Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

Федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет   
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет ИТР

Кафедра ПИн

*Лабораторная №6*

По Сети электронных вычислительных машин

Тема Работа с FTP-сервером

Руководитель

Астафьев А.В.

(фамилия, инициалы)

(подпись) (дата)

Студент ПИн-121

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись) (дата)

Муром 2024

**Лабораторная №6**

**Тема:** Работа с FTP-сервером

**Задачи:**   
Разработать FTP-клиент, осуществляющий: замену файла с указанным именем на указанный файл.

**Ход работы:**

Клиентская часть:

using BytesRoad.Net.Ftp;

namespace work

{

internal class Program

{

static FtpClient client = new FtpClient()

{

PassiveMode = true

};

const int TimeoutFTP = 30000; //Таймаут.

const string FTP\_SERVER = "127.0.0.1";

const int FTP\_PORT = 21;

const string FTP\_USER = "anonymous";

const string FTP\_PASSWORD = "";

static void Main(string[] args)

{

//Подключаемся к FTP серверу.

try

{

client.Connect(TimeoutFTP, FTP\_SERVER, FTP\_PORT);

}

catch (BytesRoad.Net.Ftp.FtpTimeoutException error)

{

Console.WriteLine("Время ожидания истекло! Сервер не отвечает. " + error.Message);

Console.ReadKey();

return;

}

catch (System.Net.Sockets.SocketException error)

{

Console.WriteLine(error.Message);

Console.ReadKey();

return;

}

try

{

client.Login(TimeoutFTP, FTP\_USER, FTP\_PASSWORD);

}

catch (BytesRoad.Net.Ftp.FtpTimeoutException error)

{

Console.WriteLine("Время ожидания истекло! Сервер не отвечает. " + error.Message);

Console.ReadKey();

return;

}

catch (System.Net.Sockets.SocketException error)

{

Console.WriteLine(error.Message);

Console.ReadKey();

return;

}

f();

}

static void f()

{

while (true)

{

Console.Write(client.GetWorkingDirectory(TimeoutFTP) + ">");

string command = Console.ReadLine();

string[] commands = command.Split(" ");

if (commands.Length > 0)

{

switch (commands[0])

{

case "exit":

if (fExit())

{

return;

}

break;

case "help": fHelp(); break;

case "show": fShow(); break;

case "cd": fCD(commands); break;

case "upload": fUpload(commands); break;

default: Console.WriteLine("Error: команда не найдена"); break;

}

}

else

{

Console.WriteLine("Error: Комманда не введена!");

}

}

}

static bool fExit()

{

Console.WriteLine("Выйти? [y/n]");

while (true)

{

char key = Console.ReadKey(false).KeyChar;

if (key == 'y')

{

return true;

}

else if(key == 'n')

{

break;

}

}

return false;

}

static void fHelp()

{

Console.WriteLine("Commands:");

Console.WriteLine("help - показывает список комманд");

Console.WriteLine("cd [dir name]- перемещает вас в указанную директорию, если она есть");

Console.WriteLine("upload [file name] [file path on pc]- обновляет указанный файл, на файл из системы");

Console.WriteLine("show - показать все файлы и папки");

Console.WriteLine("exit - выход из программы");

}

static void fCD(string[] args)

{

if(args.Length > 1)

{

try

{

client.ChangeDirectory(TimeoutFTP, args[1]);

fShow();

}

catch

{

Console.WriteLine("Error: ошибка при переходе в папку");

}

}

else

{

Console.WriteLine("Error: неуказанны параметры");

}

}

static void fShow()

{

FtpItem[] dirs = client.GetDirectoryList(TimeoutFTP);

foreach (FtpItem item in dirs)

{

Console.WriteLine(item.ItemType.ToString() + ": " + item.Name);

}

}

static void fUpload(string[] args)

{//upload 1.txt D:/test.txt

if (args.Length > 2)

{

string fileName = args[2];

if (File.Exists(fileName))

{

byte[] bytes = null;

try

{

bytes = File.ReadAllBytes(fileName);

}

catch

{

Console.WriteLine("Error: не удалось получить доступ к файлу на компьютере");

return;

