

块放在一起,当然最好是在同一个柱面上,从而减少磁盘臂的移动次数。当写一个输出文件时,文件系统就必须按照要求一次一次地分配磁盘块。如果用位图来记录空闲块,并且整个位图在内存中,那么选择与前一块最近的空闲块是很容易的。如果用空闲表,并且链表的一部分存在磁盘上,要分配紧邻着的空闲块就困难得多。

不过,即使采用空闲表,也可以采用块簇技术。这里用到一个小技巧,即不用块而用连续块簇来跟踪磁盘存储区。如果一个扇区有512个字节,有可能系统采用1KB的块(2个扇区),但却按每2块(4个扇区)一个单位来分配磁盘存储区。这和2KB的磁盘块并不相同,因为在高速缓存中它依然使用1KB的块,磁盘与内存数据之间传送也是以1KB为单位进行,但在一个空闲的系统上顺序读取文件,寻道的次数可以减少一半,从而使文件系统的性能大大改善。若考虑旋转定位则可以得到这类方案的变体。在分配块时,系统尽量把一个文件中的连续块存放在同一柱面上。

在使用i节点或任何类似i节点的系统,另一个性能瓶颈是,读取一个很短的文件也需要两次磁盘访问:一次是访问i节点,另一次是访问块。通常情况下,i节点的放置如图4-29a所示。其中,全部i节点都放在靠近磁盘开始位置,所以i节点和它指向的块之间的平均距离是柱面数的一半,这将需要较长的寻道时间。

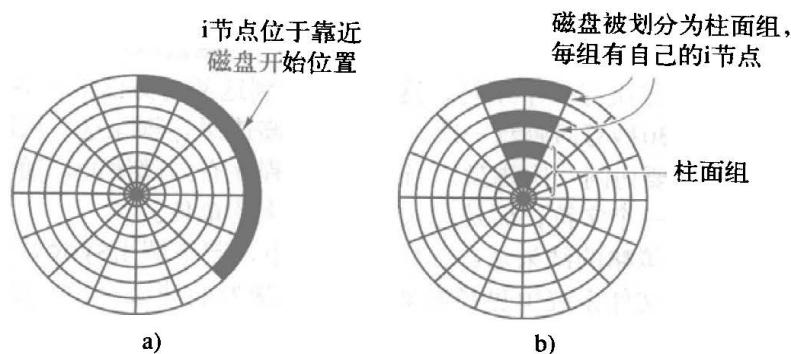


图4-29 a) i节点放在磁盘开始位置; b) 磁盘分为柱面组, 每组有自己的块和i节点

一个简单的改进方法是,在磁盘中部而不是开始处存放i节点,此时,在i节点和第一块之间的平均寻道时间减为原来的一半。另一种做法是:将磁盘分成多个柱面组,每个柱面组有自己的i节点、数据块和空闲表(McKusick 等人,1984),见图4-29b。在文件创建时,可选取任一i节点,但首先在该i节点所在的柱面组上查找块。如果在该柱面组中没有空闲的块,就选用与之相邻的柱面组的一个块。

#### 4.4.5 磁盘碎片整理

在初始安装操作系统后,从磁盘的开始位置,一个接一个地连续安装了程序与文件。所有的空闲磁盘空间放在一个单独的、与被安装的文件邻近的单元里。但随着时间的流逝,文件被不断地创建与删除,于是磁盘会产生很多碎片,文件与空穴到处都是。结果是,当创建一个新文件时,它使用的块会散布在整个磁盘上,造成性能的降低。

磁盘性能可以通过如下方式恢复:移动文件使它们相邻,并把所有的(至少是大部分的)空闲空间放在一个或多个大的连续的区域。Windows有一个程序defrag就是从事这个工作的。Windows的用户应该定期使用它。

磁盘碎片整理程序会在一个在分区末端的连续区域内有适量空闲空间的文件系统上很好地运行。这段空间会允许磁盘碎片整理程序选择在分区开始端的碎片文件,并复制它们所有的块放到空闲空间内。这个动作在磁盘开始处释放出一个连续的块空间,这样原始或其他的文件可以在其中相邻地存放。这个过程可以在下一大块的磁盘空间上重复,并继续下去。

有些文件不能被移动,包括页文件、休眠文件以及日志,因为移动这些文件所需的管理成本要大于移动它们的价值。在一些系统中,这些文件是固定大小的连续的区域,因此它们不需要进行碎片整理。这类文件缺乏灵活性会造成一些问题,一种情况是,它们恰好在分区的末端附近并且用户想减小分区的大小。解决这种问题的惟一的方法是把它们一起删除,改变分区的大小,然后再重新建立它们。