目录：

[1、Linux系统的目录结构 1](#_Toc418499242)

1.1目录层次标准FHS 1

1.2根目录结构 2

1.2.1重要子目录的说明 8

2、文件属性 [9](#_Toc418499243)

2.1文件属性概论 9

2.1.1文件类型概念 10

2.1.2常用的文件分为五种 10

2.2Linux的扩展名 12

2.3索引节点inode 13

2.4Linux系统中用户角色划分 16

2.4.1用户分为三类 16

2.5Linux系统链接的概念 17

2.5.1硬链接 18

2.5.2软链接 19

2.5.3目录链接 20

2.5.4文件删除原理 22

2.5.5文件的时间 23

**1、Linux系统的目录结构**

* 1. **目录层次标准FHS**

FHS全称（filesystem hierarchy standard），中文意思是目录层次标准，是Linux的目录规范标准。

**FHS定义了两层规范：**

第一层是“/”目录下的各个目录应该放什么文件数据，例如：/etc目录下放置系统的配置文件，以及启动的命令，例如：/etc/exports，/etc/hosts，而/bin和/sbin是放置程序及系统命令。

第二层是针对/usr（unix software resource|secondary hierarchy）和/var（variable data）防止系统日志文件等。提示：/var分区单独分出来，硬盘足够大，我们都有监控，剩余20%报警。

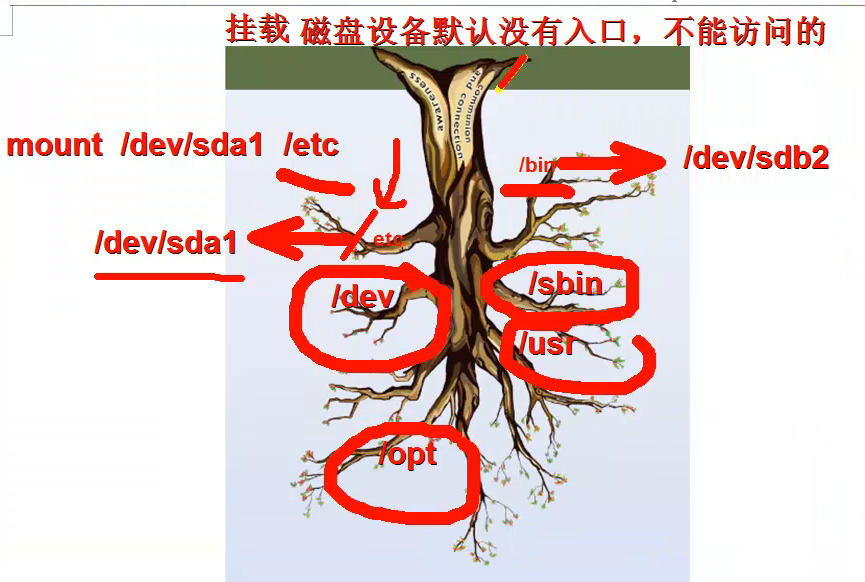
**1.2根目录结构**

逻辑上所有的目录（包括目录下的字目录）都在最高级别的目录“/”下，根（“/”）目录是Linux系统中所有目录的起点（顶点），根下面的目录及目录的子目录是一个有层次的树结构，很像一颗倒挂着的一颗大树，这和Windows系统按照硬盘分区（C：D：E：）分隔的目录结构是不同的（Windows系统可以理解为多棵树，每个分区就是一颗独立的小树）。

Linux系统中的这些不同的目录也是很特别的，不同的目录可以分布在不同的硬盘分区以及不同的硬盘设备上，这是Windows的系统难以做到的，Windows系统的不同分区都是独立存在的。

**根目录小结：**

1. Linux的所有目录结构是一个有层次的倒挂着的树形目录结构。“/”根是所有目录的起点。
2. 目录结构和物理设备是没有关联的，不同的目录数据可以跨越不同的磁盘分区或不同的磁盘设备。
3. 所有的目录都是按照一定的类别有规律的组织和命名的。（每一个目录都可以是分区）
4. Linux系统里的设备默认情况是无法访问的，需要进行挂载。
5. 挂载就是给设备设置一个访问的入口，这个入口是Linux的目录，一般使用绝对路径，这个目录入口被称之为挂载点。
6. 实现挂载的命令就是mount，卸载挂载的命令就是umount。
7. 通过df-h 查看系统的分区情况
8. 系统分区是通过配置/etc/fstab实现开机自动挂载的。

****

挂载：给设备设置一个入口（Linux的目录，绝对路径）

Linux下面设备默认是没有入口的（没门没窗户的监狱，要想访问的话就需要挂载点）

如果想访问/dev/sda1（第一块硬盘的第一个主分区） 需要一个入口/mnt，但想访问的话需要输入mount（实现挂载的命令）命令

被挂载的目录叫做挂载点。

Mount

|  |  |  |
| --- | --- | --- |
| 文件系统类型 | 设备 | 挂载点 |
| Mount | /dev/cdrom | /mnt |

