибавление цены третьего ПК

xor rdx,rdx ; очистка регистра RDX

div num1 ; /3

mov res1,rax ; запись результата в переменную результата

;invoke wsprintf,ADDR buf1,ADDR text1,res1,ADDR res1 ; преобразование данных в строку

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

;invoke ExitProcess,0

invoke wsprintf,ADDR buf1,ADDR txt1,res1

mov params.cbSize,SIZEOF MSGBOXPARAMSA ; размер структуры

mov params.hwndOwner,0 ; дескриптор окна владельца

invoke GetModuleHandle,0 ; получение дескриптора программы

mov params.hInstance,rax ; сохранение дескриптора программы

lea rax, buf1 ; адрес сообщения

mov params.lpszText,rax

lea rax,title1 ; адрес заглавия окна

mov params.lpszCaption,rax

mov params.dwStyle,MB\_USERICON ; стиль окна

mov params.lpszIcon,IDI\_ICON ; ресурс значка

mov params.dwContextHelpId,0 ; контекст справки

mov params.lpfnMsgBoxCallback,0

mov params.dwLanguageId,LANG\_NEUTRAL ; язык сообщения

lea rcx,params

invoke MessageBoxIndirect ; вызов окна с результатом работы и иконкой

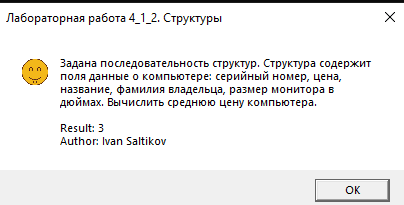
invoke ExitProcess,0

entry\_point endp ; точка выхода

end

#define IDI\_ICON 1001

IDI\_ICON ICON DISCARDABLE "1.ico"



include \masm64\include64\masm64rt.inc

IDI\_ICON EQU 1001

MSGBOXPARAMSA STRUCT ; объявление системной структуры

cbSize DWORD ?,?

hwndOwner QWORD ?

hInstance QWORD ?

lpszText QWORD ?

lpszCaption QWORD ?

dwStyle DWORD ?,?

lpszIcon QWORD ?

dwContextHelpId QWORD ?

lpfnMsgBoxCallback QWORD ?

dwLanguageId DWORD ?,?

MSGBOXPARAMSA ENDS

mSub macro a, b ;; макрос с именем mSub

fld a ;; загрузка a

fld b ;; загрузка b

fsub ;; a-b

endm ;; окончание макроса

.data

params MSGBOXPARAMSA <> ; инициализация системной структуры

hInstance dq ? ; дескриптор програми

hWnd dq ? ; дескриптор окна

hIcon dq ? ; дескриптор иконки

hCursor dq ? ; дескриптор курсора

sWid dq ? ; ширина монитора (колич. пикселей по x)

sHgt dq ? ; высота монитора (колич. пикселей по y)

classname db "template\_class",0

caption db "Результат выполнения арифметического выражения",0

title1 db "Лабораторная работа 4-2. Макросы",0

txt1 db "Написать на ассемблере программу вычисления выражения, в котором одна из переменных изменяется несколько раз.",10,10

txt2 db "Выражение: 3,5(a – b) – (a – b)/5,1",10,10

txt3 db "Автор: Ivan Salikov, KN-921d",0

buf dq 30 dup(?),0 ; буффер для вывода

buf1 dq 3 dup(?),0 ; буффер для вывода

buf2 db 80 dup(?),0 ; буффер для вывода

buf3 dq buf2,0 ; буффер для вывода

buf4 db 16 dup(?),0 ; буффер для вывода

const1 real4 3.5 ; переменная константа

const2 real4 5.1 ; переменная константа

a1 real4 3.7

b1 real4 5.2

res1 real8 0.0 ; переменная результата

res2 dq 0 ; переменная результата

.code

entry\_point proc

finit ; инициализация стека

mSub [a1],[b1] ; a-b

fmul const1 ; 3.5(a-b)

mSub [a1],[b1] ; a-b

fdiv const2 ; (a-b)/5.1

fsub ; 3.5(a-b)-(a-b)/5.1

fst res1 ; получение результата

fistp res2 ; получение результата

mov hInstance,rv(GetModuleHandle,0) ; получение и сохранение дескрипторa програми

mov hIcon, rv(LoadIcon,hInstance,10) ; загрузка и сохранение дескрипторa иконки

mov hCursor,rv(LoadCursor,0,IDC\_ARROW) ; загрузка курсора и сохранение

mov sWid,rv(GetSystemMetrics,SM\_CXSCREEN) ; получение кол. пикселей по х монитора

mov sHgt,rv(GetSystemMetrics,SM\_CYSCREEN) ; получение кол. пикселей по y монитора

call main ; вызов процедуры main

invoke ExitProcess,0

ret

entry\_point endp

main proc

LOCAL wc :WNDCLASSEX ; объявление локальных переменных

LOCAL lft :QWORD ; лок. переменные содержатся в стеке

LOCAL top :QWORD ; и существуют только во время вып. проц.

