

```

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.Collections;
using System.Threading;
using System.IO;

namespace ServerTCP_
{
    public partial class FormServer : Form
    {
        public FormServer()
        {
            InitializeComponent();
        }

        delegate void ShowMessage(String Message_);

        public void Show(string Message_)
        {
            if (lstReceive.InvokeRequired)
            {
                ShowMessage message = new ShowMessage(Show);
                lstReceive.Invoke(message, new object[] { Message_ });
                return;
            }
            lstReceive.Items.Add(Message_);
        }

        public void ShowIP(string Message_)
        {
            if (txtIP.InvokeRequired)
            {
                ShowMessage message = new ShowMessage(ShowIP);
                txtIP.Invoke(message, new object[] { Message_ });
                return;
            }
            txtIP.Text = Message_;
        }

        string returnString = "";
        char[] s;

        TcpListener listener;
        TcpClient client;
        Thread listenerThread;

        bool isStart = false;
    }
}

```

```

public void ListenMessage()
{
    try
    {
        listener = new TcpListener(IPAddress.Any, int.Parse(txtPort.Text));
        listener.Start();
        isStart = true;
        listenerThread = new Thread(new ThreadStart(WaitingConnect));
        listenerThread.Start();
    }
    catch (Exception exc)
    {
        MessageBox.Show("Không thể khởi động máy chủ ! \n" + exc.ToString());
    }
}

public void WaitingConnect()
{
    while (isStart)
    {
        IPEndPoint remote = new IPEndPoint(IPAddress.Any, 0);
        client = listener.AcceptTcpClient();
        StreamReader reader = new StreamReader(client.GetStream());
        string data = reader.ReadLine();
        if (data.Trim().Length > 0)
        {
            Show("Nhận thông điệp từ -> " + remote.Address.ToString() + " : " +
data);

            StreamWriter writer = new StreamWriter(client.GetStream());

            writer.WriteLine(data);
            writer.Flush();
        }
    }
}

private void btnStart_Click(object sender, EventArgs e)
{
    lstReceive.Items.Clear();
    new Thread(new ThreadStart(ListenMessage)).Start();
    lstReceive.Items.Add("Đang lắng nghe ... ");
}

private void lstReceive_SelectedIndexChanged(object sender, EventArgs e)
{
    if (lstReceive.SelectedItem != null)
        txtResult.Text = lstReceive.SelectedItem.ToString();
}

private void Stop_Click(object sender, EventArgs e)
{
    if (listener != null)
    {
        isStart = false;
        listener.Stop();
    }
}

```

```
}  
}  
}  
}
```

Client

IP Port

Receive Message

Server

Port

Receive Message