**1-3-RHEL7基本命令操作**

# 1.1 rhel界面基本操作

* **1.1Linux终端介绍 Shell提示符 Bash Shell基本语法。**
* **1.2基本命令的使用：ls、pwd、cd。**
* **1.3 查看系统和BIOS硬件时间。**
* **1.4 Linux如何获得帮助，Linux关机命令：shutdown、init等。**
* **1.5 Linux 7个启动级别，设置服务器在来电后自动开机。**

**单词整理：**

**Terminal ： 终端 network-scripts 网络脚本 passwd 密码文件**

**nologin 禁止登录 shutdown 关机 reboot 重启 poweroff 关机**

**Grep 过滤 localhost 本机 useradd 添加用户**

**graphical 图形**

**tty控制台终端 （）**

**pts虚拟终端 （）**

**tty1-tty6**

**如是一开始进入的是图形图面：tty1 就是图形界面。**

**图形界面切换到字符界面终端： Ctrl+Alt+F2-6**

**切换到图像界面 Alt+F1-F6**

**pts虚拟终端**

**pty（虚拟终端):**

**如果我们远程telnet到主机时,不也需要一个终端交互么？**

**是的，这就是虚拟终端pty(pseudo-tty)**

**3. pts/ptmx(pts/ptmx结合使用，进而实现pty):**

**ctrl+shift+T 新建伪终端**

**Alt + 数字键 终端切换**

**Alt + F4 关闭终端**

**使用who am i查询当前终端对应的pts号**

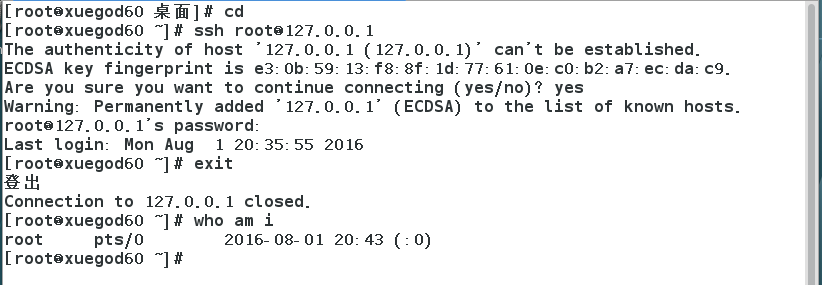
**新建一个终端 查看pts号是否一样呢？**

**放大：ctrl+shift+ +**

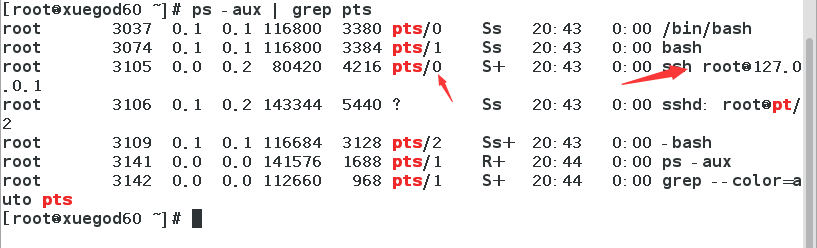
**缩小：ctrl+ -**

**查看进程： 验证ssh的终端**

**实验：使用ssh远程，查看pts**



查看进程，然后使用grep过滤



**pts(pseudo-terminal slave) 伪终端 ：通过ssh相关工具连接上后，显示出来终端为：pts**

## 1.1.2 Shell提示符：

**[root@xuegod60 ~]#**

**[san@xuegod60 ~]$**

**su - san // 切换用户**

**Pwd 在roo t目录和基本用户的路径是不同的，它们的宿主目录不同**

**用户名—主机名—当前的目录—root/普通用户**

**Hosename // 查看主机名**

## 1.1.3 Bash Shell基本语法

**如何输入命令：**

**查看所有的解释器：**

**cat /etc/shells**

**[root@xuegod60 ~]# chsh -l**

**命令的格式：**

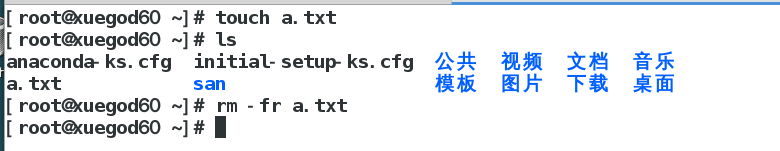
