**TCP连接各种状态**

LISTEN：侦听来自远方的TCP端口的连接请求

SYN-SENT：再发送连接请求后等待匹配的连接请求（客户端）

SYN-RECEIVED：再收到和发送一个连接请求后等待对方对连接请求的确认（服务器）

ESTABLISHED：代表一个打开的连接

FIN-WAIT-1：等待远程TCP连接中断请求，或先前的连接中断请求的确认

FIN-WAIT-2：从远程TCP等待连接中断请求

CLOSE-WAIT：等待从本地用户发来的连接中断请求

CLOSING：等待远程TCP对连接中断的确认

LAST-ACK：等待原来的发向远程TCP的连接中断请求的确认

TIME-WAIT：等待足够的时间以确保远程TCP接收到连接中断请求的确认

CLOSED：没有任何连接状态

主动端可能出现的状态：FIN\_WAIT1、FIN\_WAIT2、CLOSING、TIME\_WAIT   
被动端可能出现的状态：CLOSE\_WAIT LAST\_ACK

客户端的状态可以用如下的流程来表示：   
CLOSED->SYN\_SENT->ESTABLISHED->FIN\_WAIT\_1->FIN\_WAIT\_2->TIME\_WAIT->CLOSED

服务器的状态可以用如下的流程来表示：   
CLOSED->LISTEN->SYN收到->ESTABLISHED->CLOSE\_WAIT->LAST\_ACK->CLOSED

**建立连接时的状态变迁**

一开始，建立连接之前服务器和客户端的状态都为CLOSED。服务器创建socket后开始监听，变为LISTEN状态。客户端请求建立连接，向服务器发送SYN报文，客户端的状态变为SYN\_SENT。服务器收到客户端的报文后向客户端发送ACK和SYN报文，此时服务器的状态变为SYN\_RCVD。然后，客户端收到ACK、SYN，就向服务器发送ACK，客户端状态变为ESTABLISHED，服务器收到客户端的ACK后也变为ESTABLISHED。此时，3次握手完成，连接建立！

**断开连接时的状态变迁**

由于tcp连接是全双工的，断开连接会比建立连接麻烦一点点。客户端先向服务器发送FIN报文，请求断开连接，其状态变为FIN\_WAIT1。服务器收到FIN后向客户端发生ACK，服务器状态变为CLOSE\_WAIT。客户端收到ACK后就进入FIN\_WAIT2状态。此时连接已经断开了一半了。如果服务器还有数据要发送给客户端，就会继续发送。直到发完了，就发送FIN报文，此时服务器进入LAST\_ACK状态。客户端收到服务器的FIN后，马上发送ACK给服务器，此时客户端进入TIME\_WAIT状态，再过了2MSL长的时间后进入CLOSED状态。服务器收到客户端的ACK就进入CLOSED状态。   
至此，还有一个状态没有提及：CLOSING状态。CLOSING状态表示客户端发生了FIN，但没有收到服务器的ACK，却收到了服务器的FIN。这种情况发生在服务器发送的ACK丢包的时候，因为网络传输有时会有意外。

## CLOSED:

这个没什么好说的了，表示初始状态。

* 1
* 2

## LISTEN:

这个也是非常容易理解的一个状态，表示服务器端的某个SOCKET处于监听状态，可以接受连接了。

## SYN\_RCVD:

这个状态表示接受到了SYN报文，在正常情况下，这个状态是服务器端的SOCKET在建立TCP连接时的三次握手会话过程中的一个中间状态，很短暂，基本上用netstat你是很难看到这种状态的，除非你特意写了一个客户端测试程序，故意将三次TCP握手过程中最后一个ACK报文不予发送。因此这种状态时，当收到客户端的ACK报文后，它会进入到ESTABLISHED状态。

## SYN\_SENT:

这个状态与SYN\_RCVD遥想呼应，当客户端SOCKET执行CONNECT连接时，它首先发送SYN报文，因此也随即它会进入到了SYN\_SENT状态，并等待服务端的发送三次握手中的第2个报文。SYN\_SENT状态表示客户端已发送SYN报文。

## ESTABLISHED：

这个容易理解了，表示连接已经建立了。

* 1
* 2

## FIN\_WAIT\_1:

这个状态要好好解释一下，其实FIN\_WAIT\_1和FIN\_WAIT\_2状态的真正含义都是表示等待对方的FIN报文。而这两种状态的区别是：FIN\_WAIT\_1状态实际上是当SOCKET在ESTABLISHED状态时，它想主动关闭连接，向对方发送了FIN报文，此时该SOCKET即进入到FIN\_WAIT\_1状态。而当对方回应ACK报文后，则进入到FIN\_WAIT\_2状态，当然在实际的正常情况下，无论对方何种情况下，都应该马上回应ACK报文，所以FIN\_WAIT\_1状态一般是比较难见到的，而FIN\_WAIT\_2状态还有时常常可以用netstat看到。

## FIN\_WAIT\_2：

上面已经详细解释了这种状态，实际上FIN\_WAIT\_2状态下的SOCKET，表示半连接，也即有一方要求close连接，但另外还告诉对方，我暂时还有点数据需要传送给你，稍后再关闭连接这就是著名的半关闭的状态了，这是在关闭连接时，客户端和服务器两次握手之后的状态。在这个状态下，应用程序还有接受数据的能力，但是已经无法发送数据，但是也有一种可能是，客户端一直处于FIN\_WAIT\_2状态，而服务器则一直处于WAIT\_CLOSE状态，而直到应用层来决定关闭这个状态

## TIME\_WAIT:

表示收到了对方的FIN报文，并发送出了ACK报文，就等2MSL后即可回到CLOSED可用状态了。如果FIN\_WAIT\_1状态下，收到了对方同时带FIN标志和ACK标志的报文时，可以直接进入到TIME\_WAIT状态，而无须经过FIN\_WAIT\_2状态。

## CLOSE\_WAIT:

这种状态的含义其实是表示在等待关闭。怎么理解呢？当对方close一个SOCKET后发送FIN报文给自己，你系统毫无疑问地会回应一个ACK报文给对方，此时则进入到CLOSE\_WAIT状态。接下来呢，实际上你真正需要考虑的事情是查看你是否还有数据发送给对方，如果没有的话，那么你也就可以close这个SOCKET，发送FIN报文给对方，也即关闭连接。所以你在CLOSE\_WAIT状态下，需要完成的事情是等待你去关闭连接。

## LAST\_ACK:

这个状态还是比较容易好理解的，它是被动关闭一方在发送FIN报文后，最后等待对方的ACK报文。当收到ACK报文后，也即可以进入到CLOSED可用状态了。