//Client

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.IO;

using System.Net;

using System.Net.Sockets;

namespace WindowsFormsApp1

{

   public partial class Form1 : Form

   {

       public TcpClient m\_cli;

       public NetworkStream m\_stm;

       public Form1()

       {

           InitializeComponent();

           InitSocket();

       }

       public void InitSocket()

       {

           m\_cli = new TcpClient();

           m\_cli.Connect("localhost", 5002);

           m\_stm = m\_cli.GetStream();

           //byte[] sendbytes = Encoding.ASCII.GetBytes(str);

           //m\_stm.Write(sendbytes, 0, sendbytes.Length);

       }

       public void ReceiveFile()

       {

           int BufferSize = 1024;

           byte[] RecData = new byte[BufferSize];

           int RecBytes;

           string SaveFileName = string.Empty;

           SaveFileDialog DialogSave = new SaveFileDialog();

           DialogSave.Filter = "All files (\*.\*)|\*.\*";

           DialogSave.RestoreDirectory = true;

           DialogSave.Title = "Where do you want to save the file?";

           DialogSave.InitialDirectory = @"C:/";

           if (DialogSave.ShowDialog() == DialogResult.OK)

               SaveFileName = DialogSave.FileName;

           if (SaveFileName != string.Empty)

           {

               int totalrecbytes = 0;

               FileStream Fs = new FileStream

(SaveFileName, FileMode.OpenOrCreate, FileAccess.Write);

               while ((RecBytes = m\_stm.Read

    (RecData, 0, RecData.Length)) > 0)

               {

                   Fs.Write(RecData, 0, RecBytes);

                   totalrecbytes += RecBytes;

               }

               Fs.Close();

               System.Diagnostics.Process.Start(SaveFileName);

           }

       }

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

       {

           string message = textBox1.Text;

           byte[] sendbytes = Encoding.ASCII.GetBytes(message);

           byte[] intLength = BitConverter.GetBytes(sendbytes.Length);

           m\_stm.Write(intLength, 0, intLength.Length);

           m\_stm.Write(sendbytes, 0, sendbytes.Length);

           ReceiveFile();

       }

   }

}

//Server

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Net;

using System.Net.Sockets;

namespace ConsoleApp2

{

   class Program

   {

       static void Main(string[] args)

       {

           IPAddress it = IPAddress.Parse("127.0.0.1");

           TcpListener tclis = new TcpListener(it, 5002);

           Console.WriteLine("Server start");

           tclis.Start();

           TcpClient tccli = tclis.AcceptTcpClient();

           NetworkStream stm = tccli.GetStream();

           byte[] readInt = new byte[256];

           stm.Read(readInt, 0, 256);

           int result = BitConverter.ToInt32(readInt, 0);

           Console.WriteLine(result);

           byte[] readBuf = new byte[result];

           stm.Read(readBuf, 0, result);

           string htmlFile = Encoding.ASCII.GetString(readBuf);

           int BufferSize = 1024;

           Console.WriteLine(htmlFile);

           Console.WriteLine(htmlFile.Length);

           if (htmlFile.Length != 0)

           {

               string file = @htmlFile;

               if (File.Exists(file))

               {

                   Console.WriteLine(file);

               }

               FileStream Fs = new FileStream(file, FileMode.Open, FileAccess.Read);

               int NoOfPackets = Convert.ToInt32

            (Math.Ceiling(Convert.ToDouble(Fs.Length) / Convert.ToDouble(BufferSize)));

               Console.WriteLine(NoOfPackets);

               int TotalLength = (int)Fs.Length, CurrentPacketLength, counter = 0;

               for (int i = 0; i < NoOfPackets; i++)

               {

                   if (TotalLength > BufferSize)

                   {

                       CurrentPacketLength = BufferSize;

                       TotalLength = TotalLength - CurrentPacketLength;

                   }

                   else

                   {

                       CurrentPacketLength = TotalLength;

                   }

                   byte[] SendingBuffer = new byte[CurrentPacketLength];

                   Fs.Read(SendingBuffer, 0, CurrentPacketLength);

                   stm.Write(SendingBuffer, 0, (int)SendingBuffer.Length);

               }

               Fs.Close();

           }

           stm.Close();

       }

   }

}