

基于 UDP 的 WinSock 编程

一、实验目的

- 1、深入了解 TCP 和 UDP 的区别与联系。
- 2、掌握使用 WinSock 编写面向非连接的 Client/Server 结构的程序的基本编程方法。

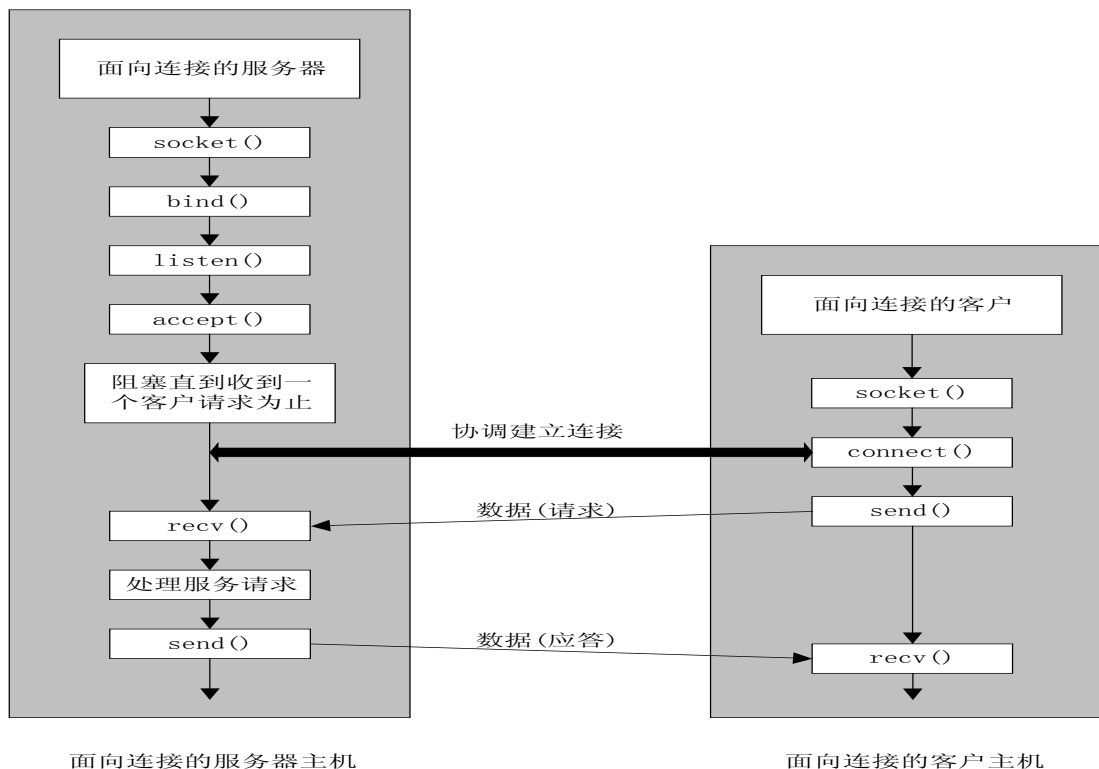
二、实验原理

在 TCP 编程实验中，我们已经学习了如何使用 WinSock 建立一个 Socket 接口，以及如何建立一个连接，并进行通信。在这一个实验里，我们将学习如何进行非连接的通信。

