

```

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;
using System.Collections;

namespace Server
{
    public partial class WindowServer : Form
    {
        private byte[] data = new byte[1024*1024];
        private int size = 1024*1024;
        private Socket server;
        private int count=0;

        public WindowServer()
        {
            InitializeComponent();
        }

        private void WindowServer_Load(object sender, EventArgs
e)
        {

        }

        private void cbChoose_CheckedChanged(object sender,
EventArgs e)
        {
            if (cbChoose.CheckState == CheckState.Checked)
            {
                txtIP.Enabled = false;
                txtPort.Enabled = false;
            }
            else

```

```

        {
            txtIP.Enabled = true;
            txtPort.Enabled = true;
        }
    }

    //Sự kiện nhấn nút Start Listening
    private void btConn_Click(object sender, EventArgs e)
    {
        server = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
        if (cbChoose.Checked)
        {
            IPEndPoint iep = new
IPEndPoint(IPAddress.Any, 9050);
            server.Bind(iep);
        }
        else
        {
            if (txtIP.Text == "" || txtPort.Text == "")
                MessageBox.Show("Input your address
first!");
            else
            {
                IPAddress ip =
IPAddress.Parse(txtIP.Text);
                IPEndPoint iep = new IPEndPoint(ip,
Convert.ToInt32(txtPort.Text));
                server.Bind(iep);
            }
        }

        txtStatus.Text = "Waiting for client...";
        server.Listen(5);
        server.BeginAccept(new AsyncCallback(AcceptConn),
server); //Bắt đầu việc chấp nhận kết nối từ client
    }
    void AcceptConn(IAsyncResult iar)
    {
        lbCount.Text = "";
        Socket oldserver = (Socket)iar.AsyncState;
    }

```

```

        Socket client = oldserver.EndAccept(iar); //Kết thúc
việc kết nối
        count++;
        server.BeginAccept(new AsyncCallback(AcceptConn),
oldserver); //Chấp nhận kết nối nếu có thêm client yêu cầu
        lbCount.Text = Convert.ToString(count);
        txtStatus.Text = "Server is connecting to client...";
        string stringData = "Welcome to my server";
        byte[] message1 =
Encoding.ASCII.GetBytes(stringData);
        client.BeginSend(message1, 0, message1.Length,
SocketFlags.None,
            new AsyncCallback(SendData), client); //Gửi
thông điệp chào mừng tới client
    }
    //Hàm gửi dữ liệu
    void SendData(IAsyncResult iar)
    {
        Socket client = (Socket)iar.AsyncState;
        int sent = client.EndSend(iar); //Kết thúc việc gửi
dữ liệu
        client.BeginReceive(data, 0, size, SocketFlags.None,
            new AsyncCallback(ReceiveData), client); //Bắt
đầu nhận dữ liệu từ socket
    }

    //Hàm nhận dữ liệu từ client gửi đến
    void ReceiveData(IAsyncResult iar)
    {
        Socket client = (Socket)iar.AsyncState;

        int recv = client.EndReceive(iar); //Kết thúc nhận
dữ liệu
        if (recv == 0)
        {
            client.Close();
            count--;
            if (count <= 0)
            {
                txtStatus.Text = "Waiting for client...";
            }
        }
    }

```

```

        lbCount.Text= Convert.ToString(count);
        server.BeginAccept(new
AsyncCallback(AcceptConn), server); //Chấp nhận kết nối mới đến
server

        return;
    }
    string receivedData =
(Encoding.ASCII.GetString(data, 0, recv));
    string recvData = receivedData.Replace(" ", "");
    if (File.Exists(recvData) && recvData != "")
    {

        StreamReader SRD = new StreamReader(recvData);
        string mess = SRD.ReadToEnd(); //Đọc tất cả
nội dung trong file mà client yêu cầu
        byte[] message2 =
Encoding.ASCII.GetBytes(mess);
        client.BeginSend(message2, 0,
message2.Length, SocketFlags.None, new AsyncCallback(SendData),
client); //Bắt đầu việc gửi nội dung văn bản sang client
    }
    else
    {
        string mess = "Your path was wrong!\r\n Please
re-send your path.";
        byte[] message2 =
Encoding.ASCII.GetBytes(mess);
        client.BeginSend(message2, 0, message2.Length,
SocketFlags.None,
            new AsyncCallback(SendData), client);
//Gửi thông điệp yêu cầu client nhập lại đường dẫn
    }
}

private void btClose_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```