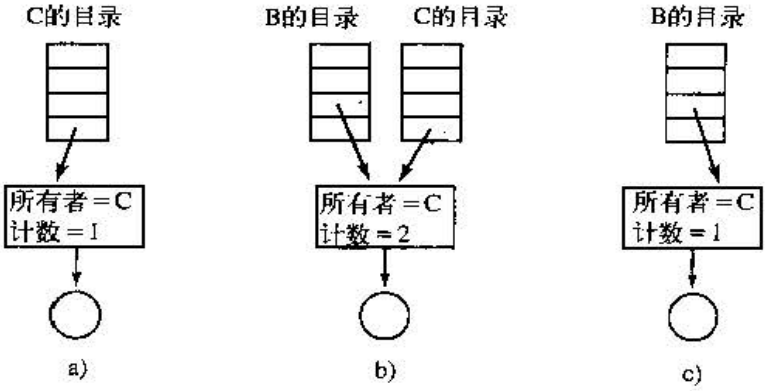

图4-17 a) 连接之前的状况；b) 创建连接之后；c) 当所有者删除文件后

对于符号连接，以上问题不会发生，因为只有真正的文件所有者才有一个指向i节点的指针。连接到该文件上的用户只有路径名，没有指向i节点的指针。当文件所有者删除文件时，该文件被销毁。以后若试图通过符号连接访问该文件将导致失败，因为系统不能找到该文件。删除符号连接根本不影响该文件。

符号连接的问题是需要额外的开销。必须读取包含路径的文件，然后要一个部分一个部分地扫描路径，直到找到i节点。这些操作也许需要很多次额外的磁盘存取。此外，每个符号连接都需要额外的i节点，以及额外的一个磁盘块用于存储路径，虽然如果路径名很短，作为一种优化，系统可以将它存储在i节点中。符号连接有一个优势，即只要简单地提供一个机器的网络地址以及文件在该机器上驻留的路径，就可以连接全球任何地方的机器上的文件。

还有另一个由连接带来的问题，在符号连接和其他方式中都存在。如果允许连接，文件有两个或多个路径。查找一指定目录及其子目录下的全部文件的程序将多次定位到被连接的文件。例如，一个将某一目录及其子目录下的文件转储到磁带上的程序有可能多次复制一个被连接的文件。进而，如果接着把磁带读进另一台机器，除非转储程序具有智能，否则被连接的文件将被两次复制到磁盘上，而不是只是被连接起来。

4.3.5 日志结构文件系统

不断进步的科技给现有的文件系统带来了更多的挑战。特别是CPU的运行速度越来越快，磁盘容量越来越大，价格也越来越便宜（但是磁盘速度并没有增快多少），同时内存容量也以指数形式增长。而没有得到快速发展的参数是磁盘的寻道时间。所以这些问题综合起来，便成为影响很多文件系统性能的一个瓶颈。为此，Berkeley设计了一种全新的文件系统，试图缓解这个问题，即日志结构文件系统（Log-structured File System，LFS）。在这一节里，我们简要说明LFS是如何工作的。如果需要了解更多相关知识，请参阅（Rosenblum 和Ousterhout，1991）。

促使设计LFS的主要原因是，CPU的运行速度越来越快，RAM内存容量变得更大，同时磁盘高速缓存也迅速地增加。进而，不需要磁盘访问操作，就有可能满足直接来自文件系统高速缓存的很大一部分读请求。所以从上面的事实可以推出，未来多数的磁盘访问是写操作，这样，在一些文件系统中使用的提前读机制（需要读取数据之前预取磁盘块），并不能获得更好的性能。

更为糟糕的情况是，在大多数文件系统中，写操作往往都是零碎的。一个50μs的磁盘写操作之前通常需要10ms的寻道时间和4ms的旋转延迟时间，可见零碎的磁盘写操作是极其没有效率的。根据这些参数，磁盘的效率降低到1%以下。

为了看看这样小的零碎写操作从何而来，考虑在UNIX文件系统上创建一个新文件。为了写这个文件，必须写该文件目录的i节点、目录块、文件的i节点以及文件本身。而这些写操作都有可能被延迟，那么如果在写操作完成之前发生死机，就可能在文件系统中造成严重的不一致性。正因为如此，i节点的写操作一般是立即完成的。

出于这一原因，LFS的设计者决定重新实现一种UNIX文件系统，该系统即使对于一个大部分由零碎的随机写操作组成的任务，同样能够充分利用磁盘的带宽。其基本思想是将整个磁盘结构化为一个日志。每隔一段时间，或是有特殊需要时，被缓冲在内存中的所有未决的写操作都被放到一个单独的段中，