**Звіт**

Автор: Салтиков Іван;

Група: КН-921д;

# Лабораторна робота №5. Рядкові команди. ММХ-команди

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

набути практичних навичок складання, налагодження та виконання програм з використанням команд ММХ для МП платформи х64.

Хід роботи

Iндивiдуальне завдання 1: Заданий текст з 8 слів, розділених пропуском. Визначити кількість повторень букви Е в кожному слові.

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

MSGBOXPARAMSA STRUCT ; объявление системной структуры

cbSize DWORD ?,?

hwndOwner QWORD ?

hInstance QWORD ?

lpszText QWORD ?

lpsztitle1 QWORD ?

dwStyle DWORD ?,?

lpszIcon QWORD ?

dwContextHelpId QWORD ?

lpfnMsgBoxCallback QWORD ?

dwLanguageId DWORD ?,?

MSGBOXPARAMSA ENDS

array STRUCT ; начало объявления пользовательской структуры

el1 dq ?

el2 dq ?

el3 dq ?

el4 dq ?

el5 dq ?

el6 dq ?

el7 dq ?

el8 dq ?

array ENDS ; конец объявления пользовательской структуры

.data?

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)

hImage dq ?

hStatic dq ?

.data

params MSGBOXPARAMSA <> ; объявление элемента системной структуры

mas1 db 'Cheerful and full of energy, enthusiastic, vivacious eagle-eyed. ' ; обрабатываемый текст

len1 equ ($-mas1)/type mas1 ; определение количества байтов в массиве mas1

countWord dq 0 ; счетчик слов

countE db 0 ; счетчик нужных символов

resE array <-1,?,?,?,?,?,?,?> ; инициализация 1 строки массива А

title1 db "Лабораторная работа 5-1. Строковые команды",0

task db "Задан текст из 8 слов, разделенных пропуском. Определить количество повторений буквы Е в каждом слове.", 0

buf1 dq 12 dup(0),0

ifmt1 db "Результат:",10,

"Слово 1: %d",10,

"Слово 2: %d",10,

"Слово 3: %d",10,

"Слово 4: %d",10,

"Слово 5: %d",10,

"Слово 6: %d",10,

"Слово 7: %d",10,

"Слово 8: %d",0

classname db "template\_class",0

BSIZE1 equ 100 ; кол-во байтов записываемых в файл

fName BYTE "result.txt",0 ; название файла для вывода

fHandle dq ? ; дескриптор файла

cWritten dq ? ; ячейки для адреса символов вывода

.code

entry\_point proc

GdiPlusBegin ; initialise GDIPlus

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

mov hImage,rv(ResImageLoad,20) ; макрос загрузки Bitmap

call main

GdiPlusEnd ; GdiPlus cleanup

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 ; hBrush 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,960 ; ширина пользовательского окна в пикселях

mov hgt,540 ; 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 title1, \

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 ; перейти в 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

.switch uMsg

.case WM\_COMMAND ; если выбрано меню

.switch wParam

.case 101 ; если выбран вывод информации о задании

.code

invoke MsgboxI,hWin,ptr$(task),"Информация о задании",MB\_OK,10

.case 102 ; если выбран вывод задания

.code

xor rsi,rsi ; очистка регистра RSI

xor rcx,rcx ; очистка регистра RCX

xor r11,r11 ; очистка регистра R11

xor r12,r12 ; очистка регистра R12

xor r13,r13 ; очистка регистра R13

lea r12,resE ; установка массива в начало

lea rsi,mas1 ; загрузка адреса массива mas1

mov r14, ' ' ; загрузка пробела

mov r15, 'e' ; загрузка символа е

mov rcx, len1 ; установка в счётчик максимальное значение букв

cld ; направление - вверх (признак DF)

cycle:

lodsb ; получаем символ

cmp rax,r15 ; сравниваем с Е

je foundE ; переход если найдена Е

cmp rax,r14 ; сравниваем с пробелом

je foundSpace ; переход если найден пробел

add rdi,1 ; переход на следующий символ

loop cycle ; переход в начало цикла

jmp \_end ; если весь текст проверен

foundE:

add r11,1 ; увеличение переменной кол-ва е

jmp cycle ; переход в начало цикла

foundSpace:

mov [r12],r11 ; запись кол-ва Е в массив результатов

add r12,8 ; перемещение на следующий элемент массива

mov r11,0 ; очистка регистра кол-ва нахождений Е

jmp cycle ; переход в начало цикла

\_end:

invoke wsprintf,ADDR buf1,ADDR ifmt1,resE.el1,resE.el2,resE.el3,resE.el4,resE.el5,resE.el6,resE.el7,resE.el8

invoke MessageBox,0,addr buf1,addr title1,MB\_OK ; вывод данных на экран

.case 103 ; если выбрано сохранение в файл

.data

msg1 db "Сохранение результата в файл.",0 ; уведомление о выходе

const1 dq -1

.code

mov rax,resE.el1

cmp rax,const1

je NoResult

jne HaveResult

NoResult:

invoke MsgboxI,hWin,"Результат вычислений отсутствует. Выполните расчёт в начале.","Ошибка",MB\_OK,10

jmp \_end2

HaveResult:

invoke MsgboxI,hWin,ptr$(msg1),"Сохранение в файл",MB\_OK,10 ; вывод окна с уведомлением

invoke CreateFile,ADDR fName,GENERIC\_WRITE,0,0,CREATE\_ALWAYS,FILE\_ATTRIBUTE\_ARCHIVE,0

mov fHandle,rax

invoke WriteFile,fHandle,ADDR buf1,BSIZE1,ADDR cWritten,0 ; запись результата в файл

\_end2:

xor rax,rax

.case 104 ; если выбран вывод информации об авторе

.data

msgtxt db "Автор: Saltikov Ivan",10,10,"Группа: KN-921d",0 ; вывод данных об авторе

.code

invoke MsgboxI,hWin,ptr$(msgtxt),"Информация об авторе",MB\_OK,10

.case 105 ; если выбран выход из программы

.data

msg db "Выход из программы.",0 ; вывод уведомления о выходе

.code

invoke MsgboxI,hWin,ptr$(msg),"Выход",MB\_OK,10

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0 ; уничтожение окна

.endsw

.case WM\_CREATE

invoke CreateWindowEx,WS\_EX\_LEFT,"STATIC",0,WS\_CHILD or WS\_VISIBLE or SS\_BITMAP,\

0,0,0,0,hWin,hInstance,0,0

mov hStatic,rax

invoke SendMessage,hStatic,STM\_SETIMAGE,IMAGE\_BITMAP,hImage ; сообщение окну

invoke LoadMenu,hInstance,100 ; загружает меню из exe-файла

invoke SetMenu,hWin,rax ; связывает меню с окном

.return 0

.case WM\_CLOSE

invoke SendMessage,hWin,WM\_DESTROY,0,0

.case WM\_DESTROY

invoke PostQuitMessage,NULL

.endsw

invoke DefWindowProc,hWin,uMsg,wParam,lParam

ret

WndProc endp

end

#define IDR\_MENU1 10000

#define IDM\_File 10001

#define IDM\_Help 10003

#define IDM\_Result 102

#define IDM\_Save 103

#define IDM\_Task 101

#define IDM\_Info 104

#define IDM\_Exit 105

#include "C:/masm64/include64/Resource.h"

10 ICON DISCARDABLE "1.ico"

20 RCDATA DISCARDABLE "12.jpg"

100 MENUEX

BEGIN

POPUP "Task"

BEGIN

MENUITEM "Show result",IDM\_Result

MENUITEM "Save to file",IDM\_Save

END

POPUP "Info"

