**Linux**

**你明知道蜷缩在床上感觉更温暖，但还是一早就起床；你明知道什么都不做比较轻松，但依旧选择追逐梦想。这就是生活，你必须坚持下去。**

## 一、基础信息

1、**GNU**

**GNU GNU’s not unix,是1984年Richard stallman 发起的自由软件运动，导致FSF诞生，从而有了GNU这个项目，开发一个操作系统及各种组件，不是UNIX操作系统，实际就是Linux系统，GNU项目是Linux内核之上的一些组件，不包括内核。**

**GPL**

**GPL general public license 许可任何人有共享和修改自由软件的自由，任何人有权利利用、改、发布。GPL就是一个协议。**

**CentOS**

**以Redhat所发布的所有源代码重新组件符合GPL的Linux，即Redhat Linux 的在编译版**

**2、图形桌面的支持**

**1）X windows**

**2）GNOME**

**3）KDE**

## 二、安装CentOS Linux

**1、分区建议：**

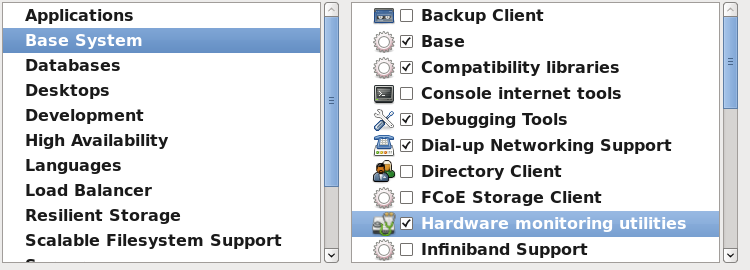
**/ 根分区剩余所有空间**

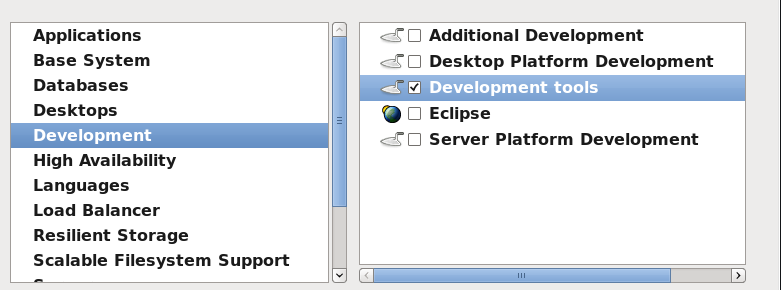
**Swap 1.5倍的物理内存**

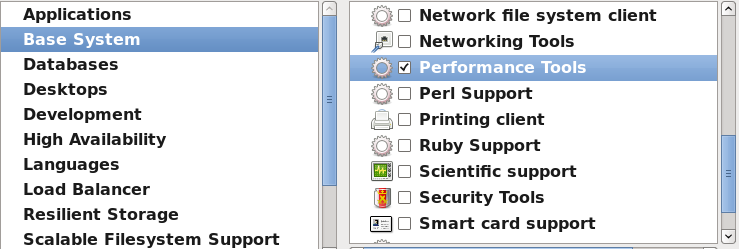
**Boot 100-200M**

**2、生产场景不同角色Linux服务器分区方案**

|  |  |  |  |
| --- | --- | --- | --- |
| **服务器角色** | **分区建议** | **优点** | **RAID方案** |
| **单机服务器**  **如8G内存，300G硬盘** | **/boot 100-200**  **Swap 16G，内存大小8G\*2**  **/80G**  **/var 20G（可以不分）**  **/data 180G（存放web及db数据）** | **数据硬盘和系统盘分开，有利于出问题时维护** | **看数据及性能要求，一般可采用raid5折中** |
| **负载均衡器（如LVS）** | **/boot 100-200M**  **Swap 内存的1-2倍**  **/** | **简单方便，只做转发数据量少** | **数据量小，重要性高，可采用RAID1** |
| **负载均衡下的RS server** | **/boot 100-200M**  **Swap 内存的1-2倍** | **简单方便，因为有多机，对数据要求低** | **数据量大，重要性不高，有性能要求，数据要求低，可采用RAID0** |
| **数据库服务器mysql及oracle如内存16G/32G** | **/boot 100-200M**  **Swap 16G 内存的1倍**  **/ 100G**  **/data 剩余（存放db数据）** | **数据盘和系统盘分开，有利于出问题时及时维护，及保持数据完整** | **视数据及性能要求主库可采用raid10/raid5,从库克采用raid0提高性能（读写分离的情况下）** |
| **存储服务器（如NFS）** | **/boot 100-200M**  **Swap 内存的1-2倍**  **/ 100G**  **/data 剩余（存放db数据）** | **此服务器不要分区太多，只做备份，性能要求低，容量大** | **可采取sata盘，raid5** |
| **监控服务器**  **Cacti,Nagios** | **/boot 100-200M**  **Swap 内存的1-2倍**  **/** | **重要性一般，数据要求也一般** | **单盘或双盘，raid1即可，三盘就raid5,看容量要求加盘即可** |







**3、安装好初步使用**

**1）重启网卡**

**/etc/init.d/network restart**

**2）网卡的配置文件位置**

**/etc/sysconfig/network-scripts/ifcfg-eth0**

**3）dd 删除整行**

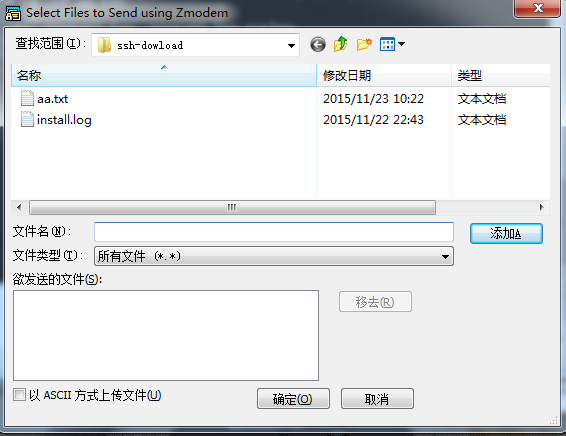
**4）ssh服务（监听本机ip对应的22端口）**

**5）用crt软件连接Linux下载文件**

**Sz –y 文件名**

**-y表示覆盖下载**

**Rz –y 回车出现如图：上传文件**



**Rz和sz命令是在dia-up networking support 包里如果没有安装可以用下面方式安装：**

**Yum install lrzsz –y**

**Yum groupinstall “dial-ip network support”**

**Rz 和sz只能上传和下载文件**

**4、Linux引导过程：**

**1）阶段一：系统BIOS检查系统并启动存放于硬盘的MBR(主引导记录)**

**2）阶段二：MBR把它自己载入内存，并启动其中的引导装载程序**

**3）阶段三：引导装载程序把内核载入到内存，然后按照顺序加载任何必须的模块并挂在“/”分区。**

**4）阶段四：内核把引导过程的控制权限转给/sbin/init程序**

**5）阶段五：引导起来的Linux系统把登陆界面呈现在用户面前。**

**MBR只有512字节大小，它包含了被称作“引导装载程序”的引导机器的代码指令，及分区列表。**

**如下图所示：**

**前446字节保存的是要在引导阶段三中加载的主引导装载程序，其中包含可执行代码和错误消息文本。接下来的64字节是分区表，其中包含4个分区的记录（每个记录的大小是16个字节）。最后2个字节作为MBR的结束符。**

**主引导记录**

分区表

引导加载程序

**S 446字节**

Magic Numberber

**64字节**

**2字节**

**当内核加载并进行初始化之后，内核就会启动系统中第一个应用程序/sbin/init**

**/sbin/init程序协调剩下的应道过程并读取/etc/inittab文件为用户配置环境**

### ****Linux系统引导过程详解****



GRUP菜单

MBR引导

开机自检（BIOS）

读取/etc/inittab配置文件

运行INIT进程

加载内核（Kernel）

执行 /etc/rc.d/rc.sysinit脚本

/etc/rc0.d/\*

/etc/rc1.d/\*

/etc/rc2/d/\*

/etc/rc3.d/\*

/etc/rc4.d/\*

/etc/rc5.d/\*

/etc/rc6.d/\*

执行/etc/rc.d/rc脚本

启动mingetty进程

**启动第一步- -加载BIOS**

**当打开计算机电源，计算机会首先加载BIOS信息，BIOS中包含了CPU的相关信息、硬盘信息、内存信息、时钟信息等**

**启动第二步- -读取MBR**

**磁盘上第0磁道第一个扇区称为MBR,也就是Master Boot Record即主引导记录，它的大小是512字节。**

**系统找到BIOS所指定的硬盘的MBR后，就会将其复制到0X7c00地址所在的物理内存中，其中被恢复到物理内存的内容就是Boot loader，而具体到你的电脑就是lilo或者grub.**

**启动第三步- -Boot loader**

**Boot loader就是在操作系统内核运行之前运行的一段小程序。**

**系统读取内存中的grub配置信息，并依照此配置信息来启动不同的操作系统。**

**启动第四步- - 加载内核**

**根据grub设定的内核映像所在路径，系统读取内存映像，并进行解压缩操作。**

**系统 解压后的内核放置在内存之中，并调用start\_kernel()函数来启动一些列的初始化函数并初始化各种设备，完成Linux核心环境的建立，至此，Linux内核已经建立起来，基于Linux的车刚需可以正常运行了**

**启动第五步- -用户层init依据inittab文件来设定运行等级**

**内核被加载后，第一个运行的程序便是/sbin/init，该文件会读取/etc/inittab文件，并依据此文件来进行初始化工作。**

**5、Linux中定义的运行级别**

|  |  |
| --- | --- |
| **运行级别** | **说明** |
| **0** | **关闭系统，该级别用来关闭系统，并切断电源** |
| **1** | **单用户文本模式，这是一种维护和应急模式** |
| **2** | **多用户文本模式，但不启动NFS** |
| **3** | **完全的多用户文本模式** |
| **4** | **未使用** |
| **5** | **完全的多用户图形模式** |
| **6** | **重新引导系统** |

**6、退出系统**

**1）exit**

**2）logout**

### ****系统的优化及初步了解****

1. **查看系统版本**

**[root@fan ~]# cat /etc/redhat-release**

**CentOS release 6.5 (Final)**

1. **一块硬盘最多有4个主分区**

**一块硬盘只能有一个扩展发分区，但这个分区内可以扩展多个逻辑分区**

**逻辑分区是从5开始，前4 是给主分区或扩展分区**

1. **IDE接口硬盘显示是hd**

**SCSI接口的硬盘显示sd**

**U盘显示是sdX**

1. **在Linux系统上划分了分区之后，还要在分区上创建文件系统**
2. **Linux 系统对分区的要求**

**（1）最少有/以及swap两个分区**

**（2）swap（交换分区）的作用：虚拟内存，swap分区的大小=1.5\*物理内存容量，当内存大于16G,虚拟内存一般给到8-16G即可。**

**（3）建议设置独立的/boot分区一般为100-200M**

**7、rpm –ivh 安装软件**

**Yum可以解决软件之间的依赖关系**

**Yum是一个包管理工具**

**Yum配置文件**

**/etc/yum.repos.d/centos-base.repo**

**默认情况下这个文件中的地址是外国的地址**

**所以需要改成国内镜像的网站**

[root@fan yum.repos.d]# wet http://mirrors.163.com/.help/CentOS6-Base-163.repo

-bash: wet: command not found

[root@fan yum.repos.d]# wget http://mirrors.163.com/.help/CentOS6-Base-163.repo

--2015-11-24 01:34:17-- http://mirrors.163.com/.help/CentOS6-Base-163.repo

Resolving mirrors.163.com... 123.58.173.185, 123.58.173.186

Connecting to mirrors.163.com|123.58.173.185|:80... connected.

HTTP request sent, awaiting response... 200 OK

Length: 2006 (2.0K) [application/octet-stream]

Saving to: 鈥淐entOS6-Base-163.repo.1鈥?

100%[=====================================================>] 2,006 --.-K/s in 0s

2015-11-24 01:34:27 (61.8 MB/s) - 鈥淐entOS6-Base-163.repo.1鈥?saved [2006/2006]

上面的内容是从国内镜像网站下载文件

cp CentOS-Base.repo CentOS-Base.repo.org 先将原始文件进行备份

在进行修改

[root@fan yum.repos.d]# cp CentOS6-Base-163.repo CentOS-Base.repo

**8、ntsysv 开启服务启动调整**

**Rsyslog 系统日志服务**

**Croud 定时日志服务**

**Ckconfig –list |grep “3:on”**

[root@fan yum.repos.d]# chkconfig --list|grep "3:on"

abrt-ccpp 0:off 1:off 2:off 3:on 4:off 5:on 6:off

abrtd 0:off 1:off 2:off 3:on 4:off 5:on 6:off

acpid 0:off 1:off 2:on 3:on 4:on 5:on 6:off

atd 0:off 1:off 2:off 3:on 4:on 5:on 6:off

auditd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

blk-availability 0:off 1:on 2:on 3:on 4:on 5:on 6:off

cpuspeed 0:off 1:on 2:on 3:on 4:on 5:on 6:off

crond 0:off 1:off 2:on 3:on 4:on 5:on 6:off

cups 0:off 1:off 2:on 3:on 4:on 5:on 6:off

haldaemon 0:off 1:off 2:off 3:on 4:on 5:on 6:off

ip6tables 0:off 1:off 2:on 3:on 4:on 5:on 6:off

iptables 0:off 1:off 2:on 3:on 4:on 5:on 6:off

irqbalance 0:off 1:off 2:off 3:on 4:on 5:on 6:off

kdump 0:off 1:off 2:off 3:on 4:on 5:on 6:off

lvm2-monitor 0:off 1:on 2:on 3:on 4:on 5:on 6:off

mcelogd 0:off 1:off 2:off 3:on 4:off 5:on 6:off

mdmonitor 0:off 1:off 2:on 3:on 4:on 5:on 6:off

messagebus 0:off 1:off 2:on 3:on 4:on 5:on 6:off

netfs 0:off 1:off 2:off 3:on 4:on 5:on 6:off

network 0:off 1:off 2:on 3:on 4:on 5:on 6:off

portreserve 0:off 1:off 2:on 3:on 4:on 5:on 6:off

postfix 0:off 1:off 2:on 3:on 4:on 5:on 6:off

rsyslog 0:off 1:off 2:on 3:on 4:on 5:on 6:off

sshd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

sysstat 0:off 1:on 2:on 3:on 4:on 5:on 6:off

**chkconfig name on/off**

**chkconfig --list查看设置开机自启动软件的命令**

**chkconfig例子**

**[root@fan ~]# chkconfig --list sshd**

**sshd 0:off 1:off 2:on 3:on 4:on 5:on 6:off**

**You have new mail in /var/spool/mail/root**

**[root@fan ~]# chkconfig sshd off**

**[root@fan ~]# chkconfig --list sshd**

**sshd 0:off 1:off 2:off 3:off 4:off 5:off 6:off**

**[root@fan ~]# chkconfig --level 3 sshd on**

**[root@fan ~]# chkconfig --list sshd**

**sshd 0:off 1:off 2:off 3:on 4:off 5:off 6:off**

**[root@fan ~]#**

**9、Linux 默认远程用户和端口**

**Root 端口是22**

**Ssh的配置文件**

**/etc/ssh/sshd\_config**

|  |  |
| --- | --- |
| **实际命令** | **命令说明** |
| **Su -** | **该命令是真正用户角色转换命令（默认是切到root）,输入root用户密码** |
| **Sudo su -** | **改命令是通过sudo权限进行角色转换（默认是切到root）,输入的是执行命令当时账号的密码，非root密码** |
| **Sudo su –oldboy** | **该命令实际意思是通过sudo，以root的权限，进行su-oldboy 用户切换，因此需要输入的是执行命令当时的账号的密码，和sudo ls /root是一样的** |

案例实践：

[root@oldboy oldboy]# useradd lifan #添加用户lifan

[root@oldboy oldboy]# passwd lifan  #设置密码123456

Changing password for user lifan.

New UNIX password:

BAD PASSWORD: it is too simplistic/systematic

Retype new UNIX password:

passwd: all authentication tokens updated successfully.

[root@oldboy oldboy]# useradd oldboy #添加用户oldboy

[root@oldboy oldboy]# passwd oldboy  #设置密码09876

Changing password for user oldboy.

New UNIX password:

BAD PASSWORD: it is too short

Retype new UNIX password:

passwd: all authentication tokens updated successfully.

[root@oldboy oldboy]# visudo   #编辑sudo文件在结尾加入如下行，使得oldboy可以通过root权限执行任务（具体意思后文会详细讲。）

oldboy  ALL=(ALL)       ALL

[root@oldboy oldboy]# su - oldboy #切到oldboy用户下。

[oldboy@oldboy ~]$ whoami

oldboy

[oldboy@oldboy ~]$ sudo su - lifan #本题的考试

We trust you have received the usual lecture from the local System

Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.

    #2) Think before you type.

    #3) With great power comes great responsibility.

Password: #先输入lifan的密码,123456

Sorry, try again.#提示密码不对。

Password: #在输入oldboy的密码,09876,没有在报错。

[lifan@oldboy ~]$ whoami

lifan

**10、内核的调优**

**将下面文字复制到/etc/sysctl.conf中**

N**et.ipv4.tcp\_fin\_timeout=2**

**Net.ipv4.tcp\_tw\_reuse=1**

**Net.ipv4.tcp\_tw\_recycle=1**

**Net.ipv4.tcp\_syncookies=1**

**Net.ipv4.tcp\_keepalive\_time=6000**

**Net.ipv4.ip\_l**ocal\_port\_range=4000 65000

Net.ipv4.tcp\_max\_syn\_backlog=16384

Net.ipv4.tcp\_max\_tw\_bluckets=36000

Net.ipv4.route.gc\_timeout=100

Net.ipv4.tcp\_syn\_retries=1

Net.ipv4.tcp\_synack\_retries=1

Net.core.somaxconn=16384

Net.core.netdev\_max\_backlog=16384

Net.ipv4.tcp\_max\_orphans=16384

防火墙的优化，在6.4上

Net.nf\_conntrack\_max=25000000

Net.netfilter.nf\_conntrack\_max=25000000

Net.netfilter.nf\_conntrack\_tcp\_timeout\_established=180

Net.netfilter.nf\_conntrack\_tcp\_timeout\_time\_wait=120

Net.netfilter.nf\_conntrack\_tcp\_timeout\_close\_wait=60

Net.netfilter.nf\_conntrack\_tcp\_timeout\_fin\_wait=120

**Sysctl –p让上面改过的内容生效**

|  |  |
| --- | --- |
| 参数 | 说明 |
| Net.ipv4.tcp\_fin\_timeout | 表示如果套接字由本端要求关闭这个参数决定了它保持在FIN-WAIT-2状态的时间，默认值是60秒 |
| Ntp.ipv4.tcp\_tw\_reuse | 表示开启重用允许将TIME-WAIT sockets重新用于新的tcp连接，默认为0，表示关闭 |
| Ntp.ipv4.tcp\_tw\_recycle | 表示开启tcp连接中TIME-WAIT sockets的快速回收  该参数对应系统路径为：/proc/sys/net/ipv4/tcp\_tw\_recycle 默认为0 表示关闭 |
| Net.ipv4.tcp\_syncookies | 表示开启SYN Cookies 功能，当出现SYN等待队列溢出时，启用cookies来处理，可防范少量SYN攻击，Centos5系列默认为1，表示开启，因此这个参数可以不添加 |
| Net.ipv4.tcp\_keepalive\_time | 表示当keepalive启用的时候，TCP发送keepalive消息的频度，缺省是2小时，改为10分钟 |
| Net.ipv4.ip\_local\_port\_range | 用来设定允许系统打开的端口号范围，即用于向外连接的端口方位。 |
| Net.ipv4.tcp\_max\_syn\_backlog | 表示SYN队列的长度，默认为1024,加大队列长度为8192，可以容纳更多等待连接的网络连接数，选项为服务器用于记录那些尚未收到客户端确认信息的连接请求的最大值。 |
| Net.ipv4.tcp\_max\_tw\_buckets | 表示保持TIME\_WAIT套接字的最大数量，如果超过这个数字TIME\_WAIT 套接字，TIME\_WAIT套接字将立刻被清除并大衣呢警告信息，默认为180000，对于apache,nignx等服务器来说可以调低一点，如：5000-30000  不同业务的服务器的也可以给大一点，比如lvs,squid  上几行的参数可以很好的减少TIME\_WAIT套接字数量，但是对于squid效果不大 |
|  |  |

**11、调试显示字符**

**Vi /etc/sysconfig/i18n**

**LANG=zh\_CN18030**

**必须牢牢记住：**

**关于如何优化Linux系统**

1. **不用root，添加普通用户，通过sudo授权管理**
2. **更改默认的远程连接ssh服务端口及禁止root用户远程连接**
3. **定时自动更新服务器时间**
4. **配置yum更新源，从国内更新源下载安装rpm包**
5. **关闭selinux及iptables（iptables工作场景如果有wan ip 一般要打开，高并发除外）**
6. **调整文件描述符的数量，进程及文件的打开都会消耗描述符**
7. **定时自动清理/var/spool/clientmquene目录垃圾文件，防止inodes节点被占满**
8. **精简开机自启动服务（crond,ssh,network,rsyslog）**
9. **Linux内核参数优化/etc/sysctl.conf执行sysctl –p生效**
10. **更改字符集支持中文，但建议还是用因为字符集，防止乱发问题，不要用中文**
11. **锁定关键系统文件**

