

从转储磁带上恢复文件系统很容易办到。首先要在磁盘上创建一个空的文件系统，然后恢复最近一次的完整转储。由于磁带上最先出现目录，所以首先恢复目录，给出文件系统的框架；然后恢复文件本身。在完整转储之后的是增量转储，重复这一过程，以此类推。

尽管逻辑转储十分简单，还是有几点棘手之处。首先，既然空闲块列表并不是一个文件，那么在所有被转储的文件恢复完毕之后，就需要从零开始重新构造。这一点可以办到，因为全部空闲块的集合恰好是包含在全部文件中的块集合的补集。

另一个问题是关于连接。如果一个文件被连接到两个或多个目录中，要注意在恢复时只对该文件恢复一次，然后要恢复所有指向该文件的目录。

还有一个问题就是：UNIX文件实际上包含了许多“空洞”。打开文件，写几个字节，然后找到文件中一个偏移了一定距离的地址，又写入更多的字节，这么做是合法的。但两者之间的这些块并不属于文件本身，从而也不应该在其上实施转储和恢复操作。核心文件通常在数据段和堆栈段之间有一个数百兆字节的空洞。如果处理不得当，每个被恢复的核心文件会以“0”填充这些区域，这可能导致该文件与虚拟地址空间一样大（例如， $2^{32}$ 字节，更糟糕可能会达到 $2^{64}$ 字节）。

最后，无论属于哪一个目录（它们并不一定局限于/dev目录下），特殊文件、命名管道以及类似的文件都不应该转储。关于文件系统备份的更多信息，请参考（Chervenak 等人，1998；Zwicky，1991）。

磁带密度不会像磁盘密度那样改进得那么快。这会逐渐导致备份一个很大的磁盘需要多个磁带的状况。当磁带机器人可以自动换磁带时，如果这种趋势继续下去，作为一种备份介质，磁带会最终变得太小。在那种情况下，备份一个磁盘的惟一的方式是在另一个磁盘上。对每一个磁盘直接做镜像是一种方式。一个更加复杂的方案，称为RAID，将会在第5章讨论。

#### 4.4.3 文件系统的一致性

影响文件系统可靠性的另一个问题是文件系统的一致性。很多文件系统读取磁盘块，进行修改后，再写回磁盘。如果在修改过的磁盘块全部写回之前系统崩溃，则文件系统有可能处于不一致状态。如果一些未被写回的块是i节点块、目录块或者是包含有空闲表的块时，这个问题尤为严重。

为了解决文件系统的不一致问题，很多计算机都带有一个实用程序以检验文件系统的一致性。例如，UNIX有fsck，而Windows用scandisk。系统启动时，特别是崩溃之后的重新启动，可以运行该实用程序。下面我们介绍在UNIX中这个fsck实用程序是怎样工作的。scandisk有所不同，因为它工作在另一种文件系统上，不过运用文件系统的内在冗余进行修复的一般原理仍然有效。所有文件系统检验程序可以独立地检验每个文件系统（磁盘分区）的一致性。

一致性检查分为两种：块的一致性检查和文件的一致性检查。在检查块的一致性时，程序构造两张表，每张表中为每个块设立一个计数器，都初始化为0。第一个表中的计数器跟踪该块在文件中的出现次数，第二个表中的计数器跟踪该块在空闲表中的出现次数。

接着检验程序使用原始设备读取全部的i节点，忽略文件的结构，只返回所有的磁盘块，从0开始。由i节点开始，可以建立相应文件中采用的全部块的块号表。每当读到一个块号时，该块在第一个表中的计数器加1。然后，该程序检查空闲表或位图，查找全部未使用的块。每当在空闲表中找到一个块时，就会使它在第二个表中的相应计数器加1。

如果文件系统一致，则每一块或者在第一个表计数器中为1，或者在第二个表计数器中为1，如图4-27a所示。但是当系统崩溃后，这两张表可能如图4-27b所示，其中，磁盘块2没有出现在任何一张表中，这称为块丢失。尽管块丢失不会造成实际的损害，但它的确浪费了磁盘空间，减少了磁盘容量。块丢失问题的解决很容易：文件系统检验程序把它们加到空闲表中即可。

有可能出现的另一种情况如图4-27c所示。其中，块4在空闲表中出现了2次（只在空闲表是真正意义上的一张表时，才会出现重复，在位图中，不会发生这类情况）。解决方法也很简单：只要重新建立空闲表即可。

最糟的情况是，在两个或多个文件中出现同一个数据块，如图4-27d中的块5。如果其中一个文件被删除，块5会添加到空闲表中，导致一个块同时处于使用和空闲两种状态。若删除这两个文件，那么在空闲表中这个磁盘块会出现两次。