

BÁO CÁO THỰC HÀNH

Môn học: Lập trình mạng căn bản

Kỳ báo cáo: Buổi 03 (Session 03)

Tên chủ đề: <LAB3 - Working with Sockets in C#>

GVHD: Nghi Hoàng Khoa

Ngày báo cáo: 09/05/2022

Nhóm: 08

1. THÔNG TIN CHUNG:

Lớp: NT106.M21.ATCL

STT	Họ và tên	MSSV	Email
1	Trương Đình Trọng Thanh	20520766	20520766@gm.uit.edu.vn
2	Lã Ngọc Ánh	20521065	20521065@gm.uit.edu.vn
3	Trần Thúy Anh	20521085	20521085@gm.uit.edu.vn
4	Nguyễn Tú Ngọc	20521655	20521655@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	Kịch bản 01	100%
2	Kịch bản 02	100%
3	Kịch bản 03	100%
4	Kịch bản 04	100%

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

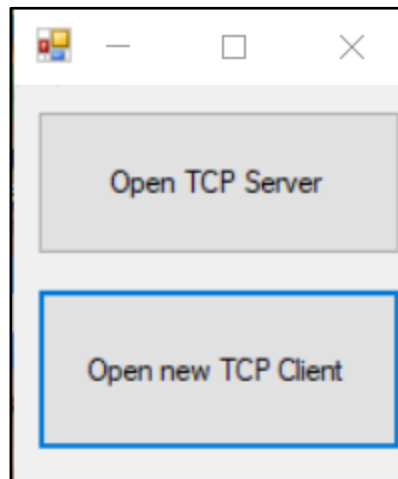
1. Kịch bản 01 – Bài 1
2. Kịch bản 02 – Bài 2
3. Kịch bản 03 – Bài 3

(3 Bài trên đã báo cáo trên lớp)

4. Kịch bản 04 – Bài 4

- Tài nguyên:
 - + TCPClient : 2 label, 2 richtextbox, 1 textbox, 1 button
 - + TCPServer : 1 richtextbox, 1 button
- Mô tả/mục tiêu: **Viết chương trình chatroom /gửi và nhận dữ liệu sử dụng TCP Client và TCP Listener. Mỗi người dùng sẽ có một tài khoản, khi một người dùng gửi tin nhắn thì tất cả mọi người còn lại đều sẽ nhận được tin nhắn đó**
- Các bước thực hiện/ Phương pháp thực hiện:

B1: Chúng tôi sẽ tạo 1 bảng tùy chọn để tạo Client - Server:

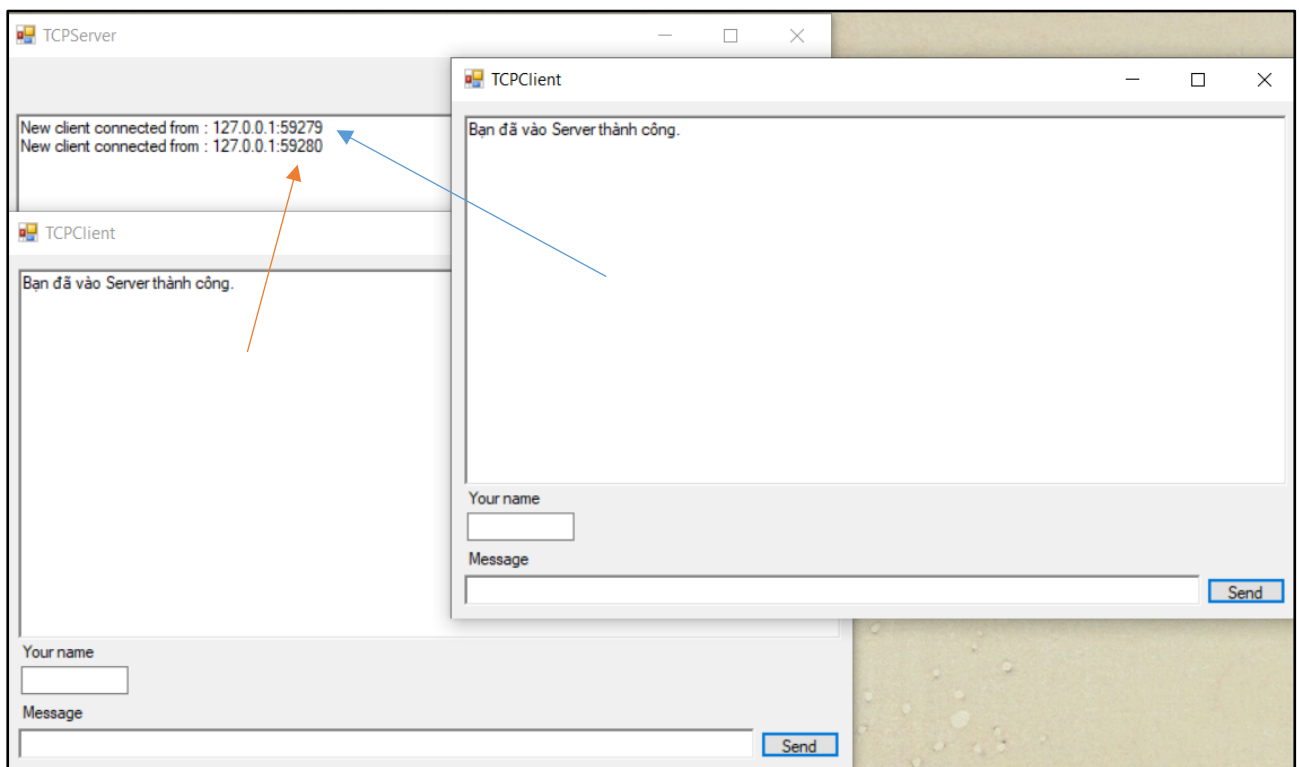


B2: Đầu tiên, ấn vào “Open TCP Server” để mở Server, sau đó ấn Nút Listen để chuẩn bị kết nối với các Client:



Sau đó ấn “Open new TCP Client” để mở Client

Nếu kết nối thành công, cả 2 Server và Client đều sẽ hiển thị thông báo sau:



Để có thể kết nối 1 Server tới n Client thì ta phải qua các hàm sau:

+ Địa chỉ bên Server:

```
}
public List<Socket> clientList;
IPEndPoint ipepServer = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 8080);
Socket listenerSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
```

+ Hàm để kết nối các Client (nút Listen):

```
private void button1_Click(object sender, EventArgs e) // Listener
{
    listenerSocket.Bind(ipepServer);
    listenerSocket.Listen(-1);
    CheckForIllegalCrossThreadCalls = false;
    for (int i = 0; i <= 100; i++)
    {
        Thread serverThread = new Thread(new ThreadStart(Listener));
        serverThread.Start();
    }
    button1.Enabled = false;
}
```

Với các hàm phụ trợ sau:

```
void Listener()
{
    int bytesReceived = 1;
    byte[] recv = new byte[1];
    clientList = new List<Socket>();
    Socket clientSocket = listenerSocket.Accept();
    clientList.Add(clientSocket);
    var endPoint = (IPEndPoint)clientSocket.RemoteEndPoint;
    string newClient = "New client connected from : " + endPoint + '\n';
    richTextBox1.Text += newClient;
    try
    {
        while (clientSocket.Connected && bytesReceived == 1)
        {
            string text = "";
            do
            {
                bytesReceived = clientSocket.Receive(recv);
                text += Encoding.UTF8.GetString(recv);
            } while (text[text.Length - 1] != '\n' && bytesReceived == 1);
            richTextBox1.Text += text;
            sendMess(text, clientSocket);
        }
    }
}
```

```

        catch
        {
            clientList.Remove(clientSocket);
            listenerSocket.Close();
            richTextBox1.Text += "A client" + endPoint + "disconnected";
        }
    }
}

```

```

1 reference
void sendMess(string s, Socket client)
{
    foreach (Socket item in clientList)
    {
        if (item != null && item != client)
            item.Send(Encoding.UTF8.GetBytes(s));
    }
}

```

+ Cách Client kết nối với Server:

```

TcpClient tcpClient = new TcpClient();
public Socket listenerSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
public IPEndPoint ipServer = new IPEndPoint(IPAddress.Any, 8080);
NetworkStream network = default(NetworkStream);
string readData = null;

```

```

private void TCPClient_Load_1(object sender, EventArgs e)
{
    IPAddress ipAddress = IPAddress.Parse("127.0.0.1");
    IPEndPoint ipEndPoint = new IPEndPoint(ipAddress, 8080);
    try
    {
        tcpClient.Connect(ipEndPoint);
        richTextBox1.Text += "Bạn đã vào Server thành công." + '\n';
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
        this.Close();
        return;
    }
    Thread threading = new Thread(GetMessenger);
    threading.IsBackground = true;
    threading.Start();
}

```

+ Hàm để gửi tin nhắn (Nút Send):

```
private void button1_Click(object sender, EventArgs e) //Send
{
    if (richTextBox2.Text != string.Empty)
    {
        if (textBox1.Text != string.Empty)
        {
            string mess = textBox1.Text + " : " + richTextBox2.Text + '\n';
            Byte[] data = Encoding.UTF8.GetBytes(mess);
            network.Write(data, 0, data.Length);
            richTextBox1.Text += mess;
            richTextBox2.Text = string.Empty;
        }
        else
        {
            MessageBox.Show("Hãy nhập Tên người dùng !", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }
}
```

Và các hàm phụ trợ sau:

```
private void messenger()
{
    if (this.InvokeRequired)
    {
        this.Invoke(new MethodInvoker(messenger));
    }
    else
    {
        richTextBox1.Text += readData;
    }
}
```

```
private void GetMessage()
{
    string returnData;
    try
    {
        while (true)
        {
            network = tcpClient.GetStream();
            var buffSize = tcpClient.ReceiveBufferSize;
            byte[] data = new byte[buffSize];
            network.Read(data, 0, buffSize);
            returnData = Encoding.UTF8.GetString(data);
            readData = returnData;
            messenger();
        }
    }
    catch
    {
        tcpClient.Close();
    }
}
```

B4: Trò chuyện với nhau và có phân biệt tên các khoản với nhau

