﻿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;

namespace Client\_Socket

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

System.Net.Sockets.TcpClient cliSocket = new System.Net.Sockets.TcpClient();

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

{

cbActionSelection.Enabled = false;

btnStopClient.Enabled = false;

}

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

{ }

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

{

try

{

lblClientCurrentStatus.Text = "Client Started successfully...";

cliSocket.Connect("127.0.0.1", 8888);

lblClientCurrentStatus.Text = "Client connected to the server successfully..";

cbActionSelection.Enabled = true;

cbActionSelection.SelectedIndex = 0;

btnClient.Enabled = false;

btnStopClient.Enabled = true;

lblDatastatus.Text = "Waiting for data transfer";

}

catch (Exception ex)

{

lblClientCurrentStatus.Text = "Server Not started, please start";

MessageBox.Show("Server not started:"+ex);

}

}

private void cbActionSelection\_SelectedIndexChanged(object sender, EventArgs e)

{

NetworkStream networkStream;

int retry = 0;

if (cbActionSelection.SelectedIndex == 1 && cbActionSelection.Text == "Send")

{

string[] fileContent = null;

lblDatastatus.Text = "Data sending..";

cbActionSelection.Enabled = false;

networkStream = cliSocket.GetStream();

try

{

openFileDialog1.ShowDialog();

lblFileName.Text = "File: " + System.IO.Path.GetFileName(openFileDialog1.FileName);

fileContent = System.IO.File.ReadAllLines(openFileDialog1.FileName.ToString());

for (int i = 0; i < fileContent.Length; i++)

{

if (fileContent[i] == "")

{

string line = "emptyline";

lblActualData.Text = "Writing empty line";

Byte[] sendBytes = frameForm(line, fileContent.Length, i);

networkStream.Write(sendBytes, 0, sendBytes.Length);

}

else

{

lblActualData.Text = fileContent[i].ToString();

Byte[] sendBytes = frameForm(fileContent[i], fileContent.Length, i);

networkStream.Write(sendBytes, 0, sendBytes.Length);

}

System.Threading.Thread.Sleep(10);

byte[] bytesFrom = new byte[10];

networkStream.Read(bytesFrom, 0, 10);

bool check = ackCheck(bytesFrom);

if (check == false)

{

retry = retry + 1;

i = i - 1;

}

if (retry == 3)

{

break;

}

}

if (retry == 3)

{

// error found while sending

lblActualData.Text = "Sending stopped due to error..";

byte[] closeFrame = frameForm("stop", 1, 1);

networkStream.Write(closeFrame, 0, closeFrame.Length);

lblActualData.Text = "File transfer stopped..";

}

else

{

// closing packet sending

lblActualData.Text = "Sending closing packet..";

byte[] closeFrame = frameForm("close", 1, 1);

networkStream.Write(closeFrame, 0, closeFrame.Length);

lblActualData.Text = "File sending finished..";

}

}

catch (Exception ex)

{

MessageBox.Show("File selection failure/file not found, please check..");

cbActionSelection.SelectedIndex = 0;

}

networkStream.Flush();

cbActionSelection.SelectedIndex = 0;

}

else if (cbActionSelection.SelectedIndex == 2 && cbActionSelection.Text == "Receive")

{

lblDatastatus.Text = "Data receiving..";

string data\_stop = "";

cbActionSelection.Enabled = false;

bool loop\_check = true;

System.IO.StreamWriter file = new System.IO.StreamWriter(@"C:\Users\Shreya\Documents\Writehere.txt");

while (loop\_check == true)

{

try

{

networkStream = cliSocket.GetStream();

byte[] bytesFrom = new byte[65536];

networkStream.Read(bytesFrom, 0, (int)cliSocket.ReceiveBufferSize);

Array.Resize(ref bytesFrom, 6 + bytesFrom[3]);

int totalPkts = bytesFrom[1];

int currentPkt = bytesFrom[2];

//if (totalPkts == currentPkt)

//{ break; }

string data = parseData(bytesFrom);

if (data == "close")

{

break;

}

if (data == "stop")

{

data\_stop = "stop";

break;

}

if (data == "emptyline")

{

lblActualData.Text = "new line writing..";

file.WriteLine("\n");

networkStream.Write(ackframeForm("Pass"), 0, ackframeForm("Pass").Length);

}

else if (data != "")

{

lblActualData.Text = data;

//have to write into a file + need to send the ack packet

networkStream.Write(ackframeForm("Pass"), 0, ackframeForm("Pass").Length);

file.WriteLine(data);

