

个人资料



访问：307185次
积分：3341
等级：BLOG > 5
排名：第10599名

原创：58篇 转载：3篇
译文：0篇 评论：7条

文章搜索

文章分类

- 数据库-MySQL (41)
- 编程语言-GO (1)
- 编程语言-C (1)
- 编程语言-Python (1)
- WEB技术-HTML5 (1)
- 操作系统-Linux (10)
- 灾备与恢复 (1)
- 存储设备-IP_SAN共享存储 (3)
- 存储设备-FC_SAN共享存储 (1)
- 存储设备-NAS网络存储 (3)
- 虚拟化技术-vmware (1)
- 网络技术-交换机 (1)
- 网络技术-路由器 (1)
- 研发管理工具 (1)

文章存档

- 2015年02月 (2)
- 2015年01月 (4)
- 2014年12月 (8)
- 2014年11月 (15)
- 2014年10月 (13)

展开

阅读排行

- mysql运维-lower_case_t (20897)
- freenas搭建nas及san网 (15837)

异步赠书：9月重磅新书升级，本本经典 程序员9月书讯 每周荐书：ES6、虚拟现实、物联网（评论送书）

linux 索引节点 inode 详解

标签：inode

2015-01-08 14:46 2943人阅读 评论(1) 收藏 举报

分类：操作系统-Linux (9)

1 inode简介

理解inode，要从文件储存说起。
文件储存在硬盘上，硬盘的最小存储单位叫做"扇区" (Sector)。每个扇区储存512字节 (相当于0.5KB)。
操作系统读取硬盘的时候，不会一个个扇区地读取，这样效率太低，而是一次性连续读取多个扇区，即一次性读取一个"块" (block)。这种由多个扇区组成的"块"，是文件存取的最小单位。"块"的大小，最常见的是4KB，即连续八个 sector组成一个 block。

文件数据都储存在"块"中，那么很显然，我们还必须找到一个地方储存文件的元信息，比如文件的创建者、文件的创建日期、文件的大小等等。这种储存文件元信息的区域就叫做inode，中文译名为"索引节点"。

2 inode的内容

- inode包含文件的元信息，具体来说有以下内容：
- * 文件的字节数
 - * 文件拥有者的User ID
 - * 文件的Group ID
 - * 文件的读、写、执行权限
 - * 文件的时间戳，共有三个：ctime指inode上一次变动的时间，mtime指文件内容上一次变动的时间，atime指文件上一次打开的时间。
 - * 链接数，即有多少文件名指向这个inode
 - * 文件数据block的位置
- 可以用stat命令，查看某个文件的inode信息：

```
[plain]
01. [root@localhost /]# stat example.txt
```

总之，除了文件名以外的所有文件信息，都存在inode之中。至于为什么没有文件名，下文会有详细解释。

3 inode的大小

inode也会消耗硬盘空间，所以硬盘格式化的时候，操作系统自动将硬盘分成两个区域。一个是数据区，存放文件数据；另一个是inode区 (inode table)，存放inode所包含的信息。

每个inode节点的大小，一般是128字节或256字节。inode节点的总数，在格式化时就给定，一般是每1KB或每2KB就设置一个inode。假定在一块1GB的硬盘中，每个inode节点的大小为128字节，每1KB就设置一个inode，那么inode table的大小就会达到128MB，占整块硬盘的12.8%。

查看每个硬盘分区的inode总数和已经使用的数量，可以使用df命令。

```
[plain]
01. [root@localhost /]# df -i
```

关闭

MySQL事件 (Event) (13379)
mysql查询语句select-子 (13263)
mysql主从复制-CHANGE (13098)
存储与服务器的连接方式 (10583)
传输工具sftp使用详解 (10129)
cgroups介绍及安装配置 (9920)
go语言操作mysql范例 (: (9819)
mysql查询语句select- (r (8434)

评论排行

mysql运维-lower_case_t (2)
MySQL事件 (Event) (1)
cgroups介绍及安装配置 (1)
linux 索引节点 inode 详解 (1)
mysql用户变量和set语句 (1)
vSphere Esxi 5.0简介及 (1)
Linux服务器时间设置及 (0)
MySQL内存表 (0)
MySQL分区表 (0)
MySQL数据库备份还原 (0)

