

两个位图分别记录空闲块和空闲i节点，这是从MINIX1文件系统继承的（大多数UNIX文件系统不使用位图，而使用空闲列表）。每一个位图的大小是一个块。如果一个块大小是1KB，那么就限制了块数和i节点数只能是8192个。块数是一个严格的限制，但是在实际应用中，i节点数并不是。

在超级块之后是i节点存储区域，它们被编号为1到某个最大值。每个i节点的大小是128字节，并且每一个i节点恰好描述一个文件。i节点包含了统计信息（包含了stat系统调用能获得的所有信息，实际上stat就是从i节点读取信息的），也包含了所有存放该文件数据的磁盘块的位置。

在i节点区后面是数据块区，所有文件和目录都存放在这个区域。对于一个包含了一个以上磁盘块的文件和目录，这些磁盘块是不需要连续的。实际上，一个大文件的块有可能遍布在整个磁盘上。

目录对应的i节点散布在磁盘块组中。如果有足够的空间，ext2会把普通文件组织到与父目录相同的块组上，而把同一个块上的数据文件组织成初始文件i节点。这个思想来自Berkeley的快速文件系统（McKusick等人，1984）。位图用于快速确定在什么地方分配新的文件系统数据。在分配新的文件块时，ext2也会给该文件预分配许多（8个）额外的数据块，这样可以减少将来向该文件写入数据时产生的文件碎片。这种策略在整个磁盘上实现了文件系统负载平衡，而且由于排列和缩减文件碎片，它的性能也很好。

要访问文件，必须首先使用一个Linux系统调用，例如open，该调用需要文件的路径名。解析路径名以解析出单独的目录。如果使用相对路径，则从当前进程的当前目录开始查找，否则就从根目录开始。在以上两种情况中，第一个目录的i节点很容易定位：在进程描述符中有指向它的指针；或者在使用根目录的情况下，它存储在磁盘上预定的块上。

目录文件允许不超过255个字符的文件名，如图10-32所示。每一个目录都由整数个磁盘块组成，这样目录就可以整体写入磁盘。在一个目录中，文件和子目录的目录项是未排序的，并且一个紧挨着一个。目录项不能跨越磁盘块，所以通常在每个磁盘块的尾部会有部分未使用的字节。

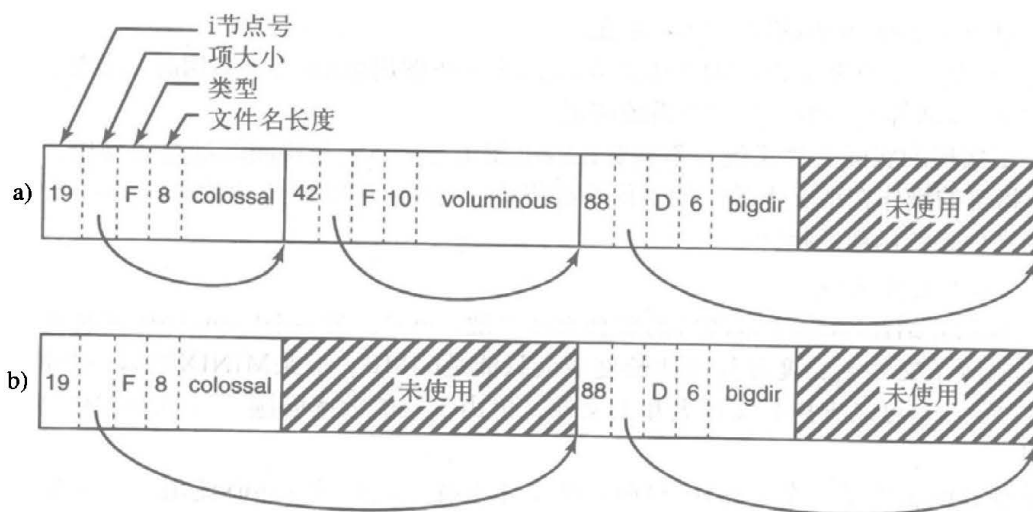


图10-32 a) 一个含有三个文件的Linux目录；b) 文件voluminous被删除后的目录

图10-32中的每个目录项由四个固定长度的域和一个可变长度的域组成。第一个域是i节点号，文件colossal的i节点号是19，文件voluminous的i节点号是42，目录bigdir的i节点号是88。接下来是rec_len域，标明该目录项的大小（以字节为单位），可能包括名字后面的一些填充。在名字以未知长度填充时，这个域被用来寻找下一个目录项。这也是图10-32中箭头的含义。接下来是类型域：文件、目录等。最后一个固定域是文件名的长度（以字节为单位），在例子中是8、10和6。最后是文件名，文件名以字节0结束，并被填充到32字节边界。额外的填充可以在此之后。

在图10-32b中，我们看到的是文件voluminous的目录项被移除后同一个目录的内容。这是通过增加colossal的域的长度，将voluminous以前所在的域变为第一个目录项的填充。当然，这个填充可以用来作为后续的目录项。

由于目录是按线性顺序查找的，要找到一个位于大目录末尾的目录项会耗费相当长的时间。因此，系统为近期访问过的目录维护一个缓存。该缓存使用文件名进行查找，如果命中，那么就可以避免费时