}

try

{

client.PutFile(TimeoutFTP, args[1], bytes);

}

catch

{

Console.WriteLine("Error: ошибка при обновлении файла");

}

}

else

{

Console.WriteLine("Error: указанного файла не существует");

}

}

else

{

Console.WriteLine("Error: неуказанны параметры");

}

}

}

}

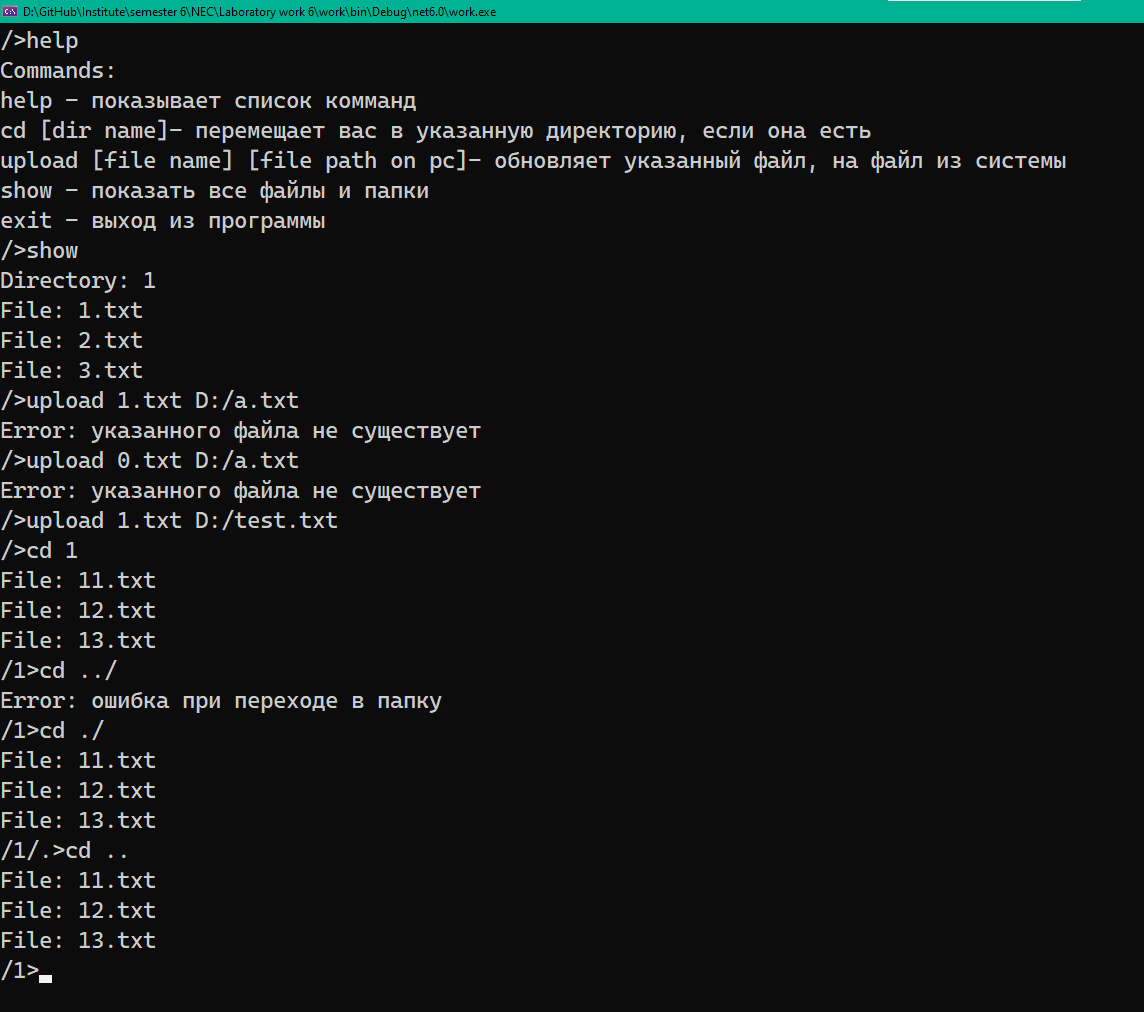


Рисунок 1 - пример работы программы.