**命令字 [选项][参数]**

**命令字： 具体执行的命令**

**选项：匹配的条件**

**参数：命令处理的对象。**

**听 仔细 Linux**



# 1.2 基本命令操作：

## 1.2.1 ls 命令

**作用：查看当前目录下有哪些文件**

**语法：ls 目录 ，如果不加目录查看当前目录**

**参数：**

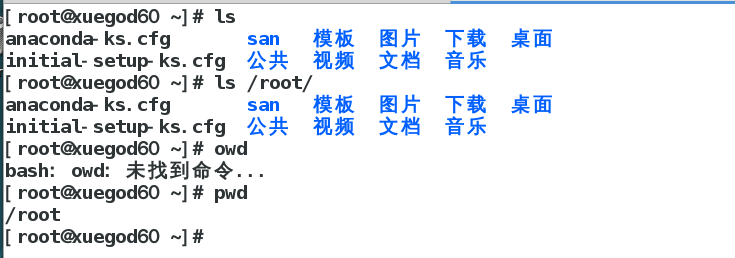
**-l 显示详细信息　　　系统里面的快捷方式#ll=ls -l**

**-a显示隐藏文件**

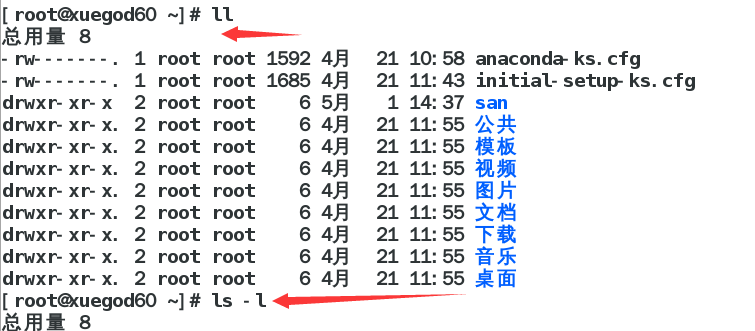
**-d 查看目录(不查看里面的内容)**

**ls -li （-l意思是长文件显示，i是显示inode节点）**

**ls -lrt 显示最近创建的文件**







**为什么有的目录文件有不同的颜色呢？**

**linux系统中不同的颜色代表了不同的文件类型**

**蓝色 目录 /etc**

**黑色 文件 /etc/passwd**

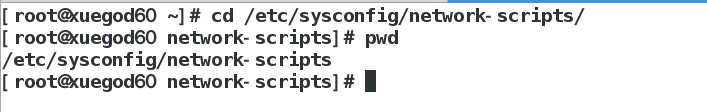
**浅蓝色 链接 /etc/grub2.cfg**

**红色 压缩包 boot.tar.gz**

**绿色 可执行文件 /etc/init.d/network**

**黑底黄字 设备文件 /dev/sda**

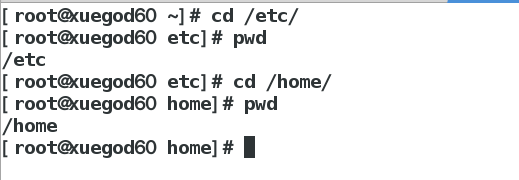
## 1.2.2 pwd命令



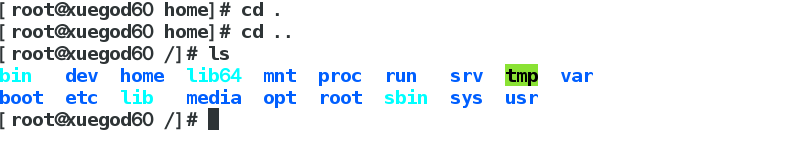
## 1.2.3 cd命令

**作用：切换目录**

**语法：cd 目录名**



**cd.. 返回到上级目录位置**



# 1.3 系统时间管理:

## 1.3.1 查看系统和BIOS硬件时间：

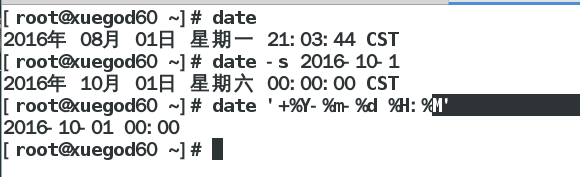
**BIOS硬件时间：**

**[root@xuegod60 ~]# hwclock**

**查看系统时间**

**[root@xuegod60 ~]# date**

## 1.3.2 修改时间



## 1.4.1 参数-h 或—help

**[root@xuegod60 ~]# date --help**

**[root@xuegod60 ~]# fdisk –h**

**[root@localhost ~]# man find**

**help 查看内部命令帮助信息**

**--help 查看外部命令帮助信息**

**1.4.2 关机命令**

**shutdown、 init 、reboot poweroff**

**shutdown**

**作用：关机，重启，定时关机**

**语法：shutdown  [选项]**

**-r     => 重新启动计算机**

**-h    => 关机**

**-h + 时间 =>定时关机**

**[root@xuegod60 ~]# shutdown -h +10**

**10分钟之后关机**

**[root@xuegod60 ~]# shutdown -h 23:30**

**指定具体的时间点进行关机**

**[root@xuegod60 ~]# shutdown -h now**

**立即关机**

**shutdown -r now  立即重启  
shutdown -r +10   #10 分钟以后重启  
