Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

Кафедра прикладной математики и кибернетики

Отчет по лабораторной работе

по дисциплине «Операционные системы»

Лабораторная работа №4

Выполнил: студент 3 курса группы ИП-811 Мироненко К. А.

Проверил: профессор кафедры ПМиК Малков Е. А.

**ОГЛАВЛЕНИЕ**

[1. Постановка задачи 3](#_Toc53520694)

[2. Описание работы программы 4](#_Toc53520695)

[3. Примеры работы программы 5](#_Toc53520696)

[*Приложение* Листинг 8](#_Toc53520697)

# Постановка задачи

**Цель:** знакомство с созданием процессов с помощью POSIX API в Linux и

Windows.

**Упражнение 1.** Протестируйте различные варианты программ, рассмотренных на Лекции 4.

**Задание:** Разработайте оболочку для выполнения (некоторых) команд командной строки Windows, используя в качестве синонимов названия команд GNU/bash.

# Описание работы программы

Используемые функции:

* std::vector<std::string> Separate(const std::string& str)

Функция, разбивающая предложение на вектор слов

* int command(std::vector<std::string> str)

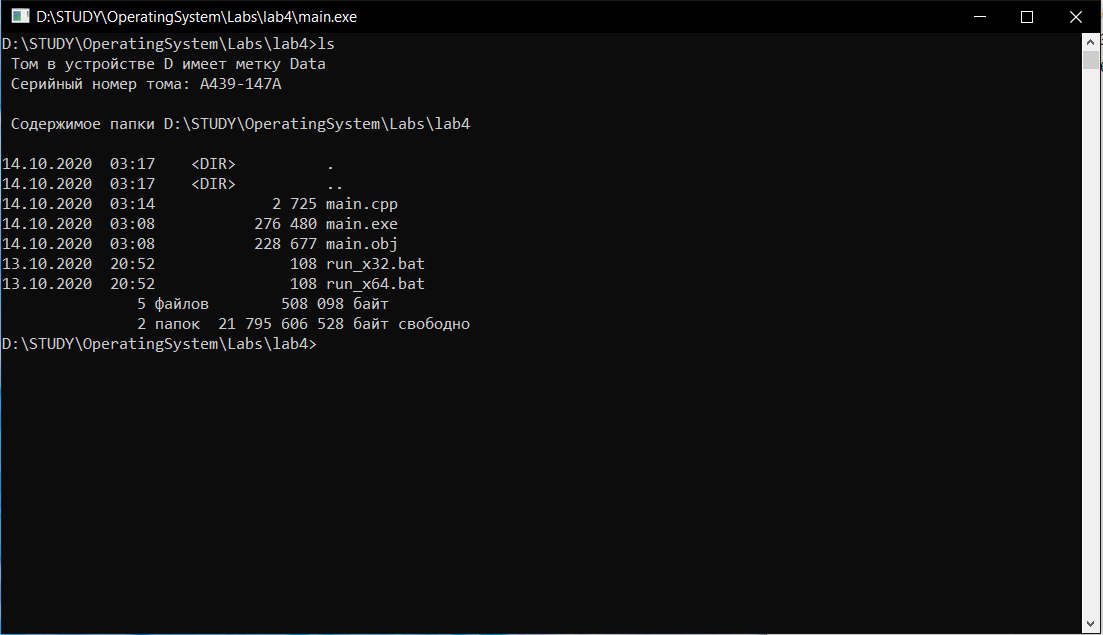
Функция, анализирующая входной вектор и выполняющая соответствующие инструкции

* \_spawnvp(int mode, const wchar\_t \*cmdname, const wchar\_t \*const \*argv) Функция, создающая и выполняющая процесс.
* GlobalMemoryStatusExФункция GlobalMemoryStatusEx получает информацию о текущей памяти компьютера (физической и виртуальной).

