

件中只有少量字节。当到达文件尾部时，该参数的值是0。如果rd_count是零或负数，复制工作就不能再进行下去，所以执行break语句，用以中断循环（否则就无法结束了）。

调用write把缓冲区的内容输出到目标文件中。第一个参数标识文件，第二个参数指定缓冲区，第三个参数指定写入多少字节，同read类似。注意字节计数是实际读出的字节数，不是BUF_SIZE。这一点是很重要的，因为最后一个缓冲区一般不会是4096，除非文件长度碰巧是4KB的倍数。

当整个文件处理完时，超出文件尾部的首次调用会把0值返回给rd_count，这样，程序会退出循环。此时，关闭两个文件，程序退出并附有正常完成的状态码。

尽管Windows的系统调用与UNIX的系统调用不同，但是Windows程序复制文件的命令行的一般结构与图4-5中的相当类似。我们将在第11章中考察Windows Vista的系统调用。

4.2 目录

文件系统通常提供目录或文件夹用于记录文件，在很多系统中目录本身也是文件。本节讨论目录、目录的组成、目录的特性和可以对目录进行的操作。

4.2.1 一级目录系统

目录系统的最简单形式是在一个目录中包含所有的文件。这有时称为根目录，但是由于只有一个目录，所以其名称并不重要。在早期的个人计算机中，这种系统很普遍，部分原因是因为只有一个用户。有趣的是，世界第一台超级计算机CDC 6600对于所有的文件也只有一个目录，尽管该机器同时被许多用户使用。这样决策毫无疑问是为了使软件设计简单。

一个单层目录系统的例子如图4-6所示。该目录中有四个文件。这一设计的优点在于简单，并且能够快速定位文件——事实上只有一个地方要查看。这种目录系统经常用于简单的嵌入式装置中，诸如电话、数码相机以及一些便携式音乐播放器等。

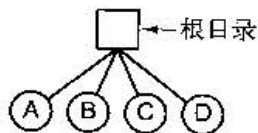


图4-6 含有四个文件的单层目录系统

4.2.2 层次目录系统

对于简单的特殊应用而言，单层目录是合适的（单层目录甚至用在了第一代个人计算机中），但是现在的用户有着成千的文件，如果所有的文件都在一个目录中，寻找文件就几乎不可能了。这样，就需要有一种方式将相关的文件组合在一起。例如，某个教授可能有一些文件，第一组文件是为了一门课程而写作的，第二组文件包含了学生为另一门课程所提交的程序，第三组文件是他构造的一个高级编译-写作系统的代码，而第四组文件是奖学金建议书，还有其他与电子邮件、短会、正在写作的文章、游戏等有关的文件。

这里所需要的是层次结构（即，一个目录树）。通过这种方式，可以用很多目录把文件以自然的方式分组。进而，如果多个用户分享同一个文件服务器，如许多公司的网络系统，每个用户可以为自己的目录树拥有自己的私人根目录。这种方式如图4-7所示，其中，根目录含有目录A、B和C，分别属于不同用户，其中有两个用户为他们的项目创建了子目录。

用户可以创建任意数量的子目录，这种能力为用户组织其工作提供了强大的结构化工具。因此，几乎所有现代文件系统都是用这个方式组织的。

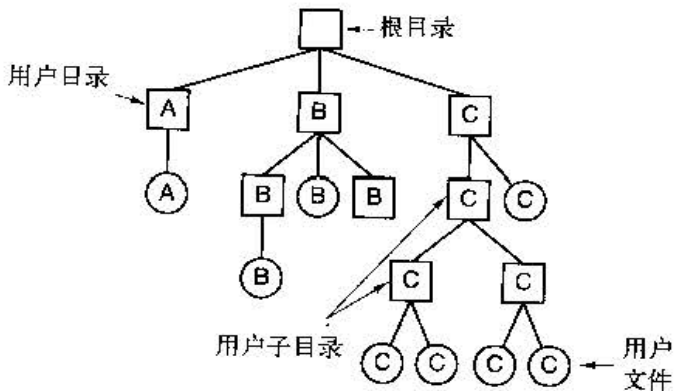


图4-7 层次目录系统

4.2.3 路径名

用目录树组织文件系统时，需要有某种方法指明文件名。常用的方法有两种。第一种是，每个文件都赋予一个绝对路径名（absolute path name），它由从根目录到文件的路径组成。例如，路径