**Chattr +i /etc/passwd /etc/shadow /etc/group /etc/gshadow /etc/inittab**

1. **清空/etc/issue，去除系统及内核版本登录前的屏幕显示**
2. **清楚无用的默认系统账户或组（非必须）**

## 三、Linux基本命令

**1）man 查看帮助**

**在查看帮助时可能内容过多想要看特定字符**

**例如查看 man find**

**因为现实的内容太多想要在内容中找到type**

**输入/type 回车就可以找到，如果想要找下一个按“n”**

**2）pwd 显示工作目录**

**[root@fan oldboy]# pwd**

**/home/oldboy**

**3）ls 列出目录内容，包括文件和子目录的名称。**

**-a 显示所有文件**

**-A 显示所有文件包括隐藏文件**

**-l 长格式输出**

**-F 在每个文件后附上一个字符以说明该文件的类型。**

**Ls –lrt 这样显示可以最快的将最近一次修改的文件显示在最下方**

**-r 表示翻转排序**

**-t 表示按修改时间排序**

**--color 以不同颜色显示文件**

**[root@fan ru]# ls --time-style=long-iso ru.txt -l**

**-rw-r--r--. 1 root root 23 2013-04-14 16:33 ru.txt**

**[root@fan ru]#**

[root@fan ~]# ls –l

total 48

-rw-------. 1 root root 1129 Nov 22 22:43 anaconda-ks.cfg

-rw-r--r--. 1 root root 13 Nov 23 05:24 ett.txt

-rw-r--r--. 1 root root 25600 Nov 22 22:43 install.log

-rw-r--r--. 1 root root 6240 Nov 22 22:42 install.log.syslog

**[root@fan ~]# ls -lhi**

**total 52K**

**273364 -rw-------. 1 root root 1.2K Nov 22 22:43 anaconda-ks.cfg**

**268001 -rw-r--r--. 1 root root 13 Nov 23 05:24 ett.txt**

**265558 -rw-r--r--. 1 root root 26 Nov 24 01:10 fan**

**260611 -rw-r--r--. 1 root root 25K Nov 22 22:43 install.log**

**260612 -rw-r--r--. 1 root root 6.1K Nov 22 22:42 install.log.syslog**

**[root@fan ~]#**



**用下面这个文件做例子**

**268001 -rw-r--r--. 1 root root 13 Nov 23 05:24 ett.txt**

**Inode: 268001**

**文件类型：- 表示这是一个普通文件**

**文件权限：文件权限为rw-r--r--**

**硬链接：表示ett.txt这个文件没有其他硬链接，以为链接数是1，就是它本身**

**文件属主：这个文件所属的用户，在这里是root**

**文件属组：这个表示文件所属的用户组，这里是root组**

**文件大小：文件大小是13个字节**

**文件修改时间：Nov 23 05:24这里的时间是该文件最后被更新的时间**

**[root@fan ~]# stat ett.txt**

**File: `ett.txt'**

**Size: 13 Blocks: 8 IO Block: 4096 regular file**

**Device: 802h/2050d Inode: 268001 Links: 1**

**Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)**

**Access: 2015-11-23 05:24:42.326997066 +0800**

**Modify: 2015-11-23 05:24:42.326997066 +0800**

**Change: 2015-11-23 05:24:42.326997066 +0800**

[root@fan ~]#

**Inode**

**中文的意思是索引节点，在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）被格式化为ext3文件系统后，一般有两部分：第一部分是Inode,第二步部分是Block.Block是用来存储实际数据用的，而inode就是用来存储这些数据属性的信息也就是ls –l的结果，inode包含的属性信息包括：大小，归属的用户组，读写权限，文件类型，修改时间。还包括指向文件实体的指针的功能，(inode节点—block的对应关系)，但是唯独不包括文件名。**

**Inode除了记录文件属性的信息外，还会为每个文件进行信息索引，所以就有了inode的数值，操作根据指令，即可通过inode的值最快的找到相对应的文件实体**

文件名

Inode

Type:regular files

-rw-r—r—

Access:此处没有文件名

Block文件的数据内容

**总结：**

1. **磁盘分区格式化文件系统后，会分为inode和block两部分内容**
2. **Inode存放文件的属性以及指向文件实体的指针，文件名不在inode里，一般上级目录的block**
3. **访问文件，通过文件名-🡪inode-🡪blocks**
4. **Inode一般情况默认大小是256k,block大小为124k,默认是4k,注意，引导分区等特殊分区除外**
5. **通过df –i 查看inode的数量及使用情况，dumpe2fs /dev/sda1 查看inode及block的大小及数量**
6. **一个文件至少占用一个inode及一个block，对多个文件文件可以占用同一个inode(硬链接)**
7. **一个block只能被一个文件使用，block太大浪费，如果文件很小block很大，剩余空间浪费无法继续被其他文件使用。**
8. **Block不是越大越好，要根据业务的文件大小进行选择，一般就是默认4k.**
9. **可以在格式化的时候改变inode及block的大小。**

**一般分区的情况block的数量要大于inode的数量**

**关于这部分的面试题：**

**企业面试题，一个100M的磁盘分区，分别写入1k的文件，以及1M的文件，分别可以写入多少？**

**a Inode是存放文件属性的默认大小是128byte(centos 5.8),256(centos 6.4)**

**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 默认分区情况下，inode是足够的，而block消耗的会更快，因此一般都是block消耗尽。假设block大小为X(1-4k),单位为K,假设文件大小为Y,当Y<X的时候，可以写多少个，当Y>=X的时候可以写多少个**

**假设Y<=X可以写入的文件的数量就是block的数量**

**Y>X可以写入的文件数量==总的block数量（Y/X）个**

**磁盘报错：“NO space left on device ”,但是df –h查看磁盘空间没满，为什么？**

**这个时候df –i查看可以发现inode已经满了**

**当我们用ls 查看某个目录或文件时，如果加上-i参数就可以看到inode节点**

**因为inode要存放文件的属性信息，因此inode是有大小的，Centos6的inode的默认大小是256字节，centos5是128字节，inode的大小在文件系统格式化之前是可以更改的，之后就无法更改。**

**总结：一个文件至少占用一个inode和一个block**

**Ext3/ext4文件系统block大小一般有1k,2k,4k其中引导分区等为1k其他普通分区为3k**

[root@fan ~]# dumpe2fs /dev/sda1 |grep -i "Block count"

dumpe2fs 1.41.12 (17-May-2010)

Block count: 204800

Reserved block count: 10240

[root@fan ~]# dumpe2fs /dev/sda1 | grep -i "Block count"

dumpe2fs 1.41.12 (17-May-2010)

Block count: 204800

Reserved block count: 10240

[root@fan ~]# dumpe2fs /dev/sda1 | grep -i "inode count"

dumpe2fs 1.41.12 (17-May-2010)

Inode count: 51200

[root@fan ~]# dumpe2fs /dev/sda1 | grep -i "inode size"

dumpe2fs 1.41.12 (17-May-2010)

Inode size: 128

**查看磁盘大小**

**[root@fan ~]# df -h**

**Filesystem Size Used Avail Use% Mounted on**

**/dev/sda2 16G 1.9G 13G 13% /**

**tmpfs 935M 0 935M 0% /dev/shm**

**/dev/sda1 194M 34M 151M 19% /boot**

**小结：**

**有关block的大小：磁盘读取数据是按block为单位读取的**

**一个block只能存放一个文件的内容，所以block并非越大越好**

**Block块越大对于单个小文件多的业务，会浪费空间，因为，一个文件无论多大都会必须占用一个inode和至少一个block，但是对于大文件，可以提升读取的效率，因为如果block太小，就要读多个block，这样就消耗磁盘IO,如果block大，则会读较少的block就读完数据，从而减少磁盘IO**

**Block 块太小又会影响硬盘读取大文件数据的效率，Block块越小，同样存储一个文件就需要更多的block，这样硬盘读取数据时就要读取多个block因此效率就越低**

**当前的生产环境一般设置4K,特殊的业务，如视频可以加大block**

**生产环境：block大小如何处理？**

**如果是大文件业务，block尽量给大一点**

**如果是小文件业务，block尽量给小一点**

**4）cd 切换目录**

**5）mkdir 建立目录**

**-p 可以是一个路径名称，用作多级目录的创建**

**[root@fan ~]# mkdir -p /a/b/c**

**[root@fan ~]# ls**

**anaconda-ks.cfg ett.txt install.log install.log.syslog**

**[root@fan ~]# cd**

**[root@fan ~]# cd /**

**[root@fan /]# ls**

**a boot dev home lib64 media opt root selinux sys usr**

**bin data etc lib lost+found mnt proc sbin srv tmp var**

**[root@fan /]# cd a/b/c**

**[root@fan c]# pwd**

**/a/b/c**

**6）rmdir 删除空目录**

**-p 递归删除空目录**

**7）file 辨别文件类型**

[root@fan ~]# file fan

fan: ASCII text

**8）cat 将文件的内容输出在显示器上**

**-n 从数字1开始对所有输出的行数编号**

**-b与-n相似只不过对于空白行不编号**

**[root@fan ~]# cat fan**

**wo**

**zheng**

**zai**

**xue**

**xi**

**linux**

**[root@fan ~]# cat -n fan**

**1 wo**

**2 zheng**

**3 zai**

**4 xue**

**5 xi**

**6 linux**

1. **more less 分屏显示文件中的内容**

**more 不能回滚，less可以回滚查看**

**10) head 显示文件的开始部分**

**-n 显示行数**

**默认是前10行**

**11）tail 显示文件的末尾部分**

**默认也是10行**

**-f 当文件不断增长时，输出后续添加的数据，用于监视日志文件的内容**

**12）三剑客：grep awk sed 它们都有过滤字符串的功能，grep最擅长，awk擅长处理列，sed擅长删除，修改，替换，添加**

**Sed擅长输出行**

**取行的例子：**

**[root@zhaofan ~]# df -h|awk '{print $5}'**

**Use%**

**3%**

**0%**

**16%**

**100%**

**[root@zhaofan ~]# df -h|awk '{print $5}'|sed -n "1p"**

**Use%**

**[root@zhaofan ~]# df -h|awk '{print $5}'|sed -n "2p"**

**3%**

**[root@zhaofan ~]#**

**Awk擅长打印列**

**Grep –v表示排除**

**如果想要保留什么就直接grep +要保留的内容**

**-E 表示可以排除**

**-i表示不区分大小写**

**Sed取各种内容**

**Sed –n(表示取消默认输出) /取的内容 /-p(表示打印) 文件**

**Sed /^ 取的内容 / -d (表示删除) 文件 把除了取的内容的其他内容显示出来**

**Awk ‘{if(NR>19&&NR<31)print$0}’ 显示文件的20-30行**

**Awk 如果想要显示某行的话：**

**例子：**

**-f 表示以什么分割**

**$0表示整行**

**Awk –f “:”’{print $1}’/etc/passwd**

**$NF:最后一个列**

**NF: 浏览记录的域的个数**

**例子:**

**[root@zhaofan scripts]# free -m|grep buffers\/**

**-/+ buffers/cache: 163 824**

**[root@zhaofan scripts]# free -m|grep buffers\/|awk '{print $NF}'**

**824**

**[root@zhaofan scripts]#**

**13)cp复制文件和目录**

**-f 删除已经存在的文件而不提示**

**14）mv 移动或更改现有文件或目录**

**-f (不给提示) 禁止交互操作**

**-i交互式操作**

**15）rm 删除文件或目录**

**-f 忽略不存在的文件，从不给提示**

**-i执行交互操作**

**-r 指示rm将参数中列出的全部目录和子目录均递归的删除**

**16) find 查找文件或目录**

**-type 文件是什么类型。类型取值：b块设备文件；c字符设备文件；d目录文件；p管道文件；f普通文件；l 符号（软）链接文件**

**-name 要搜索的文件名**

**-exec 执行命令**

**-perm 查找具有某种权限的文件**

**如下：**

[root@sersync ~]# find /usr/bin/ -type f -perm 4755 -exec ls -l {} \;

-rwsr-xr-x. 1 root root 54240 Jan 30 2012 /usr/bin/at

-rwsr-xr-x. 1 root root 71480 Dec 8 2011 /usr/bin/gpasswd

-rwsr-xr-x. 1 root root 51784 Nov 23 2013 /usr/bin/crontab

-rwsr-xr-x. 1 root root 66352 Dec 8 2011 /usr/bin/chage

-rwsr-xr-x. 1 root root 36144 Dec 8 2011 /usr/bin/newgrp

-rwsr-xr-x. 1 root root 30768 Feb 22 2012 /usr/bin/passwd

-rwsr-xr-x. 1 root root 27576 Sep 20 2013 /usr/bin/pkexec

[root@sersync ~]#

**-type c**

**File is of type c:**

**b block (buffered) special 块**

**c character (unbuffered) special 字符**

**d directory 目录**

**p named pipe (FIFO) 管道**

**f regular file**

**l symbolic link; this is never true if the -L option or the -fol-**

**low option is in effect, unless the symbolic link is broken. If**

**you want to search for symbolic links when -L is in effect, use**

**-xtype. 链接文件**

**s socket**

**D door (Solaris)**

**Find /var/ -type s –exec ls –l {}\;**

**查找显示管道文件**

**Find 中！作用是取反，-a为and表示并且，-o为or,两边有一个成立即可。**

**[root@fan ~]# find /var -type p |xargs ls -l**

**prw--w--w-. 1 postfix postfix 0 Nov 28 01:36 /var/spool/postfix/public/pickup**

**prw--w--w-. 1 postfix postfix 0 Nov 28 01:34 /var/spool/postfix/public/qmgr**

**[root@fan ~]#**

**17）du 显示文件或目录的大小**

**df 显示已经使用中的文件系统磁盘空间的使用情况。**

**dd 读取，转换并输出数据;if=输入文件；of=输出文件**

[root@fan /]# df

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/sda2 16407976 1946340 13628152 13% /

tmpfs 957248 0 957248 0% /dev/shm

/dev/sda1 198337 34160 153937 19% /boot

**18）which 命令名 查看命令的位置**

**[root@fan /]# which vi**

**/bin/vi**

**Locate /whereis /find**

**20）alias 显示设置别名**

**Vi /etc/profile**

**最后一行添加**

**Alias grep=’grep –color=auto’**

**这样以后过滤的内容都会有颜色**

[root@fan /]# alias

alias cp='cp -i'

alias l.='ls -d .\* --color=auto'

alias ll='ls -l --color=auto'

alias ls='ls --color=auto'

alias mv='mv -i'

alias rm='rm -i'

alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'

[root@fan /]#

**21) seq 打印行数**

**Seq 1 5 | tac 1-5 倒着显示出来**

**[root@fan ~]# seq 1 5 | tac**

**5**

**4**

**3**

**2**

**1**

[root@fan ~]#

**Set nu 显示文件中的行号**

**22）xargs 用于把find / ls的输出交给后面处理**

**23）uname –a –r –m print**

[root@fan ~]# uname -a

Linux fan 2.6.32-431.el6.x86\_64 #1 SMP Fri Nov 22 03:15:09 UTC 2013 x86\_64 x86\_64 x86\_64 GNU/Linux

[root@fan ~]# uname -r

2.6.32-431.el6.x86\_64

[root@fan ~]#

[root@fan ~]# uname -m

x86\_64

[root@fan ~]#

**24）halt 关机（init 0）**

**Reboot 重启（init 6）**

**Shutdown | -r now(重启) –h now(关机)**

**25）useradd 添加用户**

**Passwd 用户名 为用户设置密码**

[root@fan ~]# useradd zhaofan

l[root@fan ~]# passwd zhaofan

Changing password for user zhaofan.

New password:

BAD PASSWORD: it is WAY too short

BAD PASSWORD: is too simple

Retype new password:

passwd: all authentication tokens updated successfully.