BEGIN

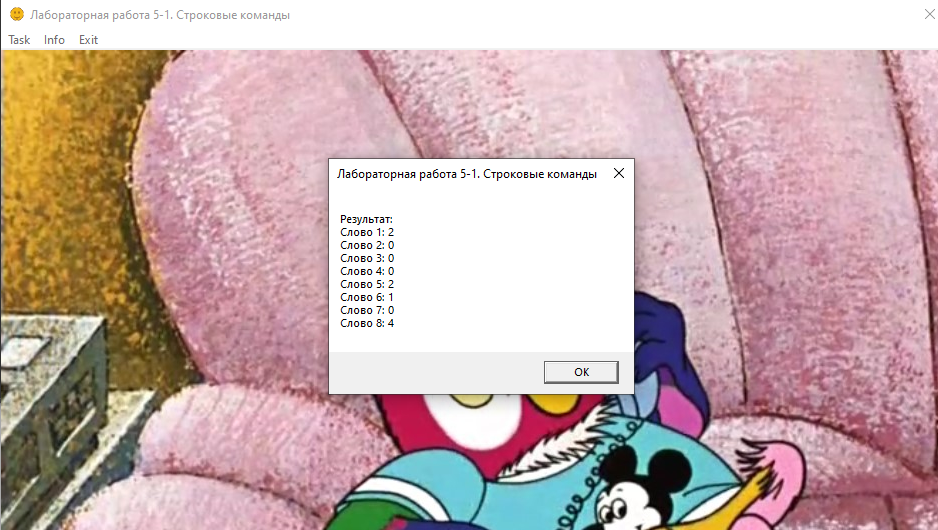
MENUITEM "Show task info",IDM\_Task

MENUITEM "Show author info",IDM\_Info

END

MENUITEM "Exit",IDM\_Exit

END



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

MSGBOXPARAMSA STRUCT ; объявление системной структуры

cbSize DWORD ?,?

hwndOwner QWORD ?

hInstance QWORD ?

lpszText QWORD ?

lpsztitle1 QWORD ?

dwStyle DWORD ?,?

lpszIcon QWORD ?

dwContextHelpId QWORD ?

lpfnMsgBoxCallback QWORD ?

dwLanguageId DWORD ?,?

MSGBOXPARAMSA ENDS

fpuMacr macro ; макрос операции a-e/b

fld \_e

fdiv \_b

fld \_a

fsubr

endm ;; окончание макроса

.data?

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)

hImage dq ?

hStatic dq ?

.data

params MSGBOXPARAMSA <> ; объявление элемента системной структуры

title1 db "Лабораторная работа 5-2. MMX команды",0

task1 db "Выполнить операции логического сложения целых чисел 2-х массивов. Если второе число больше 55, то выполнить операцию a – e/b – de, где a, b, c, d – вещественные числа; иначе – выполнить операцию a – e/b.", 0

buf1 dq 12 dup(0),0

txt1 db "Результат: %I64d",10,0

\_a real4 20.1 ; объявление переменных

\_b real4 5.2

\_c real4 2.8

\_d real4 1.3

\_e real4 6.7

res1 real4 0.0

arr1 dw 1,15,3,4 ; массив чисел arr1

len1 equ ($-arr1)/type arr1 ; размер массива arr1

arr2 dw 5,6,7,5 ; массив чисел arr2

len2 equ ($-arr2)/type arr2 ; количество чисел массива

arr3 dw (len1+len2) dup(0) ; размер буфера для чисел массивов

classname db "template\_class",0

.code

entry\_point proc

GdiPlusBegin ; initialise GDIPlus

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

mov hImage,rv(ResImageLoad,20) ; макрос загрузки Bitmap

call main

GdiPlusEnd ; GdiPlus cleanup

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 ; hBrush 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,960 ; ширина пользовательского окна в пикселях

mov hgt,540 ; 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 title1, \

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 ; перейти в 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

.switch uMsg

.case WM\_COMMAND ; если выбрано меню

.switch wParam

.case 101 ; если выбран вывод информации о задании

.code

invoke MsgboxI,hWin,ptr$(task1),"Информация о задании",MB\_OK,10

.case 102 ; если выбран вывод задания

.code

movq MM0,QWORD PTR arr1 ; загрузка массива чисел arr1

movq MM1,QWORD PTR arr2 ; загрузка массива чисел arr2

paddw MM0,MM1 ; параллельное циклическое сложение

movq QWORD PTR arr3,MM0 ; сохранение результата

pextrw rax,mm0,1 ; копирование первого слова в EAX

emms ; последняя MMX-команда

cmp eax,55 ; проверка больше ли 55

jg MoreThan55 ; если число больше 55

jmp LessThan55 ; если число меньше

MoreThan55:

fpuMacr ; a-e/b

fld \_d ; загрузка d

fmul \_e ; d\*e

fsub ; a-e/b-de

fisttp res1 ; сохранение результата

jmp \_end ; переход в конец

LessThan55:

fpuMacr ; a-e/b

fisttp res1 ; сохранение результата

\_end:

invoke wsprintf,ADDR buf1,ADDR txt1,res1

invoke MsgboxI,hWin,ADDR buf1,"Результат",MB\_OK,10

.case 104 ; если выбран вывод информации об авторе

.data

msgtxt db "Автор: Saltikov Ivan",10,10,"Группа: KN-921d",0 ; вывод данных об авторе