shutdown -r  22：22  # 22:22 以后重启**

**init 命令**

**作用：切换系统运行级别**

**语法：init 0-6**

**Linux 7个启动级别：**

**0 系统停机模式，系统默认运行级别不能设置为0，否则不能正常启动，机器关闭。  
1 单用户模式，root权限，用于系统维护，禁止远程登陆，就像Windows下的安全模式登录。  
2 多用户模式，没有NFS网络支持。  
3 完整的多用户文本模式，有NFS，登陆后进入控制台命令行模式。  
4 系统未使用，保留一般不用，在一些特殊情况下可以用它来做一些事情。例如在笔记本电脑的电池用尽时，可以切换到这个模式来做一些设置。  
5 图形化模式，登陆后进入图形GUI模式，X Window系统。  
6 重启模式，默认运行级别不能设为6，否则不能正常启动。运行init 6机器就会重启。**

## 1.4.3 启动级别配置：

**RHEL7不再使用/etc/inittab文件进行默认的启动级别配置**

**systemd使用比sysvinit的运行级更为自由的target替代。**

**第3运行级用multi-user.target替代。**

**第5运行级用graphical.target替代。**

**runlevel3.target和runlevel5.target分别是**

**指向 multi-user.target和graphical.target的符号链接。**

**切换到第3运行级**

**[root@localhost ~]# systemctl isolate multi-user.target**

**或**

**[root@localhost ~]# systemctl isolate runlevel3.target**

**切换到第5运行级**

**[root@localhost ~]# systemctl isolate graphical.target**

**或**

**[root@localhost ~]# systemctl isolate runlevel5.target**

## 设置默认的运行界别

**设置默认第三启动级别**

**[root@xuegod60 ~]# systemctl set-default multi-user.target**

**Removed symlink /etc/systemd/system/default.target.**

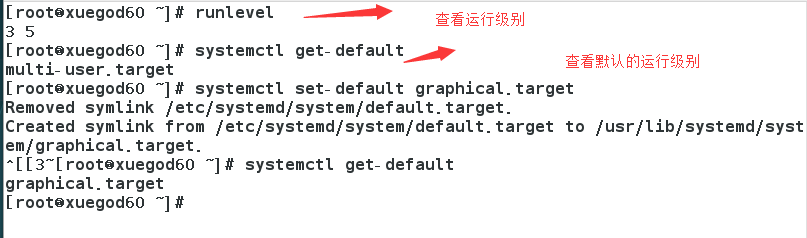
**Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/multi-user.target.**