[root@C66-moban ~]# tree -L 1 /

|  |  |
| --- | --- |
| ├── / | 第一层次结构的根、整个文件系统层次结构的根目录 |
| ├── bin | 二进制命令的路径（cp，rm，cat） |
| ├── boot | Linux引导文件：linux启动引导程序的路径  （static files of the boot loader）Linux的内核及系统引导程序所  需要的文件目录，安装系统 |
| ├── dev | 设备文件的目录，比如声卡、磁盘、光驱。。。。。 |
| ├── etc | 系统配置与服务配置文件默认路径及启动命令的路径  二进制安装包（yum，rpm）的配置文件默认路径  Fstab-----开机自动挂载的配置文件  Motd----系统登录时显示登录信息的地方  Profile----（所有的用户都生效）全局变量  服务启动命令存放的目录：/etc/init.d（yum，rpm） |
| ├── home | 普通用户家目录 |
| ├── lib | 库文件的目录Shared libraries and kernel modules  库文件存放目录。（共享库和内核模块库文件所在的命令）库文件是程序与系统打交道的文件。 |
| ├── lib64 | 64位库文件的目录 |
| ├── lost（丢失）+found（找到） | 在ext3文件系统中，当系统意外崩溃或机器意外关机，会产生一些文件碎片在这里。当系统在开机启动的过程中fsck（磁盘检查修复的工具）工具会检查这里，并修复已经损坏的文件系统。当系统发生问题。可能会有文件被移动到这个目录中，可能需要用手工的方式来修复，或移到文件到原来的位置上。 |
| ├── media | 可移除媒体（如CD-ROM）的挂载点（在FHS-2.3中出现） |
| ├── mnt | 设备临时挂载点 |
| ├── opt | 可选应用软件包，表示的是可选择的意思。 |
| ├── proc\*\*\*\*\* | 显示进程及内核信息的虚拟文件系统  cat /proc/meninfo 内存信息  cat /proc/cpuinfo CPU信息  cat /proc/mounts 挂载信息  操作系统运行时，进程信息及 |
| ├── root | 超级用户的家目录 |
| ├── sbin | 系统管理员执行的命令 |
| ├── selinux |  |
| ├── srv | 站点的具体数据，由系统提供。 |
| ├── sys | 这个目录其实跟/proc非常类似，也是一个虚拟的档案系统，主要也是记录与核心相关的资讯。 包括目前已载入的核心模组与核心侦测到的硬体装置资讯等等。 这个目录同样不占硬盘容量。 |
| ├── tmp | 临时文件的目录 |
| ├── usr | 存放第三方厂商开发的程序。 |
| └── var | 可变化的文件目录，日志、缓存等。  /var/log/messages（系统日志文件）（dmesg命令可查询）  /var/log/secure（安全的日志，会对登录信息做记录） |

**小结：重要目录文件**

/dev/ 设备目录  
/etc/  系统配置及服务配置文件、启动命令的目录  
/proc 显示内核及进程信息的虚拟文件系统  
/tmp  临时文件目录，公共厕所，客厅。  
/home 普通用户家目录，大臣的家。~  
/root  超级管理员的家目录，皇宫。  
/var   变化的目录，一般是日志文件，cache的目录。  
/usr   用户程序，及数据、帮助文件、二进制命令等的目录。  
/bin和/sbin和/usr/sbin/  用户命令的目录

**1.2.1 重要子目录的说明**

（1）/etc/sysconfig/network-scripts/ifcfg-eth0 网卡配置文件

使用setup命令修改网卡的配置就是改上述文件。

/etc/init.d/network restart 启动网卡命令（在使用此条命令时需注意ONBOOT=yes）

[root@www ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0 🡸 物理设备名，eth0表示第一块网卡

HWADDR=00:0c:29:c2:23:08 🡸 网卡的MAC地址，48位

TYPE=Ethernet 🡸 网络类型=以太网

UUID=c0433d9c-5d15-4984-9948-9ced2c57aa2d 🡸 唯一的用户标识

ONBOOT=yes 🡸 控制网卡是不是开机启动

NM\_CONTROLLED=yes 🡸 网络管理

BOOTPROTO=dhcp 🡸 其中，protoco去下列值之一：none，引导是不使用协议；static静态分配地址；bootp，使用bootp协议，或dhcp，使用DHCP协议

IPADDR=10.0.0.8 🡸IP地址。

NETMASK=255.255.255.0 🡸子网掩码，划分网络位和主机位。

DNS2=8.8.8.8 🡸DNS就是域名和IP的解析工具

GATEWAY=10.0.0.2 🡸网关---路由器的地址

DNS1=10.0.0.2

IPV6INIT=no

USERCTL=no

（2）/etc/resolv.conf DNS配置文件 （在这里配置完之后很有可能会被网卡给清掉）

DNS：domain name system 主要功能把上网的域名解析成服务器IP

HTCP：动态分配IP地址

1、客户端DNS可以在网卡配置文件里设置（ifcgf-eth0）

2、客户端DNS也可以在/etc/resolv.conf里设置

3、网卡里的设置DNS优先于/etc/resolv.conf

（3）/etc/hosts 设定用户IP与名字（或域名）的对应解析表。（相当于局域网的DNS）

作用：

1、开发、产品、测试等人员，用于通过正式的域名测试产品；

2、服务器之间的调用可以用域名（内部的DNS），方便迁移。

（4）/etc/sysconfig/network IP、掩码、网关、主机名的配置文件。

（5）/etc/fstab 记录开机要挂载的信息的文件系统的一个文件。

mount –t ext4 –o rw /dv/sda1 /mnt 命令写入fstab如下形式：

第一列：/dev/dal 第二列：/mnt 第三列： ext4 第四列：rw 第五列：0 第六列：0

光驱挂载：mount -t iso9660 /dev/cdrom /mnt

第一列：要被挂载的设备；第二列：挂载点；第三列：类型；第四列：挂载的选项（只读、可写）；第五列：提示是否备份（0为不备份）；第六列：是否开机进行扫描（0为不扫描）；

