

存放电影的另一个方法是使用大磁盘块（比如256KB），并且在每一块中放入多个帧，如图7-20b所示。这里仍然需要一个索引，但是这次不是帧索引而是块索引。实际上，该索引与图6-15中的i节点基本相同，只是可能还有额外的信息表明哪一帧处于每一块的开始，这样就有可能快速地找到指定的帧。一般而言，一个磁盘块拥有的帧的数目不见得是整数，所以需要做一些机制来处理这一问题。解决这个问题有两种选择。

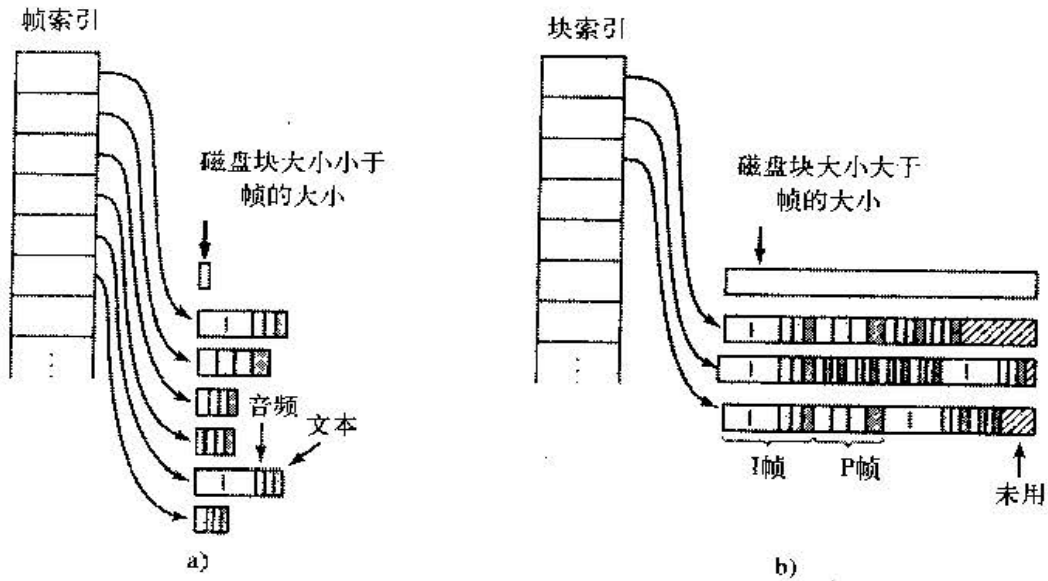


图7-20 不连续的电影存储：a) 小磁盘块；b) 大磁盘块

第一种选择如图7-20b所示，当下一帧填不满当前磁盘块的时候，则磁盘块剩余的部分就保持空闲状态。这一浪费的空间就是内部碎片，与具有固定大小页面的虚拟内存系统中的内部碎片相同。但是，这样做在一帧的中间决不需要进行寻道。

另一种选择是填充每一磁盘块到尽头，将帧分裂开使其跨越磁盘块。这一选择在帧的中间引入寻道的需要，这将损害性能，但是由于消除了内部碎片而节约了磁盘空间。

作为对比，图7-20a中小块的使用也会浪费某些磁盘空间，因为在每一帧的最后一块可能有一小部分未被使用。对于1KB的磁盘块和一部由216 000帧组成的2小时的NTSC电影，浪费的磁盘空间总共只有3.6GB中的大约108KB。图7-20b浪费的磁盘空间计算起来非常困难，但是肯定多很多，因为在一个磁盘块的尽头有时会留下100KB的空间，而下一帧是一个比它大的I帧。

另一方面，块索引比帧索引要小很多。对于256KB的块，如果帧的平均大小为16KB，那么一个块大约可以装下16个帧，所以一部由216 000帧组成的电影在块索引中只需要有13 500个索引项，与此相对比，对于帧索引则需要216 000个索引项。因为性能的原因，在这两种情形中索引都应该列出所有的帧或磁盘块（也就是说不像UNIX那样有间接块），所以块索引在内存中占用了13 500个8字节的项（4个字节用于磁盘地址，1个字节用于帧的大小，3个字节用于起始帧的帧号），帧索引则在内存中占用了216 000个5字节的项（只有磁盘地址和帧的大小），比较起来，当电影在播放时，块索引比帧索引节省了接近1MB的RAM空间。

这些考虑导出了如下的权衡：

- 1) 帧索引：电影在播放时使用大量的RAM；磁盘浪费小。
- 2) 块索引（禁止分裂帧跨越磁盘块）：RAM用量低；磁盘浪费较大。
- 3) 块索引（允许分裂帧跨越磁盘块）：RAM用量低；无磁盘浪费；需要额外寻道。

因此，这里的权衡涉及回放时RAM的使用量、自始至终浪费的磁盘空间以及由于额外寻道造成的回放时的性能损失。但是，这些问题可以用各种方法来解决。采用分页操作在需要的时候及时将帧索引装入内存，可以减少RAM的使用量。通过足够的缓冲可以屏蔽在帧传输过程中的寻道，但是这需要额外的内存并且可能还需要额外的复制操作。好的设计必须仔细分析所有这些因素，并且为即将投入的应