.code

invoke MsgboxI,hWin,ptr$(msgtxt),"Информация об авторе",MB\_OK,10

.case 105 ; если выбран выход из программы

.data

msg db "Выход из программы.",0 ; вывод уведомления о выходе

.code

invoke MsgboxI,hWin,ptr$(msg),"Выход",MB\_OK,10

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0 ; уничтожение окна

.endsw

.case WM\_CREATE

invoke CreateWindowEx,WS\_EX\_LEFT,"STATIC",0,WS\_CHILD or WS\_VISIBLE or SS\_BITMAP,\

0,0,0,0,hWin,hInstance,0,0

mov hStatic,rax

invoke SendMessage,hStatic,STM\_SETIMAGE,IMAGE\_BITMAP,hImage ; сообщение окну

invoke LoadMenu,hInstance,100 ; загружает меню из exe-файла

invoke SetMenu,hWin,rax ; связывает меню с окном

.return 0

.case WM\_CLOSE

invoke SendMessage,hWin,WM\_DESTROY,0,0

.case WM\_DESTROY

invoke PostQuitMessage,NULL

.endsw

invoke DefWindowProc,hWin,uMsg,wParam,lParam

ret

WndProc endp

end

#define IDR\_MENU1 10000

#define IDM\_File 10001

#define IDM\_Help 10003

#define IDM\_Result 102

#define IDM\_Task 101

#define IDM\_Info 104

#define IDM\_Exit 105

#include "C:/masm64/include64/Resource.h"

10 ICON DISCARDABLE "1.ico"

20 RCDATA DISCARDABLE "12.jpg"

100 MENUEX

BEGIN

POPUP "Task"

BEGIN

MENUITEM "Show result",IDM\_Result

END

POPUP "Info"

BEGIN

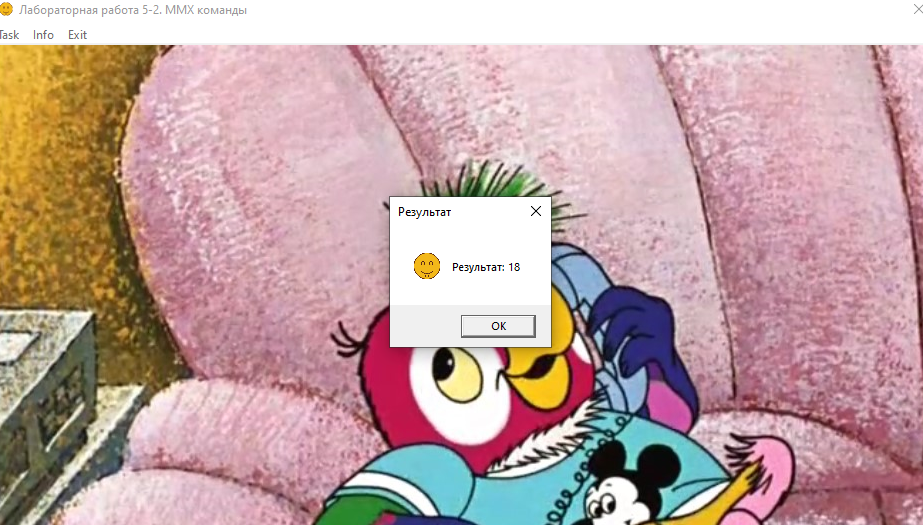
MENUITEM "Show task info",IDM\_Task

MENUITEM "Show author info",IDM\_Info

END

MENUITEM "Exit",IDM\_Exit

END



Висновки

При виконанні даної лабораторної роботи було набуто навичок при роботі з рядковими командами та MMX-команди.

Програми протестовані та виконуються без помилок.