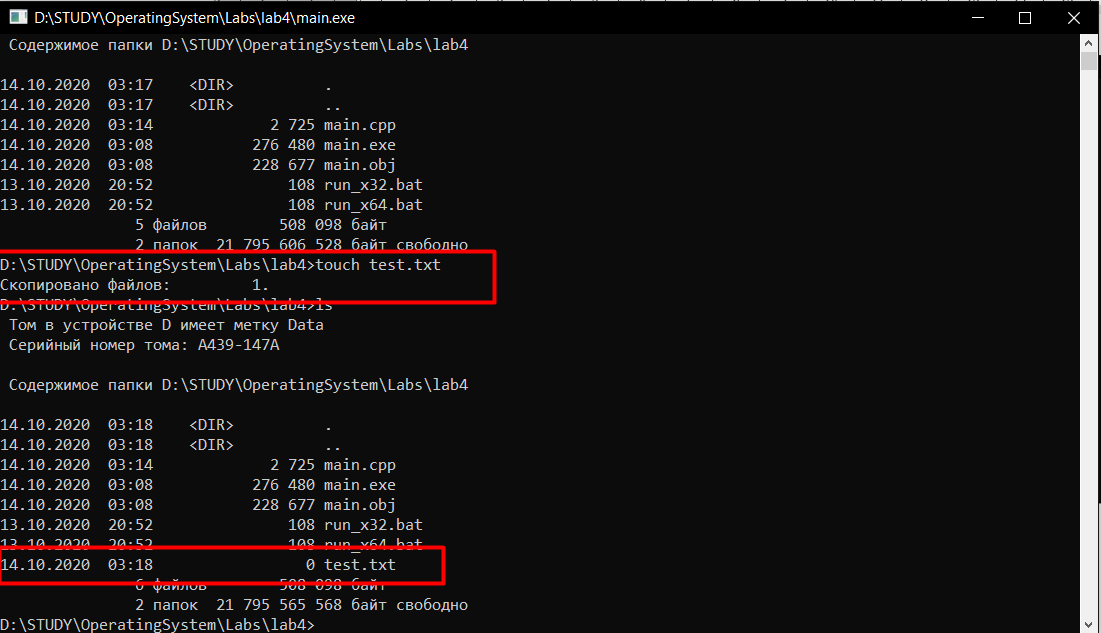
Используемые синонимы названия команд GNU/bash:

* ls – выводит список файлов и подкаталогов, находящихся в заданном каталоге.
* clear – очистка эмулятора терминала
* touch – создание файла
* rm – удаление файла