**设置默认第五启动级别**

**[root@xuegod60 ~]# systemctl set-default graphical.target**

**Removed symlink /etc/systemd/system/default.target.**

**Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/graphical.target.**



**查看当前默认的启动级别(rhel7的命令)**

**[root@xuegod60 ~]# systemctl get-default**

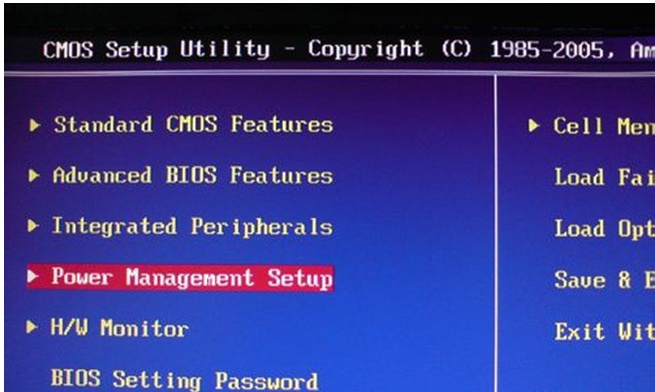
**graphical.target**

**实战，定时开机**

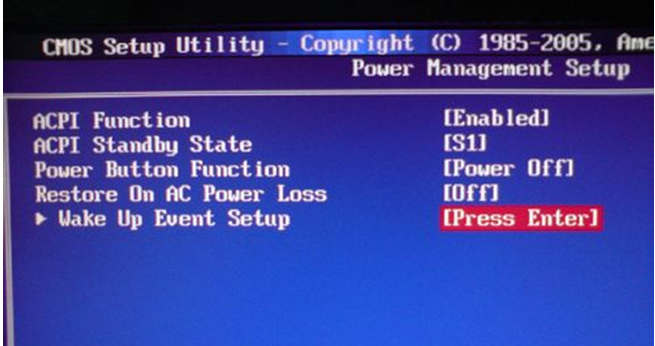
**BIOS**

**进入bios，一般是在开机后出现主板画面是按Delete这个键，部分品牌机可能按F2，F1**

**Power Management Setup，就进入电源管理设置了**



**通过回车进入这个设置后，选择Wake Up Event Setup，回车选择Press Enter**



**继续回车选择，将Disabied 更改为Enabled，然后继续回车确定。然后再继续设置时间点和日**

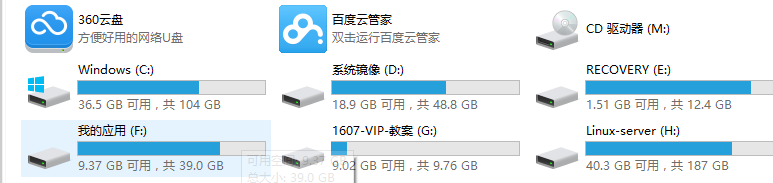






# 1.4 文件的基本管理和XFS文件系统备份恢复

* **1.1 Linux系统目录结构，相对/绝对路径。**
* **1.2 创建/复制/删除文件，rm -rf / 意外事故**
* **1.3 查看文件内容**
* **1.4 xfs文件系统的备份和恢复**



