**LAB 2**

1. Mục đích ................................................................................................................................................ 2
2. Yêu cầu ................................................................................................................................................. 2
3. Hướng dẫn ........................................................................................................................................... 2
   1. Tạo một chương trình server đơn giản ......................................................................................... 2
      1. Hướng dẫn lập trình ............................................................................................................... 2
      2. Test chương trình: .................................................................................................................. 3
      3. Trả lời câu hỏi: ....................................................................................................................... 4
   2. Tạo chương trình Server đơn giản, khi Client kết nối vào thì gởi câu chào cho client ................ 5
      1. Hướng dẫn lập trình ............................................................................................................... 5
      2. Test chương trình ................................................................................................................... 5
      3. Trả lời câu hỏi: ....................................................................................................................... 7
   3. Xây dựng chương trình client đơn giản ........................................................................................ 7
      1. Hướng dẫn lập trình ............................................................................................................... 7
      2. Test chương trình ................................................................................................................... 8
      3. Trả lời câu hỏi: ....................................................................................................................... 9
   4. Cải tiến chương trình client ........................................................................................................ 11
      1. Hướng dẫn lập trình ............................................................................................................. 11
      2. Test chương trình ................................................................................................................. 11
   5. Thêm khả năng gởi dữ liệu lên server và nhận lại dữ liệu từ server gởi về ............................... 12
      1. Hướng dẫn lập trình ............................................................................................................. 12
   6. Thêm khả năng nhận dữ liệu từ client gởi lên và gởi trả dữ liệu ngược lại client ...................... 12
      1. Hướng dẫn lập trình ............................................................................................................. 12
      2. Test chương trình ................................................................................................................. 12
      3. Trả lời câu hỏi: ..................................................................................................................... 14
4. Bài tập ............................................................................................................................................... 15

# I. Mục đích

Ban đầu làm quen với lập trình Socket

Hiểu được cách hoạt động của bộ đệm hệ thống và bộ đệm chương trình

Xây dựng chương trình client-server đơn giản

**II. Yêu cầu**

Xây dựng các chương trình client-server từ mức đơn giản nhất rồi phức tạp dần

# III. Hướng dẫn

## III.1. Tạo một chương trình server đơn giản

### III.1.1. Hướng dẫn lập trình

Để lập trình socket ta sử dụng 2 namespace:

using System.Net;

using System.Net.Sockets;

Tạo Server EndPoint, EndPoint này sẽ tham chiếu đến địa chỉ IP và Port của Server:

IPEndPoint serverEndPoint = new IPEndPoint(IPAddress.Any, 5000);

Tạo Server Socket, Socket này sẽ được kết nối với Server EndPoint:

Socket serverSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream,

ProtocolType.Tcp);

Lắng nghe kết nối trên Server Socket, khi có client kết nối tới sẽ trả về Socket để trao đổi dữ liệu với client:

serverSocket.Listen(10);

Socket clientSocket = serverSocket.Accept();

Khi client kết nối tới trên server sẽ xuất thông tin của client kết nối tới (địa chỉ IP, Port)

EndPoint clientEndPoint = clientSocket.RemoteEndPoint; Console.WriteLine(clientEndPoint.ToString());

**III.1.2. Test chương trình:**

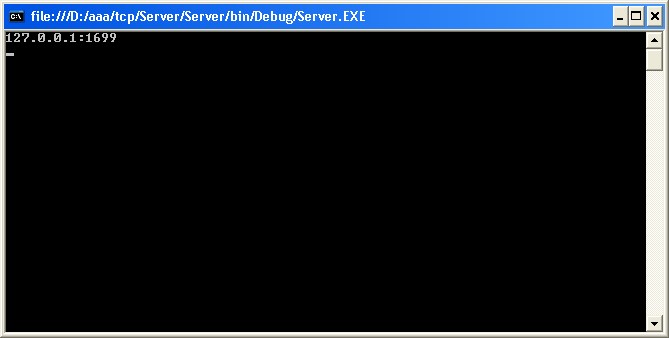
Chạy chương trình Server:



Chạy chương trình CMD và telnet vào server trên với địa chỉ IP loopback và port 5000:



Kết quả trả về trên Server:



**III.1.3. Trả lời câu hỏi:**

1. Khi telnet vào Server bằng port 5000 tại sao port client dùng để kết nối vào lại không phải là port 5000 ?

Khi kết nối TCP được tạo cần 4 cặp giá trị để xác đinh duy nhất (IP nguồn, port nguồn, IP đích, port đích)

- Server: Lắng nghe trên 1 cổng cố định (5000) để client biết nơi để kết nối (port đích).