}

else

{

networkStream.Write(ackframeForm("Fail"), 0, ackframeForm("Fail").Length);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

file.Close();

cbActionSelection.SelectedIndex = 0;

}

if (data\_stop == "stop")

{

cbActionSelection.SelectedIndex = 0;

lblActualData.Text = "File transfer failed due to checksum error..";

}

else

{

lblActualData.Text = "File received successfully..";

cbActionSelection.SelectedIndex = 0;

}

}

file.Close();

}

cbActionSelection.Enabled = true;

lblDatastatus.Text = "Data transfer finished.";

}

public byte[] frameForm(string lineData, int totalLength, int currentPktNum)

{

byte[] data\_frm = new byte[65536];

try

{

int index = 0;

data\_frm[index++] = 0xaa; //Start of the packet - unique identification

data\_frm[index++] = (byte)totalLength; //number of packets

data\_frm[index++] = (byte)currentPktNum; //current packet number

data\_frm[index++] = (byte)lineData.Length; //current data length

Byte[] dataBytes = Encoding.ASCII.GetBytes(lineData); //data

Buffer.BlockCopy(dataBytes, 0, data\_frm, index, dataBytes.Length); //copying data to original array

data\_frm[dataBytes.Length - 1 + index + 1] = calCheckSum(data\_frm, 1, dataBytes.Length + index); //checksum portion

data\_frm[dataBytes.Length - 1 + index + 2] = 0x99; //end of packet - unique identification

Array.Resize(ref data\_frm, dataBytes.Length + index + 2); //resizing the array to actual size

}

catch (Exception ex)

{

data\_frm = null;

}

return data\_frm;

}

public byte[] ackframeForm(string data)

{

byte[] ack\_frm = new byte[65536];

try

{

int index = 0;

ack\_frm[index++] = 0xaa; //Start of the packet - unique identification

ack\_frm[index++] = 0x01; //number of packets

ack\_frm[index++] = 0x01; //current packet number

ack\_frm[index++] = 0x04; //current data length

Byte[] dataBytes = Encoding.ASCII.GetBytes(data); //data

Buffer.BlockCopy(dataBytes, 0, ack\_frm, index, dataBytes.Length); //copying data to original array

ack\_frm[dataBytes.Length - 1 + index + 1] = calCheckSum(ack\_frm, 1, dataBytes.Length + index); //checksum portion

ack\_frm[dataBytes.Length - 1 + index + 2] = 0x99; //end of packet - unique identification

Array.Resize(ref ack\_frm, dataBytes.Length + index + 2); //resizing the array to actual size

}

catch (Exception ex)

{

ack\_frm = null;

}

return ack\_frm;

}

public string parseData(byte[] receiveData)

{

string actualData = "";

try

{

int totalPckts = receiveData[1]; //retriving totoal number of packets count.

int currentPktNum = receiveData[2]; //current packet number

int dataLength = receiveData[3]; //actual data length

byte chkSum = calCheckSum(receiveData, 1, receiveData.Length - 2); //Re-calculate the checksum

if (chkSum == receiveData[receiveData.Length - 2])

{

byte[] data = new byte[dataLength];

Array.Copy(receiveData, 4, data, 0, dataLength);

actualData = System.Text.Encoding.ASCII.GetString(data);

}

else

{

actualData = "";

MessageBox.Show("Checksum Error Found.");

}

}

catch (Exception ex)

{

actualData = "";

}

return actualData;

}

public byte calCheckSum(byte[] data\_frm, int startindex, int endindex)

{

byte chkSum = 0x0;

try

{

for (int i = startindex; i < endindex; i++)

{

chkSum ^= data\_frm[i];

}

}

catch (Exception ex)

{

chkSum = 0x0;

}

return chkSum;

}

public bool ackCheck(byte[] ack\_pkt)

{

bool flag = false;

try

{

string ack = System.Text.Encoding.ASCII.GetString(ack\_pkt);

if (ack.Contains("Pass") || ack.Contains("PASS") || ack.Contains("pass"))

{

flag = true;

}

else { flag = false; }

}

catch (Exception ex)

{

flag = false;

}

return flag;

}

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

{

cliSocket.Close();

lblClientCurrentStatus.Text = "Client disconnected successfully..";

btnClient.Enabled = true;

btnStopClient.Enabled = false;

}

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

{

}

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

{

}

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

{

}

}

}