在Linux当中一切都是文件

## 1.1.1 linux系统目录结构

**/ /home /root /dev /usr /etc /boot /lib /var /tmp /proc /bin /sbin**

**/ 通常称为根分区。所有的文件和目录的起始点。只有root用户对此目录拥有写权限。**

**/etc 配置文件 包含所有应用程序的配置文件，也包含启动、关闭某个特定程序的脚本，**

**例如，**

**/etc/passwd， /etc/init.d/network等。**

**/boot 存放Linux系统启动时需要加载的文件。 Kernel、grub等文件都存放在此。**

**/var 是一个可增长的目录，包含很经常变的文件。**

**例如，/var/log（系统日志）、/var/lib （包文件）**

**/root 管理员所有数据。 root用户的家目录**

**/tmp 临时文件存储位置 不能存放重要的数据**

**/usr usr表示的是unix software source /usr/src 源代码目录**

**/bin 命令 此目录包含二进制可执行文件。**

**/sbin 系统命令 ，此目录中的命令主要供系统管理员使用，以进行系统维护。**

**/dev 包含设备文件。**

**在Linux中，一切都被看做文件。终端设备、磁盘等等都被看做文件.**

**如/dev/sda。**

**/home 普通用户所有数据存放在这个目录下**

**/proc   
　　这个目录是一个虚拟的目录，它是系统内存的映射，我们可以通过直接访问这个目录来获取系统信息。**

**查看咱们的内存信息,CPU信息**

**查看咱们的内存信息,CPU信息**

**cat /proc/meminfo | grep “Mem” // 查看内存**

cat /proc/cpuinfo

**/lib 存放系统的库文件**

**lib\*\*\*.a是静态库  
lib\*\*\*.so是动态库**

**静态库在编译时被加载到二进制文件中  
动态库在运行时加载到进程的内存空间中**

**简单的说，这些库就是为了让你的程序能够正常编译运行的。**

**其作用类似于Windows里的.dll文件。几乎所有的应用程序都须要用到这些共享库。**

**小结：**

**· 本地管理员大多数情况下将额外的软件安装在/usr/local目录下,并符号连接在/usr/local/bin下的主执行程序。**

**· 系统的所有设置在/etc目录下。**

**· 不要修改根目录(“/”)或/usr目录下的任何内容，**

**目录最好和LINUX发布时保持一致。**

**· 大多数工具和应用程序安装在目录：/bin, /usr/sbin, /sbin,**

**· 所有的文件在单一的目录树下。没有所谓的“驱动符”。**

## 1.1.2 绝对路径和相对路径

**路径：在我们平时使用计算机时要找到需要的文件就必须知道文件的位置，而表示文件的位置的方式就是路径。**

**绝对路径:**

**在Linux中，绝对路径是从”/”开始的，比如/usr、/etc/passwd。如果一个路径是从/开始的，它一定是绝对路径.**

pwd 判断位置

**[root@localhost ~]# pwd 注：判断用户当前所处的位置，也就是说他到底位于哪？**

**用户当前位于/root；**

**[root@localhost ~]# cd /etc/sysconfig/network-scripts**

**注：我们以绝对路径方式进入/目录下；**

**相对路径:  
 相对路径是以 . 或 .. 开始的，**

**.表示用户当前操作所处的位置**

**.. 表示上级目录**

**在路径中**

**.表示用户当前所处的目录，..上级目录.**

**要把.和..当做目录来看。**

**./表示当前目录**

**举例实验: 实验1**

**/root**

**注：得出目录处于/root目录中；  
 [root@localhost ~]# cd .**

**注：我们进入.  
 [root@localhost ~]# pwd**

**注：判断当前用户所处的位置；  
 /root**

**注：得出在/root 目录中 ；**

**举例实验: 实验2**

**确定当前的位置是/root  
 [root@localhost ~]# cd ..**