[root@fan ~]#

**26）su – 用户名 切换用户**

**注意：切换用户时一定要有“-”否则环境变量不会变**

**27）批量给用户设置密码不产生交互**

**Echo 密码|passwd –stdin 用户名**

**28）last显示最后登录过的用户列别**

**[root@fan ~]# last**

**ru pts/0 192.168.101.56 Fri Nov 27 13:24 still logged in**

**ru pts/3 192.168.101.56 Tue Nov 24 02:30 - 13:24 (3+10:54)**

**ru pts/2 192.168.1.103 Tue Nov 24 02:24 - 12:27 (3+10:02)**

**ru pts/1 192.168.1.103 Tue Nov 24 02:06 - 12:19 (3+10:12)**

**root pts/1 192.168.1.103 Tue Nov 24 01:58 - 02:04 (00:06)**

**root pts/0 192.168.1.103 Tue Nov 24 01:52 - 11:54 (3+10:02)**

**root pts/0 192.168.101.56 Mon Nov 23 18:11 - 01:50 (07:39)**

**root pts/1 192.168.101.56 Mon Nov 23 13:50 - 18:18 (04:27)**

**root pts/0 192.168.101.56 Mon Nov 23 13:11 - 15:22 (02:11)**

**root pts/0 192.168.101.56 Mon Nov 23 12:59 - 13:04 (00:04)**

**root pts/0 192.168.101.56 Mon Nov 23 11:40 - 12:59 (01:19)**

**root pts/1 192.168.101.56 Mon Nov 23 05:12 - 11:36 (06:23)**

**root pts/0 192.168.1.103 Mon Nov 23 04:08 - 07:21 (03:12)**

**root pts/1 192.168.101.56 Mon Nov 23 02:32 - 04:51 (02:18)**

**root pts/0 192.168.101.56 Sun Nov 22 23:32 - 03:58 (04:25)**

**root pts/0 192.168.101.56 Sun Nov 22 23:14 - 23:27 (00:13)**

**root pts/0 192.168.101.56 Sun Nov 22 23:11 - 23:13 (00:02)**

**root tty1 Sun Nov 22 23:01 still logged in**

**reboot system boot 2.6.32-431.el6.x Sun Nov 22 23:00 - 14:34 (4+15:34)**

**root tty1 Sun Nov 22 22:44 - crash (00:16)**

**reboot system boot 2.6.32-431.el6.x Sun Nov 22 22:43 - 14:34 (4+15:50)**

**wtmp begins Sun Nov 22 22:43:53 2015**

**[root@fan ~]#**

**[root@fan ~]# lastlog**

**Username Port From Latest**

**root pts/1 192.168.1.103 Tue Nov 24 01:58:18 +0800 2015**

**bin \*\*Never logged in\*\***

**daemon \*\*Never logged in\*\***

**adm \*\*Never logged in\*\***

**lp \*\*Never logged in\*\***

**sync \*\*Never logged in\*\***

**shutdown \*\*Never logged in\*\***

**halt \*\*Never logged in\*\***

**mail \*\*Never logged in\*\***

**uucp \*\*Never logged in\*\***

**operator \*\*Never logged in\*\***

**games \*\*Never logged in\*\***

**gopher \*\*Never logged in\*\***

**ftp \*\*Never logged in\*\***

**nobody \*\*Never logged in\*\***

**dbus \*\*Never logged in\*\***

**vcsa \*\*Never logged in\*\***

**abrt \*\*Never logged in\*\***

**haldaemon \*\*Never logged in\*\***

**ntp \*\*Never logged in\*\***

**saslauth \*\*Never logged in\*\***

**postfix \*\*Never logged in\*\***

**sshd \*\*Never logged in\*\***

**tcpdump \*\*Never logged in\*\***

**oprofile \*\*Never logged in\*\***

**oldboy \*\*Never logged in\*\***

**zhaofan \*\*Never logged in\*\***

**ru pts/0 192.168.101.56 Fri Nov 27 13:24:57 +0800 2015**

**29）tree -L显示层数，-d列目录 查看目录的深度**

**30）cd 切换目录**

**在切换目录之后想回到刚才所在的目录最快的方法是cd –**

**Cd ~ 切换到家目录**

**31）ln 创建链接**

**32）tar 打包命令**

**Tar –zcvf 压缩文件的名字 要打包的文件目录或文件**

**[root@fan /]# tar zcvf etc.tar.gz ./etc**

**-z gzip**

**-c create 建立新的归档文件**

**-v verbose 建立归档文件**

**-f file 对普通文件操作**

**-x extract 从归档文件中解出来**

**-j bzip2 用gnu的bzip2压缩文件或解压**

**-c directory DIR 指定目录进行操作，如加压到指定目录**

**--exclude 排除不需要处理的文件或目录**

**Tar –zcvf oldboy2.tar.gz –exclude=oldboy/jeacen ./oldboy表示排除**

**Tar zxvf oldboy1.tar.gz 是解包**

**-C 指定解包到的目录**

**-p 保持文件属性**

**Tar 小结：**

**Tar gzip打包组合zcvf ,解包组合zxfv**

**bzip打包组合jcvf ,解包组合jxfv**

**z（gzip）c（create）,v(verbose),f(file),x(extract)**

**j（bzip）**

**--exclude（排除） –X（--exclude -form）,C指定目录解压**

**P保持属性**

**打包的规则：进入打包文件或目录的上级目录进行打包**

**33）cut –d指定分隔符，-f数字选列，-cq按字符 切割**

**练习：**

**[root@fan ~]# awk '{print $3" " $6}' oldboy.txt**

**oldboy 49000448**

**[root@fan ~]# cat oldboy.txt**

**i am oldboy myqq is 49000448**

**[root@fan ~]#**

**[root@fan ~]# cut -c 6-11,20- oldboy.txt**

**oldboy 49000448**

**[root@fan ~]#**

**[root@fan ~]# awk '{print $3 $5}' oldboy.txt|sed s#,myqq#" "#g**

**oldboy 49000448**

**[root@fan ~]# awk -F '[ , ]' '{print $3" " $6}' oldboy.txt**

**oldboy 49000448**

**[root@fan ~]#**

**在不指定—F分隔符的情况下，默认的分隔符是空格**

**-F表示以什么分割**

**35）wc –l 直接查看文件行数**

**[root@fan ~]# wc -l /etc/services**

**10774 /etc/services**

**[root@fan ~]#**

**所有查行号的都可以Cat –n file |taile –1**

**Grep –c “” 文件名 统计行数**

**36) lsof 查看文件的占用情况**

**Lsof –i:端口号 可以查看端口的使用情况**

**37）Linux系统下常见获取帮助的手段：**

**A.man 命令名**

**B.命令—help**

**C.额外：搜索引擎“Linux命令名”**

**D.help 命令名 内置命令帮助**

**查看是否是内置命令：type 命令名**

**重要提示：man不仅可以用于获取命令的帮助，同样可以获取到配置文件，函数的帮助。**

**38）date显示时间**

**Date +%m/%d/%y 指定date命令输出的日期和时间格式**

**[root@fan ru]# date +%m/%d/%y**

**04/14/13**

**[root@fan ru]#**

**使用-s选项可以更改系统当前的日期和时间**

**%y年**

**%x 显示日期的格式（mm/dd/yy）**

**%m 月**

**%d 天**

**[root@fan log]# date +%F**

**2013-12-05**

**[root@fan log]#**

**[root@fan ru]# date 12042129.34**

**Wed Dec 4 21:29:34 CST 2013**

**将时间修改为12月4号21点29分34秒**

**39）Echo –n 不换行输出**

**-e可以指定特殊的字符**

**40）hwclock 这样就可以把时间写道BIOS中**

**41）fdisk 磁盘分区的命令 –l 查看列表 小于2T的分区工具**

**Parted GNU分区工具 适合大于2T的分区工具**

**Mount 挂载的命令 –t 文件类型 -o 挂在的选项 –a**

**Umount 卸载的命令 –lf 强制卸载**

**Mkfs 格式化命令 –t 文件系统类型**

**42）uptime查看负载**

**43）mail –s “标题”** [**邮件地址**](mailto:邮箱@qq.com) **<文件内容**

**Mail –s “zhaofan”** [**867468837@qq.com**](mailto:867468837@qq.com) **</etc/hosts**

**Echo “正文”| Mail –s “zhaofan”** [**867468837@qq.com**](mailto:867468837@qq.com)

**Mailq 可以查看邮件队列**

**以上需要sendmail服务**

**44）文件的压缩**

**gzip 压缩或解压缩文件**

**gzip是一个使用广泛的压缩程序，若没有加上任何参数，gzip压缩完文件后悔产生.gz的压缩文件，并删除原始的文件**

**-d 将压缩文件解压**

**-r 递归的查找指定目录并压缩其中的所有文件或者解压缩**

**-t 测试，检查压缩文件是否完整**

[root@A ~]# ls

anaconda-ks.cfg install.log install.log.syslog newfile

[root@A ~]# gzip newfile

[root@A ~]# ls

anaconda-ks.cfg install.log install.log.syslog newfile.gz

[root@A ~]# gzip -d newfile.gz

[root@A ~]# ls

anaconda-ks.cfg install.log install.log.syslog newfile

[root@A ~]# cat newfile

wo shi zhaofan

[root@A ~]#

**bzip2**

**.zip2文件的压缩和解压缩程序**

**同样的也会删除原始文件**

**-d 执行解压缩**

**-t 检查指定文件的完整性，但并不对其解压缩**

**zip**

**创建zip格式的压缩文件**

**文件压缩后会另外产生具有.扩展名的压缩文件，而不会删除原有文件。**

**-d 从压缩文件内删除指定的文件**

**-P<密码> 为加密而使用的压缩密码**

**-T 检查压缩文件内的每个文件是否正确无误**

**Zip 压缩文件 原始文件**

**Unzip**

**解压缩zip文件**

**-d 把压缩文件解压到指定目录下**

**-t 检查压缩文件是否正确**

**-P<密码> 使用解压缩密码**

**45）ps命令是最常用的获的关于当前进程数据的方法**

**-a 列出带有控制终端的全部进程，不仅仅是当前用户的进程**

**-r 只列出正在运行中的进程**

**-x列出没有控制终端的进程**

**-u 列出进程的所有者**

**-f 给出进城之间的父子关系**

**-l 按长格式显示清单**

**Ps最常使用的命令是aux**

**46）top命令和ps命令的作用基本相同的，用于显示当前系统当前的进程和其他状况，但是top是一个动态显示的过程**

**Top命令执行后会将用户置于其可交互的状态中，这个会话默认是没3秒刷新一次**

**q 退出top命令的交互模式**

**47）kill 杀死进程**

**Kill -9 进程pid**

**显示后台有哪些程序运行**

**Jobs**

**Ctrl+z 是程序转入到后台，但是不会执行，而是出于停止状态，可以用bg，让其在后台执行，如果想要显示在前台用fg**

**48）uniq 去重复**

**-c 显示重复的个数**

**49）sort 排序**

**-n 按照数字排序**

**-rn 按照数字倒序**

**-k 对于某列排序**

**-t 指定分隔符**

**如果-k3,3 表示从第三个字段开始排序，到第三个字段结束**

**-k4,1,4,3从第四个字段的第一个字符开始派讯，到第四个字段的第三个字段结束**

**（.点号连接的是字符，逗号连接的是字段）**

**[root@A ~]# sort -n -t. -k3,3 -k 4.1,4.3 a.txt**

**192.168.0.151 00:0F:AF:85:6C:F6**

**192.168.0.153 00:0F:AF:85:70:03**

**192.168.1.1 00:0F:AF:81:19:1F**

**192.168.1.11 00:30:15:A3:23:B7**

**192.168.1.21 00:0F:AF:85:6C:09**

**192.168.1.152 00:0F:AF:83:1F:65**

**192.168.2.2 00:0F:AF:85:6C:25**

**192.168.2.20 00:0F:AF:85:55:DE**

**192.168.2.22 00:0F:AF:85:5C:41**

**192.168.3.3 00:0F:AF:85:70:42**

**192.168.3.10 00:30:15:A2:3B:B6**

**192.168.3.12 00:30:15:A2:3A:A1**

**[root@A ~]#**

**[root@A ~]# awk -F "/" '{print $3}' oldboy.log|sort**

**mp3.etiantian.org**

**post.etiantian.org**

**post.etiantian.org**

**www.etiantian.org**

**www.etiantian.org**

**www.etiantian.org**

**[root@A ~]# uniq**

**^[[A^H^[[B^[[B^Z**

**[1]+ Stopped uniq**

**[root@A ~]# awk -F "/" '{print $3}' oldboy.log|sort|uniq**

**mp3.etiantian.org**

**post.etiantian.org**

**www.etiantian.org**

**[root@A ~]# awk -F "/" '{print $3}' oldboy.log|sort|uniq -c**

**1 mp3.etiantian.org**

**2 post.etiantian.org**

**3 www.etiantian.org**

**[root@A ~]#**

**[root@A ~]# awk -F "/" '{print $3}' oldboy.log|sort|uniq -c|sort -rn**

**3 www.etiantian.org**

**2 post.etiantian.org**

**1 mp3.etiantian.org**

**[root@A ~]#**

**50）rename 修改文件名**

**Rename 改的内容 改成什么 对什么文件进行修改**

**51）清空历史记录**

**history –c**

**删除某一条**

**history –d 数字**

**HISTCONTROL=ignorespace 不记录敏感历史记录**

**52）driname**

**53）basename**

**54）printf**

### ****Vi编辑器****

1. **vi编辑器的三种模式：命令模式，末行模式，输入模式**
2. **输入i则可以从命令模式进入输入模式**
3. **q 如果用户只是读文件的内容而未对文件进行修改，可以使用这个末行命令退出**

**q! 强制退出**

**wq 保存退出**

**x 保存退出**

**dd 删除一行**

**p 粘贴**

**yy 复制一行**

**数字+yy 复制多行**

**U 撤销对一行所做的所有更改**

**u取消上一次操作**

**在文件中向下搜索某个字符 /字符**

**在文件中向下搜索某个字符 ？字符**

**Ctrl+r 重做**

**Shift +g 进入到文件的结尾**

**gg 进入到文件的开头**

**o 在光标的下一行打开**

**O 在光标的上一行打开**

**Ctrl+a到当前命令的最前面**

**Ctrl+e当前命令的最后面**

**set nu显示行号**

**set nonu 取消显示行号**

**$到文件当前行的结尾**

**0 到文件当前行的开头**

**：%s/old/new 替换文件中所有的字符**

**：%s/old/new 逐行确认**

**Vi编辑器的时候查找某个单词或字符：**

**输入：/字符**

**如果找下一个输入n**

**4）字符串的替换**

**Sed s#原来的内容#要更改成的内容#g 文件名 批量改文件的内容**

**-i 就会把文件中更改，不加的只是输出显示**

**Sed /文件中的某个内容/d 文件名 这样就会不显示文件中的某个内容，只显示其他的内容**

**Ls –l|sed –n ‘/^d/p‘（最好加上单引号）**

**这样就是列出以d开头的文件目录**

### ****快捷方式：****

**Ctrl+a 回到开头**

**Ctrl+e回到最后**

**Ctrl+c 结束当前进程**

**Ctrl+d结束当前shell**

**Ctrl+l 清屏==clear**

## 四、系统目录结构

1. **Linux基本目录**

|  |  |
| --- | --- |
| **目录名** | **简介** |
| **/** | **根目录，文件系统结构的起始点** |
| **/bin** | **是binary的缩写，这个目录沿袭了Unix系统的结构，存放着使用者最经常使用的命令** |
| **/dev** | **是device的缩写。这个目录下存放的是被Linux识别到的设备，其功能类似Windows下的.vxd** |
| **/etc** | **这个目录用来存放系统管理所需要的配置文件和子目录** |
| **/home** | **普通用户的宿主目录** |
| **/lib** | **这个目录里存放着系统最基本的动态链接共享库，其作用类似于windows 里的dll文件。几乎所有的应用程序都需要用到这些共享库** |
| **/lost+found** | **该目录在大多数情况下都是空的，但当突然停电或者非正常关机后，系统产生的零散文件（没有名字的），就存放在这里** |
| **/mnt** | **该目录中通常包括系统引导后需要挂在的文件系统挂在点、例如默认的光盘挂载点是/mnt/cdrom** |
| **/proc** | **这个目录是一个虚拟目录，它是系统内存的映射，用户可以通过直接访问这个目录来获取系统信息。也就是说这个目录的内容不在硬盘上而是在内存里** |
| **/root** | **如果用户是以root账户的身份登录的，这个就是超级用户的宿主目录** |
| **/sbin** | **这里的s就是super user的意思，也就是说这里存放的是超级用户使用的管理程序** |
| **/tmp** | **用来存放不同程序执行时产生的临时文件** |
| **/usr** | **这是系统中最庞大的目录，用户要用到的应用程序和文件几乎都存放在这个目录下** |
| **/var** | **这个目录中存放着那些不断扩充的东西，为了保持Linux的稳定性，那些经常被修改的文件可以放在这个目录下。一般是日志文件的目录。** |

### ****重要子目录的说明****

|  |  |
| --- | --- |
| **目录** | **说明** |
| **/etc/sysconfig/network-scripts/ifcfg-eht0** | **配置网络地址及GW等**  **/etc/init.d/network restart**  **Ifdown eth0**  **Ifup eth0** |
| **/etc/resolv.conf** | **设置Linux本地的客户端DNS文件** |
| **/etc/hosts** | **设置用户IP与名字的对应解析表，相对应本地lan内的DNS** |
| **/etc/sysconfig/network** | **可修改机器名及网卡启动** |
| **/etc/fstab** | **记录开启要mount的文件的第一个文件**  **/etc/rc.local** |
| **/etc/inittab** | **设定系统启动时init进程将把系统设置成什么样的runnlevel及加载相关的启动文件设置** |
| **/etc/exports** | **设定NFS网络文件系统用的配置文件** |
| **/etc/xinit.d** | **如果服务器时通过xinetd模式运行的，它的脚本要放在这个目录下，有些系统没有这个目录** |

### ****/etc/目录下的重要目录和文件****

|  |  |
| --- | --- |
| **文件** | **说明** |
| **/etc/issue** | **记录用户登录前显示的信息**  **[root@fan ~]# cat -n /etc/issue**  **1 CentOS release 6.5 (Final)**  **2 Kernel \r on an \m**  **3**  **[root@fan ~]#** |
| **/etc/motd（默认是空的）** | Last login: Tue Nov 24 02:30:55 2015 from 192.168.101.56  welcome to zhaofan Linux  登陆提醒 |
| **/etc/redhat-release** | **包括一行声明Redhat版本号和名称的信息**  [root@fan ~]# cat /etc/redhat-release  CentOS release 6.5 (Final)  [root@fan ~]# uname -r  2.6.32-431.el6.x86\_64  [root@fan ~]# uname -m  x86\_64  [root@fan ~]#  **[root@fan ~]# lsb\_release -a**  **LSB Version: :base-4.0-amd64:base-4.0-noarch:core-4.0-amd64:core-4.0-noarch:graphics-4.0-amd64:graphics-4.0-noarch:printing-4.0-amd64:printing-4.0-noarch**  **Distributor ID: CentOS**  **Description: CentOS release 6.5 (Final)**  **Release: 6.5**  **Codename: Final**  **[root@fan ~]#** |
| **/etc/group** | **设定用户的组名与相关xinxi** |
| **/etc/passwd** | **账号信息文件** |
| **/etc/shadow** | **密码信息文件** |
| **/etc/sudoers** | **可执行使用sudo命令的配置文件** |
| **/etc/securetty** | **设定哪些终端可以让root登陆** |
| **/etc/login.defs** | **所有用户登录时缺省配置** |
| **/etc/modprobe.conf** | **内核模块额外参数设定** |
| **/etc/syslog.conf** | **日志设置文件** |
| **/etc/rsyslog.conf** | **日志设置文件** |
| **/etc/rc.local** | **存放开机自启动内容的文件。Chkconfig一般用来管理yum、rpm包装的服务** |

### ****其他目录****

|  |  |
| --- | --- |
| **目录** | **说明** |
| **/lib** | **系统会使用到的函数库** |
| **/lib/modules** | **Kernel的相关模块** |
| **/var/lib/rpm** | **Rpm套件的安装处** |
| **/usr/bin** | **这个目录时可执行程序的目录；当我们从系统自带的软件包安装一个程序时，他的可执行文件大多会放在这个目录相似的/usr/local/sbin;** |
| **/lib64** | **库文件存放目录** |
| **/usr/sbin** | **这个目录也是可执行程序的目录，但大多存放涉及系统管理的命令；相似的目录时/sbin或/usr/local/sbin** |
| **/usr/local** | **这个目录一般是用来存放用户自编译安装软件的存放目录；一般是通过源码包安装的软件，如果没有特别指定安装目录的话，一般是安装在这个目录下；相当于c:\program file** |
| **/usr/local/bin** | **用户安装的小的应用程序，和一些在/usr/local目录下大应用程序的符号连接** |

### ****Var目录下的路径知识****

|  |  |
| --- | --- |
| **目录** | **说明** |
| **/var** | **日志文件** |
| **/var/log** | **各种系统日志存放地** |
| **/var/log/messages** | **系统信息默认日志文件，非常重要，按周自动循环**  **[root@fan ~]# ll /var/log/messages \***  **-rw-------. 1 root root 1129 Nov 22 22:43 anaconda-ks.cfg**  **-rw-r--r--. 1 root root 13 Nov 23 05:24 ett.txt**  **-rw-r--r--. 1 root root 26 Nov 24 01:10 fan**  **-rw-r--r--. 1 root root 25600 Nov 22 22:43 install.log**  **-rw-r--r--. 1 root root 6240 Nov 22 22:42 install.log.syslog**  **-rw-------. 1 root root 185127 Nov 27 11:40 /var/log/messages** |
| **/var/log/secure** | **记录登入系统存取信息的文件，按周自动轮循 系统安全的日志** |
| **/var/log/wtmp** | **记录登陆者信息的文件** |
| **/var/spool**  **/var/spool/cron/root** | **定时任务crontab默认路径，按照用户名命名的文件** |
| **/var/spool/mail** | **系统用户邮件存放目录** |
| **/var/spool/clientmqueue** | **Sedmail临时文件目录 会造成磁盘不够用**  **64位不会装sedmail,所以不会有这个问题** |

### ****/proc下的重要路径知识(内核调优)****

|  |  |
| --- | --- |
| **目录** | **说明** |
| **/proc** | **虚拟目录，是内存的映射，内核和进程的虚拟文件系统目录** |
| **/proc/version** | **内核版本** |
| **/proc/sys/kernel** | **系统内核功能** |
| **/proc/sys/net/ipv4** |  |
| **/proc/cpuinfo** | **关于处理器的信息，如类型，厂家和性能等** |
| **/proc/meminfo** | **系统内存信息** |
| **/proc/iterrupts** | **正在使用的中断，和曾将有多少个中断** |
| **/proc/loadavg** | **系统负载平均值信息（系统繁忙的情况，比较准确，但是不够细致系统性能指标），uptime的结果，值不要超过cpu的核数** |

### ****文件类型****

**Linux系统中文件类型包括：**

**普通文件、目录文件、字符设备、块设备文件、符号链接设备等等。**

**Linux中常见的文件类型：普通文件、目录文件、链接文件、管道文件、设备文件。**

**普通文件**

**1）纯文本文件（ASCII）: 内容可直接读取，数字字母等，比如配置文件，几乎都是这种类型。**

**2）二进制文件:Linux中的可执行文件比如命令文件，cat，ls，cp**

**3）数据格式文件:具有特定格式的文件**

**设备文件分为：块设备和字符设备**

**常见的字符设备：终端和串行接口。**

|  |  |
| --- | --- |
| **颜色** | **文件类型** |
| **蓝色** | **表示目录文件** |
| **绿色** | **表示可执行文件** |
| **红色** | **表示压缩文件** |
| **浅蓝色** | **表示链接文件** |
| **灰色或白色** | **表示普通文件** |
| **紫色** | **表示图形文件** |
| **红色闪烁的** | **表示链接文件但是有问题** |
| **棕色** | **表示管道文件** |
| **黄色** | **表示设备文件** |

**Linux下扩展名**

**1）.tar、.tar.gz、.tgz、.zip、.tar.bz表示压缩文件，创建命令一般为tar,gzip、zip等**

**2）.sh表示shell脚本文件，通过shell语言开发的程序**

**3）.pl表示perl语言文件，通过python语言开发的程序。**

**4）.py表示python语言文件，通过python语言开发的程序**

**5).html、htm、php 、.jsp、.do表示网页语言的文件**

**6）.conf表示系统服务的配置文件**

**7）.rpm 表示rpm安装包文件**

**运维开发语言：shell、python、 perl、 php**

**正则表达式**

**^ 表示以什么什么开头例如^zhaofan表示以zhaofan开头的**

**[^]表示非以什么开头。**

**$表示以什么什么结尾例如zhaofan$表示以zhaofan字符串结尾**

**^$意思是空行**

**[^zhaofan] ^意思是非，及不是zhaofan字符串**

**\ 让一个字符显示他本来的意义不让它其作用**

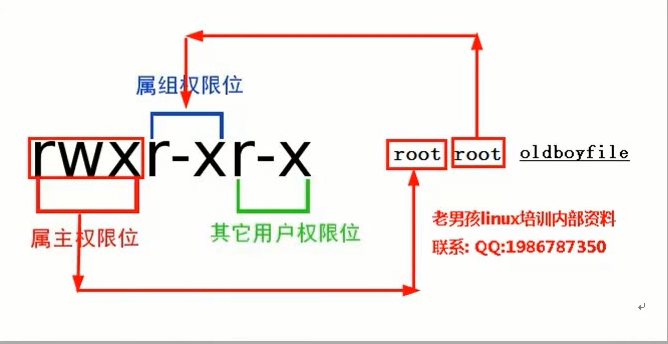
**运维思想：**

**当一个命令的输出，含有我们需要的内容的时候，我们要想到的是这个命令可能会有参数直接显示，我们需要的内容。**

**学习思想：**

**先走通后变通**

### ****文件属性知识进阶****



**文件属主（Owner）:读read 写 write 执行x**

**文件用户组（Group）：读r 写w 执行x**

**其他用户（other）：读 写 执行**

**如果权限位不可读，不可写不可执行，则用-表示**

**后面还会有特殊的权限位：t T S s X +**

#### ****改变权限属性命令****

**Chmod是用来改变文件或目录权限的命令，但只有文件的属主和超级用户root才有这种权限。通过chmod来改变文件或目录的权限有两种方法：一种是通过权限字母和操作符表达方式的方法来设置权限；另外一种是使用数字方法来设置权限。**

**如果我们想改变的仅仅是打开目录的权限，使用chmod命令时不用加任何参数。如果想把目录以下的所有文件或子目录也同时改变需要使用-R参数**

**Chmod字符方法权限表示法**

**Chmod [用户类型] [+ - =] [权限字符] 文件名**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Chmod** | **用户类型** | **操作字符** | **权限字符** | **文件或目录** |
| **U（user）** | **+（加入）** | **r** |
| **G（group）** |  |  |
| **O（other）** | **-（减去）** | **w** |
| **A（all）** | **=(设置)** | **x** |