######---🡪加播一条挂载命令

挂载：给设备设置一个入口（Linux的目录，绝对路径）

Linux下面设备默认是没有入口的（没门没窗户的监狱，要想访问的话就需要挂载点）

如果想访问/dev/sda1（第一块硬盘的第一个主分区） 需要一个入口/mnt，但想访问的话需要输入mount（实现挂载的命令）命令

被挂载的目录叫做挂载点。

Mount

|  |  |  |
| --- | --- | --- |
| 文件系统类型 | 设备 | 挂载点 |
| Mount | /dev/cdrom | /mnt |

fstab一旦配置错误，会导致服务器无法启动，如何修复？

[root@moban ~]# blkid

/dev/sda1: UUID="cb8e41b3-ffb3-454e-9ea0-451bf5c21dd3" TYPE="ext4"

/dev/sda2: UUID="a3873f14-f3ea-4da4-a8b6-8e1491dcb5e4" TYPE="swap"

/dev/sda3: UUID="9c763286-60af-4048-96a4-3b363d2ff21c" TYPE="ext4"

第一列挂载的设备，可以是设备名或UUID，或磁盘标签。

（6）/etc/rc.local ======/etc/rc.d/rc.local 这个目录来存放系统启动脚本,放服务器档案文件（用于存放开机自启动程序命令的文件）

（7）/etc/inittab系统运行级别和相关的配置文件（设定系统启动时init进程将把系统设置成什么样的runlevel运行级别）

（8）/etc/init.d/======/etc/rc.d/init.d 启动命令的目录（可以通过chkconfig 来管理）

（9）/etc/profile 全局环境配置变量

（10）/etc/profile.d 加载系统登录程序的一个目录，放一个文件（文件内容是命令）（登陆后显示的是脚本所在地）

（11）~/.bash\_profile 用户的环境变量（只对用户生效）

（12）/etc/motd 登陆后显示的是字符串

（13）/etc/issue 开机登陆前的提示。默认是系统版本提示

（14）/var/log/messages 系统日志文件

（15）/var/log/secure 安全日志文件 如pop3，ssh，

（16）/var/spool/cron/root 定时任务的配置文件

（17）proc/meminfo #<==内存信息

（18）/proc/cpuinfo #<==cpu信息

（19）/proc/mounts #<==查看设备挂载结果

（20）/etc/exports 设定NFS网络文件系统用的配置文件

（21）/etc/init.d （yum/rpm安装的软件）启动命令的路径。

（22）/etc/init.d/sshd start ====service sshd start

（23）Visudo就是修改的/etc/sudoers

（24）/usr/local 相当于：c:\program files

（25）/usr/src 内核源码存放目录，比如下面有内核源码目录。

（26）/var/spool/cron/root 定时任务的配置文件

（27）/var/spool/clientmqueue sendmail临时邮件文件目录（被小文件填满）

（28）/proc/mounts 设备的挂载信息，df –h 类似

（29）/proc/sys/net/ipv4 /etc/sysctl.conf配置的参数就对应这个路径。

（30）/etc/redhat-release Linux版本信息

（31）管理用户、用户组信息的路径

/etc/group #→设定用户的组名与相关信息※※※ chattr +i  
/etc/passwd #→帐号信息文件 ※※※※※  
/etc/shadow #→密码信息文件  
/etc/gshadow #→组密码信息文件

（32）/proc/loadavg 负载，系统繁忙程度。

[root@moban ~]# cat /proc/loadavg

1. 0.00 0.00 1/77 4897

此命令和uptime是等价的。

[root@moban ~]# uptime

13:23:16 up 16:42, 1 user, load average: 0.00, 0.00, 0.00

查看目录使用大小的命令：

[root@moban ~]# du -sh /boot

34M /boot

Free –m 时查看内存

[root@moban ~]# free -m

total used free shared buffers cached

Mem: 992 132 859 0 25 34

-/+ buffers/cache: 72 919

Swap: 767 0 767

**2、Linux文件属性**

**2.1文件属性概论**

**Linux系统中的文件或目录的属性主要包括：索引节点（inode）、文件类型、权限属性、连接数、所归属的用户和用户组、最近修改时间等内容；**

****

文字解释：

第一列：

inode，（index node），索引节点编号；它是文件或目录在磁盘里的唯一标识。Linux读取文件首先要读取到这个索引节点。

第二列：

第1个字符，代表文件类型，-普通文件，d目录，（Windows 扩展名 .jpg）

第2-10个字符，rw-r--r—文件权限（r (read读) w (write写) x (eXecute执行) -）

第11个字符，‘.’selinux相关的

第三列：

文件的硬链接数。硬链接是文件的又一个入口。

第四列：

文件对应的属主或者用户

第五列：

文件对应的用户属组或用户组。（团体）

第六列：

文件的大小

第七、八、九列：

文件被改动的时间

第十列：文件和目录的名字。

**2.1.1 文件类型概念：**

1)windows里是通过扩展名(.txt)来区分文件类型的。oldboy.txt  
2)linux里文件扩展名和文件类型没有关系。  
3)为了容易区分和兼容用户使用windows的习惯，我们也会用扩展名来表示linux里的文件类型。oldboy.sh,.oldboy.tar.gz

在linux系统中，可以说一切皆文件

**2.1.2 常用的文件分为五种：**

1. 普通文件（在最前边显示“-”），可分为3种：

1、----纯文本文件(ASCII)：内容可直接读取，数字字母等 （使用file命令查看文件类型）

例如：

[root@oldboy oldboy]# ls /etc/hosts  
/etc/hosts  
[root@oldboy oldboy]# ls /etc/motd   
/etc/motd  
[root@oldboy oldboy]# ls /etc/sysconfig/network  
/etc/sysconfig/network  
[root@oldboy oldboy]# file /etc/hosts  
/etc/hosts: ASCII text