- Client: khi tạo kết nối, hệ điều hành tự động gán cho một cổng tạm thời đây là cổng nguồn.

🡪 Client không thể dụng cổng 5000 vì server đang dùng, client cần cổng riêng để giử và nhận dữ liệu.

1. Như hình trên, giải thích vì sao có client dùng port giá trị 1699 ?

Số cổng 1699 là một cổng tạm thời (ephemeral port) được hệ điều hành của máy client tự động chọn và gán cho ứng dụng telnet tại thời điểm kết nối.

1. Có phải lúc nào client cũng mở port 1699 để kết nối với Server không ?

Không, cổng này được hệ điều hành cấp phát động và sẽ khác nhau trong mỗi lần kết nối mới. Nếu bạn đóng cửa sổ telnet và kết nối lại, rất có thể bạn sẽ thấy một số cổng khác.

1. Ta có thể chạy 2 chương trình CMD để telnet vào Server trên được không ? Giải thích.

.....................................................................................................................................

.....................................................................................................................................

## III.2. Tạo chương trình Server đơn giản, khi Client kết nối vào thì gởi câu chào cho client

### III.2.1. Hướng dẫn lập trình

Sử dụng chương trình Server đơn giản ở trên, thêm phần gởi dữ liệu xuống client

byte[] buff;

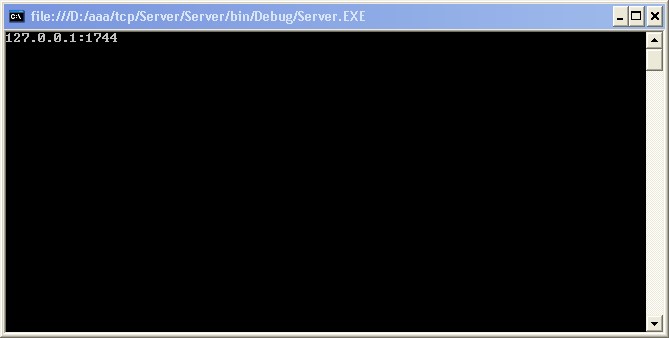
string hello = "Hello Client"; buff = Encoding.ASCII.GetBytes(hello); clientSocket.Send(buff, 0, buff.Length, SocketFlags.None); **III.2.2. Test chương trình** Chạy chương trình Server:



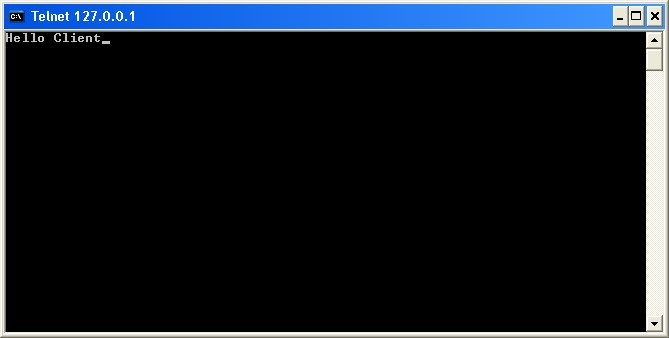
Chạy chương trình CMD và telnet vào server trên với địa chỉ IP loopback và port 5000:



Kết quả trả về trên Server:



Kết quả trả về trên client :



**III.2.3. Trả lời câu hỏi:**

1. Tạo sao trong đoạn code viết thêm vào ta không cần phải khởi tạo độ lớn ban đầu cho buffer ?

.....................................................................................................................................

.....................................................................................................................................

1. Giải thích ý nghĩa các tham số của phương thức clientSocket.Send .

...........................................................................................................................................................

