

移量。在上面的例子中,第一个记录有一个(0, 50)的头,并会提供这50个块的磁盘地址。第二个记录有一个(60, 80)的头,会提供其他20个块的磁盘地址。

每个记录的头后面跟着一个或多个对,每个对给出了磁盘地址和持续长度。磁盘地址是该磁盘块离本分区起点的偏移量;游程在行串中块的数量。在一段行串记录中需要有多少对就可以有多少对。图11-43描述了用这种方式表示的三段、9块的流。

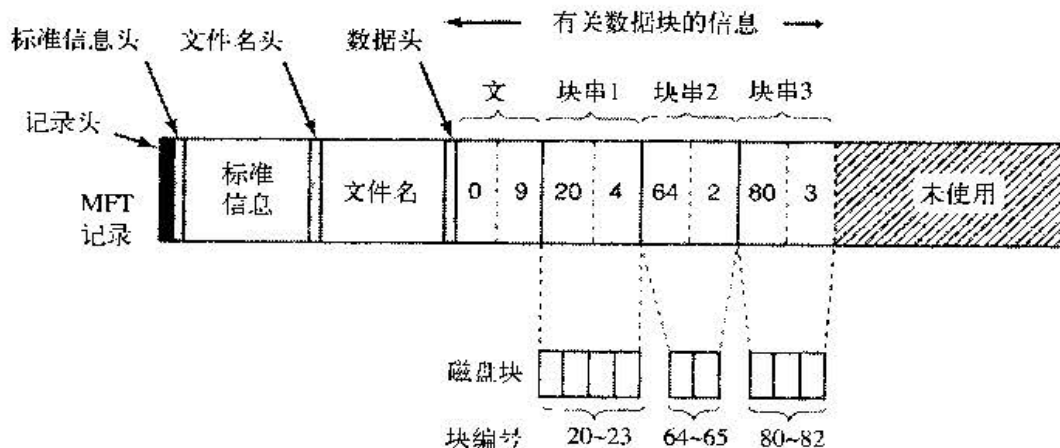


图11-43 有3个连续空间、9个块的短流的一条MFT记录

在这个图中,有一个9个块(头, 0~8)的短流的MFT记录。它由磁盘上三个行串连续块组成。第一段是块20~23,第二段是块64~65,第三段是80~82。每一个行串被记录在MFT记录中的一个(磁盘地址, 块计数)对中。有多少行串是依赖于当流被创建时磁盘块分配器在找连续块的行串时做的有多好。对于一个n块的流,段数可能是从1到n的任意值。

有必要在这里做几点说明:

首先,用这种方法表达的流的大小没有上限限制。在地址不压缩的情况下,每一对需要两个64位数表示,总共16字节。然而,一对能够表示100万个甚至更多的连续的磁盘空间。实际上,20M的流包含20个独立的包含100万个1KB的块的行串,每个都可以轻易地放在一个MFT记录中,然而一个60KB的被分散到60个不同的块的流却不行。

其次,表示每一对的直截了当的方法会占用 2×8 个字节,有压缩方法可以把一对的大减小到低于16字节。许多磁盘地址有多个高位0字节。这些可以被忽略。数据头能告诉我们有多少个高位0字节被忽略了,也就是说,在一个地址中实际上有多少个字节被用。也可以用其他的压缩方式。实际上,一对经常只有4个字节。

第一个例子是比较容易的:所有的文件信息能容纳在一个MFT记录中,如果文件比较大或者是高度碎片化以至于信息不能放在一个MFT记录当中,这时会发生什么呢?答案很简单:用两个或更多的MFT记录。从图11-44可以看出,一个文件的首MFT记录是102,对于一个MFT记录而言它有太多的行串,因而它会计算需要多少个扩展的MFT记录。比如说两个,于是会把它们的索引放到首记录中,首记录剩余的空间用来放前k个行串。

注意,图11-44包含了一些多余的信息。理论上不需要指出一个行串的结尾,因为这些信息可以从行串对中计算出来。列出这些信息是为了更有效地搜索:找到在一个给定文件偏移量的块,只需要去检查记录头,而不是行串对。

当MFT记录102中所有的空间被用完后,剩余的行串继续在MFT记录105中存放,并在这个记录中

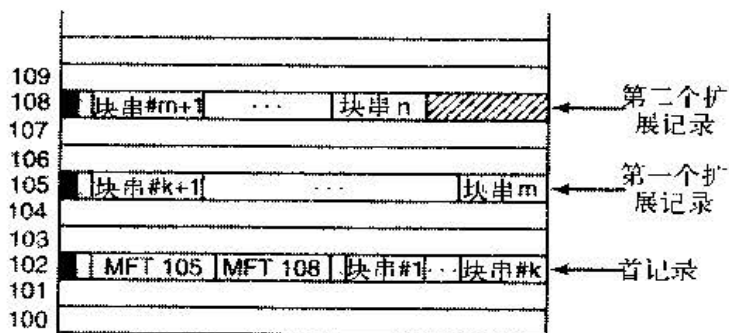


图11-44 需要三个MFT记录存储其所有行串的文件