LOCAL wid :QWORD

LOCAL hgt :QWORD

mov wc.cbSize,SIZEOF WNDCLASSEX ; колич. байтов структуры

mov wc.style,CS\_BYTEALIGNCLIENT or CS\_BYTEALIGNWINDOW ; стиль окна

mov wc.lpfnWndProc,ptr$(WndProc) ; адрес процедуры WndProc

mov wc.cbClsExtra,0 ; количество байтов для структуры класса

mov wc.cbWndExtra,0 ; количество байтов для структуры окна

mrm wc.hInstance,hInstance ; заполнение поля дескриптора в структуре

mrm wc.hIcon, hIcon ; хэндл иконки

mrm wc.hCursor,hCursor ; хэндл курсора

mrm wc.hbrBackground,0 ; цвет окна

mov wc.lpszMenuName,0 ; заполнение поля в структуре с именем ресурса меню

mov wc.lpszClassName,ptr$(classname); имя класса

mrm wc.hIconSm,hIcon

invoke RegisterClassEx,ADDR wc ; регистрация класса окна

mov wid, 900 ; ширина пользовательского окна в пикселях

mov hgt, 300 ; высота пользовательского окна в пикселях

mov rax,sWid ; колич. пикселей монитора по x

sub rax,wid ; дельта Х = Х(монитора) - х(окна пользователя)

shr rax,1 ; получение середины Х

mov lft,rax ;

mov rax, sHgt ; колич. пикселей монитора по y

sub rax, hgt ;

shr rax, 1 ;

mov top, rax ;

invoke CreateWindowEx,WS\_EX\_LEFT or WS\_EX\_ACCEPTFILES, \

ADDR classname,ADDR caption, \

WS\_OVERLAPPED or WS\_VISIBLE or WS\_SYSMENU,\

lft,top,wid,hgt,0,0,hInstance,0

mov hWnd,rax ; сохранение дескриптора окна

call msgloop

ret

main endp

msgloop proc

LOCAL msg :MSG

LOCAL pmsg :QWORD

mov pmsg, ptr$(msg) ; получение адреса структуры сообщения

jmp gmsg ; jump directly to GetMessage()

mloop:

invoke TranslateMessage,pmsg

invoke DispatchMessage,pmsg

gmsg:

test rax, rv(GetMessage,pmsg,0,0,0) ; пока GetMessage не вернет ноль

jnz mloop

ret

msgloop endp

WndProc proc hWin:QWORD,uMsg:QWORD,wParam:QWORD,lParam:QWORD

LOCAL hdc:HDC ; резервирование стека для дескриптора окна

LOCAL ps:PAINTSTRUCT ; для структуры PAINTSTRUCT

LOCAL rect:RECT ; для структуры координат RECT

LOCAL leng:QWORD

.switch uMsg

.case WM\_DESTROY

invoke PostQuitMessage,NULL

.case WM\_PAINT ; если есть смс о перерисовании

invoke BeginPaint,hWnd, ADDR ps ; вызов подготовительной процедуры

mov hdc,rax ; сохранение контекста

invoke fptoa,res1,buf3

invoke TextOut,hdc,40,35,buf3,10

;invoke wsprintf,ADDR buf1,ADDR txt1 ; преобразование данных в текст

;invoke wsprintf,ADDR buf2,ADDR txt2 ; преобразование данных в текст

;invoke wsprintf,ADDR buf,ADDR txt3,res1 ; преобразование данных в текст

;invoke TextOut,hdc,20,0,addr buf1,109 ; вывод текста в окно

;invoke TextOut,hdc,20,20,addr buf2,35 ; вывод текста в окно

;invoke TextOut,hdc,20,60,addr buf,30 ; вывод текста в окно

invoke wsprintf,ADDR buf1,ADDR txt1,ADDR txt2, ADDR txt3

mov params.cbSize,SIZEOF MSGBOXPARAMSA ; размер структуры

mov params.hwndOwner,0 ; дескриптор окна владельца

invoke GetModuleHandle,0 ; получение дескриптора программы

mov params.hInstance,rax ; сохранение дескриптора программы

lea rax, buf1 ; адрес сообщения

mov params.lpszText,rax

lea rax,title1 ; адрес заглавия окна

mov params.lpszCaption,rax

mov params.dwStyle,MB\_USERICON ; стиль окна

mov params.lpszIcon,IDI\_ICON ; ресурс значка

mov params.dwContextHelpId,0 ; контекст справки

mov params.lpfnMsgBoxCallback,0

mov params.dwLanguageId,LANG\_NEUTRAL ; язык сообщения

lea rcx,params

invoke MessageBoxIndirect ; вызов окна с результатом работы и иконкой

invoke ExitProcess,0

invoke EndPaint, hWnd, ADDR ps

.endsw ; иначе обработка по умолчанию

invoke DefWindowProc,hWin,uMsg,wParam,lParam

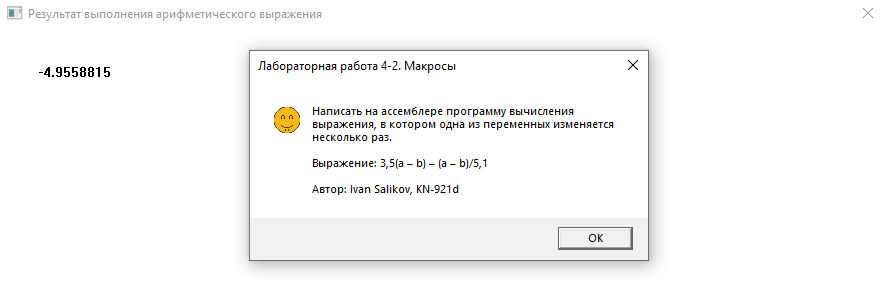
ret

WndProc endp

end

#define IDI\_ICON 1001

IDI\_ICON ICON DISCARDABLE "1.ico"



Висновки

При виконанні даної лабораторної роботи було набуто навичок по роботі з структурами та макросами.

Програми протестовані та виконуються без помилок.