**权限增减符：**

**+：添加某个权限**

**-：取消某个权限**

**=：取消其他所有权限赋予给定权限**

#### ****改变文件所属关系的命令chown****

**Chown [选项]…[所有者][:[组]] 文件…**

**方法： -R 可以递归**

**Chown 用户 文件或目录 >>仅仅授权用户**

**Chown :组 文件或目录 >>仅仅授权组**

**Chown 用户 :组 文件或目录 >>表示授权用户和组**

**强调：其中的冒号：可以用“.”替代。要授权的用户和组名必须是系统里存在的**

**Chgrp 改变用户和组 —R 可以递归**

**删除一个文件的时候是删除的文件名，而文件名是存放在上级目录的block里，所以一个目录下文件能否删除需要看上级目录的权限。**

**删除文件和文件本身的属性没有关系和文件的所在的上级目录有关系**

#### ****文件和目录权限总结：****

**文件权限总结：文件**

**可读r：表示具有读取阅读文件内容的权限**

**可写w:表示具有新增，修改文件内容的权限**

**特别提示：删除文件（修改文件名）的权限是收父目录的权限控制，和文件本身权限无关**

**可执行x:表示具有执行文件的权限（文件本身要能够执行，普通用户同时还需要具备r的权限root用户不需要r权限也能执行，当然文件本山也需要能够执行）**

**文件权限总结：目录**

**可读r:表示具有浏览目录下面文件及目录的权限，及ls dir（不能进入到目录即无法cd dir）如果没有x权限，列表时也会有问题。**

**可写w:表示具有增加。删除，或修改目录内文件名的权限（需要权限配合）**

**可执行x:表示具有进入目录的权限；及cd dir**

**Linux文件与目录权限对比说明：**

**r （读取权限）**

**对文件而言，表示具有阅读文件内容的权限**

**对目录而言，有浏览目录权限（与进入目录的权限不同，也需要x权限配合）**

**如果没有x属性，那么看不到目录内文件的属性**

**W(写权限)**

**对文件而言，表示具有新增，修改文件内容的权限（注意：删除和移动文件和文件本身属性无关，看上级目录）**

**X(执行权限)**

**对文件而言，表示具有执行文件的权限，（普通用户同时牙需要r的权限，root用户不用r也能执行，文件本身也要能执行才行）**

**对目录来说，表示具有进入目录的权限**

**再次强调：**

**当删除或移动一个文件或目录，仅仅与该文件与目录所在的上一层目录权限有关，与本身文件属性没有任何关系。**

**对于文件来说，写文件是修改文件，而不是删除文件，因此写文件是与文件的本身属性有关系。**

**Umask：控制linux系统默认权限的一个命令：**

**File:默认是644**

**Dir：默认是755**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **默认最大权限** | **Umask值** | **用户创建的权限** |
| **创建文件（Umask所有位数为偶数）时** | **666** | **022（全偶数）** | **644** |
| **偶数加减法（最大权限）-（umask权限）=（用户创建的权限）** | | |
| **创建文件（umask部分或全部为奇数）时:** | **666** | **123（部分位为奇数）** | **644（因为umask的值123中有2位都是奇数，因此在计算结果543的基础上再加101即奇数对应的文件数字权限位分别加1即可）** |
| **说明：（默认最大权限）-（umask权限）+（umask奇数对应权限位加1）=（用户创建的权限）** | | |
| **创建目录时：** | **777** | **022** | **755** |
|  | **说明：（默认最大权限）-（umask权限）=（用户创建的权限）** | | |

**Linux特殊权限suid，sgid粘滞位权限。（生产中用不到）**

1. **链接：**

**在Linux系统中。链接可以分为两种：一种为硬链接，一种为软链接。软链接相当于快捷方式**

**Ln 可以创建链接，如果不带参数，默认是硬链接**

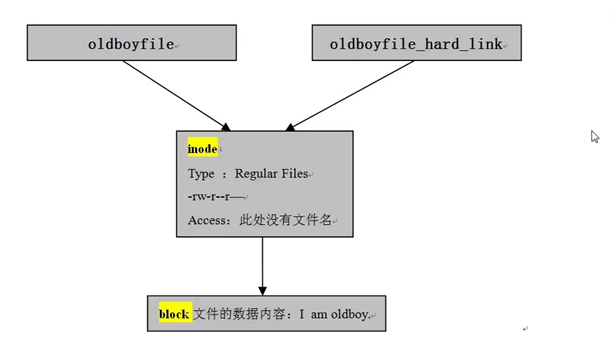
**要创建软件链接 需要-s**

**硬链接：**

**硬链接是指通过索引节点（inode）来进行链接，在Linux文件系统中，报存在磁盘分区中的文件不管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（inode index）检车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。硬链接的作用之一是允许一个文件有多个有效路径名（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“误删除”源数据。**

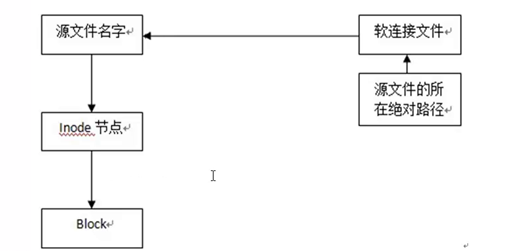
**硬链接示意图：**



**小结：**

1. **硬链接文件是具有相同inode节点的不同文件**
2. **删除硬链接文件或者删除源文件之一，文件实体并没有被删除**
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，再存放新的数据会占用这个文件的空间，包括inode和block.磁盘会做fsck检查的时候数据也会被回收。**
5. **硬链接文件是文件的又一个入口**
6. **可以通过给文件设置硬链接文件来防止文件被误删除。**
7. **通过ln 源文件 硬链接文件 即可完成创建硬链接**
8. **Rm –f 即可删除**
9. **软链接**

**软链接也称为符号链接，Linux中的软链接就类似于windows系统中的快捷方式。**



**软链接的创建：“ln –s 源文件 软链接文件“**

**查看软链接的value**

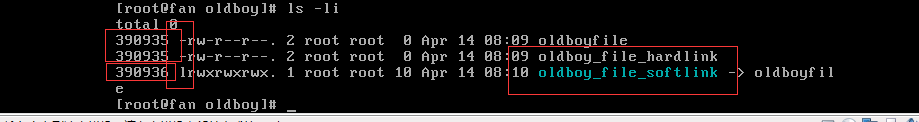
**Readlink 源文件**

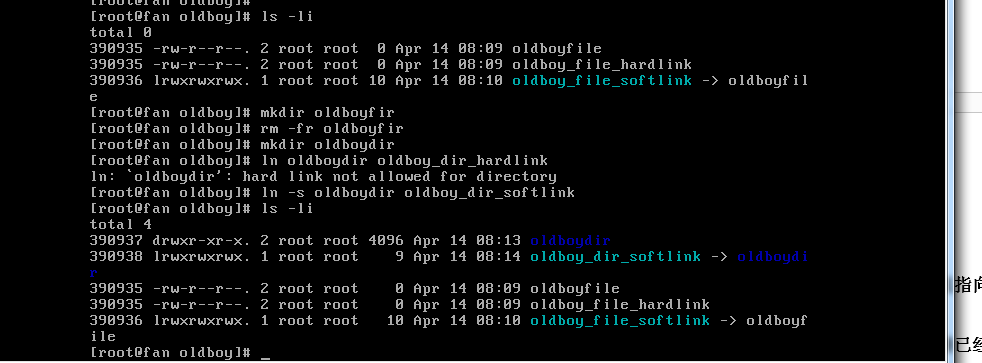
**软链接文件的inode是不同的。**

**小结：**

1. **软链接类似windows里的快捷方式，软链接里面存放的是源文件的路径，并且指向源文件。**
2. **删除源文件，软链接文件依然存在，但是无法访问指向的路径内容，即软链接已经无法使用**
3. **创建软链接的方法ln –s 源文件 软链接文件**
4. **软链接和源文件是不同类型的文件，也是不同的文件，及inode号不同**
5. **软链接的删除依然用rm –f 软链接文件**

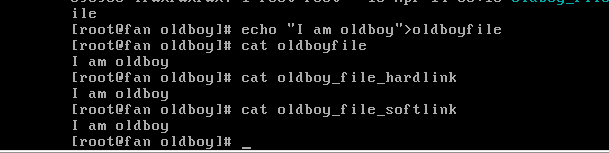
**例子**





**目录不能创建硬链接，但是可以创建软链接**

**小结：ln命令不能对目录创建硬链接，但是可以对目录创建软链接，对目录创建软链接也是生产场景运维中常用的功能。**



**结论：往源文件里写内容，它的软硬链接文件同样都有相同的内容**

[root@fan oldboy]# rm -f oldboyfile

[root@fan oldboy]# cat oldboy\_file\_hardlink

I am oldboy

[root@fan oldboy]# cat oldboy\_file\_softlink

cat: oldboy\_file\_softlink: No such file or directory

[root@fan oldboy]#

结论:删除源文件，硬链接不受影响，但是软链接不能用了

**全局结论：**

**删除软链接对源文件及硬链接无任何影响**

**删除硬链接文件对源文件及软链接文件无任何影响**

**删除源文件，对硬链接没有影响，但是会导致软链接失效，白字红底闪烁**

**同时删除源文件和硬链接，整个文件会完全删除**

**源文件和硬链接具有相同的索引节点号，可以认为是同一个文件或一个文件的多个入口**

**源文件和软链接文件索引节点不同，是不同的文件，软链接相当于源文件的快捷方式**

**有关目录链接小结：**

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环境运维中常用的技巧**
3. **目录的硬链接不能跨越文件系统**
4. **每个目录 都有一个硬链接“.”号和上级目录的链接“..”**
5. **在目录里创建一个目录，父目录的链接处会增加1（子目录里都有..来指向父目录），但是在父目录里创建文件，父目录的链接数不会增加**

**创建一个目录：/oldboy/ett**

**查看ett的硬链接：**

[root@fan ~]# ls -lid /oldboy/ett

390943 drwxr-xr-x. 2 root root 4096 Apr 14 2013 /oldboy/ett

[root@fan ~]#

**可以发现是2,原因：因为除了本身之外还有在这个目录里还有一个.**

**[root@fan ett]# ls -ali**

**total 8**

**390943 drwxr-xr-x. 2 root root 4096 Apr 14 2013 .**

**390942 drwxr-xr-x. 3 root root 4096 Apr 14 2013 ..**

**[root@fan ett]#**

**可以发现这两个的inode是一致的，所以硬链接数为2**

**而在ett目录下的..则是上一级目录的硬链接，即oldboy的硬链接为3**

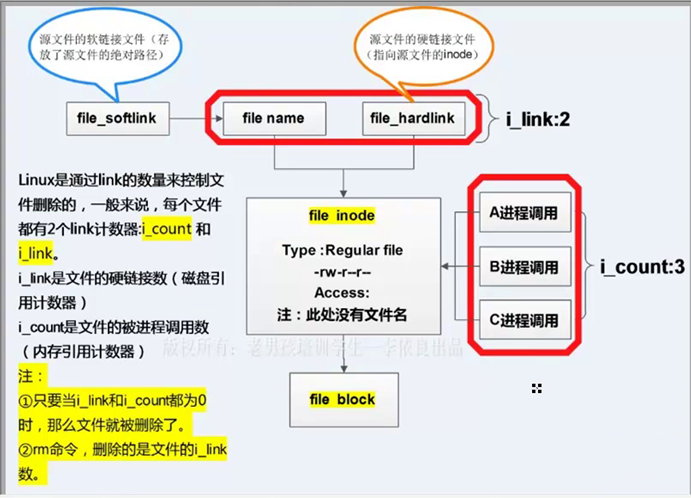
**控制文件删除:以下两个条件要同时具备，才生效。**

1. **文件的硬链接数，有一个硬链接i\_link +1,减少一个硬链接，i\_link-1,当i\_link=0时，文件就被删除**

**例：创建一个文件i\_link为这个文件创建一个硬链接i\_link=2，删除一个硬链接，i\_link-1,删除源文件i\_link=0**

1. **受进程占用控制，当有一个进程占用这个文件i\_count+1，即i\_count=1，当减少一个进程占用i\_count-1,即i\_count=0 。当没有进程调用，i\_count=0，**

**当i\_link=0并且i\_count=0时文件才会真正的被删除。**



### ****用户和用户组的介绍****

1. **用户的角色是通过UID和GID来识别的：特别是UID,在 linux系统运维中，一个UID是唯一标识一个用系统用户的账号。**

**超级用户：**

**默认是root用户，其UID和GID都是0.root用户在每台Linux操作系统中都是唯一真实存在的。在生产环境中，一般会禁止root账户通过ssh远程连接服务器，当然了，也会更改ssh端口，以增加系统安全性**

**普通用户：**

**安装系统后默认存在，且默认情况下大多数不能登陆系统，但是他们是系统正常运行不能缺少的，他们的存在主要是方便系统管理，满足相应的系统进程对文件属主的要求。**

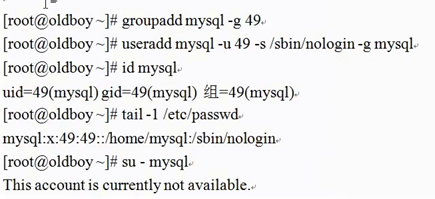
1. **Linux系统中不同用户角色对应UID说明**

|  |  |  |
| --- | --- | --- |
| **UID** | | **具备UID用户特性** |
| **0** | **超级用户** | **用户的UID为0表示这个用户为超级用户管理员账号，如果要增加一个系统管理员账号的话，只需要将该账户的UID改为0即可，当然不建议这么做，尽量用sudo替代** |
| **1-499** | **虚拟用户** | **这个方位是保留给系统使用的UID，之所以这样划分是为了防止人为建立UID和系统UID冲突，并没有其他特殊含义** |
| **500-65535** | **普通用户** | **普通账户UID，当使用useradd 创建用户时，默认情况下UID就是从500开始的。** |

**按照上面的方法创建的用户不能登陆**

**服务的运行时需要用户角色的可以不用登陆，因此工作中需要运行mysql**

**可以创建如下用户：**

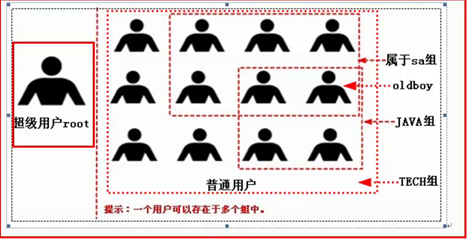


**在生产环境中，我们一般会为每个有权限管理服务器的运维人员分配一个独立的普通用户及8以上的复杂密码，并且该运维人员只能只能通过建立的这个账号登陆到系统中进行维护，当需要超级用户权限是，可以通过sudo命令方式来执行仅有root用户才允许执行的权限，当然sudo权限尽量小，还有当运维人数不多时，如2-3个，也可以直接su- 切换到超级用户root下，在执行相应的维护工作，在这里需要特别提醒一下：维护时，如果不需要root权限，就不要进入root用户下操作，以减少误操作对系统带来的损失。**

**Sudo 和su 是两个重要切换用户角色的命令。**

|  |  |
| --- | --- |
| **实际命令** | **命令说明** |
| **Su –**  **Su –root** | **该命令是真正用户角色转换命令，默认切换到root，输入的是root的密码** |
| **Sudo su** | **该命令是通过sudo权限进行角色转换（默认是切换到root）,输入执行命令当时账户的密码，非root密码** |
| **Sudo su-oldboy** | **该命令实际意思是通过sudo，以root的权限，进行su –oldboy用户切换，因此需要输入的是执行命令当时账号的密码，和sudo ls /roots是一样的** |

1. **用户和用户组的关系：一对一、一对多、多对一和多对多。如下图：**



**一对一：即一个用户可以存在一个组中，也可以是组中的唯一成员。如：root**

**一对多：即一个用户可以存在于多个用户组中，**

**多对一：多个用户可以存于一个组中，这些用户具有相同的权限。**

**多对多：多个用户可以存在多个组中。并且几个用户可以归属相同的组；其实多对多的关系是前面三条的扩展。**

1. **用户及用户组配置文件**

**Linux系统下账户文件主要有/etc/passwd 、/etc/shadow、/etc/group、/etc/gshadow四个文件。**

1. **与用户相关的配置文件**

**/etc/passwd-🡪用户的配置文件**

**Passwd文件中一行的各个字段详细说明**

|  |
| --- |
| **Root :x :0 :0 :root :root :/bin/bash** |
| **账户名称 账号密码 账号UID 账号GID 用户说明 用户家目录 shell解释器** |

1. **用户的影子口令文件/etc/shadow**

**Shadow文件的字段详细说明：**

|  |  |
| --- | --- |
| **字段名称** | **注释说明** |
| **账户名称** | **用户的账户名称** |
| **账户密码** | **用户密码，这是建国的口令** |
| **最近更改密码时间** | **从1970年1月1日起到用户最近一次更改口令的天数** |
| **禁止修改口令天数** | **从1970年1月1日起到用户可以更改密码的天数** |
| **用户必须更改口令天数** | **从1970年1月1日起到用户必须更改密码的天数** |
| **警告更改密码的期限** | **在用户密码过期之后到禁用账户的天** |
| **不活动时间** | **在用户密码过期之后到禁用账户的天数** |
| **失效时间** | **从1970年1月1日起到用户被禁止的天数** |
| **标志** | **保留** |

**5、 文件属性中的三个时间戳**

**当用touch创建一个文件的时候：**

**[root@fan ru]# stat ru.txt**

**File: `ru.txt'**

**Size: 0 Blocks: 0 IO Block: 4096 regular empty file**

**Device: 802h/2050d Inode: 273437 Links: 1**

**Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)**

**Access: 2013-04-14 16:24:57.398328395 +0800**

**Modify: 2013-04-14 16:24:57.398328395 +0800**

**Change: 2013-04-14 16:24:57.398328395 +0800**

**[root@fan ru]#**

**可以看到三个时间是一样的**

1. **当cat ru.txt后**

[root@fan ru]# stat ru.txt

File: `ru.txt'

Size: 0 Blocks: 0 IO Block: 4096 regular empty file

Device: 802h/2050d Inode: 273437 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2013-04-14 16:28:18.402843291 +0800

Modify: 2013-04-14 16:24:57.398328395 +0800

Change: 2013-04-14 16:24:57.398328395 +0800

[root@fan ru]#

**这个时候可以看出access的时间变了Modify和change的时间没有变化**

1. **当echo追加里面内容后**

**[root@fan ru]# stat ru.txt**

**File: `ru.txt'**

**Size: 11 Blocks: 8 IO Block: 4096 regular file**

**Device: 802h/2050d Inode: 273437 Links: 1**

**Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)**

**Access: 2013-04-14 16:28:18.402843291 +0800**

**Modify: 2013-04-14 16:30:12.074965394 +0800**

**Change: 2013-04-14 16:30:12.074965394 +0800**

**[root@fan ru]#**

**这个时候是Modify和change时间变了而access时间没有变化**

1. **当用vi编辑器修改里面内容后**

[root@fan ru]# stat ru.txt

File: `ru.txt'

Size: 23 Blocks: 8 IO Block: 4096 regular file

Device: 802h/2050d Inode: 273439 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2013-04-14 16:31:58.480877338 +0800

Modify: 2013-04-14 16:31:58.480877338 +0800

Change: 2013-04-14 16:31:58.483877702 +0800

[root@fan ru]#

**这个时候可以看出三个时间全变了**

**和再次用tocuh的效果是一样的三个时间都会变化**

#### ****管理用户和组命令汇总****

|  |  |
| --- | --- |
| **Useradd** | **Add user命令，执行命令可在系统中添加用户** |
| **Userdel** | **执行此命令可以删除用户及相关用户的配置文件** |
| **Passwd** | **执行此命令可为用户设置密码** |
| **Chage** | **修改用户密码有限期限** |
| **Usermod** | **修改用户的命令，可以通过usermod来修改登陆名，用户的家目录等** |
| **Id** | **查看用户的uid，gid及归属的用户组** |
| **Su** | **用户切换工具** |
| **Sudo** | **是通过另一个用户来执行命令，su是用来切换用户，然后通过切换到的用户来完成相应的任务，但sudo能在后面直接接命令执行。** |
| **Visudo** | **配置sudo权限的编辑命令也可以不用这个命令直接用vi来编辑/etc/sudoers实现，但推荐使用visudo来操作** |
| **Pwcov** | **同步用户从/etc/passwd到/etc/shadow** |
| **Groupadd** | **添加组** |

**1）/etc/skel目录**

[root@fan ~]# ll /etc/skel/ -al

total 28

drwxr-xr-x. 2 root root 4096 Nov 22 2015 .

drwxr-xr-x. 92 root root 12288 Dec 5 12:22 ..

