第13讲讲套接字Socket：Talkischeap,showmethecode

建立Socket的时候，Socket编程进行的是端到端的通信，往往意识不到中间经过多少局域网，多少路由器，因而能够设置的参数，也只能是端到端协议之上网络层和传输层的。

IPv4还是IPv6，分别对应设置为AF\_INET和AF\_INET6。

TCP协议是基于数据流的，所以设置为SOCK\_STREAM

UDP是基于数据报的，因而设置为SOCK\_DGRAM。

两端创建了Socket之后，接下来的过程中，TCP和UDP稍有不同，我们先来看TCP。

### 基于TCP的socket连接

建立连接流程：

服务端：

（1）TCP的服务端要先监听一个端口，一般是先调用bind函数，给这个Socket赋予一个IP地址和端口。

（2）调用listen函数进行监听。在TCP的状态图里面，有一个listen状态，当调用这个函数之后，服务端就进入了这个状态，这个时候客户端就可以发起连接了。

（3）在内核中，为每个Socket维护两个队列。一个是已经建立了连接的队列，这时候连接三次握手已经完毕，处于established状态；一个是还没有完全建立连接的队列，这个时候三次握手还没完成，处

于syn\_rcvd的状态。

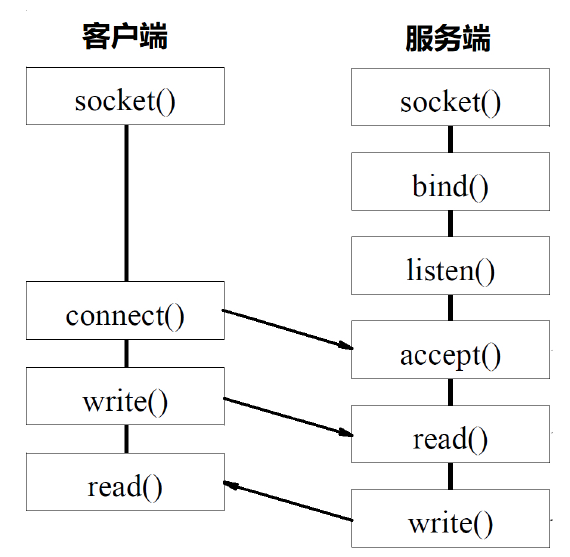
服务端调用accept函数，拿出一个已经完成的连接进行处理。如果还没有完成，就要等着。

客户端：

在服务端等待的时候，客户端可以通过connect函数发起连接。先在参数中指明要连接的IP地址和端口号，然后开始发起三次握手。内核会给客户端分配一个临时的端口。一旦握手成功，服务端的accept就会返回另一个Socket。

为什么需要端口呢？内核需要通过端口才能知道要传到你这个程序里面。

为什么需要IP地址呢？，一台机器会有多个网卡，也就会有多个IP地址，你可以选择监听所有的网卡，也可以选择监听一个网卡，这样，只有发给这个网卡的包，才会给你。



注意（也是考点）：

就是监听的Socket和真正用来传数据的Socket是两个，一个叫作**监听Socket**，一个叫作**已连接Socket**。

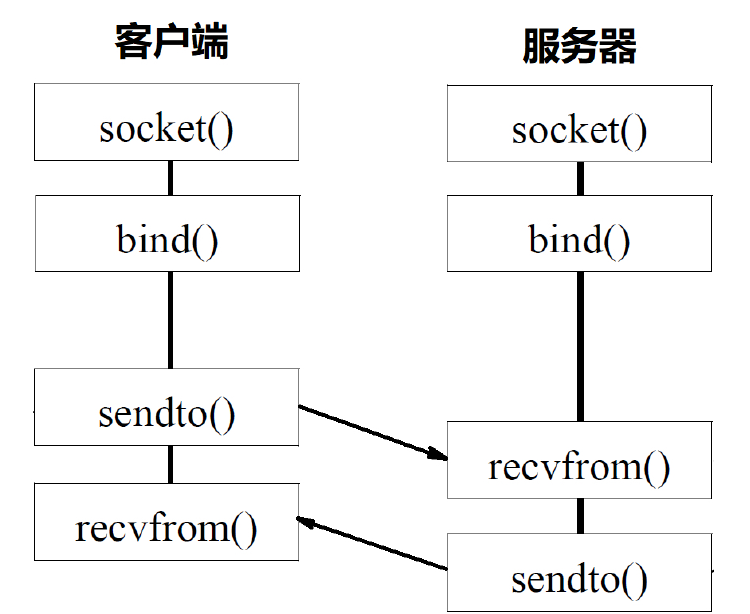
Socket就是一个文件流，Socket在Linux中就是以文件的形式存在的。除此之外，还存在文件描述符。写入和读出，也是通过文件描述符。

### 基于UDP协议的Socket程序函数调用过程

不需要三次握手，也就不需要listen和connect

Udp维护的连接没有状态，因此不需要每对连接都建立一组socket。一个socket负责和多个客户端连接。

每次通信的时候，都调用sendto和recvfrom，都可以传入IP地址和端口。



### 服务器如何增大连接数？

系统会用一个四元组来标识一个TCP连接。

{本机IP, 本机端口, 对端IP, 对端端口}

最大TCP连接数=客户

端IP数×客户端端口数

限制导致无法达到最大的连接数：**文件描述符限制、内存**

文件描述符：又上限

内存：每个TCP连接都要占用一定内存，操作系统是有限的。

1. 多进程的方式
2. 多线程的方式
3. IO复用的select轮询的方式
4. Epoll监听事件响应的方式

第四种方式可以避免第三种的轮询开销大，多线程的方式本身花费资源也大。