

2. 目录层次

文件只是所涉及的问题中的一部分。另一部分问题是目录系统。所有的分布式系统都支持有多个文件的目录。接下来的设计问题是，是否所有的用户都拥有该目录层次的相同视图。图8-37中的例子正好表达了我们的意思。在图8-37a中有两个文件服务器，每个服务器有三个目录和一些文件。在图8-37b中有一个系统，其中所有的客户（以及其他机器）对该分布式文件系统拥有相同的视图。如果在某台机器上路径/D/E/x是有效的，则该路径对所有其他的客户也是有效的。

相反，在图8-37c中，不同的机器有该文件系统的不同视图。重复先前的例子，路径/D/E/x可能在客户机1上有效，但是在客户机2上无效。在通过远程安装方式管理多个文件服务器的系统中，图8-37c是一个典型示例。这样既灵活又可直接实现，但是其缺点是，不能使得整个系统行为像单一的、旧式分时系统。在分时系统中，文件系统对任何进程都是一样的，如图8-37b中的模型。这个属性显然使得系统容易编程和理解。

一个密切相关的问题是，是否存在一个所有的机器都承认的全局根目录。获得全局根目录的一个方法是，让每个服务器的根目录只包含一个目录项。在这种情况下，路径取 /server/path 的形式，这种方式有其缺点，但是至少做到了在系统中处处相同。

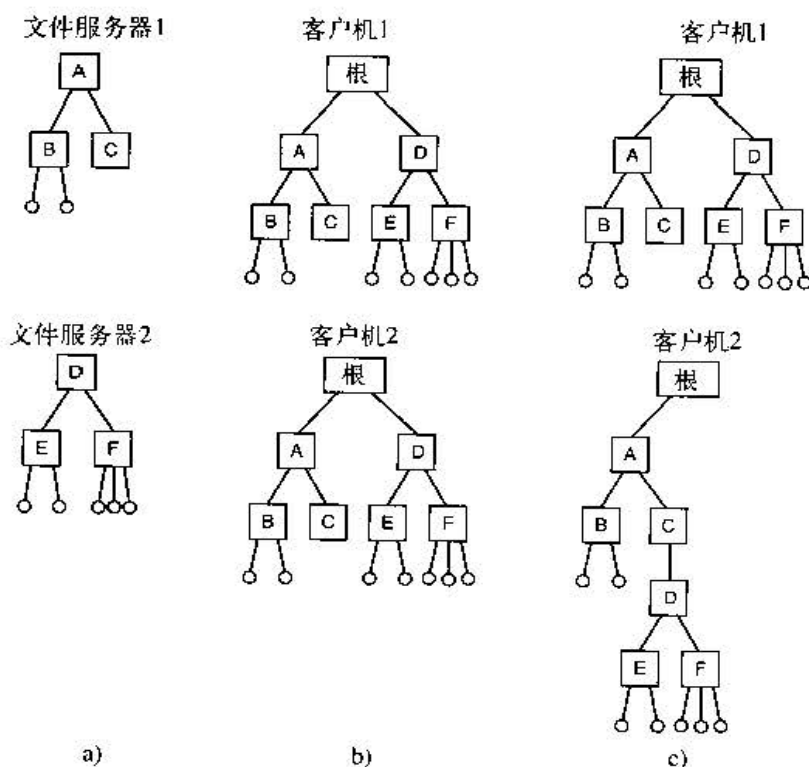


图8-37 a) 两个文件服务器。矩形代表目录，圆圈代表文件；b) 所有客户机都有相同文件系统视图的系统；c) 不同的客户机可能会有不同文件系统视图的系统

3. 命名透明性

这种命名方式的主要问题是，它不是完全透明的。这里涉及两种类型的透明性 (transparency)，并且有必要加以区分。第一种，位置透明性 (location transparency)，其含义是路径名没有隐含文件所在位置的信息。类似于 /server1/dir1/dir2/x 的路径告诉每个人，x 是在服务器1上，但是并没有说明该服务器在哪里。在网络中该服务器可以随意移动，而该路径名却不必改动。所以这个系统具有位置透明性。

但是，假设文件非常大而在服务器1上的空间又很紧张。进而，如果在服务器2上有大量的空间，那么系统也许会自动地将x从1移到服务器2上。不幸地，当整个路径名的第一个分量是服务器时，即使 dir1 和 dir2 在两个服务器上都存在，系统也不能将文件自动地移动到其他的服务器上。问题在于，让文件自动移动就得将其路径名从 /server1/dir1/dir2/x 改变成为 /server2/dir1/dir2/x。如果路径改变了，那么在内部拥有前一个路径字符串的程序就会停止工作。如果在一个系统中文件移动时文件的名称不会随之