МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ЕВФРОСИНИИ ПОЛОЦКОЙ»

Факультет информационных технологий Кафедра технологий программирования

**Отчет по выполнению лабораторной работы №3**

**по курсу «Конструирование программного обеспечения»**

«Работа со строковыми данными в языке низкого уровня»

Выполнил: студент группы 22-ИТ-зд

Шастовская М.С.

Проверил:

Скуковская А.А.

Полоцк, 2024

**Цель:** Целью работы является изучения механизма и инструментов работы со строками на языке низкого уровня.

# Задание

Написать программу со следующим алгоритмом:

− вывести на экран приглашение, и ввести с клавиатуры символьную

подстроку для поиска;

− вывести на экран еще одно приглашение, и ввести с клавиатуры

символьную строку с именем файла;

− открыть файл, прочитать его в буфер и найти в содержимом подстроку,

используя строковые инструкции;

− вывести результат поиска в виде сообщения «Найдено» или «Не найдено»;

− завершить программу.

**Решение**

includelib kernel32.lib

extrn GetStdHandle: proc

extrn ReadFile: proc

extrn WriteFile: proc

extrn CreateFileA: proc

extrn CloseHandle: proc

.data

firstMessage db 'Input string to find', 0dh, 0ah, 0

firstmlen = $ - firstMessage

secondMessage db 'Input file path', 0dh, 0ah, 0

secondmlen = $ - secondMessage

mcontain db 0dh, 0ah, 'Contain!!!', 0dh, 0ah, 0

mclen = $ - mcontain

mnotcontain db 0dh, 0ah, 'Not Contain!!!', 0dh, 0ah, 0

mnclen = $ - mnotcontain

stringBuffer byte 10 dup (?)

sblen = $ - stringBuffer

filepathBuffer byte 5 dup (?)

fpblen = $ - filepathBuffer

buffer byte 100 dup (?)

bufferLen = $ - buffer

filename byte "test.txt"

fHandle qword 0

.code

read proc

sub rsp, 56 ; 32 (shadow storage) + 8 (5-й параметр ReadFile) + 8 (byttesRead) + 8 (выравнивание)

mov r8, rcx

mov rcx, rax

mov rdx, rdi

lea r9, bytesRead

mov qword ptr [rsp + 32], 0

call ReadFile

test rax, rax

mov eax, bytesRead

jnz exit

mov rax, -

exit:

add rsp, 56

ret

bytesRead equ [rsp+40]

read endp

readFromConsole proc

sub rsp, 32

push rcx

mov rcx, -10

call GetStdHandle

pop rcx

call read

add rsp, 32

ret

readFromConsole endp

write proc

sub rsp, 56

mov rdx, rsi

mov r8, rcx

mov rcx, rax

lea r9, bytesWritten

mov qword ptr [rsp + 32], 0

call WriteFile

test rax, rax

mov eax, bytesWritten

jnz exit

mov rax, -1

exit:

add rsp, 56

ret

bytesWritten equ [rsp+40]

write endp

writeToConsole proc

sub rsp, 32

push rcx

mov rcx, -11

call GetStdHandle

pop rcx

call write

add rsp, 32

ret

writeToConsole endp

openFile proc

sub rsp, 56

mov rcx, rsi

mov rdx, 0C0000000h

xor r8, r8

xor r9, r9

mov r10, 4

mov [rsp + 32], r10

mov r10, 128

mov [rsp + 40], r10

mov [rsp + 48], r9

call CreateFileA

add rsp, 56

ret

openFile endp

closeFile proc

sub rsp, 40

mov rcx, rax

call CloseHandle

add rsp, 40

ret

closeFile endp

main proc

lea rsi, firstMessage

mov rcx, firstmlen

call writeToConsole

lea rdi, stringBuffer

mov rcx, sblen

call readFromConsole

lea rsi, secondMessage

mov rcx, secondmlen

call writeToConsole

lea rdi, filepathBuffer

mov rcx, fpblen

call readFromConsole

lea rsi, stringBuffer

mov rcx, sblen

call writeToConsole

lea rsi, filepathBuffer

mov rcx, fpblen

call writeToConsole

lea rsi, filepathBuffer

call openFile

mov fHandle, rax

mov rax, fHandle

lea rdi, buffer

mov rcx, bufferLen

call read

mov rcx, rax

lea rsi, buffer

call writeToConsole

mov rdi, offset buffer

mov rsi, offset stringBuffer

mov rcx, sblen - 2

repe cmpsb

jz notcontain

contain:

lea rsi, mcontain

mov rcx, mclen

call writeToConsole

jmp exit

notcontain:

lea rsi, mnotcontain

mov rcx, mnclen

call writeToConsole

exit:

ret

main endp

end

**Вывод:** Изучила механизм и инструменты для работы со строками на языке низкого уровня.