3. 文件压缩

NTFS支持透明的文件压缩。一个文件能够以压缩方式创建，这意味着当向磁盘中写入数据块时NTFS会自动尝试去压缩这些数据块，当这些数据块被读取时NTFS会自动解压。读或写的进程完全不知道压缩和解压在进行。

压缩流程是这样的：当NTFS写一个有压缩标志的文件到磁盘时，它检查这个文件的前16个逻辑块，而不管它们占用多少个项，然后对它们运行压缩算法，如果压缩后的数据能够存放在15个甚至更少的块中，压缩数据将写到硬盘中；如果可能的话，这些块在一个行串里。如果压缩后的数据仍然占用16个块，这16个块以不压缩方式写到硬盘中。之后，去检查第16-31块看是否能压缩到15个甚至更少的块，以此类推。

图11-46a显示一个文件。该文件的前16块被成功地压缩到了8个，对第二个16块的压缩没有成功，第三个16块也压缩了50%。这三个部分作为三个行串来写，并存储于MFT记录中。“丢失”的块用磁盘地址0存放在MFT表项中，如图11-46b所示。在图中，头（0，48）后面有五个二元组，其中，两个对应着第一个（被压缩）行串，一个对应没有压缩的行串，两个对应最后一个（被压缩）行串。

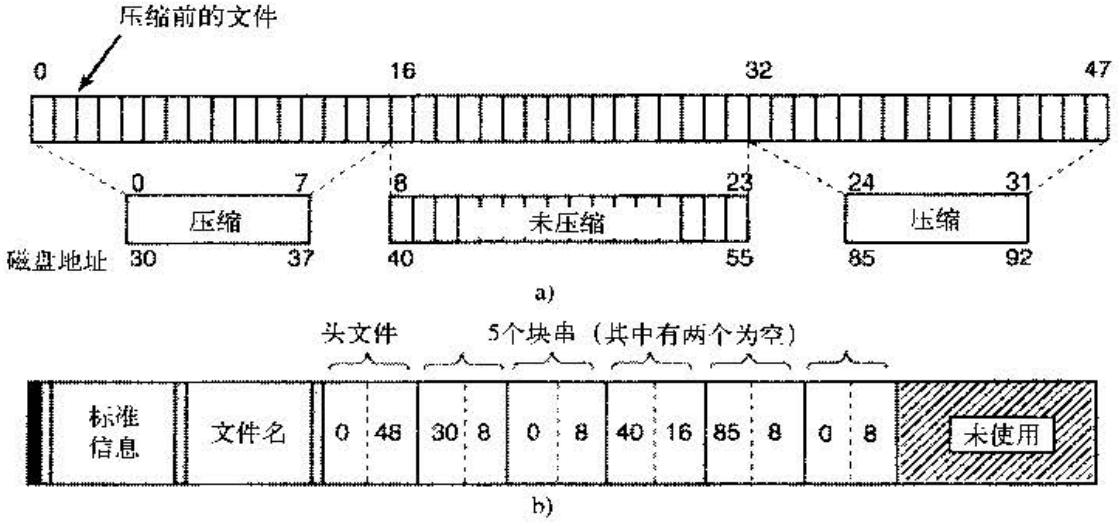

图11-46 a) 一个占48块的文件被压缩到32块的例子；b) 被压缩后文件对应的MFT记录

当读文件时，NTFS需要分辨某个行串是否被压缩过，它可以根据磁盘地址进行分辨，如果其磁盘地址是0，表明它是16个被压缩的块的最后部分。为了避免混淆，磁盘第0块不用于存储数据。同时，因为卷上的第0块包含了引导扇区，用它来存储数据也是不可能的。

随机访问压缩文件也是可行的，但是需要技巧。假设一个进程寻找图11-46中文件的第35块，NTFS是如何定位一个压缩文件的第35块区的呢？答案是NTFS必须首先读取并且解压整个行串，获得第35块的位置，之后就可以将该块传给读取它的进程。选择16个块作为压缩单元是一个折衷的结果，短了会影响压缩效率，长了则会使随机访问开销过大。

4. 日志

NTFS支持两种让程序探测卷上文件和目录变化的机制。第一种机制是调用名为NtNotifyChange Directory File的I/O操作，传递一个缓冲区给系统，当系统探测到目录或者子目录树变化时，该操作返回。这个I/O操作的结果是在缓冲区里填上变化记录的一个列表。缓冲区应该足够大，否则填不下的记录会被丢弃。

第二种机制是NTFS变化日志。NTFS将卷上的目录和文件的变化记录保存到一个特殊文件中，程序可以使用特殊文件系统控制操作来读取，即调用API NtFsControlFile并以FSCTL_ QUERY_USN_ JOURNAL为参数。日志文件通常很大，而且日志中的项在被检查之前重用的可能性非常小。

5. 文件加密

如今，计算机用来存储很多敏感数据，包括公司收购计划、税务信息、情书，数据的所有者不想把这些信息暴露给任何人。但是信息的泄漏是有可能发生的，例如笔记本电脑的丢失或失窃；使用MS-