Министерство науки и высшего образования Российской Федерации

Новосибирский Государственный технический университет

Кафедра автоматизированных систем управления



**Отчет по лабораторной работе №3**

**по дисциплине «Технологии и методы программирования»**

**«**СЕТЕВОЕ ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СОКЕТОВ**»**

Выполнил

студент группы АП-027

Савченко С.В.

г. Новосибирск

2022 г

**1. ЦЕЛЬ РАБОТЫ**

Получить навыки создания сетевых приложений на основе технологии Socket

**2. ПОСТАНОВКА ЗАДАЧИ И РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ**

**Задание 1:** разработать программу клиента, которая должна:

* запрашивать у пользователя адрес программы-сервера;
* устанавливать соединение с сервером;
* передавать на сервер имя выбранного каталога;
* принимать ответ от сервера и выводить его на экран;
* закрывать соединение с сервером.

**Задание 2:** разработать программу сервера, которая должна:

* ожидать запрос от программ клиентов на соединение;
* устанавливать соединение с клиентами, при установлении соединения передавать список логических устройств клиенту;
* принимать от клиента данные:
* если это имя каталога, передавать клиенту структуру каталога (имена файлов и подкаталогов);
* если это имя текстового файла, передавать клиенту содержимое файла.
* закрывать соединение с клиентом при получении уведомления.

Код:

|  |
| --- |
| Сервер |
| using System.Net;  using System.Net.Sockets;  using System.Text;  const string ip = "127.0.0.1";  const int port = 8080;  var tcpEndPoint = new IPEndPoint(IPAddress.Parse(ip), port);  var tspSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);  tspSocket.Bind(tcpEndPoint);  tspSocket.Listen(1);  bool quit = false;  var listener = tspSocket.Accept();  while (!quit)  {  var data = new byte[256];  var size = 0;  var str = new StringBuilder();  do  {  size = listener.Receive(data);  str.Append(Encoding.UTF8.GetString(data, 0, size));  }  while (listener.Available > 0);  if (str.ToString().Contains("exit"))  {  quit = true;  }  else if (File.Exists(str.ToString()))  {  var fileInfo = File.ReadAllText(str.ToString());  listener.Send(Encoding.UTF8.GetBytes(fileInfo));  }  else if (Directory.Exists(str.ToString()))  {  var files = Directory.GetFiles(str.ToString());  var dir = Directory.GetDirectories(str.ToString());  foreach (var d in dir)  {  listener.Send(Encoding.UTF8.GetBytes(d + "\n"));  }  foreach (var item in files)  {  listener.Send(Encoding.UTF8.GetBytes(item + "\n"));  }  }  else  {  listener.Send(Encoding.UTF8.GetBytes("command not found\n"));  listener.Send(Encoding.UTF8.GetBytes("You can:\nView the constents of the file\nView the contents of the directory"));  }  }  listener.Shutdown(SocketShutdown.Both);  listener.Close(); |
| Клиент |
| using System.Net;  using System.Net.Sockets;  using System.Text;  const string ip = "127.0.0.1";  const int port = 8080;  var tcpEndPoint = new IPEndPoint(IPAddress.Parse(ip), port);  var tspSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);  tspSocket.Connect(tcpEndPoint);  bool quit = false;  while (!quit)  {  Console.Write("Enter info: ");  var message = Console.ReadLine();  var data = Encoding.UTF8.GetBytes(message);  var buffer = new byte[256];  var size = 0;  var answer = new StringBuilder();  tspSocket.Send(data);  do  {  size = tspSocket.Receive(buffer);  answer.Append(Encoding.UTF8.GetString(buffer));  }  while (tspSocket.Available > 0);  if (message.Contains("exit"))  {  quit = true;  }  Console.WriteLine(answer.ToString());  }  tspSocket.Shutdown(SocketShutdown.Both);  tspSocket.Close(); |

