



图4-8 UNIX目录树

4.2.4 目录操作

不同系统中管理目录的系统调用的差别比管理文件的系统调用的差别大。为了了解这些系统调用有哪些及它们怎样工作，下面给出一个例子（取自UNIX）。

1) **create**。创建目录。除了目录项“.”和“..”外，目录内容为空。目录项“.”和“..”是系统自动放在目录中的（有时通过mkdir程序完成）。

2) **delete**。删除目录。只有空目录可删除。只包含目录项“.”和“..”的目录被认为是空目录，这两个目录项通常不能删除。

3) **opendir**。目录内容可被读取。例如，为列出目录中全部文件，程序必须先打开该目录，然后读其中全部文件的文件名。与打开和读文件相同，在读目录前，必须打开目录。

4) **closedir**。读目录结束后，应关闭目录以释放内部表空间。

5) **readdir**。系统调用readdir返回打开目录的下一个目录项。以前也采用read系统调用来读目录，但这方法有一个缺点：程序员必须了解和处理目录的内部结构。相反，不论采用哪一种目录结构，readdir总是以标准格式返回一个目录项。

6) **rename**。在很多方面目录和文件都相似。文件可换名，目录也可以。

7) **link**。连接技术允许在多个目录中出现同一个文件。这个系统调用指定一个存在的文件和一个路径名，并建立从该文件到路径所指名字的连接。这样，可以在多个目录中出现同一个文件。这种类型的连接，增加了该文件的i节点（i-node）计数器的计数（记录含有该文件的目录项数目），有时称为硬连接（hard link）。

8) **unlink**。删除目录项。如果被解除连接的文件只出现在一个目录中（通常情况），则将它从文件系统中删除。如果它出现在多个目录中，则只删除指定路径名的连接，依然保留其他路径名的连接。在UNIX中，用于删除文件的系统调用（前面已有论述）实际上就是unlink。

最主要的系统调用已在上面列出，但还有其他一些调用，如与目录相关的管理保护信息的系统调用。

关于连接文件的一种不同想法是符号连接。不同于使用两个文件名指向同一个内部数据结构来代表一个文件，所建立的文件名指向了命名另一个文件的小文件。当使用第一个文件时，例如打开时，文件系统沿着路径，找到在末端的名字。然后它使用该新名字启动查找进程。符号连接的优点在于它能够跨越磁盘的界限，甚至可以命名在远程计算机上的文件，不过符号连接的实现并不如硬连接那样有效率。