2、二进制文件(binary)：Linux中的可执行文件，比如命令文件,cat ls cp。

3、数据格式的文件(data)：具有特定格式的文件（存放系统的临时信息、日志等文件。）

例：

[root@oldboy oldboy]# file /var/log/wtmp   
/var/log/wtmp: data  
[root@oldboy oldboy]# file /var/log/messages  
/var/log/messages: ASCII English text  
[root@oldboy oldboy]# file /var/log/lastlog   
/var/log/lastlog: data

Lastlog=====/var/log/lastlog

2）目录文件（以d开头的文件）

[root@oldboy oldboy]# ls -lF  
总用量 8  
-rw-r--r-- 1 root root    0 10月 16 01:28 1  
-rw-r--r-- 1 root root    0 10月 16 01:28 10  
-rw-r--r-- 1 root root    0 10月 16 01:28 2  
-rw-r--r-- 1 root root    0 10月 16 01:28 3  
-rw-r--r-- 1 root root    0 10月 16 01:28 4  
-rw-r--r-- 1 root root    0 10月 16 01:28 5  
-rw-r--r-- 1 root root    0 10月 16 01:28 6  
-rw-r--r-- 1 root root    0 10月 16 01:28 7  
-rw-r--r-- 1 root root    0 10月 16 01:28 8  
-rw-r--r-- 1 root root    0 10月 16 01:28 9  
-rw-r--r-- 1 root root    0 10月 16 04:56 a  
drwxr-xr-x 2 root root 4096 10月 16 04:49 abc/  
-rw-r--r-- 1 root root    0 10月 16 04:56 a.txt  
drwxr-xr-x 2 root root 4096 10月 16 04:49 da/  
[root@oldboy oldboy]#   
[root@oldboy oldboy]# ls -lF|grep /$  
drwxr-xr-x 2 root root 4096 10月 16 04:49 abc/  
drwxr-xr-x 2 root root 4096 10月 16 04:49 da/  
[root@oldboy oldboy]# ls -lp|grep /$   
drwxr-xr-x 2 root root 4096 10月 16 04:49 abc/  
drwxr-xr-x 2 root root 4096 10月 16 04:49 da/

3）字符设备与块设备文件

1. 字符以C开头的文件，一般为串行端口的接口设备，如猫等串口设备
2. 块设备以b开头的文件，存储数据的设备，例如：磁盘、光驱等都属于块设备

4）套接口文件（以S开头的文件）用于网络之间数据连接的。

5）符号链接文件（以L开头的文件）\*\*\*\*\*

**文件类型小结：**

1. - regular file普通文件（纯文本，二进制文件，数据文件。）

可用命令（touch vi cp mv rm ）\*\*\*\*\*

1. d  directory 目录 \*\*\*\*\*

可用命令（mkdir cp mv rm rmdir）

1. l link符号链接或者软链接文件 \*\*\*\*\*

可用命令（ln、ln -s)

1. c字符文件，b块设备文件

* character  串口设备 拨号猫
* b block  硬盘，光驱 \*\*\*\*\*

1. s socket套接字文件，进程之间通信会用到。
2. p 管道文件

**2.2 Linux的扩展名**

1. 源码tar .tar.gz .tgz .zip .tar.bz
2. .sh表示shell脚本文件，通过shell语言开发程序
3. .pl表示perl语言文件，通过Perl语言开发程序
4. .py表示python语言文件，通过python语言开发程序
5. .html/.htm/php/.jsp.do表示网页语言文件
6. .conf表示系统服务的配置文件
7. .rpm表示rpm安装包文件

**2.3索引节点inode**

中文意思是索引节点（index node）。在每个Linux存储设备或存储设备的分区（存储设备可以是硬盘、软盘、U盘）设备被格式化为ext4文件系统后，一般都有两部分：

第一部分是inode（很多个）包含文件属性以及指向文件实体的指针。

存储空间C6：256字节 C5：128字节

存放：inode就是用来存储这些数据属性信息的（也就是ls –l的结果），inode包含的文件属性信息有文件大小、属主、归属的用户组、读写权限、文件类型，修改时间，还包含指向文件实体的指针的功能（inode节点—block对应关系）等，但是inode唯独不包含文件名。

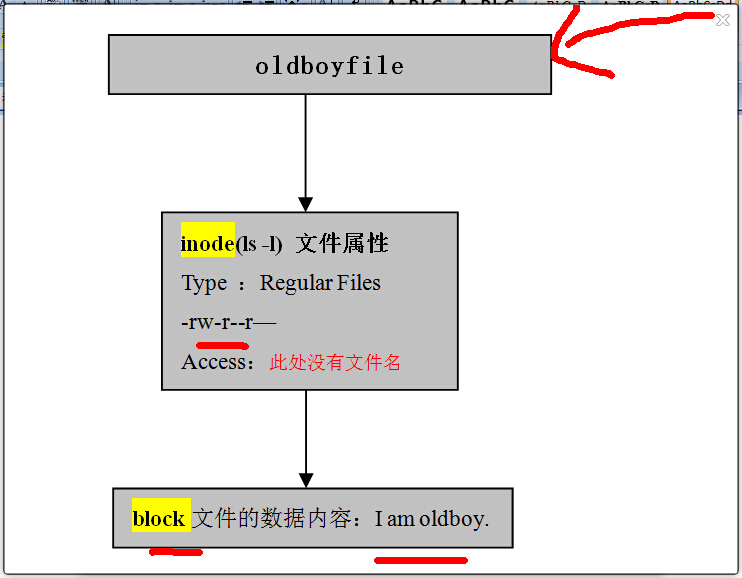
第二部分是block（很多个）存放实际数据的。1k，2k，4k

磁盘---🡪分区---🡪格式化---🡪挂载。

房子---🡪隔断---🡪装修---🡪安个门窗---🡪入住。

Inode：index node （索引节点）

磁盘-----分区------格式化（创建文件系统）

****

[root@oldboy oldboy]# dumpe2fs /dev/sda3|grep -i "inode size"   
dumpe2fs 1.41.12 (17-May-2010)  
Inode size:               256  
[root@oldboy oldboy]# dumpe2fs /dev/sda3|grep -i "block size"  
dumpe2fs 1.41.12 (17-May-2010)  
Block size:               4096

标准的inode为：256；标准的block为：4096.