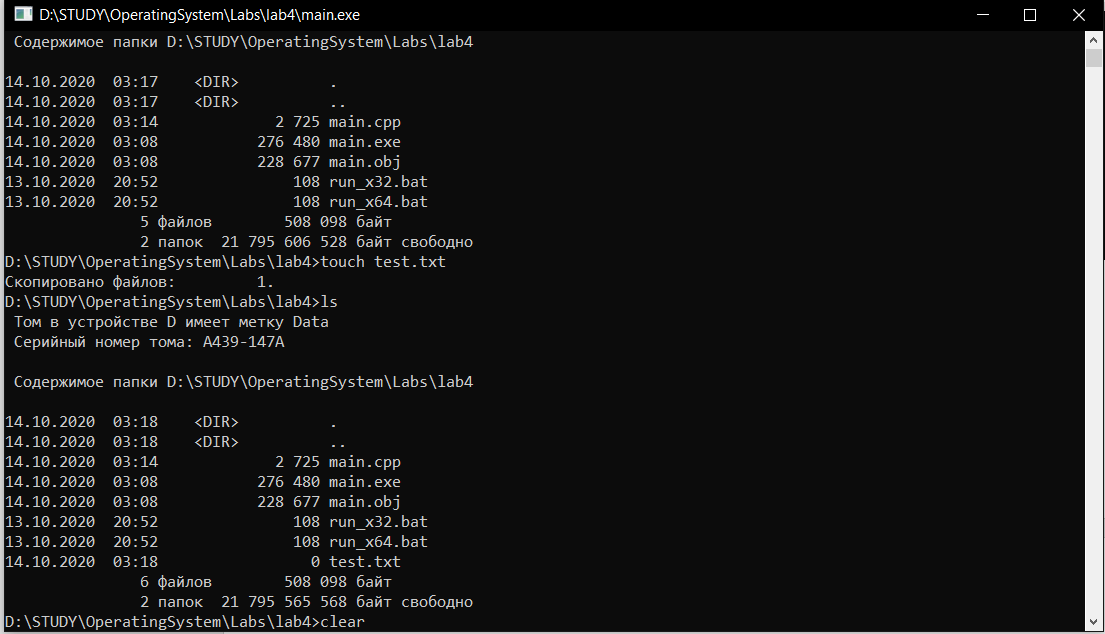
# Примеры работы программы

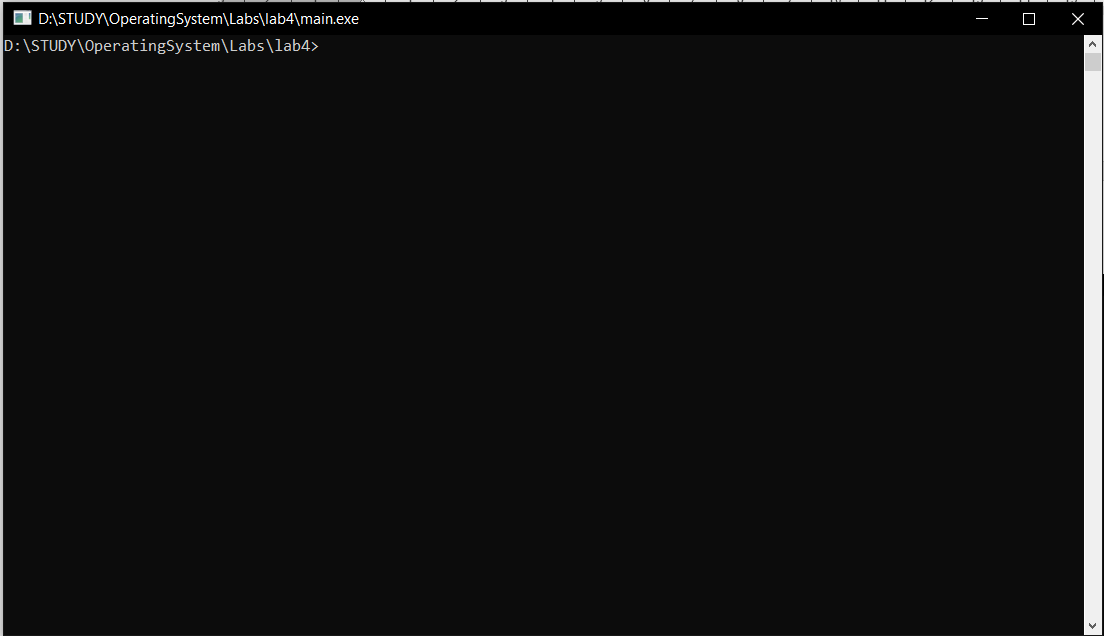


(Синоним ls)

