

连续磁盘空间分配方案有两大优势。首先,实现简单,记录每个文件用到的磁盘块简化为只需记住两个数字即可:第一块的磁盘地址和文件的块数。给定了第一块的编号,一个简单的加法就可以找到任何其他块的编号。

其次,读操作性能较好,因为在单个操作中就可以从磁盘上读出整个文件。只需要一次寻找(对第一个块)。之后不再需要寻道和旋转延迟,所以,数据以磁盘全带宽的速率输入。可见连续分配实现简单且具有较高的性能。

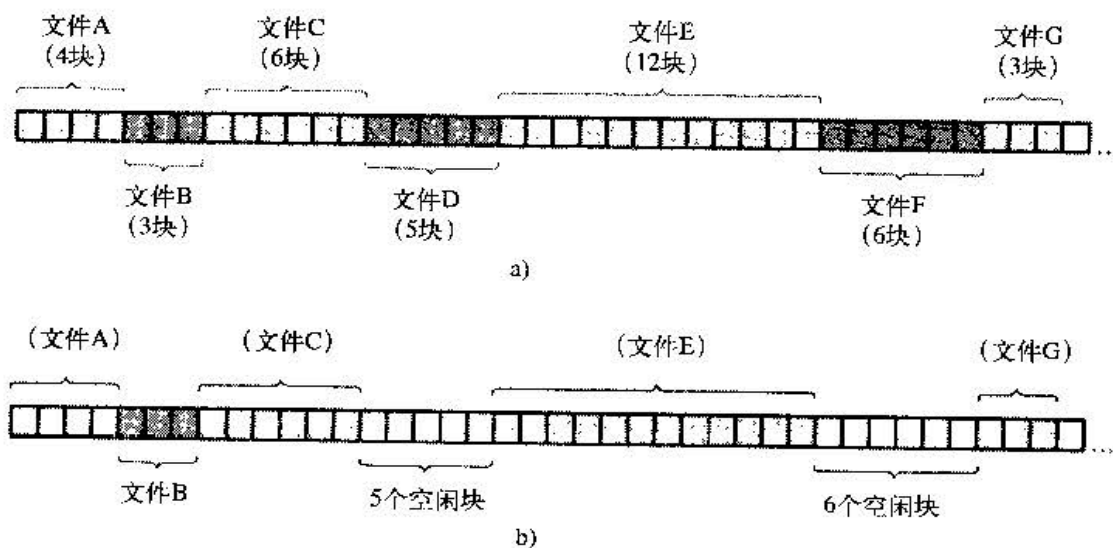


图4-10 a) 为7个文件连续分配空间; b) 删除文件D和F后磁盘的状态

但是,连续分配方案也同样有相当明显的不足之处:随着时间的推移,磁盘会变得零碎。为了了解这是如何发生的,请考察图4-10b。这里有两个文件(D和F)被删除了。当删除一个文件时,它占用的块自然就释放了,在磁盘上留下一堆空闲块。磁盘不会在这个位置挤压掉这个空洞,因为这样会涉及复制空洞之后的所有文件,可能会有上百万的块。结果是,磁盘上最终既包括文件也有空洞,如图4-10中所描述的那样。

开始时,碎片并不是问题,因为每个新的文件都在先前文件的磁盘结尾写入。但是,磁盘最终会被充满,所以要么压缩磁盘,要么重新使用空洞中的空闲空间。前者由于代价太高而不可行;后者需要维护一个空洞列表,这是可行的。但是,当创建一个新的文件时,为了挑选合适大小的空洞存入文件,就有必要知道该文件的最终大小。

设想这样一种设计的结果:为了录入一个文档,用户启动了文本编辑器或字处理软件。程序首先询问最终文件的大小会是多少。这个问题必须回答,否则程序就不能继续。如果给出的数字最后被证明小于文件的实际大小,该程序会终止,因为所使用的磁盘空洞已经满了,没有地方放置文件的剩余部分。如果用户为了避免这个问题而给出不实际的较大的数字作为最后文件的大小,比如,100 MB,编辑器可能找不到如此大的空洞,从而宣布无法创建该文件。当然,用户有权下一次使用比如50MB的数字再次启动编辑器,如此进行下去,直到找到一个合适的空洞为止。不过,这种方式看来不会使用户高兴。

然而,存在着一种情形,使得连续分配方案是可行的,而且,实际上这个办法在CD-ROM上被广泛使用着。在这里所有文件的大小都事先知道,并且在CD-ROM文件系统的后续使用中,这些文件的大小也不再改变。在本章的后面,我们将讨论最常见的CD-ROM文件系统。

DVD的情况有些复杂。原则上,一个90分钟的电影可以编码成一个独立的、大约4.5GB的文件。但是文件系统所使用的UDF (Universal Disk Format) 格式,使用了一个30位的数来代表文件长度,从而把文件大小限制在1GB。其结果是,DVD电影一般存储在3个或4个1GB的连续文件中。这些构成一个逻辑文件(电影)的物理文件块被称作extents。

正如第1章中所提到的,在计算机科学中,随着新一代技术的出现,历史往往重复着自己。多年前,连续分配由于其简单和高性能(没有过多考虑用户友好性)被实际用在磁盘文件系统中。后来由于讨厌