

LAB 6

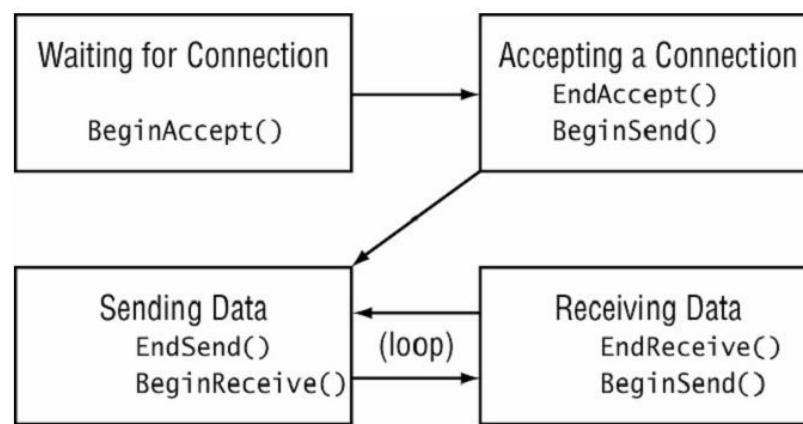
I. Lý thuyết: Lập trình socket bất đồng bộ

II. Bài tập:

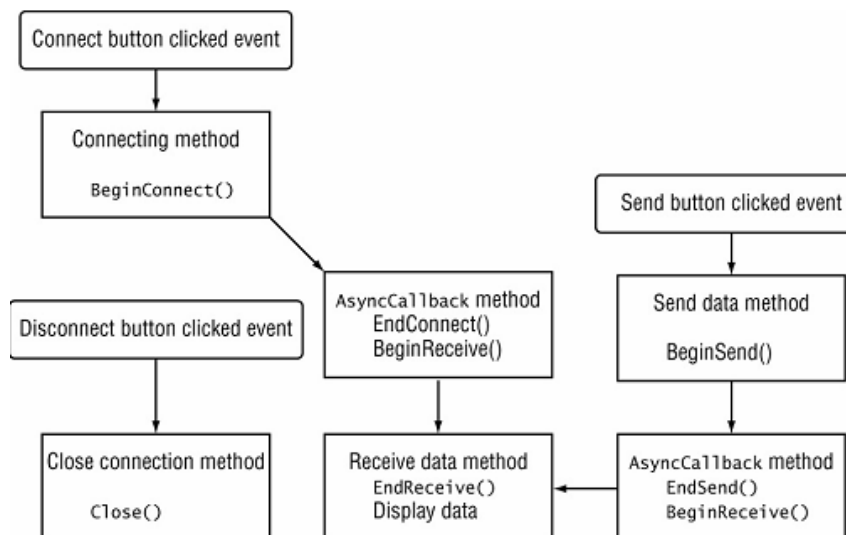
1) Viết chương trình chat Server-Client bằng Winforms sử dụng socket bất đồng bộ.

Mô tả chương trình: Client kết nối và gửi dữ liệu đến server, server nhận và gửi lại cho client.

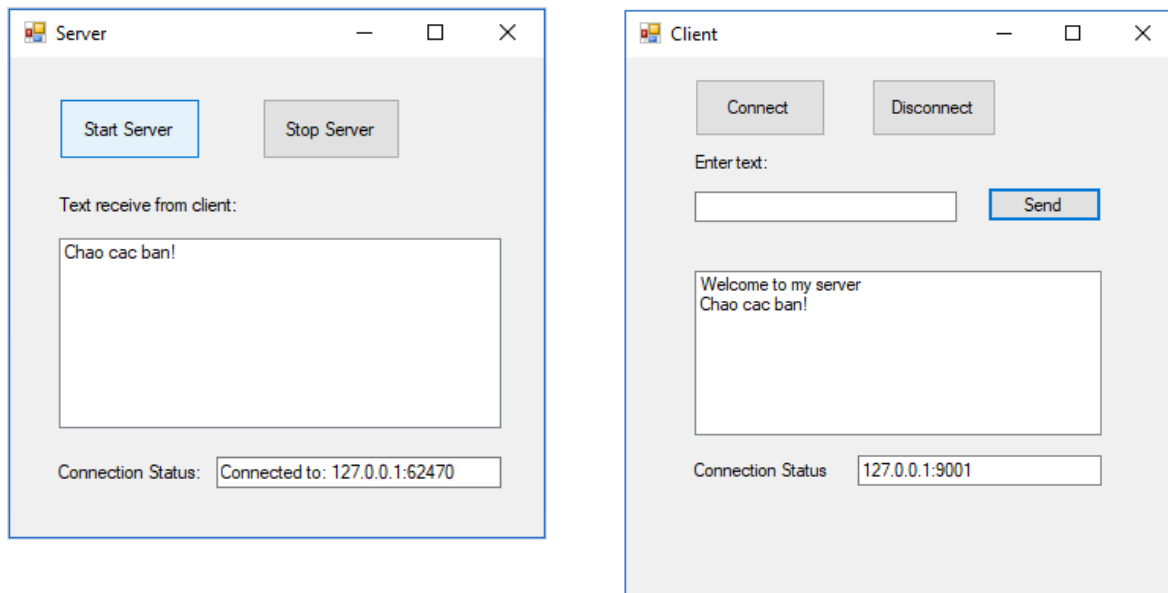
- Sơ đồ xử lý phía server:



- Sơ đồ xử lý phía client:



Giao diện của chương trình



Hướng dẫn:

Các chức năng phía Server:

- Nút Start Server: bắt đầu lắng nghe, chờ client kết nối
- Nút Stop Server: dừng socket phía server
- Text receive (Listbox): nhận và hiển thị thông điệp nhận được từ client
- Connection Status (Textbox): hiển thị thông tin client kết nối

```
//Phương thức thực thi của sự kiện nhấn nút Start Server
private void btnStart_Click(object sender, EventArgs e)
{
    server = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
    ProtocolType.Tcp);
    IPEndPoint iep = new IPEndPoint(IPAddress.Any, 9001);
    server.Bind(iep);
    server.Listen(5);
    server.BeginAccept(new AsyncCallback(AcceptCallback), server);
    //Cập nhật Connection Status
    ...
}

//Phương thức callback khi kết thúc việc nhận kết nối
void AcceptCallback(IAsyncResult ar)
{
    ...
    //Bắt đầu gửi câu chào client
    ...
}

//Phương thức callback khi kết thúc việc gửi dữ liệu
void SendCallback(IAsyncResult ar)
{
    ...
    //Bắt đầu nhận dữ liệu từ client gửi đến
    ...
}
```

```
//Phương thức callback khi kết thúc việc nhận dữ liệu
void ReceiveCallback(IAsyncResult ar)
{
    ...
    //Xuất dữ liệu nhận được lên listbox
    ...
    //Bắt đầu gửi dữ liệu lại cho client
    ...
}
```

Các chức năng phía Slient:

- Nút Connect: Tạo socket và kết nối đến server
- Nút Disconnect: Ngắt kết nối với server
- Nút Send: Gửi thông điệp trong textbox cho server
- Listbox nhận thông điệp phản hồi từ server
- Connection Status (Textbox): hiển thị thông tin của server

```
//Phương thức thực thi của sự kiện nhấn nút Connect
private void btnConn_Click(object sender, EventArgs e)
{
    UpdateStatus("Connecting...");
    Socket server = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
    ProtocolType.Tcp);
    IPEndPoint ipep = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 9001);
    server.BeginConnect(ipep, new AsyncCallback(ConnectCallback), server);
}
//Phương thức callback khi kết thúc việc kết nối đến server
void ConnectCallback(IAsyncResult ar)
{
    ...
    //Cập nhật connection status
    ...
    //Bắt đầu nhận câu chào của server
    ...
}

//Phương thức callback khi kết thúc việc nhận dữ liệu
void ReceiveCallback(IAsyncResult ar)
{
    ...
    //Xuất câu chào của server lên listbox
    ...
}

//Phương thức thực thi của sự kiện nhấn nút Send
private void btnSend_Click(object sender, EventArgs e)
{
    ...
    //Bắt đầu gửi dữ liệu trong textbox đến server
    ...
}

//Phương thức callback khi kết thúc việc gửi dữ liệu
void SendCallback(IAsyncResult ar)
{
    ...
    //Bắt đầu nhận lại dữ liệu từ server
    ...
}
```

Chú ý: Nếu thực thi chương trình gặp lỗi “*cross-thread operation not valid*” khi truy cập các control của form thì có thể giải quyết bằng 2 cách

- Đặt `Form.CheckForIllegalCrossThreadCalls = false;`
- Sử dụng delegate và phương thức Invoke

2) Cải tiến chương trình

- Server cho phép thiết đặt port number lắng nghe
- Client cho phép nhập địa chỉ IP và port của server

3) Cải tiến chương trình để cho phép nhiều client kết nối cùng lúc

- Server hiển thị danh sách (địa chỉ IP và port) của client đã kết nối
- Khi client gửi thông điệp, server hiển thị và chỉ gửi lại cho client đã gửi.