

情形下，目录项会更短：只有文件名和节点号。这种方法参见图4-14b。后面我们会看到，与把属性存放到目录项中相比，这种方法更好。图4-14中的两种处理方法分别对应Windows和UNIX，在后面我们将讨论它们。

到目前为止，我们已经假设文件具有较短的、固定长度的名字。在MS-DOS中，文件有1~8个字符的基本名和1~3字符的可选扩展名。在UNIX V7中文件名有1~14个字符，包括任何扩展名。但是，几乎所有的现代操作系统都支持可变长度的长文件名。那么它们是如何实现的呢？

最简单的方法是给予文件名一个长度限制，典型值为255个字符，然后使用图4-14中的一种设计，并为每个文件名保留255个字符空间。这种处理很简单，但是浪费了大量的目录空间，因为只有很少的文件会有如此长的名字。从效率考虑，我们希望有其他的结构。

一种替代方案是放弃“所有目录项大小一样”的想法。这种方法中，每个月录项有一个固定部分，这个固定部分通常以目录项的长度开始，后面是固定格式的数据，通常包括所有者、创建时间、保护信息以及其他属性。这个固定长度的头的后面是实际文件名，可能是如图4-15a中的正序格式放置（如SPARC机器）^①。在这个例子中，有三个文件，project-budget、personnel和foo。每个文件名以一个特殊字符（通常是0）结束，在图4-15中用带叉的矩形表示。为了使每个目录项从字的边界开始，每个文件名被填充成整数个字，如图4-15中带阴影的矩形所示。

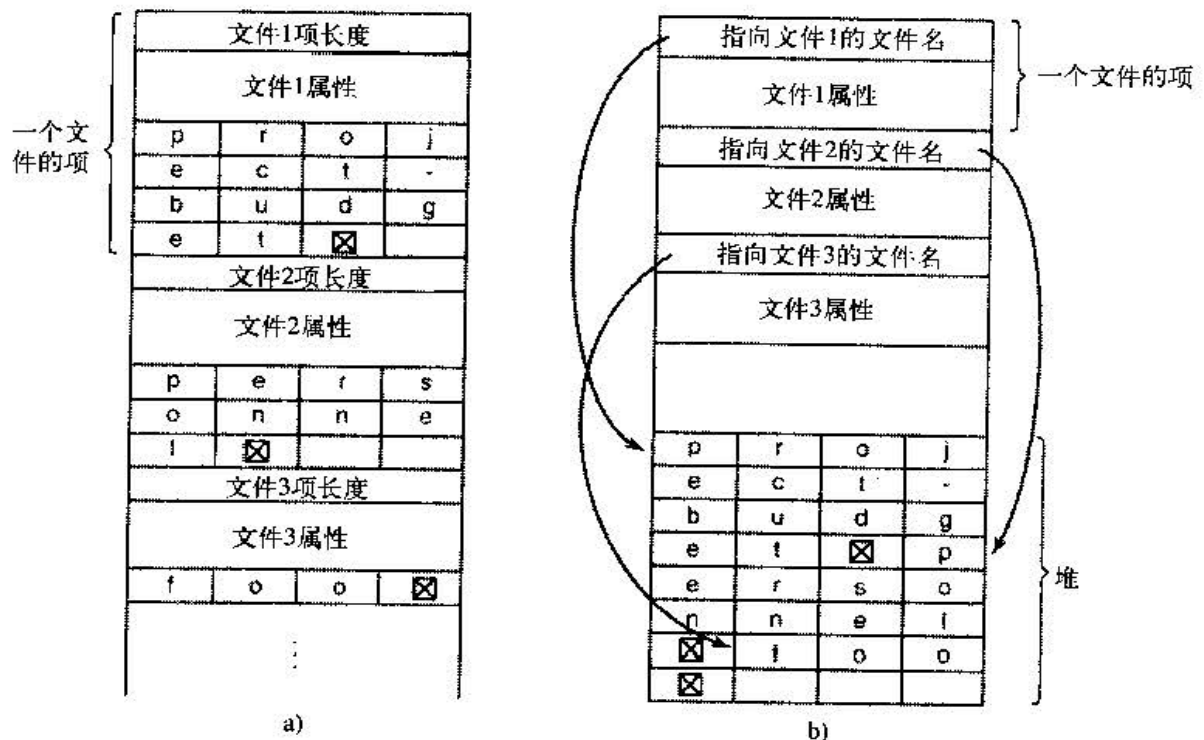


图4-15 在目录中处理长文件名的两种方法：a) 在行中；b) 在堆中

这个方法的缺点是，当移走文件后，就引入了一个长度可变的空隙，而下一个进来的文件不一定正好适合这个空隙。这个问题与我们已经看到的连续磁盘文件的问题是一样的，由于整个目录在内存中，所以只有对目录进行紧凑操作才可节省空间。另一个问题是，一个目录项可能会分布在多个页面上，在读取文件名时可能发生页面故障。

处理可变长度文件名字的另一方法是，使目录项自身都有固定长度，而将文件名放置在目录后面的堆中，如图4-15b所示。这一方法的优点是，当一个文件目录项被移走后，另一个文件的目录项总是可以适合这个空隙。当然，必须对堆进行管理，而在处理文件名时页面故障仍旧会发生。另一个小优

① 处理机中的一串字符存放的顺序有正序 (big-endian) 和逆序 (little-endian) 之分。正序存放就是高字节存放在前低字节在后，而逆序存放就是低字节在前高字节在后。例如，十六进制数为A02B，正序存放就是A02B，逆序存放就是2BA0。——译者注