Результат работы для вывода содержания директорий и файлов представлен на рис. 1

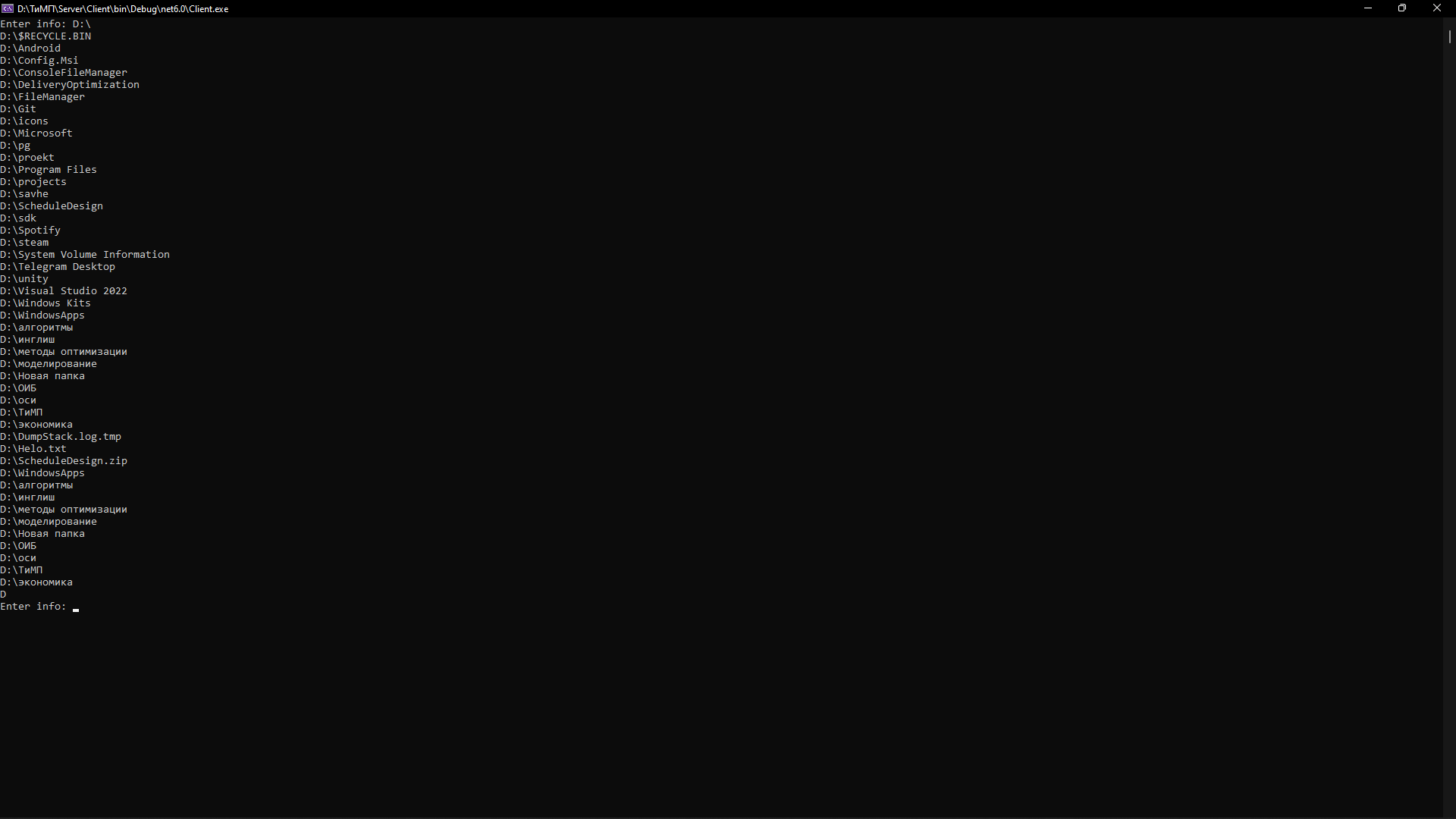
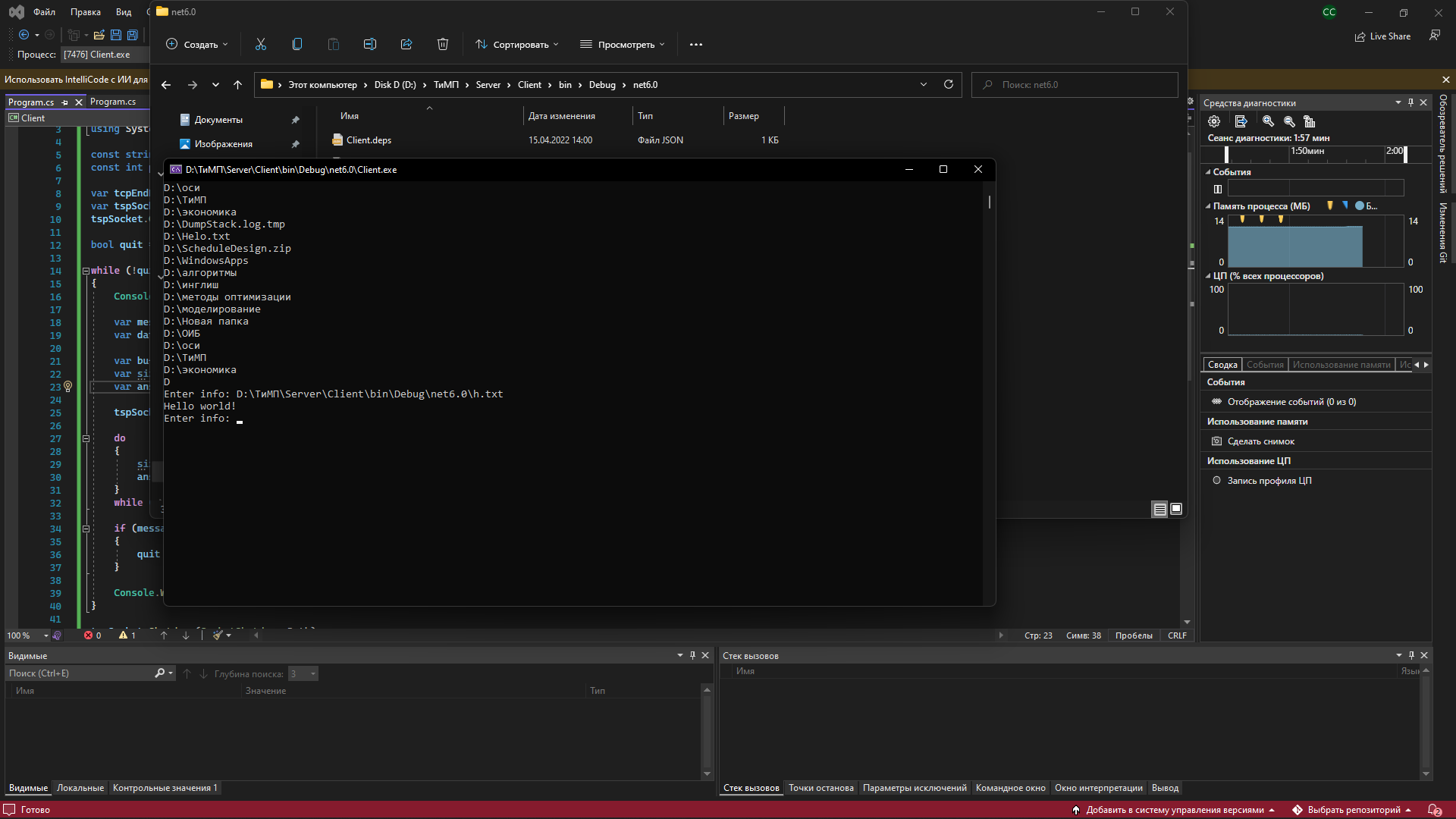