-rw-r--r--. 1 root root 18 Jul 18 21:19 .bash\_logout

-rw-r--r--. 1 root root 176 Jul 18 21:19 .bash\_profile

-rw-r--r--. 1 root root 124 Jul 18 21:19 .bashrc

**/etc/skel目录是用来存放新用户配置文件的目录，当我们添加新用户时，这个目录所有的文件会自动被复制到新添加用户的家目录下；默认情况下，/etc/skel目录下的所有文件都是隐藏的。通过修改添加删除/etc/skel/目录下的文件我们可以为新创建的用户提供统一，标准、初始化用户环境。**

**2）/etc/login.defs**

**3）/etc/default/useradd**

[root@fan ~]# cat /etc/default/useradd

# useradd defaults file

GROUP=100

HOME=/home

INACTIVE=-1

EXPIRE=

SHELL=/bin/bash

SKEL=/etc/skel

CREATE\_MAIL\_SPOOL=yes

[root@fan ~]#

#### ****Useradd用户参数选项****

|  |  |
| --- | --- |
| **-c comment** | **新账号password档的说明栏（注释）** |
| **-d home\_dir** | **新账号每次登陆时所在的home\_dir预设置为ddefault\_home内login名称，并当成登入时目录名称** |
| **-e expire\_date** | **账号终止日期。日期的格式定位MM/DD/YY** |
| **-f inactive\_days** | **账号过期几日后永久停权当值为0时账号则立刻被停权，而当值等于-1时，则关闭此功能，预设值为-1** |
| **-g initial\_group** | **Group名称或数字来作为用户登入起始用户组。用户组名必须为系统现有存在的名称。用户组数字也必须为现有存在的用户组，预设的用户数字为1** |
| **-G group，[……]** | **定义此用户为多个不同groups的成员每个用户组使用逗号分隔。用户组名同-g选项的限制。默认为用户的其实用户组** |
| **-m** | **用户目录如果不存在则自动建立** |
| **-M** | **不建立用户家目录，优先于/etc/login.defs文件设定** |
| **-s shell** | **用户登陆使用的shell名称。默认值为不填写这样系统会帮你指定预设的登入shell（根据/etc/default/useradd 预设的值）** |
| **-u uid** | **用户的ID值。这个值必须是唯一的。除非用-o选项。数字不可为负值。** |

**实例：**

useradd etiantian

[root@fan ~]# ls -ld /home/etiantian

drwx------. 2 etiantian etiantian 4096 Jan 2 14:42 /home/etiantian

[root@fan ~]# grep etiantian /etc/passwd

etiantian:x:503:503::/home/etiantian:/bin/bash

**查看用户的过期以及详细信息**

root@fan ~]# chage -l root

Last password change : Nov 22, 2015

Password expires : never

Password inactive : never

Account expires : never

Minimum number of days between password change : 0

Maximum number of days between password change : 99999

Number of days of warning before password expires : 7

[root@fan ~]#

**Useradd –D就是改变/etc/default/useradd文件的内容。**

[root@fan ~]# cat /etc/default/useradd

# useradd defaults file

GROUP=100

HOME=/home

INACTIVE=-1

EXPIRE=

SHELL=/bin/bash

SKEL=/etc/skel

CREATE\_MAIL\_SPOOL=yes

[root@fan ~]#

**Groupadd涉及的文件包括：**

**/etc/group 用户组相关文件**

**/etc/gshadow 用户组加密相关文件**

[root@fan ~]# groupadd k1 -g 1000

[root@fan ~]# tail -1 /etc/group

k1:x:1000:

[root@fan ~]#

**Passwd命令常用参数**

**--stdin 从stdin读入密码（用非交互式的方式设置密码）**

[root@fan ~]# echo "123456"|passwd --stdin ru

Changing password for user ru.

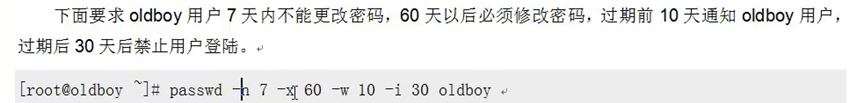
passwd: all authentication tokens updated successfully.

[root@fan ~]#

[root@fan ~]# passwd -l ru

Locking password for user ru.

passwd: Success





Userdel

删除用户-r可以把家目录也删除

Groupdel 如果该组是某个用户的主组时不能被直接删除

Usermod 修改用户信息

和useradd命令的参数基本一致。

-L冻结密码

-U取消冻结

### 关于sudo



1. 时间戳的位置：

6.4的位置是在/var/db/sudo

5.8的位置是在/var/runn/sudo

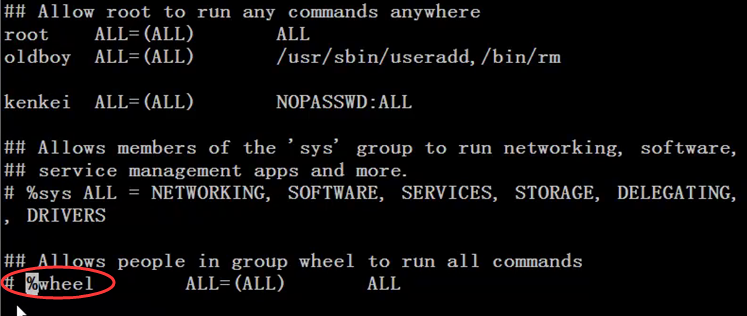
2）sudo –l 列出用户在主机上可用的和被禁止的命令；当配置好sudo授权规则之后，可以用这个参数来查看授权情况

3）sudo的配置文件/etc/sudoers

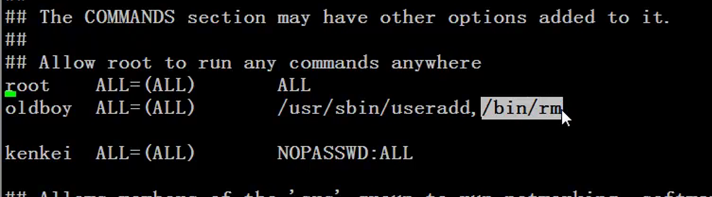
Visudo=vi /etc/sudoers

在98行（98gg可直接跳转到98行）

Visudo –c 检查语法



如果是组的话用%号开头



回顾别名和具体授权配置的关系

|  |  |  |  |
| --- | --- | --- | --- |
| 用户或组 | 主机 | 可以切换的用户角色 | 命令 |
| Root | All= | (all) | All |
| User\_Alias ADMINS = jsmith,mikem,%groupname | Host\_Alias  FILESERVERS =fs1,fs2 | Runas\_Alias op=root | Cmnd\_Alias SERVERS=/sbin/services,/sbin/chkconfig |

Sudo配置注意事项：

1. 命令别名下的成员必须是文件或目录的绝对路径
2. 别名名称是包含大写字母、数字、下划线，如果是字母都要大写。
3. 一个别名下有多个成员，成员与成员之间，通过半角“，”号分隔；成员必须是有效实际存在的
4. 别名成员受到别名类型Host\_Alias、User\_Alias、Runas\_Alias、Comnd\_Alias制约，定义什么类型的别名，就要有什么类型的成员相配。
5. 别名规则是每行算一个规则，如果一个别名规则一行容不下时，可以通过“\”来续行。
6. 指定切换的用户要用（）括起来，如果省略括号，则默认为root用户，如果括号里是ALL,则代表可以切换到所有用户
7. 如果不需要密码直接运行命令的应该加NOPASSWD参数
8. 禁止某类程序或命令执行，在命令动作前面加“！”，放在允许命令的后面
9. 用户组必须加“%“

## ****五、正则表达式****

### ****基础正则表达式****

**1、简单的说正则表达式就是一套处理字符串的规则和方法，以行为单位对字符串进行处理，通过特殊的符号的辅助，我们可以快速的过滤，替换某些特定的字符串**

**2、正则表达式实际就是一些特殊字符，赋予了他特定的含义。**

**1）^word 搜索以word开头的**

**2）world$ 搜索以world机尾的**

**3） . 代表且只能代表任意一个字符**

**4）\ 转移符号，让有着特殊身份意义的字符，失去意义，只作为符号**

[root@fan ~]# grep -n . oldboy.log

1:i am oldboy linux teacher

2:i like chinese chess,table tennis.

3:My blog is http://oldboy.blog.51cto.com

4:My qq is 49000448

5:my god,my name is not oldbey,but OLDBOY.

6:is blog not boog

7:not boog

[root@fan ~]#

[root@fan ~]# grep "\." oldboy.log

i like chinese chess,table tennis.

My blog is http://oldboy.blog.51cto.com

my god,my name is not oldbey,but OLDBOY.

[root@fan ~]#

**此处的点号表示任意单个字符，-n是给过滤出的每一行加行号**

1. **\* 重复0个或多个前面的一个字符**

**[root@fan ~]# grep "490\*448" oldboy.log**

**My qq is 49000448**

**[root@fan ~]# grep "^490\*448" oldboy.log**

**[root@fan ~]# grep "490\*448" oldboy.log**

**My qq is 49000448**

**[root@fan ~]# echo "49000000448,48448" >>oldboy.log**

**[root@fan ~]# grep "490\*448" oldboy.log**

**My qq is 49000448**

**49000000448,48448**

**[root@fan ~]# grep "490.\*448" oldboy.log**

**My qq is 49000448**

**49000000448,48448**

**[root@fan ~]# grep ".\*" oldboy.log**

**i am oldboy linux teacher**

**i like chinese chess,table tennis.**

**My blog is http://oldboy.blog.51cto.com**

**My qq is 49000448**

**my god,my name is not oldbey,but OLDBOY.**

**is blog not boog**

**not boog**

**49000000448,48448**

**[root@fan ~]#**

1. **[] 字符集合的重复特殊字符的符号。**

[root@fan ~]# grep "b[lo]og" oldboy.log

My blog is http://oldboy.blog.51cto.com

is blog not boog

not boog

[root@fan ~]#

**8）[^world] 表示不是world的内容（不包含^后任意一个字符的都会被匹配）**

**[root@fan ~]# grep "[^qq]" oldboy.log**

**i am oldboy linux teacher**

**i like chinese chess,table tennis.**

**My blog is http://oldboy.blog.51cto.com**

**My qq is 49000448**

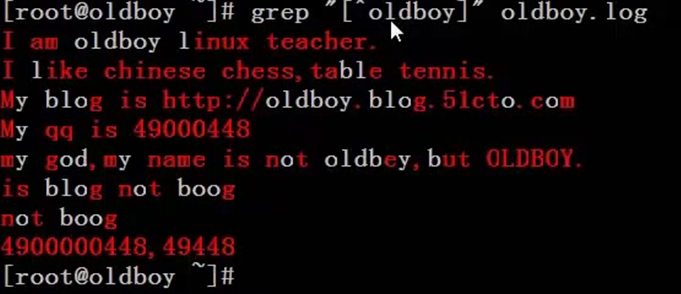
**my god,my name is not oldbey,but OLDBOY.**

**is blog not boog**

**not boog**

**49000000448,48448**

**[root@fan ~]#**



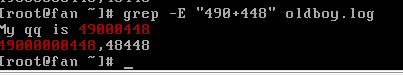
**9）a\{n,m\}重复n到m次前一个重复的字符，如果用egrep可以去掉斜线**

**a\{n,\}重复至少n次，前一个重复的字符**

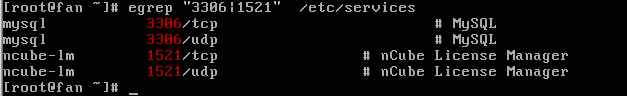
**\{n\}重复次，前一个重复的字符**

### ****扩展正则表达式****

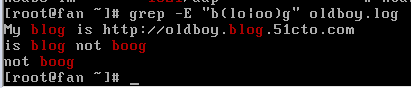
1. **+ 表示重复一个或一个以上前面的字符**



1. **？ 重复0个或一个前面的字符**
2. **| 用或的方式查找多个符合的字符串**



**4） () 找出“用户组“字符串**



**课前练习：**

**取出IP地址：**

第一种方法：

[root@fan ~]# ifconfig eth0 |grep "inet addr:"|cut -d ":" -f2 |cut -d " " -f1

192.168.101.62

第二种方法：

**[root@fan ~]# ifconfig eth0 |grep "inet addr:"|awk -F ":" '{print $2}' | awk -F " " '{print $1}'**

**192.168.101.62**

**[root@fan ~]#**

**第三种方法：**

**[root@fan ~]# ifconfig eth0 |grep "inet addr:"|awk -F "[: ]+" '{print $4}'**

**192.168.101.62**

**[root@fan ~]#**

**+表示重复前面的多个作为一个分隔符**

**第四种方法：**

**[root@fan ~]# ifconfig eth0 | sed -n "2p"|awk -F "[: ]+" '{print $4}'**

**192.168.101.62**

**第五种方法：**

**[root@fan ~]# ifconfig eth0|awk -F "[: ]+" 'NR==2 {print $4}'**

**192.168.101.62**

**[root@fan ~]#**

**第六种方法：**

**[root@fan ~]# ifconfig eth0| sed -n '/inet add/p' |sed 's#^.\*addr:##g'|sed 's# Bc.\*##g'**

**192.168.101.62**

**[root@fan ~]#**

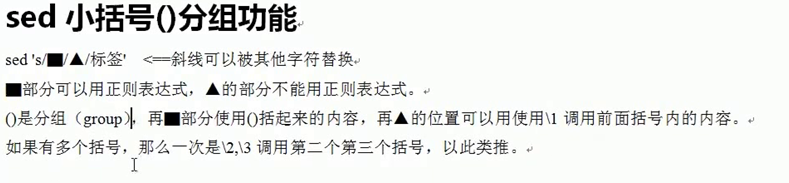
**第七种方法：**

**[root@fan ~]# ifconfig eth0 | sed -n 's#^.\*addr:\(.\*\) Bcast.\*$#\1#gp'**

**192.168.101.62**

**[root@fan ~]#**

**http://oldboy.blog.51cto.com/2561410/950730**



练习：

[root@fan ~]# cat b.txt

ddd kkk

[root@fan ~]# sed 's#\(ddd\) \(kkk\)#\1#g' b.txt

ddd

[root@fan ~]#

[root@fan ~]# sed 's#\(ddd\) \(kkk\)#\1 \2#g' b.txt

ddd kkk

**[root@fan ~]# ifconfig eth0|sed -n 's#^.\*dr:\(.\*\) Bcast:\(.\*\) Ma.\*$#\1 \2#gp'**

**192.168.101.62 192.168.101.255**

**[root@fan ~]#**

**[root@fan ~]# ifconfig eth0|sed -n 's#^.\*net\(.\*\) 00.\*$#\1#gp'**

**HWaddr**

**[root@fan ~]# ifconfig eth0|sed -n 's#^.\*net \(.\*\) 00.\*$#\1#gp'**

**HWaddr**

**[root@fan ~]# ifconfig eth0|sed -n 's#^.\*net \(.\*\) 00.\*$#\1#gp'**

**HWaddr**

**[root@fan ~]#**

**取出etiantian 文件的权限：**

**[root@fan ~]# stat etiantian |sed -n '4p'|awk -F "[/(]" '{print $2}'**

**0644**

**[root@fan ~]#**

**[root@fan ~]# stat etiantian |sed -n 's#^.\*s: (\(.\*\)/-rw.\*$#\1#gp'**

**0644**

**[root@fan ~]#**

**[root@fan ~]# stat -c %a etiantian**

**644**

**[root@fan ~]#**

**取消空行输出：**

**[root@fan oldboy]# sed '/^$/d' test.txt**

**oldboy**

**xizi**

**xiaochao**

**[root@fan oldboy]#**

**[root@fan oldboy]# grep -v "^$" test.txt**

**oldboy**

**xizi**

**xiaochao**

**[root@fan oldboy]#**

**[root@fan oldboy]# awk /^[^$]/ test.txt**

**oldboy**

**xizi**

**xiaochao**

**[root@fan oldboy]#**

## ****六、系统定时任务****

**每个进程启动时，系统会指定一个唯一的数字给每个进程，这个数字称为进程id或pid.**

**每个进程启动都可能以两种方式存在：一个前台，一个后台。**

**一般的系统网络服务都是以后台方式存在的，且都会常驻在系统中，直到刮泥或者主动杀死时**

**第一个带有进程或者守护进程的服务。Crond定时任务服务（守护进程）。**

**Linux的定时任务crond(crontab)服务**

**Crond是Linux系统中用来定期执行命令或者指定程序任务或者软件**

**Crond服务默认情况下（每分钟）会检查系统中是否有需要执行的定时任务，如果有就会根据事先定义好的规则来执行这个定时任务。**

**Linux定时任务分类：**

1. **系统自身的定期执行的任务**
2. **用户执行的定时任务**

### ****Linux的定时任务分类****

1. **at适合执行一次就结束的跳读任务。突发性的任务。依赖服务atd.**
2. **Acacron适合于非7\*24小时开机的服务器准备的。开机执行的。**

**检测停机期间没有执行的任务，在开机后一次执行一遍。**

**3）Crond服务默认情况下（每分钟）会检查系统中是否有需要执行的定时任务，如果有就会根据事先定义好的规则来执行这个定时任务。**

**提示：我们说的crond是一个定时任务的守护进程，而crontab这个是一个命令是用户用来设置定时任务的命令。**

**Crond服务是企业生产工作中常用的重要服务，at和anacron很少使用，可以忽略**

**几乎每个服务器都会用到crond服务**

### ****定时任务的使用说明：****

|  |  |  |
| --- | --- | --- |
| **参数** | **含义** | **指定示例** |
| **-l(字母)** | **查看crontab文件内容提示：l可以理解为list的缩写** | **Crontab –l** |
| **-e** | **编辑crontab文件内容，提示：e可以裂解为edit的缩写** | **Crontab –e** |
| **-i** | **删除crontab文件内容，删除前会提示确认，用的很少** | **Crontab -ri** |
| **-r** | **删除crontab文件内容，用的很少** | **Crontab –r** |
| **-u user** | **指定使用的用户执行任务** | **Crontab –u boy -l** |
| **特别强调：-i,-r参数在生产中很少用，没什么需求必须要用-e进去编辑即可。** | | |

**查看特定用户的定时任务：**

**[root@fan ~]# crontab -u ru -l**

**no crontab for ru**

**[root@fan ~]#**

**使用者权限及定时任务文件**

|  |  |
| --- | --- |
| **文件** | **说明** |
| **/etc/cron.deny** | **该文件中所列用户不允许使用crontab命令** |
| **/etc/cron.allow** | **该文件中所列用户允许使用crontab命令，优先于/etc/cron.deny** |
| **/var/spool/cron** | **所有用户crontab配置文件默认都存在此目录，文件以用户名命名** |

### ****定时任务格式****

定时任务的log是在/var/log/cron

**Crontab 用户定时任务规则一般分为6个段每个段通过空格来分隔。系统的定时任务分为8段空来来分隔，前五段为时间的设定段，第六段为索要执行的命令或脚本任务段。**

**01 \* \* \* \* cmd**

**提示：**

1. **cmd为要执行的命令或脚本。**
2. **每个段之间必须要有一个空格。**
3. **-minute(0-59)**

**-hour(0-23)**

**-day of mount (1-31)**

**-mount (1-12)**

**-day of week(0-6)**

|  |  |  |
| --- | --- | --- |
| **段** | **含义** | **取值范围（整数）** |
| **第一段** | **代表分钟** | **00-59** |
| **第二段** | **代表小时** | **00-23** |
| **第三段** | **代表日，天** | **01-31** |
| **第四段** | **代表月份** | **01-12** |
| **第五段** | **代表星期，周几** | **0-7（0和7都代表周日）** |
| **提示：时间记忆口诀：分时日月周。** | | |

**Crontab语法格式中特殊符号含义如下表**

|  |  |
| --- | --- |
| **特殊符号** | **含义** |
| **\*** | **\*号，表示任意时间都可以，实际就是“每”时间的意思。举例：00 23 \* \* \* cmd 表示每月每周每日的23:00都执行cmd任务。需要注意的是：每个时间位上的\*表示每，如果位上是\*就是该位上时间的取值范围，例如小时上的\*等驾驭00-23** |
| **-** | **减号，表示分隔符，表示一个时间范围，区间段，如17-19点，每天17,18,19的00分执行任务。00 17-19 \* \* \* cmd 就是17，18,19点整点分别执行的意思** |
| **，** | **逗号，表示分割时段的意思，30 17,18,19 \* \* \*/bin/sh /scripts/oldboy.sh 表示每天17,18,19点的半点时刻执行/scripts/oldboy.sh脚本。也可以和“-”结合使用例如：30 3-5,17-19 \* \* \* /bin/sh/scripts/oldboy.sh** |
| **/n** | **N代表数字，即每个n单位时间,例如每十分钟执行一次任务可以写成：\*/10 \* \* \* \* cmd其中\*/10,\*的范围是0-59，也可以写成0-59/10** |

**练习：**

[root@fan scripts]# crontab -e

\*/5 \* \* \* \* /sbin/ntpdate time.windows.com >/dev/null 2&1

#print a name

\* \* \* \* \* echo zhaofan >>/server/log/zhaofan.log

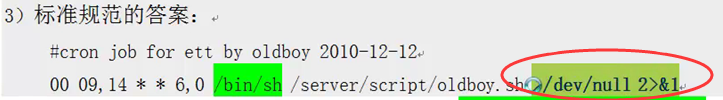
00 9,14 \* \* 0,6 /bin/sh /server/scripts/fan.sh >>/var/spool/cron/root

