

§4. Lập trình socket

Các bài học trước đã cung cấp nhiều kiến thức mà công nghệ .NET hỗ trợ trong lập trình mạng. Từ bài này, kiến thức sẽ sâu hơn. Trước hết là các socket. Bài này tìm hiểu:

- Thế nào là socket? Có những kiểu socket nào?
- Lớp `System.Net.Sockets.Socket` của .NET.
- Các tham số để thiết lập socket.
- Áp dụng các quyền hạn khác nhau với socket.

Sẽ có hai ví dụ ứng dụng điển hình:

- Ứng dụng server dùng TCP socket đồng bộ.
- Ứng dụng server dùng TCP socket bất đồng bộ.

1. Thế nào là socket

Mỗi socket là một đầu mối liên lạc của thực thể chạy ứng dụng mạng. Socket của thực thể này, nối với socket của thực thể khác, qua mạng. Thông qua kết nối giữa socket với socket, ta có thể truyền dữ liệu hai chiều. Socket là một cách bao gói chung các giao thức tầng mạng và tầng chuyển vận.

Socket được phát triển bởi các nhà khoa học thuộc đại học bang California tại Berkeley, nhằm vào môi trường UNIX và dựa trên tư tưởng phát triển hết sức đơn giản. Trước khi dùng một tài nguyên, nó cần phải được mở trước với một quyền hạn cụ thể nào đó, chỉ định bởi các tham số. Khi đã mở rồi, tài nguyên có thể đọc ra, hoặc ghi vào. Sau cùng, tài nguyên cần phải được đóng lại để báo cho hệ điều hành biết rằng tác vụ đã kết thúc.

Socket cũng là một tài nguyên, và cần được hệ điều hành quản lý. Để quản lý được, socket (cũng như các tài nguyên khác) cần có một **bộ mô tả** (descriptor, hay handle) dưới dạng giống như con trỏ tập tin, con trỏ bộ nhớ... để truy cập tới nội dung cụ thể bên trong.

Bộ mô tả có chu kỳ sống gồm ba giai đoạn: *open/create*, *receive/send*, và *close*.

Các kiểu socket

Có hai kiểu socket cơ bản gồm: các stream socket và các datagram socket. Ngoài ra còn có thêm một loại socket “chung chung”, gọi là “raw socket”.

Các stream socket

- Một stream socket là một socket dạng kết nối thường xuyên, có hướng (connection-oriented), mà nó bao hàm một luồng byte đẳng hướng. Có nghĩa là, ứng dụng có thể thực hiện cả hai chiều, truyền và nhận.
- Một stream socket tự thực hiện các tác vụ sửa lỗi, giao nhận và bảo toàn thứ tự các gói tin và tự xử lý trùng lặp.
- Một stream socket phù hợp cho việc truyền dữ liệu với khối lượng lớn
- Một stream socket sử dụng giao thức TCP để bảo đảm chất lượng phiên truyền.
- Một phiên truyền dẫn dùng stream socket cần phải thiết lập kênh truyền trước.
- Thủ tục kết nối các stream socket luôn phải tường minh: socket A yêu cầu kết nối tới socket B; socket B có thể chấp nhận hoặc từ chối.

Các stream socket thường được ưa dùng hơn so với datagram socket khi cần truyền gửi dữ liệu với khối lượng lớn, và đặc biệt là khi đòi hỏi đảm bảo giao nhận.

Các datagram socket

- Một datagram socket là một socket dạng kết nối không thường xuyên, vô hướng (connectionless). Có nghĩa là, không thực sự có một kết nối tường minh nào được khởi tạo.
- Truyền gửi bằng datagram socket không tin cậy bằng stream socket. Nhưng ở một số ứng dụng, việc đợi khởi tạo một kết nối là không thể chấp nhận được, chẳng hạn như các ứng dụng đồng bộ hóa thời gian giữa server và client. Những ứng dụng như thế tốt nhất là dùng datagram socket để giảm thiểu thông tin phụ trội và giảm thiểu thời gian trễ trong xử lý các thông tin này và cũng giảm thời gian tạo kết nối.
- Một datagram socket sử dụng giao thức UDP để truyền gửi dữ liệu từ client tới server. Thường có sự hạn chế khối lượng dữ liệu và cũng bảo đảm giao nhận, không có sự đáp trả từ server nếu dữ liệu bị mất trên đường truyền.

Các raw socket

Raw socket là một dạng socket mà nó trực tiếp nhận các gói tin, bỏ qua các tầng TCP và UDP trong giao thức phân tầng TCP/IP và gửi chúng thẳng tới ứng dụng.

Với các raw socket như thế, các gói tin được truyền nhận dưới dạng nguyên thủy, không có sự thêm bớt hay cời bỏ các header, và đến thẳng ứng dụng.