Рисунок 1 – Вывод директорий и файлов диска D

Результат вывода содержимого txt файла представлен на рис. 2



Содержимое файла D:\ТиМП\Server\Client\bin\Debug\net6.0\h.txt представлено на рис. 3

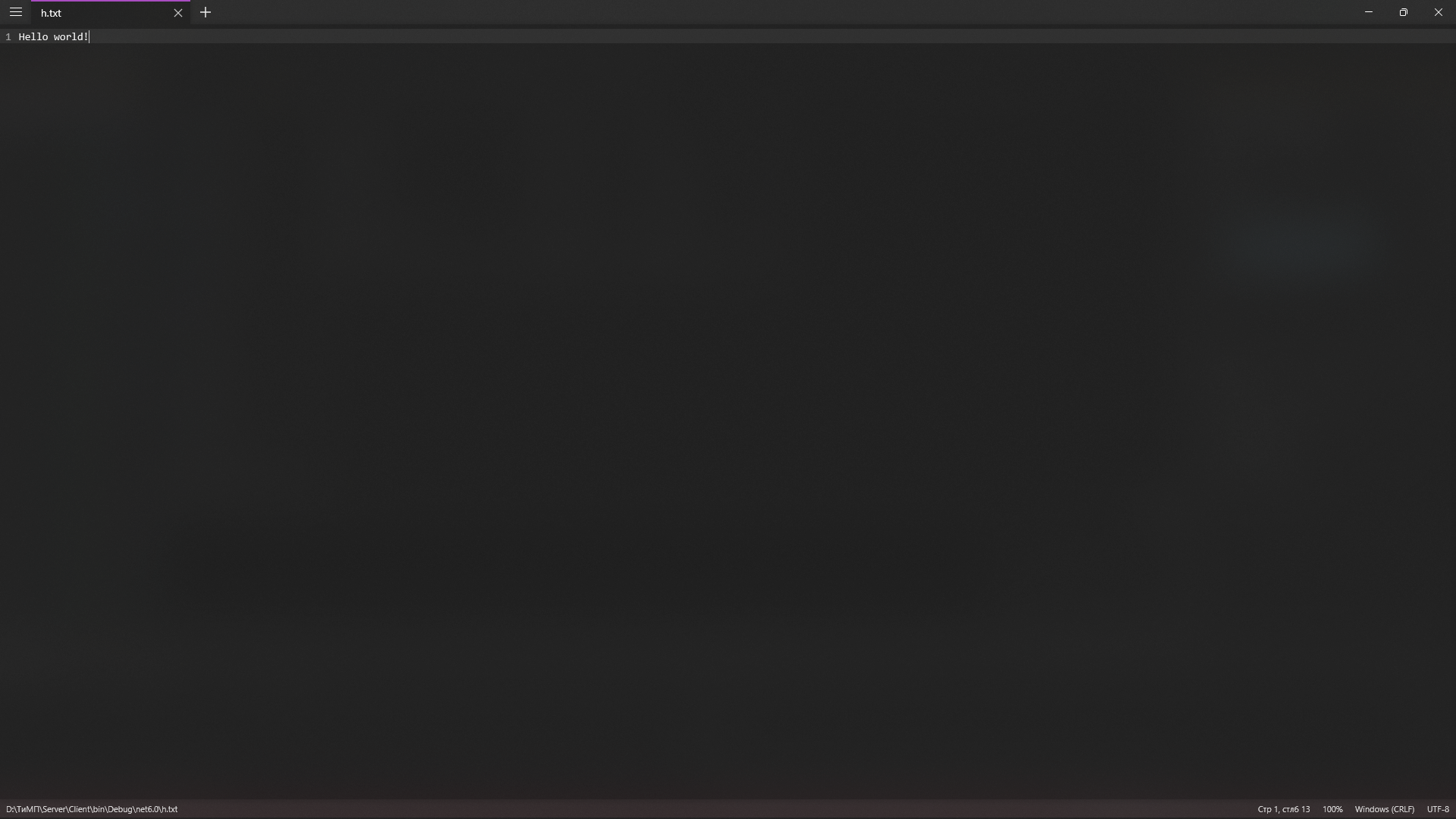


Рисунок 3 – Содержимое D:\ТиМП\Server\Client\bin\Debug\net6.0\h.tx