[root@fan scripts]# crontab -l |tail -2

\* \* \* \* \* echo zhaofan >>/server/log/zhaofan.log

00 9,14 \* \* 0,6 /bin/sh /server/scripts/fan.sh >>/var/spool/cron/root

[root@fan scripts]#



**定时任务要领：**

**1）加必要注释：写定时任务规则时尽可能的加上注释（最好是英文注释）**

**2）执行定时任务时，如果是执行脚本请尽量在脚本前面带上/bin/sh**

**3）定时任务命令结尾加>/dev/null 2>&1**

**4）脚本一般要放在特定的目录下保存**

**定时任务（一般是脚本任务）规则的机尾最好加上>/dev/null 2 >&1等内容，如果需要打印日志，则可以追加到指定的日志文件里（此时不要和/dev/null 同时存在）尽量不要留空。如果任务时命令的话，结尾使用“>dev/null 2 >&1”时要有检查的手段。**

**其中>表示重定向，/dev/null为特殊的字符设备文件。表示黑洞设备或空设备。2>&1表示让标准错误和标准输出一样，本命令内容即把脚本的正常和错误输出都重定向到/dev/null，即不记录任何输出。**

**>/dev/null 2 >&1 等价于 1>/dev/null 2>/dev/null**

**如果定时任务规则结尾不加>dev/null 2>&1 等命令配置，有可能有大量输出信息，时间长了，可能由于系统为开启邮件服务而导致邮件临时目录/var/spool/clientmqueue文件数猛增的隐患，大量文件会占用大量磁盘inode节点（每个文件占一个inode）以至于磁盘inode满而无法写入正常数据**

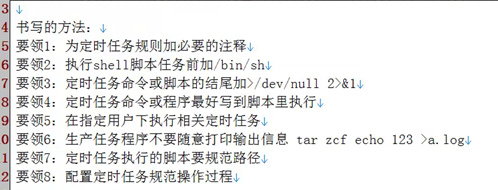
**Echo 123 1>a.log 2 >b.log**

**上面的意思是”Echo 123”当这个命令对的时候写到a.log。当写错的时候到b.log**

**Echo 1 1>a.log 2>b.log===echo 123 >a.log 2>&1**

**技巧：**

1. **进入打包文件或目录的上级目录进行打包、打包的时候不要加v输出**
2. **定时任务时按小时，打包里的日期也要精确到小时**
3. **命令操作成功是定时任务可以成功的大前提**
4. **规范脚本存放的目录**
5. **定时任务打包或处理其他工作，不要在屏幕输出，可以达到日志文件里。**
6. **命令放定时任务里，时间的%必须要转义。\%.**



**例子：**

**#####tar /etc/services by fan######**

**00 2 \* \* \* /bin/sh /server/scripts/dabao.sh >dev/null 2>&1**

**[root@fan scripts]# crontab -l|tail -2**

**#####tar /etc/services by fan######**

**00 2 \* \* \* /bin/sh /server/scripts/dabao.sh >dev/null 2>&1**

**[root@fan scripts]# cat /server/scripts/dabao.sh**

**cd /etc/**

**tar zcf /tmp/services\_2013-12-05-12.tar.gz ./services**

**[root@fan scripts]#**

### ****生产场景如何调试crontab定时任务****

1. **增加执行任务频率调试任务（某些任务不能用于生产环境）**

**强调：有些计划任务时不允许频繁执行的。例如：定时往数据库里插入数据，这样的任务就要在测试机上测试。然后部署到正式线上，这样正式工作出问题的机会就少了**

1. **调整系统时间调试任务。（不能用于生产环境）**
2. **通过脚本日志输出调试定时任务**

**在脚本中加入日志输出，然后把输出打到指定的日志中，然后观察日志内容结果，看是否执行或正确执行。或像下面的内容把脚本结果定向到一个log文件，重定向》即可不需要>>追加，这样日志就不会一直变大**



1. **注意一些任务命令带来的问题**

**\*/1 \* \* \* \* echo “===” >>/tmp/oldboy.log >/dev/null 2 >&1 这是隐蔽的无法正常正确执行的任务配置，原因是前面多了>>或者去掉结尾的>/dev/null 2 >$1**

**5、注意环境变量导致的定时任务故障**

**在调试java 程序任务的时候，注意环境变量，把环境变量的定义加到脚本里**

1. **通过crond定时任务服务日志调试定时任务**

[root@fan ~]# tail -f /var/log/cron

Jan 2 07:44:01 fan CROND[28481]: (root) CMD (echo zhaofan >>/server/log/zhaofan.log)

Jan 2 07:45:01 fan CROND[28486]: (root) CMD (echo zhaofan >>/server/log/zhaofan.log)

Jan 2 07:45:01 fan CROND[28487]: (root) CMD (/sbin/ntpdate time.windows.com >/dev/null 2&1)

Jan 2 07:46:01 fan CROND[28498]: (root) CMD (echo zhaofan >>/server/log/zhaofan.log)

Jan 2 07:47:01 fan CROND[28501]: (root) CMD (echo zhaofan >>/server/log/zhaofan.log)

Jan 2 07:48:01 fan CROND[28504]: (root) CMD (echo zhaofan >>/server/log/zhaofan.log)

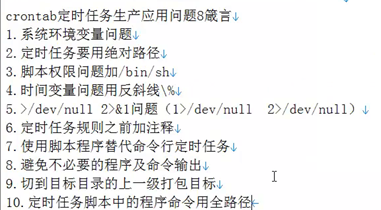
Jan 2 07:49:01 fan CROND[28507]: (root) CMD (echo zhaofan >>/server/log/zhaofan.log)

Jan 2 07:50:01 fan CROND[28514]: (root) CMD (/usr/lib64/sa/sa1 1 1)

Jan 2 07:50:01 fan CROND[28515]: (root) CMD (/sbin/ntpdate time.windows.com >/dev/null 2&1)

Jan 2 07:50:01 fan CROND[28518]: (root) CMD (echo zhaofan >>/server/log/zhaofan.log)

Jan 2 07:51:02 fan CROND[28559]: (root) CMD (echo zhaofan >>/server/log/zhaofan.log)



### ****Crontab生产问题案例及解决过程****

**面试题:在系统维护的时候，创建文件提示“No space left on device ” 请问你这是什么故障？**

**解答：磁盘空间block满了或者inode被占满了**

**解决方案：**

**1）做一个删除定时任务**

**2）定时任务加上>/dev/null 2>&1**

**定时清理clientmqueue目录垃圾文件防止占满磁盘空间：**

**手动清理方法：**

**Find /var/spool/clinetmqueue/ -type f |xargs rm –f**

**定时清理：**

**Echo “find /var/spool/clientmqueue/ -type f |xargs rm -f”>/server/scripts/del\_sys\_file.sh**

**Cat /server/scripts/del\_sys\_file.sh**

**find /var/spool/clientmqueue/ -type f |xargs rm –f**

**echo “00 00 \* \* \* /bin/sh /server/scripts/del\_sys\_file.sh>/dev/null 2>&1”>>/var/spool/cron/root**

**strace这个命令特别重要**

## ****七、Linux网络配置****

### ****修改主机名****

**首先hostname 可以查看主机名还有就是可以修改主机名，但只是临时生效**

**Hostname + 主机名**

**如果想要永久生效需要修改/etc/sysconfig/network**

[root@fanlinux ~]# cat /etc/sysconfig/network

NETWORKING=yes

HOSTNAME=fan

[root@fanlinux ~]#

**同时要修改/etc/hosts文件**

[root@fanlinux ~]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

[root@fanlinux ~]#

**修改主机名的步骤：**

1. **hostname +主机名**
2. **vi /etc/sysconfig/network**
3. **vi /etc/hosts**

**注意：改完后，执行/etc/init.d/network restart 或 source /etc/sysconfig/network**

**也可以通过setup修改主机名**

**注意：如果主机名有问题，服务器会启动非常慢，因为很多服务都需要用到**

### ****网卡配置****

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

DEVICE=eth0

TYPE=Ethernet

ONBOOT=yes （网卡开机自启）

NM\_CONTROLLED=yes

BOOTPROTO=dhcp

USERCTL=no

PEERDNS=yes

IPV6INIT=no

HWADDR=00:0c:29:13:f5:eb

DNS1=8.8.8.8

You have mail in /var/spool/mail/root

[root@fanlinux ~]#

**Linux一个网卡可以配多个地址**

**Ifconfig ehth0:0-154 IP地址 mask 掩码**

**在6.4系统中**

**网卡的配置文件中的地址会和/etc/resolve.conf自动同步，如果修改resolve.con件，重启网卡，会自动修改成和/etc/sysconfig/network-scripts/ifcfg-eth0配置相同**

**查看网关**

[root@fanlinux ~]# route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.101.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

0.0.0.0 192.168.101.1 0.0.0.0 UG 0 0 0 eth0

[root@fanlinux ~]#

**当前面的都无法到达时就会选择最后一条路由**

[root@fanlinux ~]# netstat -rn

Kernel IP routing table

Destination Gateway Genmask Flags MSS Window irtt Iface

192.168.101.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

0.0.0.0 192.168.101.1 0.0.0.0 UG 0 0 0 eth0

You have mail in /var/spool/mail/root

[root@fanlinux ~]#

**删除默认网关的命令  
route del default gw 网关地址**

**添加默认网关的命令**

**Route add default gw 网关地址**

**这些修改是临时生效**

**网关生效的第一文件是：**

**/etc/sysconfig/network-scripts/ifcfg-eht0**

**网关生效的第二文件：**

**/etc/sysconfig/network**

**临时生效是：**

**Router add default gw**

**Dig** [www.baidu.com](http://www.baidu.com)

**可以查看地址的解析情况**

[root@fanlinux ~]# dig www.baidu.com

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6\_4.6 <<>> www.baidu.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47283

;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 5, ADDITIONAL: 5

;; QUESTION SECTION:

;www.baidu.com. IN A

;; ANSWER SECTION:

www.baidu.com. 740 IN CNAME www.a.shifen.com.

www.a.shifen.com. 600 IN A 119.75.217.109

www.a.shifen.com. 600 IN A 119.75.218.70

;; AUTHORITY SECTION:

a.shifen.com. 416 IN NS ns3.a.shifen.com.

a.shifen.com. 416 IN NS ns1.a.shifen.com.

a.shifen.com. 416 IN NS ns2.a.shifen.com.

a.shifen.com. 416 IN NS ns5.a.shifen.com.

a.shifen.com. 416 IN NS ns4.a.shifen.com.

;; ADDITIONAL SECTION:

ns1.a.shifen.com. 49 IN A 61.135.165.224

ns2.a.shifen.com. 460 IN A 180.149.133.241

ns3.a.shifen.com. 55 IN A 61.135.162.215

ns4.a.shifen.com. 203 IN A 115.239.210.176

ns5.a.shifen.com. 421 IN A 119.75.222.17

;; Query time: 35 msec

;; SERVER: 202.106.0.20#53(202.106.0.20)

;; WHEN: Fri Jan 3 06:09:32 2014

;; MSG SIZE rcvd: 260

You have mail in /var/spool/mail/root

[root@fanlinux ~]#

**还有就是nslookup**

[root@fanlinux ~]# nslookup

> www.baidu.com

Server: 202.106.0.20

Address: 202.106.0.20#53

Non-authoritative answer:

www.baidu.com canonical name = www.a.shifen.com.

Name: www.a.shifen.com

Address: 119.75.217.109

Name: www.a.shifen.com

Address: 119.75.218.70

**排除问题：**

traceroute to www.baidu.com (119.75.218.70), 30 hops max, 60 byte packets

1 192.168.101.1 (192.168.101.1) 1.280 ms 1.893 ms 1.801 ms

**Tcpdump –n icmp –i eth0**

## ****八、磁盘基础知识****

[root@fanlinux ~]# free -m

total used free shared buffers cached

Mem: 1869 876 992 0 126 538

-/+ buffers/cache: 211 1657

Swap: 3999 0 3999

[root@fanlinux ~]#

**提示：**

1. **Linux系统的特性是将系统不用的物理内存缓存起来，因此，992不是系统的真实内存**
2. **系统真正的内存是1657**
3. **Buffers为写入缓冲区，sync将缓冲区数据写入磁盘，cache为读出缓冲区**

**磁盘相关名字：**

**Disk 磁盘**

**Head 磁头**

**Sector扇区**

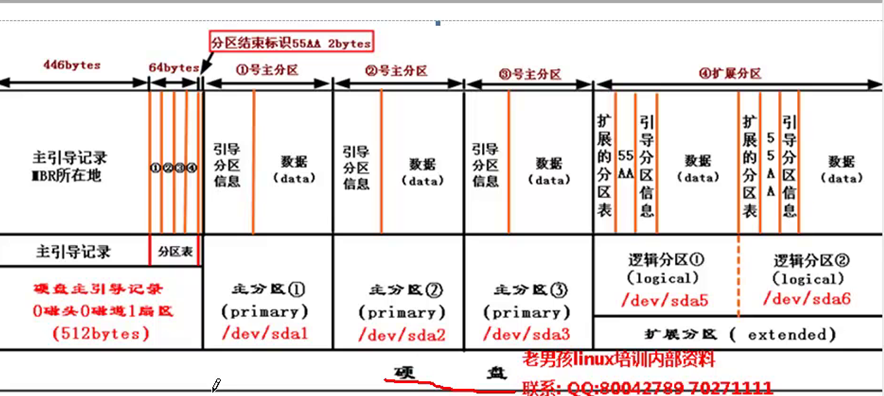
**Track 磁道**

**Cylinder柱面**

**Units单元块（一个柱面的大小）**

**Block 磁盘块**

**Inode 索引节点**



**主引导记录MBR**

**磁盘的0磁头0磁道1扇区，是硬盘上的一个非常重要的位置。在这个扇区中存放有硬盘的主引导记录MBR和硬盘的分区表DPT**

**磁盘分区的重点：**

1. **磁盘分区的实质就是针对上述0磁道1扇区的前446字节后面接下来的64bytes的分区表进行设置，分区工具有fdisk（适用于小于2T的磁盘分区），parted（适用于大于2T的磁盘分区）**
2. **一块磁盘的分区表仅有64bytes大小，仅支持四个分区表信息（主分区+扩展分区）**
3. **磁盘分区的最小单位为柱面（cylikder）**
4. **扩展分区不能直接用，还需要在扩展分区的基础上创建逻辑分区才行**
5. **扩展分区有自己的分区表，因此，扩展分区下面的逻辑分区可以有多个。磁盘在使用前一般需要进行分区，当然如果不分区直接格式化使用也是没问题的，但这不是常见情况。磁盘分区有主分区，扩展分区和逻辑分区之分。一个硬盘最多有四个分区表信息，其中一个主分区的位置可以用一个扩展分区替代。且一个硬盘只能有一个扩展分区，在扩展分区上可以创建多个逻辑分区（IDE磁盘逻辑分区大概编号可以是5-63，SATA（编号5-15））**

### ****Fdisk****

**Fdisk –l 查看分区**

[root@fan ~]# fdisk -l

Disk /dev/sda: 21.5 GB, 21474836480 bytes

255 heads, 63 sectors/track, 2610 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x000817b0

Device Boot Start End Blocks Id System

/dev/sda1 \* 1 26 204800 83 Linux

Partition 1 does not end on cylinder boundary.

/dev/sda2 26 2101 16669696 83 Linux

/dev/sda3 2101 2611 4096000 82 Linux swap / Solaris

Disk /dev/sdb: 1073 MB, 1073741824 bytes

255 heads, 63 sectors/track, 130 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

You have new mail in /var/spool/mail/root

[root@fan ~]#

[root@fan ~]# fdisk /dev/sdb

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel

Building a new DOS disklabel with disk identifier 0x831f5d0c.

Changes will remain in memory only, until you decide to write them.

After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): m

Command action

a toggle a bootable flag

b edit bsd disklabel

c toggle the dos compatibility flag

d delete a partition

l list known partition types

m print this menu

n add a new partition

o create a new empty DOS partition table

p print the partition table

q quit without saving changes

s create a new empty Sun disklabel

t change a partition's system id

u change display/entry units

v verify the partition table

w write table to disk and exit

x extra functionality (experts only)

Command (m for help): n

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): 1

First cylinder (1-130, default 1):

Using default value 1

Last cylinder, +cylinders or +size{K,M,G} (1-130, default 130): +100M

Command (m for help): p

Disk /dev/sdb: 1073 MB, 1073741824 bytes

255 heads, 63 sectors/track, 130 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x831f5d0c

Device Boot Start End Blocks Id System

/dev/sdb1 1 14 112423+ 83 Linux

Command (m for help): n

Command action

e extended

p primary partition (1-4)

e

Partition number (1-4): 5^H^H

Value out of range.

Partition number (1-4): 2

First cylinder (15-130, default 15):

Using default value 15

Last cylinder, +cylinders or +size{K,M,G} (15-130, default 130):

Using default value 130

Command (m for help): p

Disk /dev/sdb: 1073 MB, 1073741824 bytes

255 heads, 63 sectors/track, 130 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x831f5d0c

Device Boot Start End Blocks Id System

/dev/sdb1 1 14 112423+ 83 Linux

/dev/sdb2 15 130 931770 5 Extended

Command (m for help): n

Command action

l logical (5 or over)

p primary partition (1-4)

l

First cylinder (15-130, default 15):

Using default value 15

Last cylinder, +cylinders or +size{K,M,G} (15-130, default 130): +400M

Command (m for help): p

Disk /dev/sdb: 1073 MB, 1073741824 bytes

255 heads, 63 sectors/track, 130 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x831f5d0c

Device Boot Start End Blocks Id System

/dev/sdb1 1 14 112423+ 83 Linux

/dev/sdb2 15 130 931770 5 Extended

/dev/sdb5 15 66 417658+ 83 Linux

Command (m for help): n

Command action

l logical (5 or over)

p primary partition (1-4)

l

First cylinder (67-130, default 67):

Using default value 67

Last cylinder, +cylinders or +size{K,M,G} (67-130, default 130):

Using default value 130

Command (m for help): l

0 Empty 24 NEC DOS 81 Minix / old Lin bf Solaris

1 FAT12 39 Plan 9 82 Linux swap / So c1 DRDOS/sec (FAT-

2 XENIX root 3c PartitionMagic 83 Linux c4 DRDOS/sec (FAT-

3 XENIX usr 40 Venix 80286 84 OS/2 hidden C: c6 DRDOS/sec (FAT-

4 FAT16 <32M 41 PPC PReP Boot 85 Linux extended c7 Syrinx

5 Extended 42 SFS 86 NTFS volume set da Non-FS data

6 FAT16 4d QNX4.x 87 NTFS volume set db CP/M / CTOS / .

7 HPFS/NTFS 4e QNX4.x 2nd part 88 Linux plaintext de Dell Utility

8 AIX 4f QNX4.x 3rd part 8e Linux LVM df BootIt

9 AIX bootable 50 OnTrack DM 93 Amoeba e1 DOS access

a OS/2 Boot Manag 51 OnTrack DM6 Aux 94 Amoeba BBT e3 DOS R/O

b W95 FAT32 52 CP/M 9f BSD/OS e4 SpeedStor

c W95 FAT32 (LBA) 53 OnTrack DM6 Aux a0 IBM Thinkpad hi eb BeOS fs

e W95 FAT16 (LBA) 54 OnTrackDM6 a5 FreeBSD ee GPT

f W95 Ext'd (LBA) 55 EZ-Drive a6 OpenBSD ef EFI (FAT-12/16/

10 OPUS 56 Golden Bow a7 NeXTSTEP f0 Linux/PA-RISC b

11 Hidden FAT12 5c Priam Edisk a8 Darwin UFS f1 SpeedStor

12 Compaq diagnost 61 SpeedStor a9 NetBSD f4 SpeedStor

14 Hidden FAT16 <3 63 GNU HURD or Sys ab Darwin boot f2 DOS secondary

16 Hidden FAT16 64 Novell Netware af HFS / HFS+ fb VMware VMFS

17 Hidden HPFS/NTF 65 Novell Netware b7 BSDI fs fc VMware VMKCORE

18 AST SmartSleep 70 DiskSecure Mult b8 BSDI swap fd Linux raid auto

1b Hidden W95 FAT3 75 PC/IX bb Boot Wizard hid fe LANstep

1c Hidden W95 FAT3 80 Old Minix be Solaris boot ff BBT

1e Hidden W95 FAT1

Command (m for help): p

Disk /dev/sdb: 1073 MB, 1073741824 bytes

255 heads, 63 sectors/track, 130 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x831f5d0c

Device Boot Start End Blocks Id System

/dev/sdb1 1 14 112423+ 83 Linux

/dev/sdb2 15 130 931770 5 Extended

/dev/sdb5 15 66 417658+ 83 Linux

/dev/sdb6 67 130 514048+ 83 Linux

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

You have new mail in /var/spool/mail/root

[root@fan ~]# fdisk -l

Disk /dev/sda: 21.5 GB, 21474836480 bytes

255 heads, 63 sectors/track, 2610 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x000817b0

Device Boot Start End Blocks Id System

/dev/sda1 \* 1 26 204800 83 Linux

Partition 1 does not end on cylinder boundary.

