



图4-27 文件系统状态：a) 一致；b) 块丢失；c) 空闲表中有重复块；d) 重复数据块

文件系统检验程序可以采取相应的处理方法是，先分配一空闲块，把块5中的内容复制到空闲块中，然后把它插到其中一个文件之中。这样文件的内容未改变（虽然这些内容几乎可以肯定是不对的），但至少保持了文件系统的一致性。这一错误应该报告，由用户检查文件受损情况。

除检查每个磁盘块计数的正确性之外，文件系统检验程序还检查目录系统。此时也要用到一张计数器表，但这时是一个文件（而不是一个块）对应于一个计数器。程序从根目录开始检验，沿着目录树递归下降，检查文件系统每个目录。对每个目录中的每个文件，将文件使用计数器加1。要注意，由于存在硬连接，一个文件可能出现在两个或多个目录中。而遇到符号连接是不计数的，不会对目标文件的计数器加1。

在检验程序全部完成后，得到一张由i节点号索引的表，说明每个文件被多少个月录包含。然后，检验程序将这些数字与存储在文件i节点中的连接数目相比较。当文件创建时，这些计数器从1开始，随着每次对文件的一个（硬）连接的产生，对应计数器加1。如果文件系统一致，这两个计数应相等。但是，有可能出现两种错误，即i节点中的连接计数太大或者太小。

如果i节点的连接计数大于目录项个数，这时即使所有的文件都从目录中删除，这个计数仍是非0，i节点不会被删除。该错误并不严重，却因为存在不属于任何目录的文件而浪费了磁盘空间。为改正这一错误，可以把i节点中的连接计数设成正确值。

另一种错误则是潜在的灾难。如果同一个文件连接两个目录项，但其i节点连接计数只为1，如果删除了任何一个目录项，对应i节点连接计数变为0。当i节点计数为0时，文件系统标志该i节点为“未使用”，并释放其全部块。这会导致其中一个目录指向一未使用的i节点，而很有可能其块马上就被分配给其他文件。解决方法同样是把i节点中连接计数设为目录项的实际个数值。

由于效率上的考虑，以上的块检查和目录检查经常被集成到一起（即仅对i节点扫描一遍）。当然也有一些其他检查方法。例如，目录是有明确格式的，包含有i节点数目和ASCII文件名，如果某个月录的i节点编号大于磁盘中i节点的实际数目，说明这个目录被破坏了。

再有，每个i节点都有一个访问权限项。一些访问权限是合法的，但是很怪异，比如0007，它不允许文件所有者及所在用户组的成员进行访问，而其他的用户却可以读、写、执行此文件。在这类情况下，有必要报告系统已经设置了其他用户权限高于文件所有者权限这一情况。拥有1000多个目录项的目录也很可疑。为超级用户所拥有，但放在用户目录下，且设置了SETUID位的文件，可能也有安全问题，因为任何用户执行这类文件都需要超级用户的权限。可以列出一长串特殊的情况，尽管这些情况合法，但报告给用户却是必要的。

以上讨论了防止因系统崩溃而破坏用户文件的问题，某一些文件系统也防止用户自身的误操作。如果用户想输入

```
rm *.o
```

删除全部以.o结尾的文件（编译器生成的目标文件），但不幸键入了

```
rm * .o
```