

# MÔN: LẬP TRÌNH MẠNG

## Bài thực hành số 6:

### LẬP TRÌNH ỨNG DỤNG CLIENT – SERVER

### BẮT ĐỒNG BỘ

#### I. Mục tiêu:

- Nắm vững cách sử dụng các phương thức bắt đồng bộ của socket để tăng tính linh hoạt cho chương trình client – server.
- Viết được chương trình theo mô hình Client – Server.

#### II. Bài tập mẫu:

Chương trình server xử lý nhiều client cùng lúc.

```
class SelectTcpSrvr
{
    public static void Main()
    {
        ArrayList sockList = new ArrayList(2);
        ArrayList copyList = new ArrayList(2);
        Socket main = new Socket(AddressFamily.InterNetwork,
                                SocketType.Stream, ProtocolType.Tcp);
        IPEndPoint iep = new IPEndPoint(IPAddress.Any, 9050);
        byte[] data = new byte[1024];
        string stringData;
        int recv;
        main.Bind(iep);
        main.Listen(2);
        Console.WriteLine("Waiting for 2 clients...");
        Socket client1 = main.Accept();
        IPEndPoint iep1 = (IPEndPoint)client1.RemoteEndPoint;
        client1.Send(Encoding.ASCII.GetBytes("Welcome to my server"));
        Console.WriteLine("Connected to {0}", iep1.ToString());
        sockList.Add(client1);
        Console.WriteLine("Waiting for 1 more client...");
        Socket client2 = main.Accept();
        IPEndPoint iep2 = (IPEndPoint)client2.RemoteEndPoint;
        client2.Send(Encoding.ASCII.GetBytes("Welcome to my server"));
        Console.WriteLine("Connected to {0}", iep2.ToString());
        sockList.Add(client2);
        main.Close();
        while(true)
        {
```

```

copyList = new ArrayList(sockList);
Console.WriteLine("Monitoring {0} sockets...", copyList.Count);
Socket.Select(copyList, null, null, 10000000);
foreach(Socket client in copyList)
{
    data = new byte[1024];
    recv = client.Receive(data);
    stringData = Encoding.ASCII.GetString(data, 0, recv);
    Console.WriteLine("Received: {0}", stringData);
    if (recv == 0)
    {
        iep = (IPEndPoint)client.RemoteEndPoint;
        Console.WriteLine("Client {0} disconnected.", iep.ToString());
        client.Close();
        sockList.Remove(client);
        if (sockList.Count == 0)
        {
            Console.WriteLine("Last client disconnected, bye");
            return;
        }
    }
    else
        client.Send(data, recv, SocketFlags.None);
}
}
}
}

```

### Chương trình client.

```

class SelectTcpClient
{
    public static void Main()
    {
        Socket sock = new Socket(AddressFamily.InterNetwork,
            SocketType.Stream, ProtocolType.Tcp);
        IPEndPoint iep = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 9050);
        byte[] data = new byte[1024];
        string stringData;
        int recv;
        sock.Connect(iep);
        Console.WriteLine("Connected to server");
        recv = sock.Receive(data);
        stringData = Encoding.ASCII.GetString(data, 0, recv);
        Console.WriteLine("Received: {0}", stringData);
        while(true)
        {
            stringData = Console.ReadLine();
            if (stringData == "exit")
                break;
        }
    }
}

```

```

        data = Encoding.ASCII.GetBytes(stringData);
        sock.Send(data, data.Length, SocketFlags.None);
        data = new byte[1024];
        recv = sock.Receive(data);
        stringData = Encoding.ASCII.GetString(data, 0, recv);
        Console.WriteLine("Received: {0}", stringData);
    }
    sock.Close();
}
}

```

### III. Bài tập phải làm:

Bài 1. Viết chương trình:

- a. Server chờ nhận kết nối từ nhiều client. Server nhận giá trị client gửi tới và tính tổng. Sau đó gửi trả kết quả về cho client.
- b. Client kết nối tới server. Sau đó gửi một mảng số nguyên cho server. Nhận kết quả từ server và hiển thị.

Bài 2. Viết chương trình:

- a. Server chờ nhiều client kết nối tới. nhận username và password của người dùng. Sau đó xử lý đăng nhập. Trả kết quả đăng nhập thành công hay thất bại cho client.
- b. Client kết nối tới server. Sau đó gửi username và password cho server. Nhận kết quả xử lý đăng nhập từ server.