**Inode小结：学会给阶段性的知识做小节是学好Linux运维的好习惯。**

1、磁盘分区格式化ext4文件系统后会生成一定数量的inode和block。

2、Inode是索引节点，作用是存放文件的属性信息以及作为文件的索引（指向文件的实体）

3、Ext3/ext4文件系统的block存放的是文件的实际内容。

4、Inode是一块存储空间，C6非启动分区inode默认大小256字节，C5是128字节。

5、Inode是一串数字，不同的文件对应的inode（一串数字）在文件系统里是唯一的。

6、Inode相同的文件，互为硬链接文件，实质还是一个文件。（文件的又一个入口）

7、一个文件被创建后至少要占用一个inode和一个block（一个inode和block只能被一个文件使用）

8、Block的大小一般有1k，2k，4k几种，其中引导分区等为1K，其他普通分区为4K（C6）

9、如果一个文件很大，可能占多个block，如果文件很小0.1K，至少占一个4K的block，并且剩余空间浪费了。

10、如果一个文件很小，也至少占一个block，并且剩余空间不可以使用。

11、Inode大小和总量查看

（1）

[root@C66-moban ~]# dumpe2fs /dev/sda3|egrep -i "block size|inode size"

dumpe2fs 1.41.12 (17-May-2010)

Block size: 4096

Inode size: 256

（2）

[root@C66-moban ~]# dumpe2fs /dev/sda3|egrep -i "block count|inode count"

dumpe2fs 1.41.12 (17-May-2010)

Inode count: 462384

Block count: 1849088

Reserved block count: 92454

12、查看inode的总量和使用量目录 df –I

[root@C66-moban ~]# df -i

Filesystem Inodes IUsed IFree IUse% Mounted on

/dev/sda3 462384 54628 407756 12% /

tmpfs 127032 1 127031 1% /dev/shm

/dev/sda1 51200 38 51162 1% /boot

13、如何生成及制定inode大小 mkfs.ext4 –b 2048 –l 256 /dev/sdb

Inode知识网友资料：

http://www.cnblogs.com/thinksasa/archive/2013/04/10/3013445.html

<http://www.cnblogs.com/adforce/p/3522433.html>

**Linux有关block知识：**

1、磁盘读取数据是按block为单位读取的。

2、一个文件可能占用多个block，每读取一个block就会消耗一次磁盘I/0

3、如果要提升磁盘I0性能，那么就要尽可能一次读取数据尽量的多。

Block设置为4K。

4、一个block只能存放一个文件的内容，无论内容有多小。如果block 4K，那存放1K的文件，剩余3K就浪费了。

5、Block并非越大越好，block太大对于小文件存放就会浪费磁盘空间，例如：1000K的文件，block为4K，占用250个block，block为1K，占1000个block，访问效率谁更高？消耗I0分别额外250次和1000次。

6、大文件（大于4K）一般设置block大一点，小文件（小4K）一般设置block小一点。

7、Block太大例如4K，文件都是0.1K的，大量浪费磁盘空间。

8、Block太大例如1K，文件都是1000K，消耗IO过多。

9、Block的设置也是格式化分区的时候，mkfs.ext4 –b 2048 –I 256 /dev/sdb

10、文件较大时，block设置大一些会提升磁盘访问效率。

11、Ext3/ext4文件一般设置为4K

=======================================================

Block要大于inode

Inode count:              51200  
Block count:              204800

=======================================================

1、企业面试题：一个100M的磁盘分区，分别写入0.5K的文件及写入1M的文件，分别可以写多少个？为什么？

解答思想：先答四点知识  
a.Inode是存放文件属性信息的，默认大小128byte(c58)，256byte(c64)。  
b.Block 是存放文件实际内容的，默认大小1K（boot）或4K（非系统分区默认给4K）。  
c.一个文件至少要占用一个inode及一个block。  
d. 默认分区常规情况下，inode数量是足够的。而block数量消耗的会更快。  
正确解答：  
1、默认分区常规情况下，对大文件来讲inode是足够的。而block数量消耗的会更快，BLOCK为4K的情况，1M的文件不会有磁盘浪费情况，所以文件数量大概为100/1=100个。  
2、对于小文件0.1k，inode会消耗的更快。默认分区的时候block数量是大于inode数量的。  
每个小文件都会占用一个inode和一个block。所以最终文件的数量：  
inode会先消耗完，文件总量是inode的数量。

2、企业案例：如果向磁盘写入数据提示如下错误：No space left on device，通过df -h查看磁盘空间，发现没满，请问可能原因是什么？

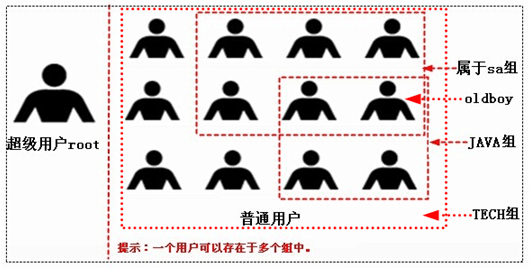
解答：可能是inode数量被消耗尽了。（系统里小文件太多了）

CentOS5系列的系统会默认安装Sendmail服务，因此邮件临时存放地点的路径为/var/spool/clientmqueue/。  
CentOS6默认情况下没有安装Sendmail服务，而是改装了Postfix服务，因此邮件临时存放地点的路径为/var/spool/postfix/maildrop/。

