HỌC PHẦN: **HỆ THỐNG PHÂN TÁN**
BÀI: **LAB_02**
THỜI GIAN: **4 TIẾT**

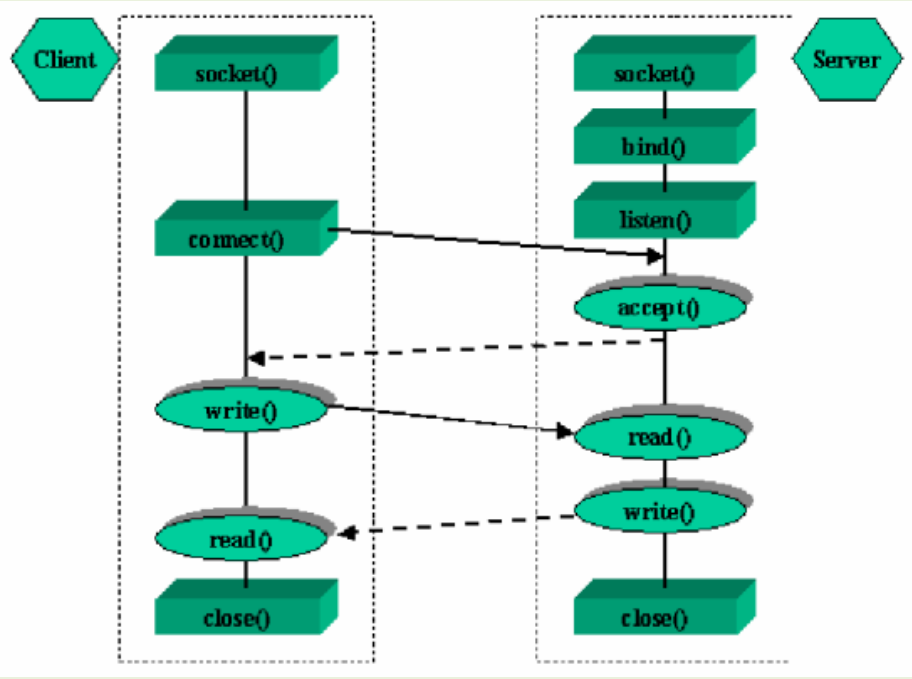
Nội dung

Giao tiếp giữa các tiến trình sử dụng cơ chế giao tiếp ***Socket***

Ngôn ngữ: Java

Hướng dẫn và yêu cầu

1) **Khái niệm Socket:**
   - Là một giao diện lập trình ứng dụng mạng.
   - Thông qua giao diện này chúng ta có thể lập trình điều khiển sự giao tiếp giữa 2 tiến trình đặt trên một máy hoặc giữa hai hay nhiều máy.
   - Socket là sự trừu tượng hóa ở mức cao, có thể tưởng tượng nó như thiết bị truyền thông 2 chiều gửi – nhận dữ liệu giữa hai tiến trình trên một máy hoặc hai máy khác nhau.
2) **Đặc điểm của Socket**
   - có một đường kết nối ảo giữa hai tiến trình.
   - Một trong hai tiến trình phải đợi tiến trình kia yêu cầu kết nối.
   - Có thể được sử dụng để liên lạc theo mô hình client – server.
   - Trong mô hình client / server thì server lắng nghe và chấp nhận một yêu cầu kết nối.
   - Các gói dữ liệu gửi đi theo tuần tự.
3) **Số hiệu cổng của Socket**
   - để có thể thực hiện cuộc giao tiếp một trong hai quá trình phải công bố số hiệu cổng của socket mà mình sử dụng.
   - mỗi cổng giao tiếp thể hiện một địa chỉ xác định trong hệ thống. Khi tiến trình được gán một số hiệu cổng, nó có thể nhận dữ liệu gửi đến cổng này từ các tiến trình khác.
   - Tiến trình còn lại cũng yêu cầu tạo ra một socket.
4) **Thư viện sử dụng:**
   - using System.Net;
   - using System.Net.Sockets;
   - using System.IO;
5) **Mô hình giao tiếp**



6) **Ví dụ**: tạo ra hai project, project 1 dùng để tạo tiến trình server, project 2 dùng để tạo tiến trình client.

project 1, tiến trình server có lớp sau:

```
class Server
 {
    static void Main(string[] args)
    {

        IPEndPoint iep = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 2010);
          Socket server = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
          ProtocolType.Tcp);
        server.Bind(iep);
        server.Listen(10);
        Console.WriteLine("Cho ket noi tu client");
        Socket client = server.Accept();
        //**********************************
        NetworkStream ns = new NetworkStream(client);
        StreamReader sr = new StreamReader(ns);
        StreamWriter sw = new StreamWriter(ns);
        Console.WriteLine("Chap nhan ket noi tu:{0}", client.RemoteEndPoint.ToString());
        string s = "Chao ban den voi Server";
        sw.WriteLine(s);
        sw.Flush();

        while (true)
        {
            s = sr.ReadLine();
            Console.WriteLine("Client gui len:{0}",s);
            //Neu chuoi nhan duoc la Thoat thi thoat
            if (s.ToUpper().Equals("THOAT")) break;
            //Gui tra lai cho client chuoi s
            s = s.ToUpper();
            sw.WriteLine(s);
            sw.Flush();

        }
        client.Shutdown(SocketShutdown.Both);
        client.Close();
        server.Close();


    }
 }
```

project 2, tiến trình Client có lớp sau:

```
class Client
 {
    static void Main(string[] args)
    {
        IPEndPoint iep = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 2010);
          Socket client = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
          ProtocolType.Tcp);
        client.Connect(iep);

        NetworkStream ns = new NetworkStream(client);
        StreamReader sr = new StreamReader(ns);
        StreamWriter sw = new StreamWriter(ns);
        string s = sr.ReadLine();
        Console.WriteLine("Server gui:{0}", s);
        string input;
        while (true)
        {
            input = Console.ReadLine();
            sw.WriteLine(input);
```

```
            sw.Flush();
            if (input.ToUpper().Equals("THOAT")) break;
            s = sr.ReadLine();
            Console.WriteLine("Server gui:{0}", s);

        }
        client.Disconnect(true);
        client.Close();

    }
}
```

7) Giải thích cụ thể từng đoạn code trong ví dụ ở mục 6.

8) ***Cách chạy chương trình:***
Chạy trên một máy: chạy tiến trình Server trước, sau đó chạy tiến trình Client.
Chạy trên hai máy khác nhau: chép tiến trình Server lên máy A, để tiến trình Client trên máy B. sử dụng lệnh run -> cmd -> ipconfig trên máy A để biết địa chỉ IP của máy A. Dùng địa chỉ IP này thay cho địa chỉ "127.0.0.1" trong cả hai tiến trình. Cuối cùng, chạy chương trình server trước rồi chạy client.

9) Hãy cải tiến chương trình ở ví dụ trên để có được một chương trình "***Chat"*** giữa 2 máy (2 người)

10) Hãy cải tiến chương trình ở mục 9 để có được chương trình ***"Chat"*** giữa nhiều máy (sử dụng cơ chế đa luồng trong lớp System.Thread và `using System.Diagnostics` đã học ở các bài Lab trước.