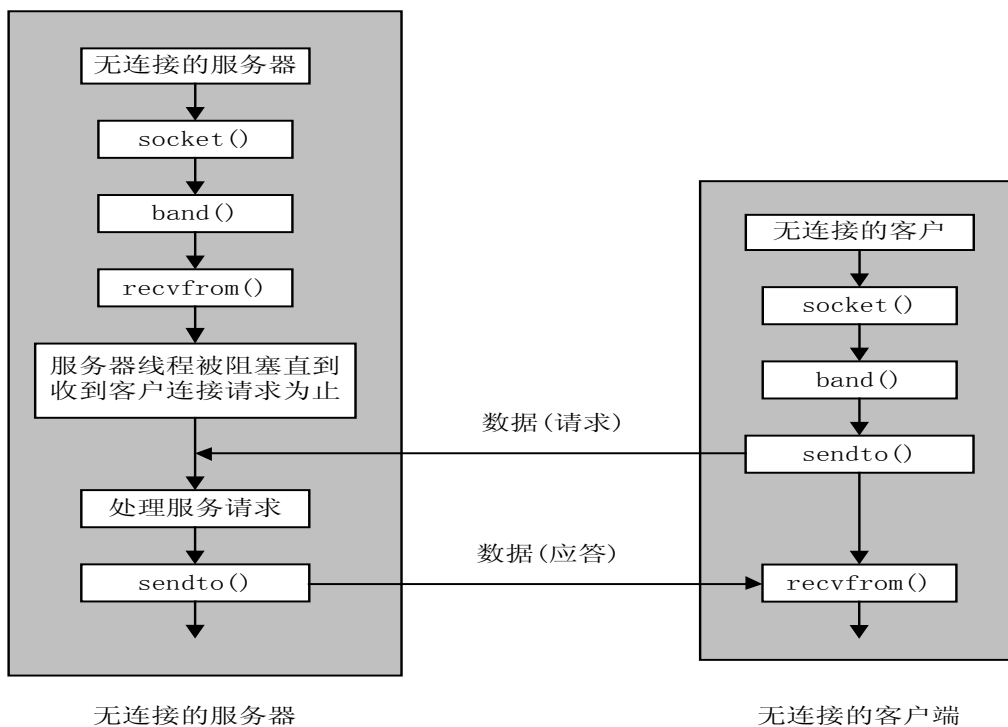
传输层和应用层之间进行交换的数据称为报文 (Message)，而在传输层和网络层之间进行交换的数据被称为数据报 (Datagram)。传输层可以使用传输控制协议 (TCP) 来封装数据，也可以使用用户数据报协议 (UDP) 来封装数据。TCP 协议面向连接，使用字节流传送服务，是可靠的；而 UDP 协议面向非连接，使用数据报服务，是非可靠的。TCP 协议提供高可靠性的传输，UDP 协议提供高效的传输。在实际应用中，它们有其各自所适应的场合。

在无连接的 Client/Server 结构中，服务器使用 socket 和 bind 函数调用建立和连接 socket。由于此时的 socket 是无连接的，服务器使用 recvfrom() 函数从 socket 接收数据。客户端也只调用 bind() 函数而不调用 connect 函数。注意，无连接的协议不在两个端口之间建立点到点的连接。因此 sendto() 函数要求程序在一个参数中指明目的地址。recvfrom() 函数不需要建立连接，它对到达相连协议端口的任何数据作出响应。当 recvfrom() 函数从 socket 收到一个数据报时，它将保存发送此数据包的进程的网络地址以及数据包本身。程序 (服务器和客户) 用保存的地址去确定发送 (客户) 进程。在必要的条件下，服务器将其应答数据报送到从 recvfrom() 函数调用中所得到的网络地址中去。

使用有连接协议时的 Client/Server 结构程序模型



使用无连接协议时的 Client/Server 结构程序模型



三、实验操作

- 1、选择一个操作系统环境 (Linux 或者 WindowsNT), 编制 UDP/IP 通信程序, 完成一定的通信功能。
- 2、在发送 UDP 数据包时做一个循环, 连续发送 100 个数据包; 在接受端统计丢失的数据包。

四、思考题

- 1、说明面向连接的客户端和面向非连接的客户端在建立 socket 时有什么区别。
- 2、说明面向连接的客户端和面向非连接的客户端在收发数据时有什么区别。在非连接的客户端又是如何判断数据发送结束的。
- 3、使用 socket API 开发无连接的客户端程序和服务器程序时, 各需要哪些不同的函数。
- 4、比较面向连接的通信和无连接通信, 它们各有什么优点和缺点? 适合在何种场合下使用?