**2.4 Linux系统中用户角色划分**

在Linux系统中用户是分角色的，由于角色不同，权限和所完成的任务也不同，对于Linux系统来说，用户的角色是通过UID和GID识别的，特别是UID，在Linux系统运维工作中，一个UID是唯一标识一个系统用户的账号（相当于我们的身份证），用户系统账号的名称（如oldboy）其实给人（管理员）看的，Linux系统能够识别的仅仅是UID和GID这样的数字。

**用户和组的对应关系：**



用户的角色是通过UID和GID识别的。

用户的UID就相当于我们的身份证一样。

用户名就相当于我们的名字。

UID User Identify 用户ID，相当于我们的身份证，在系统中是唯一的。

GID Group Identify 组 ID,相当于我们的家庭或者学校。

企业工作中：没有

**2.4.1 用户分为三类：**

超级用户：

默认是root用户，其UID和GID均为0，root比喻为皇帝，root家目录为皇宫，操作系统为全国。

普通用户：

这类用户一般是由具备系统管理员root的权限的运维的运维或系统管理人添加的。普通用户比喻皇帝的臣民，干坏事的，国家有法律管束你，为普通用户

虚拟用户：

与真实普通用户区分开来，这类用户最大的特点就是安装系统后默认就会存在的，并且默认情况大多数不能登录系统。

**创建文件：必须满足用户和组的要求。**

**程序运行：必须满足用户和组的要求。（虚拟用户）相当于傀儡。**

用户的范围：

超级用户：UID GID 0

普通用户：500+

虚拟用户：1-499

**2.5 Linux系统链接的概念**

在Linux系统中，链接可分为两种：一种为硬链接（hard link），另一种为软链接符号链接（symbolic link or soft link）。前面讲解过In这个命令就是创建链接文件的，在默认不带参数的情况下，习性ln命令创建的链接是硬链接。

如果使用ln –s创建链接则为软链接，前面文件类型为L（字母L）的是软链接。

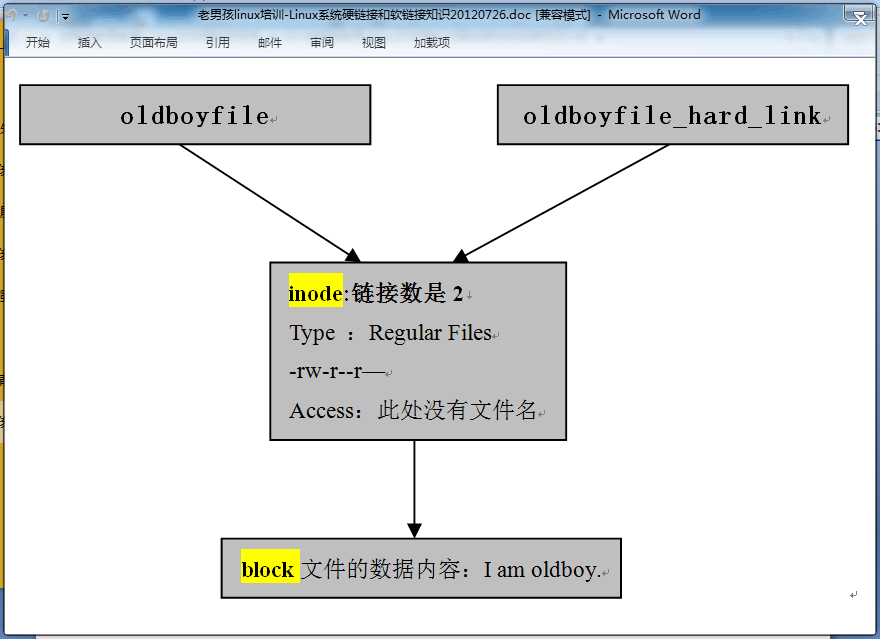
**2.5.1 硬链接**

硬链接是指通过索引节点（inode）来进行链接，在Linux（ext2,ext3,ext4）文件系统中，保存在磁盘分区中的文件不管是什么类型都会给它分配一个编号，这个编号被称为索引节点编号（index inode）简称inode，即在系统中文件的编号。

