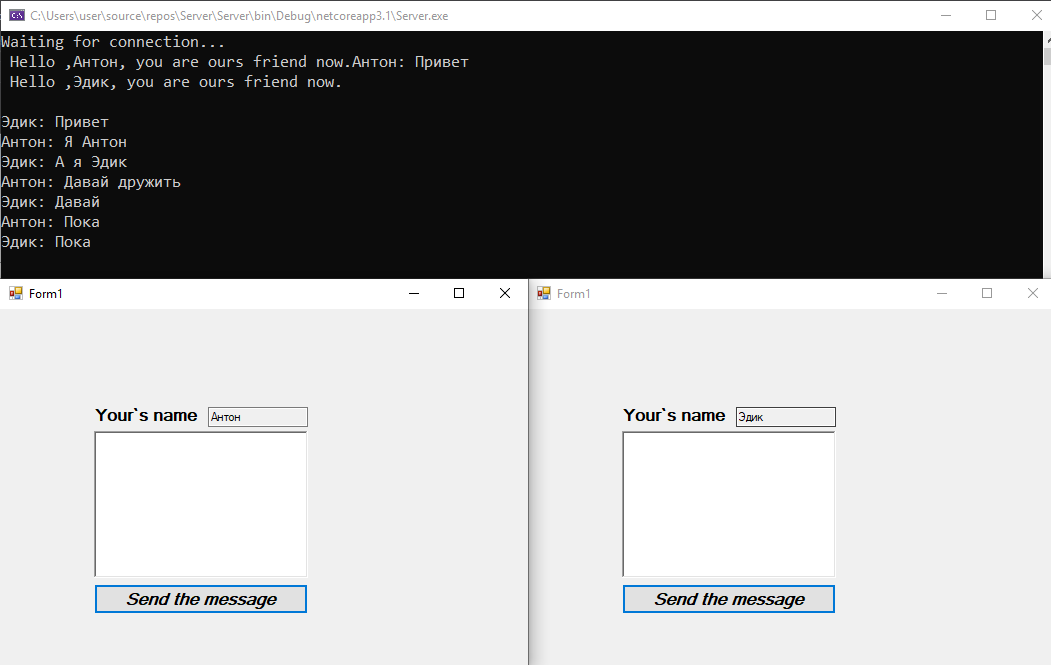
**Отчет:**



**Простейший чат на протоколе TCP, реализованный с помощью сокетов.**

1. Ожидание подключений к серверу
2. Ввод клиента через приложение имя и текста
3. Отображение уведомлений о подключенных пользователях и их его сообщения.

**Код программы сервера:**

using System;

using System.Net.Sockets;

using System.Net;

using System.Text;

using System.Threading;

namespace std

{

public class ClientObject

{

public Socket client;

public ClientObject(Socket tcpClient)

{

client = tcpClient;

}

public void Process()

{

try

{

byte[] data = new byte[64]; // буфер для получаемых данных

while (true)

{

// получаем сообщение

StringBuilder builder = new StringBuilder();

int bytes = 0;

do

{

bytes = client.Receive(data);

builder.Append(Encoding.Unicode.GetString(data, 0, bytes));

}

while (client.Available > 0);

string message = builder.ToString();

Console.WriteLine(message);

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

}

class Program

{

const int port = 8888;

static void Main(string[] args)

{

IPEndPoint ipPoint = new IPEndPoint(IPAddress.Parse("127.0.0.1"), port);

Socket listenSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

try

{

listenSocket.Bind(ipPoint);

listenSocket.Listen(10);

Console.WriteLine("Waiting for connection...");

while (true)

{

Socket handler = listenSocket.Accept();

ClientObject clientObject = new ClientObject(handler);

// создаем новый поток для обслуживания нового клиента

Thread clientThread = new Thread(new ThreadStart(clientObject.Process));

clientThread.Start();

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

}

}

}

**Код программы клиента:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Net.Sockets;

using System.Threading;

using System.Net;

namespace std

{

public partial class Form1 : Form

{

string userName;

const int port = 8888;

int flag1 = 1;

const string address = "127.0.0.1";

Socket socket;

public Form1()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void Form1\_Load\_1(object sender, EventArgs e)

{

try

{

IPEndPoint ipPoint = new IPEndPoint(IPAddress.Parse(address), port);

socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

socket.Connect(ipPoint);

}

catch { }

}

private void button1\_Click\_1(object sender, EventArgs e)

{

int flag;

flag = 1;

userName = textBox1.Text;

textBox1.ReadOnly = true;

try

{

if (flag1 == 1)

{

string message = " Hello ," + userName +", you are ours friend now.";

byte[] data = Encoding.Unicode.GetBytes(message);

socket.Send(data);

flag1 = 0;

}

while (flag == 1)

{

Console.Write(userName + ": ");

// ввод сообщения

string message1 = richTextBox1.Text;

string message2 = userName + ": " + message1;

byte[] data1 = Encoding.Unicode.GetBytes(message2);

socket.Send(data1);

richTextBox1.Clear();

flag = 0;

}

}

catch (Exception ex)

{

richTextBox1.Text = ex.Message;

}

}

private void textBox1\_TextChanged\_1(object sender, EventArgs e)

{

}

private void richTextBox1\_TextChanged\_1(object sender, EventArgs e)

{

}

}

}