

tcp要点学习断开连接Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_tcp_E8_A6_81_E7_82_B9_E5_c103_644917.htm 主要部分，四次握手：断开连接其实从我的角度看不区分客户端和服务端，任何一方都可以调用close(or closesocket)之类的函数开始主动终止一个连接。这里先暂时说正常情况。当调用close函数断开一个连接时，主动断开的一方发送FIN(finish报文给对方。有了之前的经验，我想你应该明白我说的FIN报文时什么东西。也就是一个设置了FIN标志位的报文段。FIN报文也可能附加用户数据，如果这一方还有数据要发送时，将数据附加到这个FIN报文时完全正常的。之后你会看到，这种附加报文还会有很多，例如ACK报文。我们所要把握的原则是，TCP肯定会力所能及地达到最大效率，所以你能够想到的优化方法，我想TCP都会想到。当被动关闭的一方收到FIN报文时，它会发送ACK确认报文(对于ACK这个东西你应该很熟悉了)。这里有个东西要注意，因为TCP是双工的，也就是说，你可以想象一对TCP连接上有两条数据通路。当发送FIN报文时，意思是说，发送FIN的一端就不能发送数据，也就是关闭了其中一条数据通路。被动关闭的一端发送了ACK后，应用层通常会检测到这个连接即将断开，然后被动断开的应用层调用close关闭连接。我可以告诉你，一旦当你调用close(or closesocket)，这一端就会发送FIN报文。也就是说，现在被动关闭的一端也发送FIN给主动关闭端。有时候，被动关闭端会将ACK和FIN两个报文合在一起发送。主动关闭端收到FIN后也发送ACK，然后整个连接关闭(事实上还没完全关闭，只是

关闭需要交换的报文发送完毕)，四次握手完成。如你所见，因为被动关闭端可能会将ACK和FIN合到一起发送，所以这也算不上严格的四次握手---四个报文段。给出一个正常关闭时的windump信息：14:00:38.819856 IP cd-zhangmin.1748 gt. cd-zhangmin.1748: F 1:1(0) ack 2 win 2920 14:00:38.864412 IP cd-zhangmin.1748 > 220.181.37.55.80: . ack 2 win 65535 补充细节：关于以上的四次握手，我补充下细节：1. 默认情况下(不改变socket选项)，当你调用close(or closesocket，以下说close不再重复)时，如果发送缓冲中还有数据，TCP会继续把数据发送完。2. 发送了FIN只是表示这端不能继续发送数据(应用层不能再调用send发送)，但是还可以接收数据。3. 应用层如何知道对端关闭？通常，在最简单的阻塞模型中，当你调用recv时，如果返回0，则表示对端关闭。在这个时候通常的做法就是也调用close，那么TCP层就发送FIN，继续完成四次握手。如果你不调用close，那么对端就会处于FIN_WAIT_2状态，而本端则会处于CLOSE_WAIT状态。这个可以写代码试试。4. 在很多时候，TCP连接的断开都会由TCP层自动进行，例如你CTRL C终止你的程序，TCP连接依然会正常关闭，你可以写代码试试。特别的TIME_WAIT状态：从以上TCP连接关闭的状态转换图可以看出，主动关闭的一方在发送完对对方FIN报文的确认(ACK)报文后，会进入TIME_WAIT状态。TIME_WAIT状态也称为2MSL状态。什么是2MSL？MSL即Maximum Segment Lifetime，也就是报文最大生存时间，引用中的话：“它(MSL)是任何报文段被丢弃前在网络内的最长时间。”那么，2MSL也就是这个时间的2倍。其实我觉得没必要把这个MSL的确切含义搞明白，你所需要明白的是，

当TCP连接完成四个报文段的交换时，主动关闭的一方将继续等待一段时间(2-4分钟)，即使两端的应用程序结束。你可以写代码试试，然后用netstat查看下。为什么需要2MSL？根据和中的说法，有两个原因：其一，保证发送的ACK会成功发送到对方，如何保证？我觉得可能是通过超时计时器发送。这个就很难用代码演示了。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com