在Linux文件系统中，多个文件名指向同一个索引节点是正常且允许的，这种情况的文件就称为硬链接，硬链接文件就相当于文件的另一个入口。

作用：允许一个文件拥有多个有效路径名（多个入口），这样用户就可以建立硬链接到重要的文件，以防止“误删”源数据。

硬链接：多个文件名指向同一个索引节点，硬链接文件就相当于文件的另一个入口。

****

**硬链接知识小结：**

1、具有相同inode节点号的多个文件互为硬链接文件。

2、删除硬链接文件或者删除源文件任意之一，文件实体并未被删除。

3、只有删除了源文件及所有对应的硬链接文件，文件实体才会被删除。

4、当所有的硬链接文件及源文件被删除后，在存放新的数据会占用这个文件的空间，或者磁盘fsck检查的时候，删除的数据也会被系统回收。

5、硬链接文件就是文件的另一个入口（相当于超市的前、后门一样）

6、可以通过给文件设置硬链接文件，来防止重要文件被误删。

7、通过执行命令“In源文件 硬链接文件”，即可完成创建硬链接。

8、硬链接文件可以用rm命令删除。

9、对于静态文件（没有进程正在调用的文件）来讲，当对应硬链接数为0（i\_link），文件就被删除，i\_link的查看方法（ls –l 结果的第三列就是）

**硬链接实例：**

[root@moban data]# mkdir /test/oldboy -p

[root@moban data]# cd /test/oldboy/

[root@moban oldboy]# touch oldboyfile

[root@moban oldboy]# ls -lhi

总用量 0

140102 -rw-r--r-- 1 root root 0 10月 16 18:08 oldboyfile

[root@moban oldboy]# ln oldboyfile oldboyfile\_hard\_link

[root@moban oldboy]# ls -lhi

总用量 0

140102 -rw-r--r-- 2 root root 0 10月 16 18:08 oldboyfile

140102 -rw-r--r-- 2 root root 0 10月 16 18:08 oldboyfile\_hard\_link

[root@moban oldboy]# echo "I am oldboy" >oldboyfile

[root@moban oldboy]# cat oldboyfile\_hard\_link

I am oldboy

[root@moban oldboy]# ln oldboyfile oldboyfile\_hard\_link\_1

[root@moban oldboy]# ls -lhi

总用量 12K

140102 -rw-r--r-- 3 root root 12 10月 16 18:10 oldboyfile

140102 -rw-r--r-- 3 root root 12 10月 16 18:10 oldboyfile\_hard\_link

140102 -rw-r--r-- 3 root root 12 10月 16 18:10 oldboyfile\_hard\_link\_1

**2.5.2 软链接**

软链接（soft link）

执行命令“In –s 源文件 软链接文件”，即可完成创建软链接，误区：创建软链接文件是需要存在的，要创建的软链接文件是不能存在的，是要用In命令创建的。

Linux里的软链接文件就类似于Windows系统中的快捷方式，实际上可以理解为一个文本文件。

**软链接实例：**

[root@moban oldboy]# mkdir /application/apache2.2.23 -p

[root@moban oldboy]# ln -s /app

app/ application/

[root@moban oldboy]# ln -s /application/apache2.2.23/ /application/apache

[root@moban oldboy]# ls -l /application/

总用量 4

lrwxrwxrwx 1 root root 26 10月 16 18:30 apache -> /application/apache2.2.23/

drwxr-xr-x 2 root root 4096 10月 16 18:29 apache2.2.23

[root@moban oldboy]# readlink /application/apache

/application/apache2.2.23/

**软链接知识小结**：

1、软链接类似Windows的快捷方式（可以通过readlink查看其指向）

2、软链接类似一个文本文件，里面存放的是源文件的路径，指向源文件实体。

3、删除源文件，软链接文件依然存在，但是无法访问指向的源文件路径内容了。

4、失效的时候一般是白字红底闪烁提示，oldboy\_soft\_link\_file ->oldboy\_file

5、执行命令“In –s 源文件软链接文件”，即可完成创建软链接（目标不能存在）。

6、软链接和源文件是不同类型的文件，也是不同的文件。Inode号也不相同。

7、删除软连接文件可以用rm命令。

**2.5.3 目录链接**

目录的软链接在老男孩的生产场景中更经常用到，硬链接用的不多。

**实例：**

[root@moban oldboy]# mkdir oldboydir

[root@moban oldboy]# ln oldboydir oldboydir\_hard\_link

ln: "oldboydir": 不允许将硬链接指向目录

[root@moban oldboy]# ln -s oldboydir oldboydir\_soft\_link

[root@moban oldboy]# ls -lhi

总用量 4.0K

140103 drwxr-xr-x 2 root root 4.0K 10月 16 18:39 oldboydir

140108 lrwxrwxrwx 1 root root 9 10月 16 18:41 oldboydir\_soft\_link -> oldboydir

140102 -rw-r--r-- 1 root root 0 10月 16 18:35 oldboyfile

140107 lrwxrwxrwx 1 root root 10 10月 16 18:32 oldboyfile\_soft\_link -> oldboyfile

.开头的文件是隐藏文件。 Ls –a 可以找出来。

[root@oldboy test]# ls -lhid oldboy  oldboy/oldboydir/..  
143595 drwxr-xr-x 3 root root 4.0K 10月 16 08:38 oldboy  
143595 drwxr-xr-x 3 root root 4.0K 10月 16 08:38 oldboy/oldboydir/..  
[root@oldboy test]# ls -lhid /etc/rc3.d/ /etc/rc.d/rc3.d  
132254 drwxr-xr-x. 2 root root 4.0K 10月 16 00:33 /etc/rc3.d/  
132254 drwxr-xr-x. 2 root root 4.0K 10月 16 00:33 /etc/rc.d/rc3.d  
[root@oldboy test]# ls -lhid /etc/init.d/ /etc/rc.d/init.d/  
132250 drwxr-xr-x. 2 root root 4.0K 10月 16 00:22 /etc/init.d/  
132250 drwxr-xr-x. 2 root root 4.0K 10月 16 00:22 /etc/rc.d/init.d/

**有关目录链接的小结：**

（1）对于目录，不可以创建硬链接，但可以创建软链接。

（2）对于目录的软链接是生产场景运维中常用的技巧

（3）目录的硬链接不能跨越文件系统（从硬链接原理可以理解）

（4）每个目录下面都有一个硬链接“.”号，和对应上级目录的硬链接“..”

（5）在父目录里创建一个子目录，父目录的连接数增加1（子目录里都有..来指向父目录）但是再父目录里创建文件，父目录的链接数不会增加。

凯哥讲解目录链接：

[root@moban ~]# mkdir heihei ###---首先创建heihei父目录，依次创建下面子目录

[root@moban ~]# ll -hid heihei/ ###--查看heihei目录链接个数，发现有8个

261880 drwxr-xr-x 8 root root 4.0K 10月 16 23:27 heihei/

[root@moban heihei]# ll –hi ###---父目录下共计6个子目录

总用量 24K

265857 drwxr-xr-x 2 root root 4.0K 10月 16 23:27 haha

265924 drwxr-xr-x 2 root root 4.0K 10月 16 23:27 hehe

264558 drwxr-xr-x 2 root root 4.0K 10月 16 23:27 kaka

265856 drwxr-xr-x 2 root root 4.0K 10月 16 23:27 lala

264330 drwxr-xr-x 2 root root 4.0K 10月 16 23:27 lili

265914 drwxr-xr-x 2 root root 4.0K 10月 16 23:27 xixi

[root@moban ~]# ll -hid heihei heihei/. heihei/haha/.. heihei/hehe/.. heihei/kaka/.. heihei/lala/.. heihei/lili/.. heihei/xixi/..