## III.3. Xây dựng chương trình client đơn giản

### III.3.1. Hướng dẫn lập trình

Để lập trình socket ta using 2 namespace:

using System.Net;

using System.Net.Sockets;

Tạo Server EndPoint, EndPoint này sẽ tham chiếu đến địa chỉ IP và Port của Server:

IPEndPoint serverEndPoint = new IPEndPoint(IPAddress.Loopback, 5000);

Tạo socket dùng để kết nối với server:

Socket serverSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream,

ProtocolType.Tcp);

Console.WriteLine("Dang ket noi voi server..."); serverSocket.Connect(serverEndPoint);

Nếu kết nối thành công, nhận câu chào từ server và hiển thị câu chào lên màn hình:

if (serverSocket.Connected)

{

Console.WriteLine("Ket noi thanh cong voi server ...");

byteReceive = serverSocket.Receive(buff, 0, buff.Length, SocketFlags.None);

str = Encoding.ASCII.GetString(buff, 0, byteReceive)

Console.WriteLine(str);

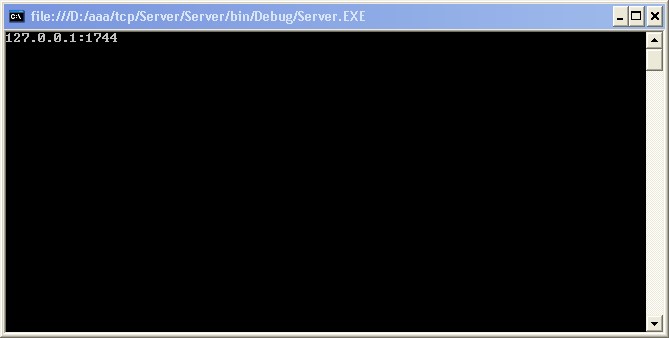
}

**III.3.2. Test chương trình**

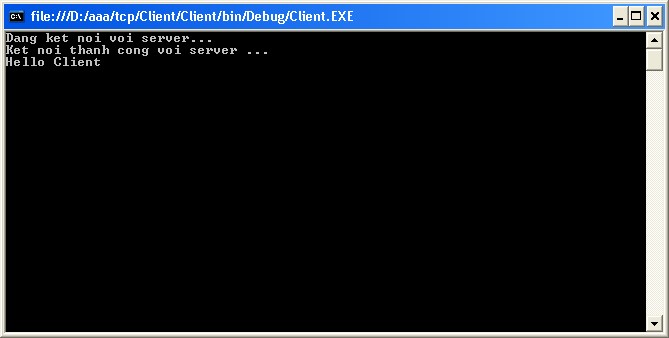
Chạy chương trình Server:



Kết quả trả về trên Server:



Kết quả trả về trên client :



**III.3.3. Trả lời câu hỏi:**

1. Khi client kết nối đến server mà server chưa chạy hoạc là đường truyền mạng bị rớt thì có hiện tượng gì xảy ra ?, giải thích.

..................................................................................................................................... .....................................................................................................................................

1. Từ đoạn code ở chương trình client: str = Encoding.ASCII.GetString(buff, 0, byteReceive)

Nếu ta thay byteReceive bằng buff.Lengh thì kết quả sẽ như thế nào? Giải thích.

..................................................................................................................................... .....................................................................................................................................

## III.4. Cải tiến chương trình client

### III.4.1. Hướng dẫn lập trình

Chương trình client này khi kết nối đến server mà server chưa bật hoạc đường truyền bị lỗi thì chương trình sẽ bị lỗi.

Sửa lại chương trình bằng cách thêm try-catch vào:

try {

serverSocket.Connect(serverEndPoint);

}

catch (SocketException se)

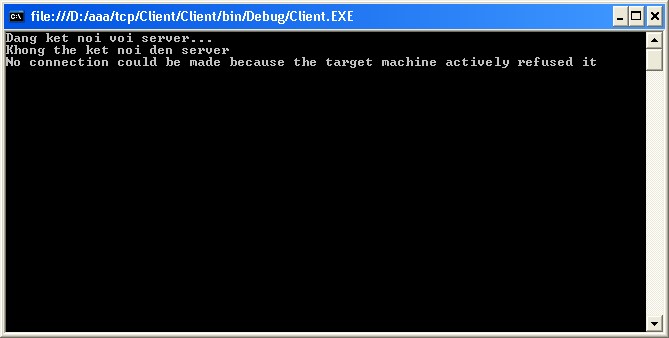
{

Console.WriteLine("Khong the ket noi den server"); return;

}

### III.4.2. Test chương trình

Chạy chương trình ở client mà không chạy server:



## III.5. Thêm khả năng gởi dữ liệu lên server và nhận lại dữ liệu từ server gởi về

### III.5.1. Hướng dẫn lập trình

Việc này được thực hiện bằng cách thêm vòng lặp vô hạn

while (true)