Các raw socket ít được dùng trong các ứng dụng loại trừ các ứng dụng mức hệ thống như ping, traceroute...

Các port

Một port, hay cổng, được xác định nhằm giải quyết vấn đề liên lạc đồng thời giữa nhiều ứng dụng khác nhau. Mỗi máy tính có nhiều port mang các số hiệu khác nhau để phân biệt.

Socket là sự kết hợp giữa địa chỉ IP của một máy tính cụ thể và một số hiệu cổng. Vì địa chỉ IP là duy nhất trên Internet, và các số hiệu port cũng phân biệt với nhau trên mỗi máy. Do đó, socket cũng được phân biệt trên Internet, và cho phép tiến trình trên máy này làm việc chính xác với tiến trình trên máy kia.

Một số dịch vụ đặc biệt được Internet Assigned Numbers Authority (IANA) gán vĩnh viễn cho một số hiệu cổng đặc biệt (well-known port numbers).

Một máy tính có thể đồng thời chạy nhiều ứng dụng cùng truyền nhận qua mạng. Dù chỉ có một (hoặc vài) địa chỉ IP, thì dựa vào việc kết hợp với số hiệu cổng, không bao giờ có sự nhầm lẫn trong giao nhận dữ liệu giữa các ứng dụng đang chạy.

Một ứng dụng client-server điển hình thường có hai phiên bản phần mềm, một cho server luôn chạy để đợi phục vụ các kết nối đến, một cho client có thể đề nghị mở kết nối đến server vào bất cứ lúc nào. Đương nhiên, client phải biết địa chỉ IP và cổng phục vụ của phần mềm server để nhận được dịch vụ cần thiết. Khi muốn kết nối với server, một quy trình sau diễn ra:

- Client: Yêu cầu kết nối tới địa chỉ IP và số hiệu cổng trên server
- Server: Chấp nhận hay hủy bỏ yêu cầu kết nối. Nếu chấp nhận, nó dàn xếp một socket mới để phục vụ riêng cho client. Báo cho client về socket mới này, rồi quay lại đợi yêu cầu kết nối khác.

- Client/Server bắt đầu phiên truyền gửi mong muốn. Một trong hai bên có thể đóng kết nối.

2. Hỗ trợ socket trong .NET

Như đã đề cập. Không gian System.Net.Sockets của .NET bao hàm nhiều lớp con, gồm: MulticastOption, NetworkStream, TcpClient, TcpListener, UdpClient, SocketException và Socket.

Lớp con Socket cung cấp chức năng cơ bản cho một ứng dụng dùng socket.

Lớp System.Net.Sockets.Socket

Lớp socket đóng vai trò quan trọng trong lập trình mạng, thực hiện các hành động trên cả server và client. Hầu hết các lời gọi phương thức tới lớp này là việc thực hiện các phép kiểm tra an ninh cần thiết, rồi sau đó ủy nhiệm toàn bộ công việc còn lại cho Windows Sockets API. Một số thuộc tính lớp quan trọng gồm:

Thuộc tính	Mô tả
AddressFamily	Lấy họ địa chỉ của socket.
Available	Lấy chiều dài toàn bộ dữ liệu cần đọc.
Blocking	Lấy hoặc thiết lập giá trị biểu thị chế độ chặn của socket .
Connected	Cho biết socket có còn kết nối với phía kia hay không.
LocalEndPoint	Lấy giá trị điểm cuối cục bộ (Ứng dụng cục bộ).
ProtocolType	Lấy kiểu giao thức mà socket đang dùng.
SocketType	Lấy kiểu socket.

Các phương thức quan trọng của lớp System.Net.Sockets.Socket gồm:

Phương thức	Mô tả
Accept()	Tạo mới một socket để đáp ứng một yêu cầu kết nối.
Bind()	Kết hợp một socket với một ứng dụng cục bộ (local endpoint) để đợi kết kết nối đến.
Close()	Buộc socket phải tự đóng chính nó.
Connect()	Thiết lập kết nối với đầu kia.
GetSocketOption()	Trả về giá trị của một SocketOption.

IOControl()	Thiết lập cấp thấp chế độ làm việc cho socket. Phương thức này cung cấp truy xuất mức thấp tới mức thành phần cấu trúc tạo thành socket.
Listen()	Chuyển socket sang chế độ lắng nghe, chỉ áp dụng cho ứng dụng server.
Receive()	Nhận dữ liệu từ socket.
Poll()	Thăm dò tình trạng socket.
Select()	Kiểm tra trạng thái làm việc của một hay nhiều socket.
Send()	Gửi dữ liệu tới socket.
SetSocketOption()	Thiết lập một giá trị tùy chọn SocketOption.
Shutdown()	Chấm dứt truyền nhận với một socket.