using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Sockets;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp9

{

class Program

{

private static readonly Socket ClientSocket = new Socket

(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

private const int PORT = 27001;

static void Main()

{

Console.Title = "Client";

ConnectToServer();

RequestLoop();

Exit();

}

private static void ConnectToServer()

{

int attempts = 0;

while (!ClientSocket.Connected)

{

try

{

attempts++;

Console.WriteLine("Connection attempt " + attempts);

// Change IPAddress.Loopback to a remote IP to connect to a remote host.

ClientSocket.Connect(IPAddress.Parse("10.2.11.19"), PORT);

}

catch (SocketException)

{

Console.Clear();

}

}

Console.Clear();

Console.WriteLine("Connected");

}

private static void RequestLoop()

{

Console.WriteLine(@"<Type ""exit"" to properly disconnect client>");

var sender = Task.Run(() =>

{

while (true)

{

SendRequest();

}

});

var receiver = Task.Run(() =>

{

while (true)

{

ReceiveResponse();

}

});

Task.WaitAll(sender, receiver);

}

/// <summary>

/// Close socket and exit program.

/// </summary>

private static void Exit()

{

SendString("exit"); // Tell the server we are exiting

ClientSocket.Shutdown(SocketShutdown.Both);

ClientSocket.Close();

Environment.Exit(0);

}

private static void SendRequest()

{

Console.Write("Send a request: ");

string request = Console.ReadLine();

SendString(request);

if (request.ToLower() == "exit")

{

Exit();

}

}

/// <summary>

/// Sends a string to the server with ASCII encoding.

/// </summary>

private static void SendString(string text)

{

byte[] buffer = Encoding.ASCII.GetBytes(text);

ClientSocket.Send(buffer, 0, buffer.Length, SocketFlags.None);

}

private static void ReceiveResponse()

{

var buffer = new byte[2048];

int received = ClientSocket.Receive(buffer, SocketFlags.None);

if (received == 0) return;

var data = new byte[received];

Array.Copy(buffer, data, received);

string text = Encoding.ASCII.GetString(data);

Console.WriteLine(text);

}

}

}