261880 drwxr-xr-x 8 root root 4.0K 10月 16 23:27 heihei

261880 drwxr-xr-x 8 root root 4.0K 10月 16 23:27 heihei/.

261880 drwxr-xr-x 8 root root 4.0K 10月 16 23:27 heihei/haha/..

261880 drwxr-xr-x 8 root root 4.0K 10月 16 23:27 heihei/hehe/..

261880 drwxr-xr-x 8 root root 4.0K 10月 16 23:27 heihei/kaka/..

261880 drwxr-xr-x 8 root root 4.0K 10月 16 23:27 heihei/lala/..

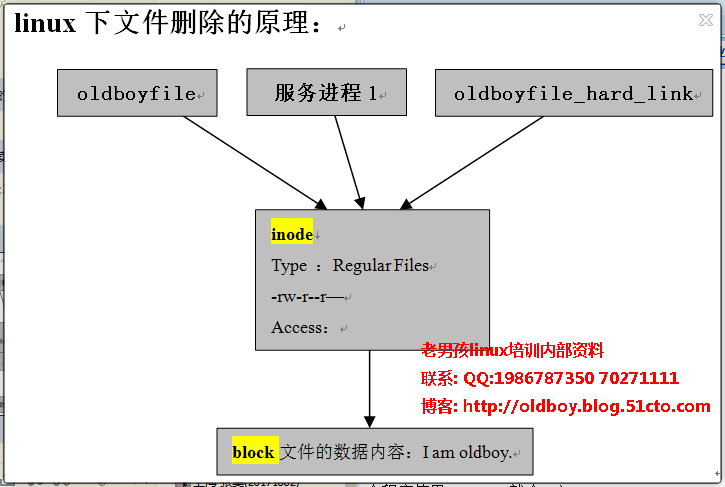
261880 drwxr-xr-x 8 root root 4.0K 10月 16 23:27 heihei/lili/..

261880 drwxr-xr-x 8 root root 4.0K 10月 16 23:27 heihei/xixi/..

个人理解，linux系统中每个目录最少有两个（一个是创建的，一个是隐藏的.）所以linux目录最少都有两个链接。

**2.5.4 文件删除原理**

1. i\_link硬链接数为0
2. i\_count数为0（程序引用计数，有一个程序使用i\_count就会+1）

****

只要文件的索引节点还有一个以上的硬链接，只删除其中一个硬链接（即仅仅删除了该文件的链接指向）并不影响索引节点本身和其它的链接（即数据文件实体并未被删除），只有当文件的最后一个链接被删除后，此时如果有新数据要存储到硬盘上时或者系统通过类似fsck做磁盘检查的时候，被删除文件的数据块及目录的链接才会被释放，空间被新数据占用并覆盖。此时，数据就再也无法找回了，Linux系统中，删除静态文件的条件是与之相关的所有硬链接均被删除。

企业面试题：描述linux下软链接和硬链接的区别（面试题）。（记时2分钟）

解答：  
在linux系统中，链接分两种 ：一种被称为硬链接（Hard Link），另一种被称为符号链接或软链接（Symbolic Link）。  
1)默认不带参数情况下，ln命令创建的是硬链接，带-s参数的ln命令创建的是软链接。  
2)硬链接文件与源文件的inode节点号相同，而软链接文件的inode节点号与源文件不同。3)ln命令不能对目录创建硬链接，但可以创建软链接，对目录的软链接会经常被用到。  
4)删除软链接文件,对源文件及硬链接文件无任何影响；  
5)删除文件的硬链接文件，对源文件及软链接文件无任何影响；  
6)删除链接文件的源文件，对硬链接文件无影响，会导致其软链接失效（红底白字闪烁状）；  
7)同时删除源文件及其硬链接文件，整个文件才会被真正的删除。  
8)很多硬件设备中的快照功能，使用的就类似硬链接的原理。  
9)软链接可以跨文件系统，硬链接不可以跨文件系统。

企业面试答题必胜思路：

介绍软硬链接的概念。

对于文件的软硬链接的区别。

对于目录的软硬链接的区别。

李小龙说过一句话，他不怕一千个动作只练一遍的人，而怕一个动作练一千遍的人。

**2.5.5文件的时间**

**Access:访问时间 find -atime**

**Modify:修改时间，内容发生变化 find –mtime**

**Change:变化时间，包含Modify，权限、属主、用户组 find -ctime**

Access: 2015-08-08 20:33:23.477999996 +0800（访问时间）

**对应查看命令：cat less more**  
Modify: 2015-08-02 11:29:55.309023177 +0800（修改时间）

**对应查看命令：vi/echo/sed**  
Change: 2015-08-02 03:19:28.902006774 +0800（变化时间）

**对应查看命令：vi/echo/sed/chmod/chown**

**显示一个标准的时间命令：**

**ls -lrt --time-style=long-iso**

**[root@oldboy ~]# ls -lrt --time-style=long-iso  
总用量 4  
drwxr-xr-x 5 oldboy oldboy 4096 2015-04-16 17:05 tools  
-rw-rw-r-- 1 oldboy oldboy    0 2015-08-08 21:01 a**