/dev/sda2 26 2101 16669696 83 Linux

/dev/sda3 2101 2611 4096000 82 Linux swap / Solaris

Disk /dev/sdb: 1073 MB, 1073741824 bytes

255 heads, 63 sectors/track, 130 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x831f5d0c

Device Boot Start End Blocks Id System

/dev/sdb1 1 14 112423+ 83 Linux

/dev/sdb2 15 130 931770 5 Extended

/dev/sdb5 15 66 417658+ 83 Linux

/dev/sdb6 67 130 514048+ 83 Linux

[root@fan ~]#

### ****文件系统****

**文件系统是基于存储设备的，如果硬盘或光盘，并且包含文件物理位置的维护；文件系统也可以说是虚拟数据或网络数据存储的方法，比如NFS,MFS，GFS,目前Linux的文件系统有很多种实现，例如：UFS(基于BSD的UNIX文件系统)、ext2,ext3,ext4,zfs，xfs和Reiserfs**

**文件系统选型简单介绍**

1. **sas/SATA硬盘文件系统选择**
2. **reiserfs大量小文件业务首选**
3. **xfs 数据库mysql业务，门户案例**
4. **ext4 视频下载，流媒体，小文件业务**
5. **ext2 蓝汛的cache业务**

**对于Centos5.x常规应用就是默认的ext3文件系统即可**

**对于Centos6.x常规应用就是默认的ext4文件系统即可**

**Centos默认的文件系统都是比较优秀的。**

**Xfs.reiserfs这两个文件都需要单独安装的，默认不支持的，维护不方便**

**大并发并不是过多的做磁盘的优化，而是在前面增加缓存**

**，架构师能力。**

1. **SSD文件系统选择**

**Ext4/reiserfs可以作为ssd文件系统，但未对ssd做优化，不能充分发挥ssd的性能并影响使用时间**

**分区的实质:修改分区表，划分起始柱面号和扇区号**

**一个磁盘最多4个主分区，3p+1e,2p+1e,1p+1e,1p,2p,3p**

**最多一个扩展分区，操作系统限制**

**最多有4个主分区加扩展分区，磁盘限制**

**扩展分区不能直接用，必须要划分逻辑分区**

**Fdisk 分区的原理，就是修改64字节分区表，不能对大于2T的磁盘分区**

**Parted GUN分区工具，可以适合各种分区情况，当然也支持2T以上的**

## ****九、NFS服务****

**NFS网络文件系统的使用很像windows系统的网络共享、安全功能、网络驱动器映射，这也是和linux里的samba服务类似**

### ****NFS应用场景****

**在企业集群架构的工作场景中，NFS网络文件系统一般被用来存储共享视频、图片、附件等静态资源文件（一般把网站用户上传的文件都放到NFS共享里，例如：BBS产品的图片，附件、头像，注意网站程序不放在NFS共享里）。大公司或门户除了使用NFS外，还可能会使用MFS,GFS,FASTFS,TFS等分布式文件系统。**

1. **什么是RPC**

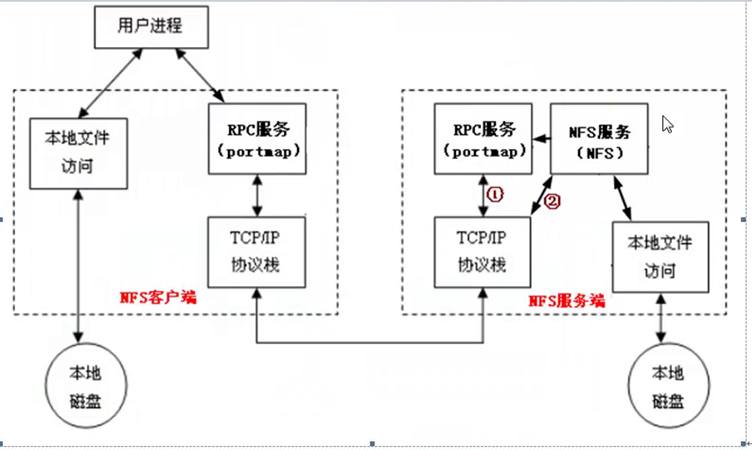
**因为端口不固定，这样一来就会造成NFS客户端与NFS服务端的通讯障碍，因为客户端必须要知道NFS服务器端的数据传输端口才能进行通信交互数据。**

**要解决上面的通讯问题困扰，就需要远程过程调用RPC服务来帮忙，NFS的RPC服务最主要的功能就是记录每个NFS功能所对应的端口号，并在NFS客户端请求时将该端口和功能对应的信息传递给请求数据的NFS客户端，从而可以确保客户端可以连接到正确的NFS端口上去，达到实现数据传输交互数据目的。这个RPC服务很类似NFS服务端和客户端之间的一个中介。**

**启动NFS服务之前要先启动rpc服务**

**Centos5:rpc服务是portmap,Centos6:rpc服务是rpcbind**

**原理图：**



### ****NFS软件及配置****

**要部署NFS服务，需要安装下面的软件包**

**1、Nfs-utils：这个是NFS服务主程序**

**包括rpc.nfsd、rpc.mountd两个daemons和相关文档说明及执行命令文件等**

1. **portmap：Centos5.x下面的RPC的主程序（Centos6.x下名字为rpcbind）**

**NFS可以被视为一个RPC程序，在启动任何一个RPC程序之前，需要做好端口和功能的对应映射工作，这个映射工作就是由portmap（或rpcbind）服务来完成的，因此在提供NFS服务之前必须先启动portmap(rpcbind)服务**

**检查系统版本**

**Cat /etc/redhat-release**

**Uname –r**

**Uname –m**

**安装nfs**

**Yum install nfs-utils portmap rpcbind –y 来安装，或者通过系统光盘rpm包安装命令如：rpm –ivh nfs-utils-1.2.3-36.e16.x86\_64.rpm**

**Yum groupinstall “NFS file server”**

**查看当前Centos6.4下nfs软件安装情况**

[root@nfsclient ~]# rpm -qa nfs-utils portmap rpcbind

rpcbind-0.2.0-11.el6.x86\_64

nfs-utils-1.2.3-64.el6.x86\_64

[root@nfsclient ~]#

**可以看出来RPC服务用的端口是111**

[root@fan ~]# rpcinfo -p localhost

program vers proto port service

100000 4 tcp 111 portmapper

100000 3 tcp 111 portmapper

100000 2 tcp 111 portmapper

100000 4 udp 111 portmapper

100000 3 udp 111 portmapper

100000 2 udp 111 portmapper

[root@fan ~]#

[root@fan ~]# /etc/init.d/rpcbind stop

Stopping rpcbind: [ OK ]

[root@fan ~]# rpcinfo -p localhost

rpcinfo: can't contact portmapper: RPC: Remote system error - Connection refused

[root@fan ~]#

出现上面错的原因是rpc服务没有开启

**开启nfs服务**

[root@fan ~]# /etc/init.d/nfs start

Starting NFS services: [ OK ]

Starting NFS quotas: [ OK ]

Starting NFS mountd: [ OK ]

Starting NFS daemon: [ OK ]

Starting RPC idmapd: [ OK ]

[root@fan ~]#

[root@fan ~]# rpcinfo -p localhost

program vers proto port service

100000 4 tcp 111 portmapper

100000 3 tcp 111 portmapper

100000 2 tcp 111 portmapper

100000 4 udp 111 portmapper

100000 3 udp 111 portmapper

100000 2 udp 111 portmapper

100011 1 udp 875 rquotad

100011 2 udp 875 rquotad

100011 1 tcp 875 rquotad

100011 2 tcp 875 rquotad

100005 1 udp 43455 mountd

100005 1 tcp 33998 mountd

100005 2 udp 50453 mountd

100005 2 tcp 33530 mountd

100005 3 udp 51069 mountd

100005 3 tcp 39559 mountd

100003 2 tcp 2049 nfs

100003 3 tcp 2049 nfs

100003 4 tcp 2049 nfs

100227 2 tcp 2049 nfs\_acl

100227 3 tcp 2049 nfs\_acl

100003 2 udp 2049 nfs

100003 3 udp 2049 nfs

100003 4 udp 2049 nfs

100227 2 udp 2049 nfs\_acl

100227 3 udp 2049 nfs\_acl

100021 1 udp 37021 nlockmgr

100021 3 udp 37021 nlockmgr

100021 4 udp 37021 nlockmgr

100021 1 tcp 53035 nlockmgr

100021 3 tcp 53035 nlockmgr

100021 4 tcp 53035 nlockmgr

You have new mail in /var/spool/mail/root

[root@fan ~]#

**NFS客户端配置**

**NFS只需要开启rpcbind服务**

[root@nfsclient ~]# /etc/init.d/rpcbind start

Starting rpcbind: [ OK ]

[root@nfsclient ~]# chkconfig rpcbind on

[root@nfsclient ~]# chkconfig --list rpcbind

rpcbind 0:off 1:off 2:on 3:on 4:on 5:on 6:off

[root@nfsclient ~]#

**NFS配置文件：**

[root@fan ~]# ll /etc/exports

-rw-r--r--. 1 root root 0 Jan 12 2010 /etc/exports

**配置配置文件：**

[root@fan ~]# cat /etc/exports

#####shared data for bbs by zhaofan ######

/data 192.168.101.0/24(rw,sync)

[root@fan ~]# /etc/init.d/nfs reload

Rw表示权限

Sync表示写入到硬盘而不是缓存里

Reload 表示平滑重启，这样不会影响正在访问的用户

**检查NFS服务是否配置正确：**

[root@fan ~]# showmount -e localhost

Export list for localhost:

/data 192.168.101.0/24

[root@fan ~]#

**在客户端检查NFS**

**[root@nfsclient ~]# showmount -e 192.168.101.62**

**Export list for 192.168.101.62:**

**/data 192.168.101.0/24**

**[root@nfsclient ~]#**

**检查端口是否有问题**

**[root@nfsclient ~]# telnet 192.168.101.62 111**

**Trying 192.168.101.62...**

**Connected to 192.168.101.62.**

**Escape character is '^]'.**

**在客户端进行挂在**

**[root@nfsclient ~]# mount -t nfs 192.168.101.62:/data /mnt**

**[root@nfsclient ~]#**

**挂在后可以通过df –h查看**

**[root@nfsclient ~]# df -h**

**Filesystem Size Used Avail Use% Mounted on**

**/dev/sda3 18G 1.6G 16G 10% /**

**tmpfs 654M 0 654M 0% /dev/shm**

**/dev/sda1 194M 29M 155M 16% /boot**

**192.168.101.62:/data 16G 1.9G 13G 13% /mnt**

**[root@nfsclient ~]#**

**注意：权限分为：共享权限和目录权限两部分，只有两部分都满足了才能具有权限**

**客户端默认在共享的nfs里写文件是文件的创建者是nfsnobody**

**-rw-r--r--. 1 nfsnobody nfsnobody 0 Dec 21 03:22 fan.txt**

**[root@fan data]# cat /var/lib/nfs/etab**

**/data 192.168.101.0/24(rw,sync,wdelay,hide,nocrossmnt,secure,root\_squash,no\_all\_squash,no\_subtree\_check,secure\_locks,acl,anonuid=65534,anongid=65534,sec=sys,rw,root\_squash,no\_all\_squash)**

**[root@fan data]#**

**[root@fan data]# grep 65534 /etc/passwd**

**nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin**

**[root@fan data]#**

### ****NFS配置文件路径****

|  |  |
| --- | --- |
| **NFS常用路径** | **说明** |
| **/etc/exports** | **NFS服务主配置文件，配置NFS具体共享服务的地点，默认内容为空** |
| **/user/sbin/exportfs** | **NFS服务的管理命令。例如：可以加载NFS配置生效，还可以配置NFS共享目录，即无需/etc/exports实现共享**  **Exportfs –rv ====/etc/init.d/nfs reload**  **Exportfs 不但可以加载配置生效，也可以通过命令直接共享目录，越过/etc/exports,但是重启失效** |
| **/usr/sbin/showmount** | **常用来在客户端查看NFS配置及挂载结果的命令** |
| **/var/lib/nfs/etab** | **NFS配置文件的完整参数设定的文件（有很多没有配置但是默认就有的NFS参数）** |
| **/var/lib/nfs/xtab** | **适合centos5记录查看曾经挂载过的NFS客户端的信息，包括IP地址等6没有** |

**配置/etc/exports每个项**

1. **NFS共享目录：为NFS服务端要共享的世纪目录，要永绝对路径，注意共享目录的本地权限，如果需要读写共享，一定要让本地目录被NFS客户端的用户（nsnobody）可以读写**
2. **NFS客户端地址：为NFS服务端授权的可访问共享目录的NFS客户端地址，可以为单独的IP地址或主机名，域名等，也可以为整个网段地址，还可以用“\*”,来匹配所有客户端服务器可以访问，这里所谓的客户端一般来说是前段的业务服务器**
3. **参数**

|  |  |
| --- | --- |
| **参数名称** | **参数用途** |
| **Rw** | **表示读写** |
| **Ro** | **表示只读** |
| **Sync** | **请求和写入数据时，数据同步到NFS server的硬盘后才会返回** |
| **Async** | **请求或写入数据时，先返回请求，再将数据写入到内存缓存和硬盘中，即异步写入数据。此参数可以提升NFS性能，但是会降低数据的安全性**  **了解同步异步：电商的秒杀活动就是异步** |
| **No\_root\_squash** | **访问NFS server 共享目录的用户如果是root的话它对该共享的目录有root权限，这个配置原本为误判客户端准备的，用户应避免使用** |
| **Root\_squash** | **对于访问NFS server 共享目录的用户如果是root的话，则它的权限将被压缩为匿名用户，同时它的UID和GID通常会成为nobody或nfsnobody账号身份** |
| **All\_squash** | **不管访问NFS server 共享目录的用户身份如何，他的权限都将被压缩为匿名用户，同时它的UID和GID都会变成nobody或nfsnobody账号身份，在多个NFS客户端同时读写NFS server 数据时，这个参数特别有用**  **配置NFS 生产环境技巧**   1. **确保所有服务器对NFS 共享目录具备相同的权限** 2. **all\_squash把所有客户端都压缩成匿名用户** 3. **就是anonuid，anongid指定的UID和GID的用户**   **2、所有的客户端和服务端都需要有一个相同的UID和GID的用户，即nfsnobody（UID必须相同）** |
| **Anonuid** | **参数一anon\*开头即指anonymous匿名用户，这个用户的UID设置通常为nfsnobody或nobody的UID值，当然我们也可自己设置这个UID但是UID必须存在于/etc/passwd中** |
| **Anongid** | **同anonuid，区别及时把uid换成gid** |

**实验：**

**实现当多台客户端同时挂一台NFS server 时，无论从哪个客户端写入数据，其他客户端同样可以读写，即让所有的NFS 客户端写入到NFS server 的文件或目录在NFS server 上的用户和组都是同一个名称oldboy.**

**解答：3台机器，2台客户端，一台服务端**

1. **在建立一个NFS 共享用户，所有机器都存在，并且uid,gid相同**

**Useradd –u 555 oldboy**

**服务端：**

**[root@fan ~]# cat /etc/exports**

**############shiyan#########**

**/data 192.168.101.0/24(rw,sync,anonuid=555,anongid=555,all\_squash)**

**[root@fan ~]#**

1. **客户端进行挂载**

### ****关于NFS挂载****

**想要将要NFS的挂载的开机的时候就挂载上：**

1. **把挂载的命令放在/etc/rc.local文件中**

**缺点：偶尔开机挂载不上，工作中除了开机自启动配置**

1. **把挂载放在fstab中**

**缺点：NFS网络文件系统尽量不要放在fstab,fstab优先于网络启动，此时还连接不上NFS服务器**

**即使是本地文件系统，也要注意fstab最后两列，要设置0 0 否则会导致无法启动服务器的问题。**

**一般不再fstab中配置网络系统的挂载，本地的可以**

**把NFS rpc服务的启动命令的和挂在的命令均放在/etc/rc.local 然后在通过nagios监控软件监控开机后的挂载情况**

**查看服务端挂载时的默认参数**

**Cat /var/lib/nfs/etab**

**查看客户端挂载时的默认挂载参数：**

**Cat /proc/mounts**

**Mount 挂载及fstab文件的参数表格**

|  |  |  |
| --- | --- | --- |
| **参数** | **参数功能** | **默认参数** |
| **Fg**  **bg** | **当在客户端执行挂载时，可选择是前台（fg）还是在后台（bg）执行。若前台执行，则mount会持续尝试挂载，直到成功或挂载超时为止，若在后台执行，则mount会在后台持续多次进行mount，而不会影响到前台的其他程序操作，如果网络连接不稳定，或是服务器常常需要开关机，建议使用bg比较稳妥** | **Fg** |
| **Soft**  **Hard** | **上面不是谈到挂载时会使用RPC呼叫么？如果是hard参数，则当服务器主机离线，则RPC会持续呼叫，知道对方回复联机为止，如果是soft的话，那RPC会在超时后重复呼叫，而非持续呼叫，因此系统的演示会不再这么显示，同样，如果服务器可能开开关关，建议用soft** | **Hard** |
| **Intr** | **当使用hard方式挂载时，若加上intr这个参数，则当RPC持续呼叫时，呼叫可以被中断** | **没有** |
| **Rsize**  **Wsize** | **读出与写入的区块大小，这个设置可以影响客户端与服务器传输数据的缓冲存储量。一般来说，如果在局域网内，并且客户端与服务器端都具有足够的内存，这个值可以设置大一点，但是这个值也不要设置太大，最好是实现网络能够传输最大值为限** | **这个参数可以用于优化** |

### ****NFS客户端mount挂载优化****

**Wsize**

**Rsize**

**这个两个参数是用于优化的**

**Time dd if=/dev/zerom of=/mnt/testfile bs=9k count =2000 这条命令可以用于测试速度 计算一个任务用的时间**

**NFS网络系统优化挂在的参数建议：**

**Centos5.8 X86\_64 服务器和客户端环境**

**Mount –t nfs –o noatime,nodiratime,nosuid,noexec,nodev,rw,bg,soft,rsize=32768,wsize=32768 10.0.0.18:/data/ /mnt**

**Nosuide 和nodev是安全参数，rw默认就有**

**Centos6.5 X86\_64 服务器和客户端环境**

**Mount –t nfs –o noatime,nodiratime,nosuid,noexec,rw,bg,hard,intr 10.0.0.18:/data /mnt**

**6.5中rsize和wsize默认大小是131072**

**生产场景NFS共享存储优化：**

1. **硬件：sas/ssd磁盘，买多块，raid0/raid10 网卡好**
2. **服务器端：/data 10.0.0.0/24(rw,async,all\_squash,anonuid=555,anongid=555)**
3. **客户端挂载：rsize,wsize,noatime,nodirtime,nosuid,noexec,soft(hard,intr)**

**Mount –t nfs –o nosuid,noexec，noatime,nodiratime,rsize=131072,wsize=131072 10.0.0.7:/data /mnt**

**如果为了精简下面方式是最佳选择：**

**Mount –t nfs –o noatime,nodiratime,rsize=131072,wsize=131072 10.0.0.7:/data /mnt**

1. **内核优化**

**Net.core.wmem\_default=8388608**

**Net.core.rmeem\_default=8388608**

**Net.core.rmem\_max=16777216**

**Net.core.wmen\_max=16777216**

### ****Showmount命令说明****

**Showmount命令一般用于从NFS客户端检查NFS服务器端共享目录的情况**

**常用参数**

|  |  |  |
| --- | --- | --- |
| **短格式** | **长格式** | **用途及实例结果** |
| **-e** | **--exports** | **显示NFS服务器输出的目录列表** |
| **-d** | **--directories** | **显示NFS服务器中提供共享的目录** |
| **-a** | **--all** | **以ip:/dir格式显示NFS服务器的IP地址和可被挂载的目录** |

**Exportfs命令介绍**

**Exportfs –rv===/etc/init.d/nfs reload 加载配置生效**

**通过exportfs命令，我们可以管理当前NFS共享的文件系统列表**

**当我们在启动NFS服务之后，此时如果修改了/etc/exports就要重新启动nfs,使修改的配置生效。这个时候我们就可以用exportfs命令来完成。**

**如果rpcbind服务是停止的或有问题的**

**[root@fan ~]# /etc/init.d/rpcbind stop**

**Stopping rpcbind: [ OK ]**

**[root@fan ~]# rpcinfo -p localhost**

**rpcinfo: can't contact portmapper: RPC: Remote system error - Connection refused**

**[root@fan ~]#**

**如果nfs服务是停止的话**

**应显示如下**

**[root@fan ~]# /etc/init.d/nfs stop**

**Shutting down NFS daemon: [ OK ]**

**Shutting down NFS mountd: [ OK ]**

**Shutting down NFS quotas: [ OK ]**

**Shutting down NFS services: [ OK ]**

**Shutting down RPC idmapd: [ OK ]**

**[root@fan ~]# rpcinfo -p localhost**

**program vers proto port service**

**100000 4 tcp 111 portmapper**

**100000 3 tcp 111 portmapper**

**100000 2 tcp 111 portmapper**

**100000 4 udp 111 portmapper**

**100000 3 udp 111 portmapper**

**100000 2 udp 111 portmapper**

**[root@fan ~]#**

### ****NFS故障及解决办法总结****

**1、**

### ****小结****

**和NFS相关命令及相关目录：**

1. **Mount(-t –o -a)、umount(-l -f),showmount(-e),exportfs(-rv)，rpcinfo(-p),ping(-c -t) telnet**
2. **/etc/fstab，/etc/exports，/var/lib/nfs/etab,/proc/mounts,/etc/mtab**

### ****NFS服务生产场景应用说明****

**NFS服务可以让不同客户端挂载使用同一个目录，作为共享存储使用，这样可以保证不同节点客户端数据的一致性，在集群架构环境中经常会用到。**

**NFS的有点：**

1. **简单，容易上手，容易掌握，数据是在文件系统之上的**
2. **方便，部署快速，维护简单**
3. **可靠，从软件层面上看，数据可靠性高，经久耐用**