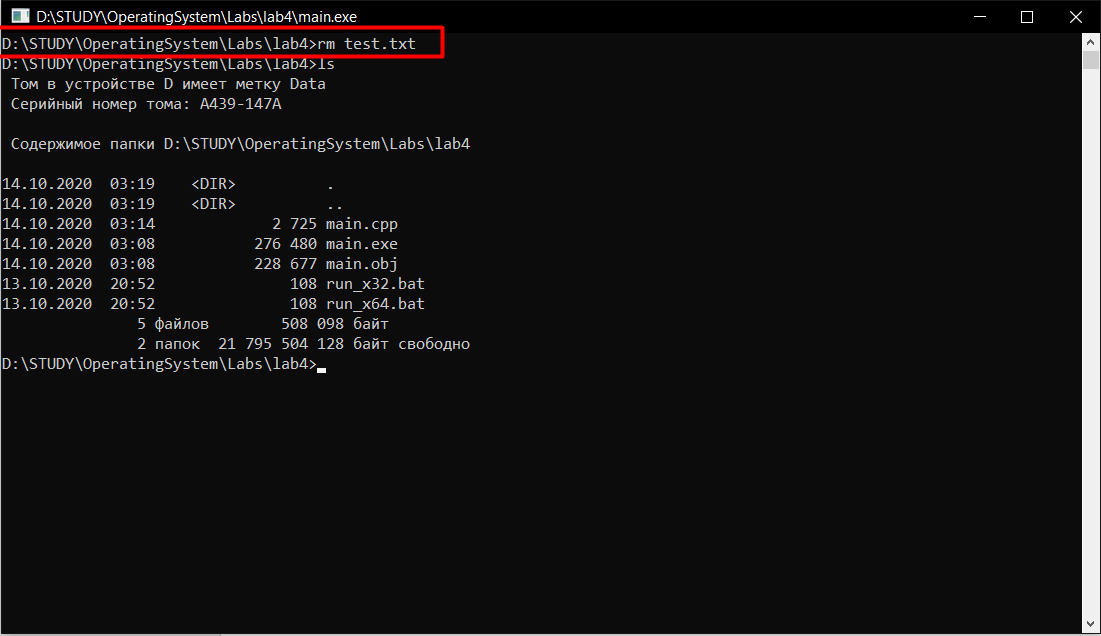


(Синоним touch)





(Синоним clear)



(Синоним rm)

# Приложение. Листинг

**main.cpp**

#include <windows.h>

#include <stdio.h>

#include <iostream>

#include <locale.h>

#include <process.h>

#include <string>

#include <vector>

std::vector<std::string> Separate(const std::string& str)

{

std::vector<std::string> vec;

std::string s;

for (auto &i : str)

{

if (i == ' ')

{

if (s.size() != 0)

{

vec.push\_back(s);

s.clear();

}

}

else

s.push\_back(i);

}

if (s.size() != 0){

vec.push\_back(s);

s.clear();

}

return vec;

}

int command(std::vector<std::string> str)

{

if (str[0] == "ls")

{

if (str.size() == 1)

{

char \*arg[] = {(char \*)"cmd", (char \*)"/c", (char \*)"dir", NULL};

\_spawnvp(P\_WAIT, arg[0], arg);

}

else if (str.size() == 2)

{

char \*arg[] = {(char \*)"cmd", (char \*)"/c", (char \*)"dir", (char \*)str[1].c\_str(), NULL};

\_spawnvp(P\_WAIT, arg[0], arg);

}

else

printf("[ ls ] Неверное количество аргументов!");

}

else if (str[0] == "clear")

{

if (str.size() == 1){

char \*arg[] = {(char \*)"cmd", (char \*)"/c", (char \*)"CLS", NULL};

\_spawnvp(P\_WAIT, arg[0], arg);

}

else

printf("[ clear ] Неверное количество аргументов!");

}

else if (str[0] == "touch")

{

if (str.size() == 2){

char \*arg[] = {(char \*)"cmd", (char \*)"/c", (char \*)"copy NUL ", (char \*)str[1].c\_str(), NULL};

\_spawnvp(P\_WAIT, arg[0], arg);

}

else

printf("[ touch ] Неверное количество аргументов!");

}

else if (str[0] == "rm")

{

if (str.size() == 2){

char \*arg[] = {(char \*)"cmd", (char \*)"/c", (char \*)"del ", (char \*)str[1].c\_str(), NULL};

\_spawnvp(P\_WAIT, arg[0], arg);

}

else

printf("[ rm ] Неверное количество аргументов!");

}

else if (str[0] == "exit")

return 1;

else

{

printf("[ ERROR ] Неизвестная команда!");

}

return 0;

}

int main(int argc, char\* argv[])

{

setlocale(LC\_ALL, "");

system("cls");

TCHAR curPath[MAX\_PATH] = "";

GetCurrentDirectory(MAX\_PATH, curPath);

bool flag = 0;

while (!flag)

{

std::string str;

printf("%s>", curPath);

getline(std::cin, str, '\n');

flag = command(Separate(str));

}

system("cls");

return 0;

}