

探索linux系统中限制对目录作硬连接的原因 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/267/2021_2022__E6_8E_A2_E7_B4_A2linu_c103_267178.htm 我们都知道，linux系统中的硬连接有两个限制：不能跨越文件系统和不允许普通用户对目录作硬连接。至于第一个限制，很好理解，而第二个就不那么好理解了。我们对任何一个目录用ls -l命令都可以看到其连接数至少是2，这也说明了系统中是存在硬连接的，而且命令ln -d也可以让超级用户对目录作硬连接，这些都说明了系统限制对目录进行硬连接只是一个硬性规定，并不是逻辑上不允许或技术上的不可行。那么操作系统为什么要进行限制呢？答案可能有两个。先来说第一个，如果引入了对目录的硬连接就有可能在目录中引入循环，那么在目录遍历的时候系统就会陷入无限循环当中。也许您会说，符号连接不也可以引入循环吗，那么为什么不限制目录的符号连接呢？原因就在于在linux系统中，每个文件(目录也是文件)都对应着一个inode结构，其中inode数据结构中包含了文件类型(目录，普通文件，符号连接文件等等)的信息，也就是说操作系统在遍历目录时可以判断出符号连接，既然可以判断出符号连接当然就可以采取一些措施来防范进入过大的循环了，系统在连续遇到8个符号连接后就停止遍历，这就是为什么对目录符号连接不会进入死循环的原因了。但是对于硬连接，由于操作系统中采用的数据结构和算法限制，目前是不能防范这种死循环的。在说明第二个原因之前，我们先来看看文件的dentry结构在系统空间中长什么样子和它们是怎么存放在系统空间的。dentry结构主要包含了文件名，文件的inode号，

指向父目录dentry结构的指针和其他一些与本次讨论无关的指针，这里关键是那个指向父目录的指针；系统中所有的dentry结构都是按杂凑值存放在杂凑表中的，这里的杂凑算法很重要，它是取文件名和文件的父目录dentry结构的地址一起杂凑运算出杂凑值的。现在我们假设有两个目录 /a和/b，其中/b是我们通过ln -d命令建立起来的对/a的硬连接。这个时候内核空间中就会存在一个/a的dentry结构和一个/b的dentry结构，由上面的知识可知，/a和/b目录下面的每一个文件或目录都各自有对应的dentry结构(因为虽然/a目录下面的文件名没有改变，但是因为dentry结构有指向父目录dentry的指针和计算杂凑值时考虑了父目录dentry结构的地址，这个时候dentry结构就分身乏术了)，而且这种继承还会影响到所有子目录下面的文件，这样下来就会浪费很多系统空间了，特别是如果被硬连接的目录中存在大量文件和子目录的时候就更加明显了。这也许是第二个原因。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com