{

str = Console.ReadLine(); buff = Encoding.ASCII.GetBytes(str);

serverSocket.Send(buff, 0, buff.Length, SocketFlags.None); buff = new byte[1024];

byteReceive = serverSocket.Receive(buff, 0, buff.Length, SocketFlags.None);

str = Encoding.ASCII.GetString(buff, 0, byteReceive);

Console.WriteLine(str); }

## III.6. Thêm khả năng nhận dữ liệu từ client gởi lên và gởi trả dữ liệu ngược lại client

### III.6.1. Hướng dẫn lập trình

Việc này được thực hiện bằng cách thêm vòng lặp vô hạn

while (true)

{

buff = new byte[1024];

byteReceive = clientSocket.Receive(buff, 0, buff.Length, SocketFlags.None);

str = Encoding.ASCII.GetString(buff, 0, byteReceive); Console.WriteLine(str);

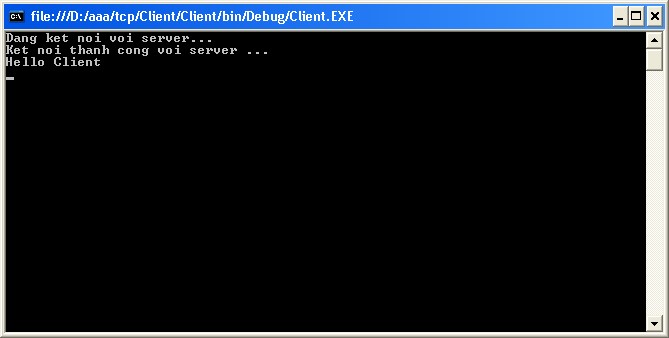
clientSocket.Send(buff, 0, byteReceive, SocketFlags.None); }

### III.6.2. Test chương trình

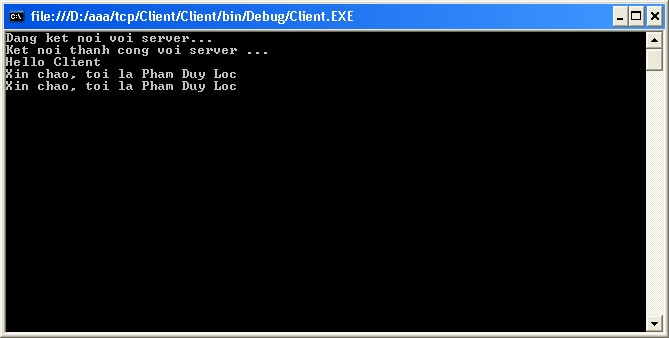
Chạy chương trình server:



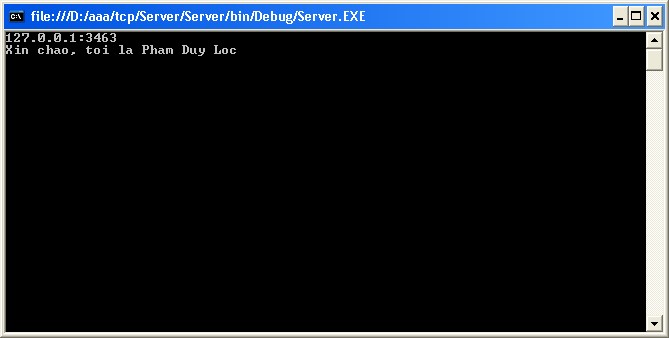
Chạy chương trình client:



Nhập dữ liệu và gởi lên server:



Kết quả hiển thị trên server:



**III.6.3. Trả lời câu hỏi:**

1. Khi thay đổi kích thước bộ đệm từ 1024 xuống 10 thì có hiện tượng gì xảy ra ?. Giải thích

.....................................................................................................................................

.....................................................................................................................................

1. Nếu ta không reset lại bộ đệm thì có hiện tượng gì xảy ra? Giải thích.

..................................................................................................................................... .....................................................................................................................................

# IV. Bài tập

1. Chương trình trên khi tắt client thì trên server sẽ bị lỗi, hãy cải tiến lại chương để khi client tắt đột ngột trên server sẽ thông báo client đã ngắt kết nối
2. Cải tiến chương trình: khi client nhập chữ “exit” thì sẽ tắt client, trên server thông báo client đã ngắt kết nối
3. Viết chương trình thực hiện các phép tính số học, việc thực hiện tính toán được thực hiện trên server, phía client sẽ nhập vào các toán hạng và các phép tính