绝对路径的缺点是文件名太长并且不方便。因为这个原因，Linux允许用户和进程把他们当前工作的目录标识为工作目录，这样路径名就可以相对于工作目录命名，这种方式命名的目录名叫做相对路径。例如，如果/usr/ast/books/mos3是工作目录，那么shell命令

```
cp chap-10 backup-10
```

和长命令cp /usr/ast/books/mos3/chap-10/usr/ast/books/mos3/ backup-10 的效果是一样的。

一个用户要使用属于另一个用户的文件或者使用文件树结构里的某个文件的情况是经常发生的。例如，两个用户共享一个文件，这个文件位于其中某个用户所拥有的目录中，另一个用户需要使用这个文件时，必须通过绝对路径才能引用它（或者通过改变工作目录的方式）。如果绝对路径名很长，那么每次输入时将会很麻烦。为了解决这个问题，Linux提供了一种指向已存在文件的目录项，称作链接（link）。

以图10-24a为例，两个用户Fred和Lisa一起工作来完成一个项目，他们需要访问对方的文件。如果Fred的工作目录是/usr/fred，他可以使用/usr/lisa/x来访问Lisa目录下的文件x。Fred也可以如图10-24b所示的方法，在自己目录下创建一个链接，然后他就可以用x来代替/usr/lisa/x了。

在上面的例子中，我们说在创建链接之前，Fred引用Lisa的文件x的惟一方法是使用绝对路径。实际上这并不正确，当一个目录被创建出来时，有两个目录项“.”和“..”被自动创建出来存放在该目录中，前者代表工作目录自身，而后者表示该目录的父目录，也就是该目录所在的目录。这样一来，在/usr/fred目录中访问Lisa的文件x的另一个路径是：../lisa/x。

除了普通的文件之外，Linux还支持字符特殊文件和块特殊文件。字符特殊文件用来建模串行I/O设备，比如键盘和打印机。如果打开并从/dec/tty中读取内容，等于从键盘读取内容，而如果打开并向/dev/lp中写内容，等于向打印机输出内容。块特殊文件通常有类似于/dev/hd1的文件名，它用来直接向硬盘分区中读取和写入内容，而不需要考虑文件系统。一个偏移为k字节的read操作，将会从相应分区开始的第k个字节开始读取，而完全忽略i节点和文件的结构。原始块设备常被一些建立（如mkfs）或修补（如fsck）文件系统的程序用来进行分页和交换。

许多计算机有两块或更多的磁盘。银行使用的大型机，为了存储大量的数据，通常需要在一台机器上安装100个或更多的磁盘。甚至在PC上也至少有两块磁盘——一块硬盘和一个光盘驱动器（如DVD）。当一台机器上安装了多个磁盘的时候，如何处理它们就是一个问题。

一个解决方法是在每一个磁盘上安装自包含的文件系统，使它们之间互相独立。考虑如图10-25a所示的解决方法，有一个硬盘C:和一个DVD D:，它们都有自己的根目录和文件。如果使用这种解决方法，除了默认盘外，使用者必须指定设备和文件，例如，要把文件x复制到目录d中（假设C:是默认盘），应该使用命令

```
cp D:/x /a/d/x
```

这种方法被许多操作系统使用，包括MS-DOS、Windows 98和VMS。

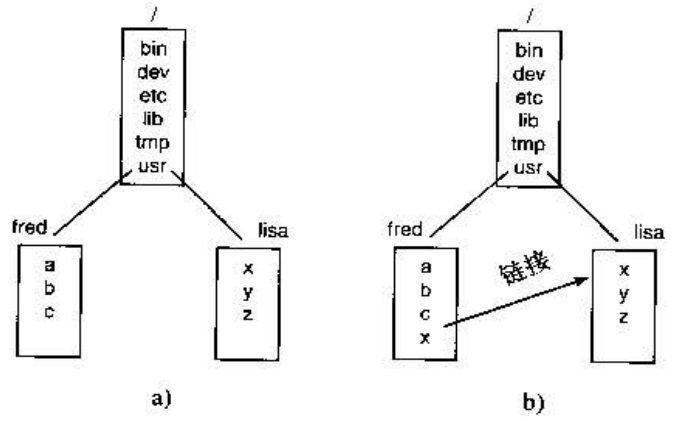

图10-24 a) 链接前；b) 链接后

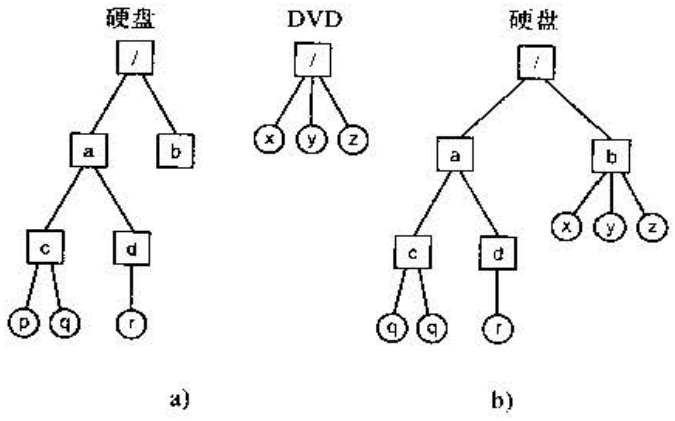

图10-25 a) 分离的文件系统；b) 挂载之后

Linux的解决方法是允许一个磁盘挂载到另一个磁盘的目录树上，比如，我们可以把DVD挂载在目录/b上，构成如图10-25b所示的文件系统。挂载之后，用户能够看见一个目录树，而不再需要关心文件