```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net;
using System.Net.Sockets;
using System.IO;

namespace Client
{
    public partial class WindowClient : Form
    {
        private Socket client;
        private const int size =1024*1024;
        private byte[] data = new byte[size];
        public WindowClient()
        {
            InitializeComponent();
        }

        private void btConnect_Click(object sender, EventArgs e)
        {
            if (client == null)
            {
                Socket newsock = new
Socket(AddressFamily.InterNetwork,
                           SocketType.Stream, ProtocolType.Tcp);
                if (txtIP.Text == "" || txtPort.Text == "")
                    MessageBox.Show("Input your address first!");
                else
                {

                           IPEndPoint iep = new
IPEndPoint(IPAddress.Parse(txtIP.Text),
Convert.ToInt32(txtPort.Text));
                           newsock.BeginConnect(iep, new
AsyncCallback(Connected), newsock); //Bắt đầu việc kết nối từ
server
```

```
                }
            }
            else
            {
                MessageBox.Show("You are on connection");
            }
        }
        //Hàm kết nối client với server
        void Connected(IAsyncResult iar)
        {
            try
            {
                client = (Socket)iar.AsyncState;
                client.EndConnect(iar); //Kết thúc việc kết nối
                txtStatus.Text = "Connected to: " +
client.RemoteEndPoint.ToString();
                client.BeginReceive(data, 0, size,
SocketFlags.None,
                        new AsyncCallback(ReceiveData), client);
//Bắt đầu nhận dữ liệu từ socket
            }
            catch (SocketException se)
            {
                string str;
                str = "\nConnection failed, is the server
running?\n" + se.Message;
                MessageBox.Show(str);
            }
        }
        //Hàm nhận dữ liệu được gửi qua từ server
        void ReceiveData(IAsyncResult iar)
        {
            Socket remote = (Socket)iar.AsyncState;
            int recv = remote.EndReceive(iar); //Kết thúc việc
nhận dữ liệu
            string stringData = Encoding.ASCII.GetString(data, 0,
recv);

            txtShow.Text = stringData; //Hiển thị dữ liệu nhận
được
        }
```

```
        private void btDisconn_Click(object sender, EventArgs e)
        {

            if (client != null)
            {
                client.Close();
                client = null;
                txtStatus.Text = "no connection";
            }
            else
            {
                string noti = "Connect first!";
                MessageBox.Show(noti); //Thông báo khi client
được kết nối
            }
        }

        private void btSendStr_Click(object sender, EventArgs e)
        {
            if (client != null)
            {
                if (txtDir.Text != "")
                {
                    byte[] message =
Encoding.ASCII.GetBytes(txtDir.Text);
                    txtDir.Clear();
                    client.BeginSend(message, 0, message.Length,
SocketFlags.None,
                                new AsyncCallback(SendData), client);
//Bắt đầu gửi dữ liệu từ socket
                }

                else
                {
                    string noti = "Input your path first.";
                    MessageBox.Show(noti);
                }
            }
            else
            {
                string noti = "Connect first!";
                MessageBox.Show(noti);
```

```
            }
        }

        //Hàm gửi dữ liệu đi
        void SendData(IAsyncResult iar)
        {
            try
            {
                txtShow.Clear();
                Socket remote = (Socket)iar.AsyncState;
                int sent = remote.EndSend(iar); //EndSend()
                remote.BeginReceive(data, 0, size,
SocketFlags.None,
                        new AsyncCallback(ReceiveData), remote);
//Bắt đầu nhận dữ liệu từ socket
            }
            catch (SocketException se)
            {
                MessageBox.Show(se.ToString());
            }
        }

        private void btClose_Click(object sender, EventArgs e)
        {
            if (client == null)
            {
                Close();
            }
            else
            {
                string noti = "Disconnect first.";
                MessageBox.Show(noti);
            }
        }

}
```