**NFS的局限性：**

1. **局限性是存在单点故障，如果NFS server 宕机了，所有客户端都不能访问共享目录。不过这个在后期会通过负载均和及高可用方案弥补**
2. **在高并发的场合，NFS效率性能有限（一般几千万一下PV的网站不是瓶颈，除非网站架构太差，2千万PV/日）**
3. **客户端认证时基于ip和主机名的，权限是根据ID来实现的，安全性一般（踊跃内网则问题不大）**
4. **NFS数据是明文的，对数据完整性不作验证**
5. **多台机器挂载NFS服务器时，连接管理维护麻烦，尤其NFS服务器端出问题后，所有的NFS客户端都挂掉状态（测试环境可以使用autofs自动挂在解决）**
6. **耦合度太高==》架构师的重要职责===解耦。网站扩展性才更好**

**总结fstab文件**

### ****客户端自动挂在（autofs）部署方法****

**为什么会需要autofs**

**Autofa可以实现当用户访问的时候再挂载，如果没有用户访问指定时间内，就自动卸载**

**可以解决NFS服务器和客户端紧密耦合的问题，缺点：是用户请求才挂载，所以开始请求的瞬间效率较差。**

**Autf用户测试环境，或者并发很低的生产环境（加目录漫游lDAP）中，一般的大并发场景不用它**

## ****十、SSH服务****

### ****SSH介绍****

**SSH是专为远程登录会话和其他网络服务提供的安全协议。利用SSH协议可以有效的防止远程管理过程中的信息泄露问题，在当前的生产环境中，绝大多数企业普遍采用SSH协议服务来替代传统的不安全的远程联机服务软件，如：telnet(23端口)**

**在默认状态下，SSH服务提供两个服务功能：一个是提供类似telnet远程联机服务器的服务，即上面提到的SSH服务；另一个类似FTP服务的sft-server,借助SSH协议来传输数据的，提供更安全的SFTP服务。**

**特别提醒：SSH（SSH命令）还包括一个很有用的远程安全拷贝命令scp,也是通过SSH协议工作的**

**SSH结构**

**SSH服务由服务端软件和客户端组成，SSH服务默认使用22端口提供服务**

**查看服务端上的ssh**

**[root@fan ~]# rpm -aq openssh**

**openssh-5.3p1-94.el6.x86\_64**

**[root@fan ~]# rpm -qa openssh openssl openssl是用来加密的**

**openssh-5.3p1-94.el6.x86\_64**

**openssl-1.0.1e-15.el6.x86\_64**

**[root@fan ~]# rpm -aq | grep openss\***

**openssl098e-0.9.8e-17.el6.centos.2.x86\_64**

**compat-opensm-libs-3.3.5-3.el6.x86\_64**

**openssh-clients-5.3p1-94.el6.x86\_64**

**openssh-5.3p1-94.el6.x86\_64**

**openssh-server-5.3p1-94.el6.x86\_64**

**openssl-1.0.1e-15.el6.x86\_64**

**[root@fan ~]#**

**OpenSSH同时支持SSH1.X和2.X。用SSH2.X的客户端程序不能连接的SSH1.X的服务程序上。**

**SSh服务端是一个守护进程（demon），它在后台运行并响应来自客户端的连接请求。**

**SSH服务端的进程名为sshd,负责实时监听远程ssh客户端的连接请求，并处理。SSh服务就是我们基础系统优化中保留开机自动启动的服务之一**

**SSH客户端包括ssh程序，以及向scp（远程拷贝）、slogin（远程登录）、sftp（安全FTP文件传输）等相应程序**

**SSH服务的认证类型：基于口令的安全验证，基于秘钥的安全认证。公钥一般放在服务器端，私钥放在自己电脑上**

**基于密钥的安全认证方式是指，需要一开密钥，也就是必须先建立一对密钥对，然后把公用密钥（public key）放在需要访问的目标服务器上，另外，还需要把私有密钥（Private key）放到SSH的客户端或对应的服务器上。**

**两个linux系统之间用ssh连接**

**Ssh –p端口 用户名@IP地址**

**面试题：已知端口，如果用命令查找这个端口对应的服务**

**如果给一个端口如何查看这个端口对应的服务：**

**[root@fan ~]# lsof -i:22**

**COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME**

**sshd 1365 root 3u IPv4 12534 0t0 TCP \*:ssh (LISTEN)**

**sshd 1365 root 4u IPv6 12536 0t0 TCP \*:ssh (LISTEN)**

**sshd 41166 root 3r IPv4 127866 0t0 TCP 192.168.101.87:ssh->192.168.101.56:12858 (ESTABLISHED)**

**sshd 42327 root 3r IPv4 134073 0t0 TCP 192.168.1.104:ssh->192.168.1.102:9658 (ESTABLISHED)**

**[root@fan ~]#**

**或者**

**[root@fan ~]# netstat -lntup|grep 22**

**tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN 1365/sshd**

**tcp 0 0 :::22 :::\* LISTEN 1365/sshd**

**udp 0 0 0.0.0.0:68 0.0.0.0:\* 42265/dhclient**

**[root@fan ~]#**

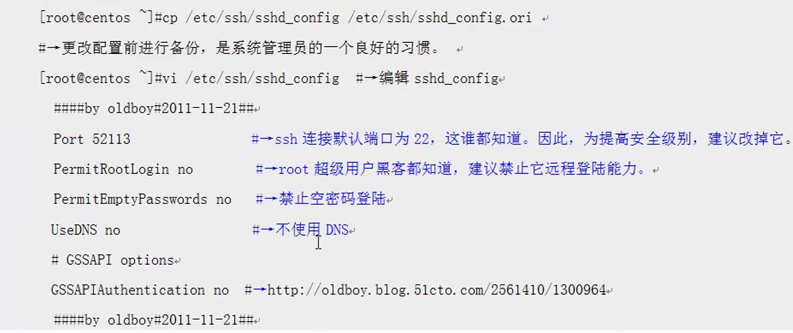
### ****更改SSH默认登陆配置****

**修改SSH服务的运行参数，是通过修改配置文件/etc/ssh/sshd\_config实现的**

**一般来说，ssh服务使用默认的配置已经能够很好的工作的。如果对安全要求不高，仅提供ssh服务的情况，可以不需要修改任何配置。**

OpenSSH的配置通常放在:/etc/ssh/sshd\_config

PermitRootLogin no # 禁止root用户登陆  
StrictModes yes # 检查密钥的用户和权限是否正确，默认打开的   
RSAAuthentication yes # 启用 RSA 认证  
AuthorizedKeysFile .ssh/authorized\_keys # 验证公钥的存放路径  
PubkeyAuthentication yes # 启用公钥认证  
PasswordAuthentication no # 禁止密码认证，默认是打开的。



**UseDNS no**

**GSSAPIAuthentication no**

**上面的这两个参数有助于解决连接ssh是缓慢的问题**

**小结：**

1. **切换到别的服务器ssh –p端口号 user@ip**
2. **到其他机器执行命令（不会切到机器上） ssh –p 22 user@ip 命令（命令要全路径）**

**[root@fan yum.repos.d]# ssh -p 52113 root@192.168.101.115 /sbin/ifconfig**

**root@192.168.101.115's password:**

**Connection closed by 192.168.101.115**

**You have new mail in /var/spool/mail/root**

**[root@fan yum.repos.d]# ssh -p 52113 root@192.168.101.115 /sbin/ifconfig**

**root@192.168.101.115's password:**

**eth0 Link encap:Ethernet HWaddr 00:0C:29:20:2C:A5**

**inet addr:192.168.101.115 Bcast:192.168.101.255 Mask:255.255.255.0**

**inet6 addr: fe80::20c:29ff:fe20:2ca5/64 Scope:Link**

**UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1**

**RX packets:89894 errors:0 dropped:0 overruns:0 frame:0**

**TX packets:13398 errors:0 dropped:0 overruns:0 carrier:0**

**collisions:0 txqueuelen:1000**

**RX bytes:20408034 (19.4 MiB) TX bytes:1264768 (1.2 MiB)**

**lo Link encap:Local Loopback**

**inet addr:127.0.0.1 Mask:255.0.0.0**

**inet6 addr: ::1/128 Scope:Host**

**UP LOOPBACK RUNNING MTU:16436 Metric:1**

**RX packets:9 errors:0 dropped:0 overruns:0 frame:0**

**TX packets:9 errors:0 dropped:0 overruns:0 carrier:0**

**collisions:0 txqueuelen:0**

**RX bytes:872 (872.0 b) TX bytes:872 (872.0 b)**

**[root@fan yum.repos.d]#**

1. **当第一次连接的时候，本地会产生一个秘钥文件~/.ssh/known\_hosts（服务器多时会有多个秘钥）**

**下面是~/.ssh/known\_hosts文件中的内容**

**[root@fan ~]# cat ~/.ssh/known\_hosts**

**192.168.101.115 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAq/wh0cp/z1B4HNP3O+HaK/A073AcSLgC+QG+s50iGz88oi3382F+7Xy5g6A+0IIKBv0BhRtKEkl0/fuRNZaqp3A2mdAl60IJv+nDJhSP1YKbb1HYwneN1azGDoLWUbl2YvlSyqubqeE2ji3qHmFQv4d5QiC80wTOsaWst82/VJQ07KbqWXDVZs5/um5+WbMsNsCpchIitH9VB4mjvraY6hZ5COUc6l/IbSp+92epGj2jcB1FZM21Hzv4kxv3tinAxql0l+20YGO932q8bLjqRvkOdoBYbO84u7D+xtoOdXIvVzdQazTS1qAq/of+8Qnk4fId87vniWICtNrUnw+MgQ==**

**[root@fan ~]#**

### ****SSH客户端附带的远程拷贝命令scp****

**Scp的基本语法使用：scp 全称是：scp –secure copy**

**有两种方式:**

**一种是推 ：push**

**是将数据从自己的服务器推到另外一个服务器上**

**[root@fan data]# scp -P 52113 /data/fan.txt root@192.168.101.115:/data**

**root@192.168.101.115's password:**

**fan.txt 100% 0 0.0KB/s 00:00**

**一种是拉：pull**

**是将数据从别的服务器放到自己的服务器上**

**[root@nfsclient data]# scp -P 52113 root@192.168.101.87:/data/aaa /data**

**root@192.168.101.87's password:**

**aaa 100% 0 0.0KB/s 00:00**

**[root@nfsclient data]# ls**

**aaa fan.txt**

**[root@nfsclient data]#**

**Scp 为远程拷问文件或目录的命令**

**-P 接端口，默认是22端口时可以省略**

**-r 表示拷贝目录**

**-p表示在拷贝前后保持文件或目录属性**

**@前为用户名，@后为连接服务器的IP地址**

**小结：**

**Scp是加密的远程拷贝，可以把数据从一台机器推送到另外一台机器，可以其他服务器数据拷贝到本地执行命令的服务器，但是每次都是全量拷贝（rsync增量拷贝），因此效率不高。**

### ****SSh服务附带的SFTP功能服务****

**[root@fan ~]# sftp -oPort=52113 root@192.168.101.115**

**Connecting to 192.168.101.115...**

**root@192.168.101.115's password:**

**sftp>**

**默认登陆之后是在登陆用户的家目录**

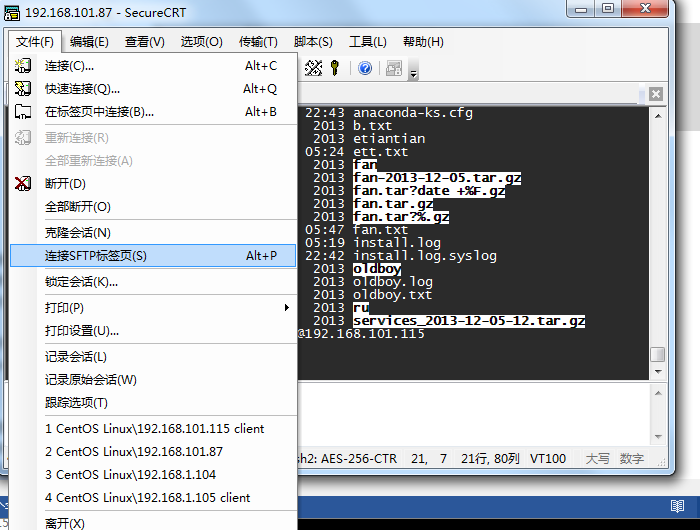
**小结：**

**1、sftp -oPort=52113 root@192.168.101.115**

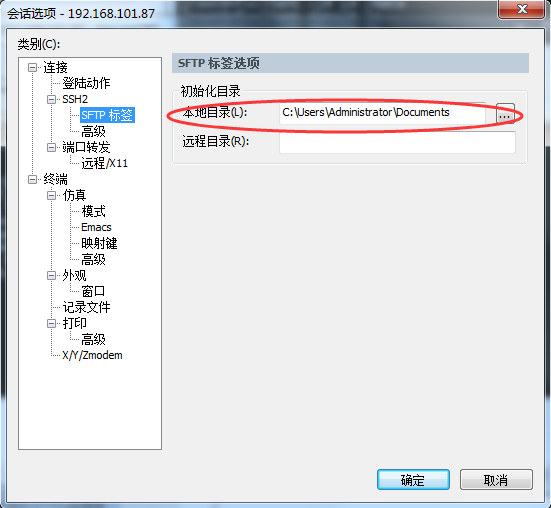
**2、上传put加客户端本地路径，也可以指定路径上传**

**3、下载get服务端的内容，下载到本地的当前目录**

**Windows客户端直接连接方法：**



**下载的地址**



## ****十一、Rsync数据同步工具****

**Rsync简介：**

**Rsync是一款开源的、快速的、多功能的，可实现全量及增量的本地或远程数据同步备份的优秀工具。Rsync软件适用于linux/unix、windows等多种操作系统**

**Rsync具有可使本地和远程两台主机之间的数据快速复制同步镜像、远程备份的功能，这个功能类似于ssh带的scp命令，但是优于scp命令的功能，scp每次都是全量拷贝，而rsync可以增量拷贝。当然，rsync还可以在本地主机的不同分区或目录之间全量及增量的复制数据，这又类似于cp命令，但同样优于cp命令，cp每次都是全量拷贝，而rsync可以增量拷贝文件。**

**小提示：利用rsync还可以实现删除文件和目录的功能，这又相当于rm命令**

### ****Rsync服务在生产环境中的应用****

1. **两台服务器之间数据的同步**
2. **把所有客户服务器数据同步到备份服务器，生产场景集群架构服务器备份方案**
3. **Rsync集合inotify的功能做实时的数据同步**

### ****Rsync的工作方式****

**Rsync大致使用三种主要的传输数据的方式，分别是：**

1. **单主机本地之间的数据传输（此时类似于cp命令的功能）**
2. **借助rcp,ssh等通道来传输数据（此时类似于scp命令的功能）**
3. **以守护进程（socket）的方式传输数据（这个是rsync自身重要的功能）**

### ****本地数据传输模式****

**Rsync [option…]src [dest]**

**语法说明：**

1. **rsync为同步的命令**
2. **[option]为同步时参数选项**
3. **SRC为源，即待拷贝的分区，文件或目录等**
4. **[DEST]为目的的分区，文件或目录等**

**例子：**

**[root@A ~]# rsync /etc/hosts /tmp/**

**[root@A ~]# cd /tmp**

**[root@A tmp]# ls**

**hosts**

**[root@A tmp]# cat hosts**

**127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4**

**::1 localhost localhost.localdomain localhost6 localhost6.localdomain6**

**192.168.1.100 A**

**192.168.1.101 B**

**192.168.1.102 C**

**[root@A tmp]#**

**-avz参数是最常用的**

**[root@A tmp]# rsync -avz /etc/hosts /tmp/**

**sending incremental file list**

**hosts**

**sent 150 bytes received 31 bytes 362.00 bytes/sec**

**total size is 206 speedup is 1.14**

**[root@A tmp]# rsync -avz /etc/hosts /tmp/**

**sending incremental file list**

**sent 30 bytes received 12 bytes 84.00 bytes/sec**

**total size is 206 speedup is 4.90**

**[root@A tmp]# rsync -avz /etc/hosts /tmp/**

**sending incremental file list**

**sent 30 bytes received 12 bytes 84.00 bytes/sec**

**total size is 206 speedup is 4.90**

**[root@A tmp]#**

**用rsync删除**

**注意标注的地方必须有/**

**[root@A /]# mkdir null**

**[root@A /]# rsync -r --delete /null/ /data/**

**[root@A /]# cd /data**

**[root@A data]# ls**

**[root@A data]#**

**[zhaofan@A ~]$ rsync -avz -e 'ssh -p 52113' /etc/hosts zhaofan@192.168.1.101:~**

**sending incremental file list**

**hosts**

**sent 67 bytes received 37 bytes 208.00 bytes/sec**

**total size is 206 speedup is 1.98**

**[zhaofan@A ~]$**

### ****客户端常用命令的参数：****

**-v –verbose 详细模式输，传出时的进度等信息**

**-z –compress 传输时进行压缩以提高传输效率**

**-a –archive归档模式，表示以递归方式传输文件，并保持所有文件属性，等于rtopgDL**

**-r 对子目录以递归模式，及目录下的所有目录都同样传输，注意是小写r**

**-t --times保持文件属性信息**

**-o –owner 保持文件属主信息**

**-p -perms保持文件权限**

**-g --group保持文件属组信息**

**-P --progress 显示同步的过程及传输时的进度信息**

**-D –device 保持设备文件信息**

**-l --links 保留软件链接**

**-e –rsh=COMAND 使用信道协议，指定替代rsh的shell程序**

### ****以守护进程（socket）的方式传输数据（重点）****

****

**小结：**

**服务端**

**1、vi /etc/rsyncd.conf 该文件不存在需要自己创建（用户rsync，目录，非系统虚拟用户及密码文件）**

**2、创建共享目录 /zhaofan**

**3、创建rsync用户，并且授权访问 /zhaofan**

**4、创建密码文件，复制配置文件里的路径，然后添加密码内容**

**虚拟用户名：密码**

1. **密码文件的权限为600**

**以上是服务端的配置**

1. **rsync --daemon然后放入/etc/rc.local启动服务**
2. **tail /var/log/rsyncd.log**

**客户端**

1. **密码文件和服务端没有任何关系，命令执行**

**---password-file=/etc/rsync.password内容：密码和服务端密码一样**

**2、/etc/rsync.password 600**

**3、同步**

**推：拉：**

**Access via remote shell:**

**Pull: rsync [OPTION...] [USER@]HOST:SRC... [DEST]**

**Push: rsync [OPTION...] SRC... [USER@]HOST:DEST**

**Access via rsync daemon:**

**Pull: rsync [OPTION...] [USER@]HOST::SRC... [DEST]**

**rsync [OPTION...] rsync://[USER@]HOST[:PORT]/SRC... [DEST]**

**Push: rsync [OPTION...] SRC... [USER@]HOST::DEST**

**rsync [OPTION...] SRC... rsync://[USER@]HOST[:PORT]/DEST**

**排错：**

1. **防火墙和selinux（setenforce getenforce ）/etc/init.d/iptable stop**
2. **/var/log/rsyncd.log**
3. **整个部署流程整理考虑排查**
4. **操作习惯当做一件大事**

**Rsync的排除功能：**

**下面是客户端排除：**

rsync -avz /data/ --exclude ==(填写要排除的文件) rsync\_backup@192.168.1.100::zhaofan

**上面是排除一个文件，如果要排除多个文件**

rsync -avz /data/ --exclude =={文件1，文件2 }rsync\_backup@192.168.1.100::zhaofan

**如果排除连续的文件**

rsync -avz /data/ --exclude =={ 1..2 }rsync\_backup@192.168.1.100::zhaofan

rsync -avz /data/ --exclude-from=paichu.log rsync\_backup@192.168.1.100::zhaofan

**服务端排除（一般不用）**

**exclude =a b 加入到/etc/rsyncd.conf**

**需要记住的命令：**

**[root@fan zhaofan]# ps -ef|grep rsync**

**root 2700 1 0 11:24 ? 00:00:00 rsync --daemon**

**root 28362 2672 0 12:48 pts/0 00:00:00 grep rsync**

**[root@fan zhaofan]#**

**[root@fan zhaofan]# lsof -i:873**

**COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME**

**rsync 2700 root 4u IPv4 15461 0t0 TCP \*:rsync (LISTEN)**

**rsync 2700 root 5u IPv6 15462 0t0 TCP \*:rsync (LISTEN)**

**[root@fan zhaofan]#**

**Pkill rsync 就可以停止rsync服务**

**Rsync推送的场景：**

1. **备份—delete风险**

**本地有啥，远端有啥，本地没有的远端有也要删除，服务端的目录数可能丢失**

**Rsync拉取企业工作场景：**

**1、代码发布，下载—delete风险**

**远端有啥，本地有啥，远端没有的本地有也要删除，本地的目录数可能丢失**

### ****Rsync无差异同步的生产场景应用****

**一般是有需要两台服务器之间，必须要求数据一致，且时时性又不是很高的情况，如两台负载均衡下面的web服务器之间的同步，或者高可用双机配置之间的同步，rsync无差异同步非常危险，而且，有很多的替代方案，因此，生产场景没有特殊的需求，硬应避免使用。**