https://mp.weixin.qq.com/s/Hbo40D9AFcyR96xdHi4H5g

# rm删除文件之后，空间就被释放了吗？

在Linux，你是不是曾经天真的以为，使用rm删除一个文件，占用的空间就释放了？事情可能不是常常如人意。

### **产生一个指定大小的随机内容文件**

我们先看一下当前各个挂载目录的空间大小：



我这里挑选了其中一个结果展示（你可以选择任一挂载目录），接下来准备在/boot下生成一个文件。

首先我们产生一个50M大小的文件：

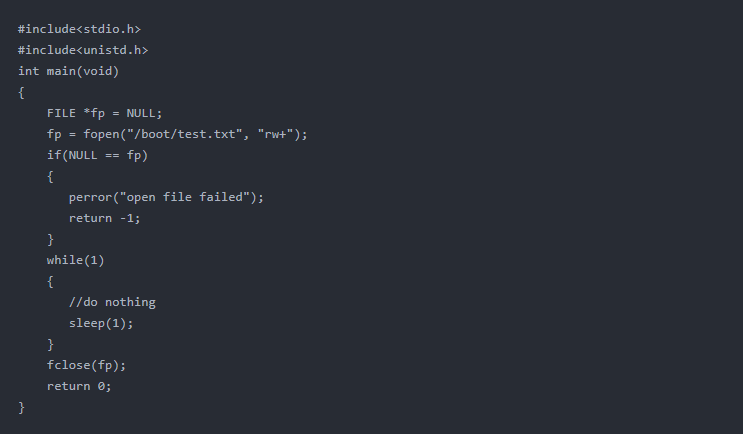


至此，我们产生了一个50M大小的文件，再看boot下：



这里你不用关心到底多了多少，你只需要关注，/boot下的文件增多了。

测试程序：



至于程序本身，也没干啥实际的事情，就是打开一个文件，然后一直循环。编译并运行：



打开另外一个窗口，删掉test.txt：



再看一下boot空间：



咦？空间大小怎么一点都没变！！明明使用rm把它删除了啊？

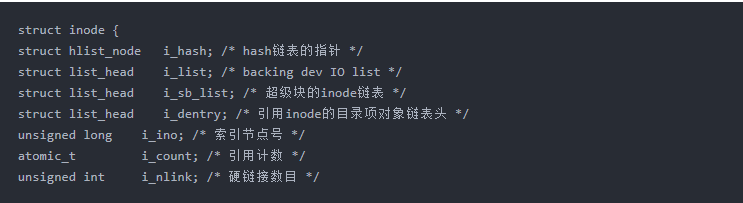
我们把openFile程序停掉，再看看：



乖乖，空间马上就释放掉了，也就是按照预期，我们的文件被删除了。

### **一个文件什么情况下才会被删除？**

实际上，只有当一个文件的引用计数为0（包括硬链接数）的时候，才可能调用unlink删除，只要它不是0，那么就不会被删除。所谓的删除，也不过是文件名到 inode 的链接删除，只要不被重新写入新的数据，磁盘上的block数据块不会被删除，因此，你会看到，即便删库跑路了，某些数据还是可以恢复的。换句话说，当一个程序打开一个文件的时候（获取到文件描述符），它的引用计数会被+1，rm虽然看似删除了文件，实际上只是会将引用计数减1，但由于引用计数不为0，因此文件不会被删除。



关于里面的细节，还有很多内容（如硬链接数量也会影响文件是否被删除），这里不一一展开。

### **如何释放已经被删除文件占用的空间？**

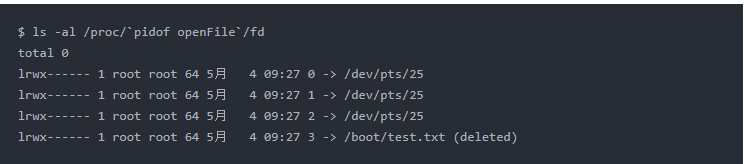
关于释放，前面已经说了，重启打开该文件的进程即可。但是有没有方法找到哪些文件被删除了，但还是被某些进程打开了呢？

自然是有方法的：



其中被标记为deleted的文件，就是这样的一些文件。

其实在前面的例子中，我们也可以很容易观察到(openFile程序运行，test.txt文件被删除)：



看见没有，test.txt后面还有deleted字样。

既然我们都说了，这样的情况下文件是没有被删除的，那么还能不能恢复呢？实际上还是可以读取的。

### **总结**

实际上对于这种文件被删除了，常常出现于程序的日志文件中，可能你有一个定时任务去清理程序产生的日志文件，但是如果程序本身忘记关闭句柄，就会导致磁盘空间得不到释放，最终就是你认为文件都被删除了，但是磁盘却依然被占着。所以，养成好习惯，打开文件后，不用时，记得关闭文件描述符。

如果发现明明已经删除了大量文件，但是空间却并没有恢复正常，那么不妨看看是不是还有程序打开了这些文件。