**LAB 02**

**1. Tạo một chương trình Server cơ bản**

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

Do port của server được cài đặt cố định là ‘5000’, còn port của client do hệ điều hành chọn ngẫu nhiên.

***2) Như hình trên, giải thích vì sao có client dùng port giá trị 1699?***

Cổng từ 1024 đến 49151 là cổng đã được đăng ký bởi IANA (Internet Assigned Numbes Authority) hoặc ICANN (Internet Corporation for Assighned Names and Numbers), dùng cho các ứng dụng hoặc giao thức nhất định.

***3) Có phải lúc nào client cũng mở port 1699 để kết nối với Server không?***

Không. Nếu ứng dụng hoặc client được cài đặt cổng 1699 tĩnh thì mỗi lần kết nối sẽ đến server thì sẽ dùng cổng đó.

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

Có. Vì server có thể nhận nhiều kết nối đồng thời trên port 5000, và mỗi CMD sẽ được cấp một port ngẫu nhiên khác nhau để kết nối đến Server.

**2. Tạo chương trình Server cơ bản, khi Client kết nối vào thì gửi câu chào cho Client**

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

Vì phương thức Encoding.UTF8.GetBytes(hello) đã tự động tính toán và tạo ra một buffer có kích thước đúng bằng với nội dung tin nhắn, đảm bảo hiệu quả và không lãng phí bộ nhớ.

***2) Giải thích ý nghĩa các tham số của phương thức clientSocket.Send.***

Phương thức clientSocket.Send() dùng để gửi dữ liệu, phương thức này có một số đối số quan trọng sau:

* buff: mảng chứa dữ liệu cần gửi
* 0: vị trí bắt đầu gửi trong mảng (offset)
* buff.Lenght: số byte sẽ gửi đi (size)
* SocketFlags.None: không dùng tùy chọn đặc biệt nào khi gửi dữ liệu

**3. Xây dựng chương trình Client đơn giản**

***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.***

Client sẽ không kết nối được và phát sinh lỗi (Exception) — thường là lỗi SocketException với thông báo: “No connection could be made because the target machine actively refused it”. Vì server chưa khởi động → port đó chưa được “lắng nghe” (listen), hệ điều hành bên server sẽ trả lại gói RST (Reset) báo rằng không có ứng dụng nào đang mở cổng này. Do đó, socket ở client ném ra ngoại lệ (exception).

***2) 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.***

Khi thay *byteReceive* bằng *buff.Lengh,* chuỗi str sẽ bao gồm cả dữ liệu rác (null bytes) còn sót lại trong buff từ các lần nhận trước. Vì buff có kích thước cố định là 1024 byte, nhưng thực tế có thể bạn chỉ nhận vài chục byte dữ liệu. Các byte thừa (chưa ghi dữ liệu mới) sẽ là giá trị mặc định 0 (ký tự '\0' trong ASCII).

Nguyên nhân là vì: Khi nhận dữ liệu qua TCP, số byte nhận được mỗi lần không cố định. Hàm Receive() chỉ trả về số byte thực tế có trong buffer tại thời điểm đó. Nếu đọc cả buff.Length thì đang yêu cầu giải mã toàn bộ 1024 byte, trong đó nhiều byte không chứa dữ liệu hợp lệ.

=> Vì vậy, Encoding.ASCII.GetString() sẽ cố gắng chuyển đổi cả phần chưa sử dụng của buffer, dẫn đến: ký tự rác, dòng trống, dữ liệu cũ từ lần trước còn sót lại.

**6. Thêm khả năng nhận dữ liệu từ Client gửi lên và gửi trả dữ liệu ngược lại Client**

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

Khi thay đổi kích thước bộ đệm từ 1024 xuống 10 thì dữ liệu truyền từ server gửi tới client hay từ client tới server sẽ chỉ nhận được giới hạng là 10 byte dữ liệu nên sẽ có sự sai lệch về dữ liệu khi truyền giữa client với server.

Nếu giới hạn byte chỉ có 10 thì dữ liệu nhận được chỉ có 10 byte. Các dữ liệu còn lại vẫn còn được lưu trong bộ nhớ đệm nếu client gửi dữ liệu có số byte lớn hơn 10.

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

Nếu không reset lại bộ đệm thì trong bộ đệm vẫn còn lưu dữ liệu bên trong nếu như client gửi dữ liệu có số byte quá lớn. Điều này sẽ dấn đến dữ liệu từ server gửi về client sẽ bị sai lệch.