最新评论

mysql运维-lower_case_table_name
Kiven_Huang: 如果前面建表时 lower_case_table_names=0,后面修改lower_case_table_name...
MySQL事件 (Event)
雪叮头: 学习了。
mysql运维-lower_case_table_name
黄杉: lower_case_table_names, 个人还是觉得要大小写敏感才好啊。
linux 索引节点 inode 详解
Dablelv: 请问博主, linux中是如何根据文件名称找到对应的inode索引节点的?
mysql用户变量和set语句
ly_00001: mysql> set @varA = 3, @varB = @varA; mysql> selec...
cgroups介绍及安装配置使用详解
sinat_34001490: 在重启cgconfig的时候报了一个这样的错误, Error: failed to parse fil...
vSphere Esxi 5.0简介及安装配置
梦里茶: 博主你好, 在vsphere client 里安装了ubuntu后, 还需要设置什么才能上网呢?
sysbench-0.4, sysbench-0.5编译
jesseyoung: @lly520zhiyong:请参考: http://blog.csdn.net/jesseyoung...
sysbench-0.4, sysbench-0.5编译
jesseyoung: @lly520zhiyong:你好, 还没整理, 稍后添加, 谢谢关注
sysbench-0.4, sysbench-0.5编译
hello-world2016: 请问有没有PostgreSQL的测试例子呢?

```
[root@localhost ~]# df -h
Filesystem              Inodes   IUsed   IFree IUse% Mounted on
/dev/mapper/VolGroup-lv_root 3276800 143748 3133052 5% /
tmpfs                    126544   7       126537 1% /dev/shm
/dev/sda1                 128016   38      127978 1% /boot
/dev/mapper/VolGroup-lv_home 1199520  28      1199492 1% /home
/dev/mapper/VolGroup-LogVol03 1921360 27143 1894217 2% /opt
```

查看每个inode节点的大小, 可以用如下命令:

```
[plain]
01. [root@localhost ~]# dumpe2fs -h /dev/sda1 | grep "Inode size"
```

```
[root@localhost ~]# dumpe2fs -h /dev/sda1 | grep "Inode size"
dumpe2fs 1.41.12 (17-May-2010)
Inode size: 128
```

由于每个文件都必须有一个inode, 因此有可能发生inode已经用光, 但是硬盘还未存满的情况。这时, 就无法在硬盘上创建新文件。

4 inode号码

每个inode都有一个号码, 操作系统用inode号码来识别不同的文件。

这里值得重复一遍, Unix/Linux系统内部不使用文件名, 而使用inode号码来识别文件。对于系统来说, 文件名只是inode号码便于识别的别称或者绰号。表面上, 用户通过文件名, 打开文件。实际上, 系统内部这个过程分成三步: 首先, 系统找到这个文件名对应的inode号码; 其次, 通过inode号码, 获取inode信息; 最后, 根据inode信息, 找到文件数据所在的block, 读出数据。

使用ls -li命令, 可以看到文件名对应的inode号码:

```
[plain]
01. [root@localhost ~]# ls -li test.txt
02. 6970 test.txt
```

5 目录文件

Unix/Linux系统中, 目录 (directory) 也是一种文件。打开目录, 实际上就是打开目录文件。

目录文件的结构非常简单, 就是一系列目录项 (dirent) 的列表。每个目录项, 由两部分组成: 所包含文件的文件名, 以及该文件名对应的inode号码。

ls命令只列出目录文件中的所有文件名:

```
[plain]
01. [root@localhost ~]# ls /
02. backup boot etc lib lost+found misc net proc sbin srv tmp var
03. bin dev home lib64 media mnt opt root selinux sys usr
```

ls -li命令列出整个目录文件, 即文件名和inode号码:

```
[plain]
01. [root@localhost ~]# ls -li /
02. 2359297 backup 2 home 14372 misc 131073 root 262145 tmp
03. 393217 bin 262144 lib 1703937 mnt 1835009 sbin 3014657 usr
04. 2 boot 2490369 lib64 14376 net 1 selinux 1048577 var
05. 4 dev 11 lost+found 2 opt 917505 srv
06. 2097153 etc 524289 media 1 proc 1 sys
```

如果要查看文件的详细信息, 就必须根据inode号码, 访问inode节点, 读取信息。ls -li命令列出文件的详细信息。

```
[plain]
01. [root@localhost ~]# ls -li /
```

6 硬链接

一般情况下, 文件名和inode号码是"一一对应"关系, 每个inode号码对应一个文件名。但是, Unix/Linux系统, 允许多个文件名指向同一个inode号码。这意味着, 可以用不同的文件名访问同样的内容; 对文件内容进行修改, 会影响

