

# python网络编程

## 14.1.1 socket 模块

在网络编程中的一个基本组件就是套接字 (socket)。套接字主要是两个程序之间的“信息通道”。程序可能（通过网络连接）分布在不同的计算机上，通过套接字相互发送信息。在Python中的大多数的网络编程都隐藏了socket模块的基本细节，并且不直接和套接字交互。

套接字包括两个：服务器套接字和客户机套接字。创建一个服务器套接字后，让它等待连接。这样它就在某个网络地址处（IP地址和一个端口号的组合）监听。

处理客户端套接字通常比处理服务器端套接字容易，因为服务器必须准备随时处理客户端的连接，同时还要处理多个连接，而客户机只是简单地连接，完成事务，断开连接。在这章后面，我使用socketserver类族和Twisted框架来处理服务器端的编程。

一个套接字就是一个socket模块中的socket类的实例。它的实例化需要3个参数：第1个参数是地址族（默认是socket.AF\_INET）；第2个参数是流（socket.SOCK\_STREAM，默认值）或数据报（socket.SOCK\_DGRAM）套接字<sup>①</sup>；第3个参数是使用的协议（默认是0，使用默认值即可）。对于一个普通的套接字，不需要提供任何参数。

服务器端套接字使用bind方法后，再调用listen方法去监听这个给定的地址。客户端套接字使用connect方法连接到服务器在connect方法中使用的地址与bind方法中的地址相同（在服务器端，能实现很多功能，比如使用函数socket.gethostname得到当前主机名）。在这种情况下，一个地址就是一个格式为（host, port）的元组，其中host是主机名（比如www.example.com），port是端口号（一个整数）。listen方法只有一个参数，即服务器未处理的连接的长度（即允许排队等待的连接数目，这些连接在停止接收之前等待接收）。

服务器端套接字开始监听后，它就可以接受客户端的连接。这个步骤使用accept方法来完成。这个方法会阻塞（等待）直到客户端连接，然后该方法就返回一个格式为（client, address）的元组，client是一个客户端套接字，address是一个前面解释过的地址。服务器能处理客户端到它满意的程度，然后调用另一个accept方法开始等待下一个连接。这个过程通常都是在一个无限循环中实现的。

---

**注意** 这种形式的服务器端编程称为阻塞或者是同步网络编程。14.3节会介绍一些非阻塞或者叫做异步网络编程的例子，例子中还使用了线程来同时处理多个客户机。

---