关闭

影响到所有文件名；但是，删除一个文件名，不影响另一个文件名的访问。这种情况就被称为“硬链接”（hard link）。

ln命令可以创建硬链接：

```
[plain]
01. [root@localhost /]# ln 源文件 目标文件
```

运行上面这条命令以后，源文件与目标文件的inode号码相同，都指向同一个inode。inode信息中有一项叫做“链接数”，记录指向该inode的文件名总数，这时就会增加1。反过来，删除一个文件名，就会使得inode节点中的“链接数”减1。当这个值减到0，表明没有文件名指向这个inode，系统就会回收这个inode号码，以及其所对应block区域。

这里顺便说一下目录文件的“链接数”。创建目录时，默认会生成两个目录项：“.”和“..”。前者的inode号码就是当前目录的inode号码，等同于当前目录的“硬链接”；后者的inode号码就是当前目录的父目录的inode号码，等同于父目录的“硬链接”。所以，任何一个目录的“硬链接”总数，总是等于2加上它的子目录总数（含隐藏目录），这里的2是父目录对其的“硬链接”和当前目录下的“硬链接”。

7 软链接

除了硬链接以外，还有一种特殊情况。文件A和文件B的inode号码虽然不一样，但是文件A的内容是文件B的路径。读取文件A时，系统会自动将访问者导向文件B。因此，无论打开哪一个文件，最终读取的都是文件B。这时，文件A就称为文件B的“软链接”（soft link）或者“符号链接”（symbolic link）。

这意味着，文件A依赖于文件B而存在，如果删除了文件B，打开文件A就会报错：“No such file or directory”。这是软链接与硬链接最大的不同：文件A指向文件B的文件名，而不是文件B的inode号码，文件B的inode“链接数”不会因此发生变化。

ln -s命令可以创建软链接。

```
[plain]
01. [root@localhost /]# ln -s 源文件或目录 目标文件或目录
```

8 inode的特殊作用

由于inode号码与文件名分离，这种机制导致了一些Unix/Linux系统特有的现象。

1. 有时，文件名包含特殊字符，无法正常删除。这时，直接删除inode节点，就能起到删除文件的作用。
2. 移动文件或重命名文件，只是改变文件名，不影响inode号码。
3. 打开一个文件以后，系统就以inode号码来识别这个文件，不再考虑文件名。因此，通常来说，系统无法从inode号码得知文件名。

第3点使得软件更新变得简单，可以在不关闭软件的情况下进行更新，不需要重启。因为系统通过inode号码，识别运行中的文件，不通过文件名。更新的时候，新版文件以同样的文件名，生成一个新的inode，不会影响到运行中的文件。等到下一次运行这个软件的时候，文件名就自动指向新版文件，旧版文件的inode则被回收。

9 实际问题

在一台配置较低的Linux服务器（内存、硬盘比较小）的/data分区内创建文件时，系统提示磁盘空间不足，用df -h命令查看了一下磁盘使用情况，发现/data分区只使用了66%，还有12G的剩余空间，按理说不会出现这种问题。后来用df -i查看了一下/data分区的索引节点(inode)，发现已经用满(Used=100%)，导致系统无法创建新目录和文件。

查找原因：

/data/cache目录中存在数量非常多的小字节缓存文件，占用的Block不多，但是占用了大量的inode。

解决方案：

1. 删除/data/cache目录中的部分文件，释放出/data分区的一部分inode。
2. 用软连接将空闲分区/opt中的newcache目录连接到/data/cache，使用/opt分区的inode来缓解/data分区inode不足的问题：

```
[plain]
01. [root@localhost /]# ln -s /opt/newcache /data/cache
```

原文地址：<http://blog.csdn.net/jesseyoung/article/details/42524813>

博客主页：<http://blog.csdn.net/jesseyoung>

关闭

顶 踩
2 0



上一篇 传输工具sftp使用详解
下一篇 linux 权限掩码函数 umask 介绍

相关文章推荐

- linux stat命令参数及用法详解---linux显示inode节...
- Presto的服务治理与架构在京东的实践与应用--王...
- 文件索引节点
- 深入掌握Kubernetes应用实践--王渊命
- Linux 索引节点
- Python基础知识汇总
- EXT2目录文件索引节点的操作方法
- Android核心技术详解
- VFS的索引节点
- Retrofit 从入门封装到源码解析
- Linux读取目录函数readdir以及inode结构简介
- 自然语言处理工具Word2Vec
- linux下inode深入浅出
- 解决Linux索引节点(inode)用满导致故障的方法
- Linux中的索引节点、硬连接和连接计数
- Linux 中索引节点的理解



查看评论

1楼 Dablelv 2016-11-10 16:46发表



请问博主，linux中是如何根据文件名称找到对应的inode索引节点的？

您还没有登录,请[登录](#)或[注册](#)



* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved



关闭