**注：我们切入/root的上级目录  
 [root@localhost /]# pwd**

**注：判断当前用户所处的位置。  
 /**

**注：用户当前位于/（根目录）中；**

## 1.2. 创建/查看/复制/删除 文件和文件夹

**1.2.1-1 创建文件和文件夹**

## touch

**作用：常用来创建空文件**

**语法： touch 文件名**

**[root@xuegod60 ~]# touch a.txt**

**将/temp/test.txt 文件拷贝到 / mut/test.txt 两个目录下都有形同的文件test.txt,复制后不提示**

**\mv /temp/test.txt /mnt/test.txt**

**/bin/mv /temp/test.txt /mnt/test.txt // 这样就不会提示了**

**Alias命令是显示别名的，显示哪些命令有别名，系统默认给的加mv 后加参数**

**Alias rm=‘echo “rm cmd is not allowd to use”’// 这样定义别名后使用rm命令就会提示不被使用**

## seq

Seq是生成序列行，seq 100 > /temp/test.txt就会在/temp/test.txt文件里面写入100行数，1~100，这个命令是生成序列数

## mkdir

**作用：创建目录**

**语法：mkdir 目录名**

**[root@xuegod60 ~]# mkdir test**

**[root@xuegod60 ~]# mkdir -p test/a/b/c**

**-p 再创建时连通父目录一起创建**

# 1.5 查看操作

## cat

**作用：查看文件内容**

**语法：cat 文件名**

**cat /etc/resolve.conf // 查询配置的DNS server**

cat –n nginx.conf 也可以显示行号（包括空行）

cat –n /etc/services|tail -1 显示文件有多少行

**分页查看**

## more + 文件名

**例：**

**more /etc/passwd**

**按下回车刷新一行，按下空格刷新一屏**

**q　退出**

## less　　查看文件名字

**q　退出**

**使用光标键可以向上翻页**

**linux中more与less的区别**

**more:不支持后退，但几乎不需要加参数，空格键是向下翻页，Enter键是向下翻一行，在不需要后退的情况下比较方便。**

**less：支持前后翻滚，既可以向上翻页（pageup按键），也可以向下翻页（pagedown按键）。，空格键是向下翻页，Enter键是向下翻一行**

## head

**-n　数字 显示多少行**

**[root@xuegod60 ~]# head -3 /etc/passwd**

**root:x:0:0:root:/root:/bin/bash**

**bin:x:1:1:bin:/bin:/sbin/nologin**

**daemon:x:2:2:daemon:/sbin:/sbin/nologin**

## tail

**从第后一行开始，查看文件，默认显示最后10行**

**-n 显示多少行**

**[root@xuegod60 ~]# useradd nginx**

**[root@xuegod60 ~]# tail -1 /etc/passwd**

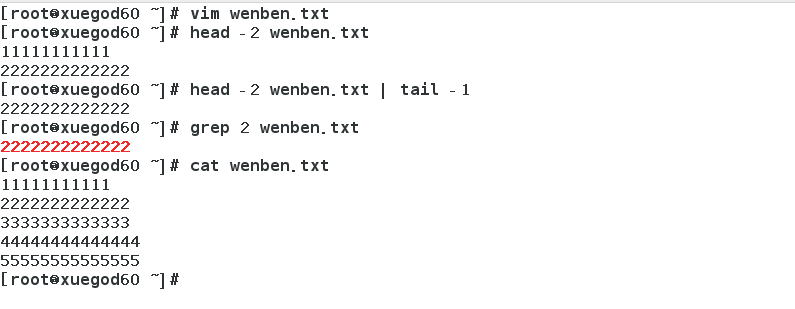
**nginx:x:1002:1002::/home/nginx:/bin/bash**

**[root@xuegod60 ~]#**

**-f 动态显示数据（不关闭）　　常用来查看日志**

**小实验:**

**Tail -n 100 /temp/test.txt // 显示默认后一百行 ，tail 默认是显示10行**



## cp

复制文件：

**作用：复制文件**

**语法：cp 源文件 目标文件**

**-r 包含子目录和文件 。**

**-p 保留原属性**

**[root@xuegod60 ~]# cp -r /boot/grub2/ /root/**

**重命名：**

**mv**

**语法：mv 源：文件或目录名 目标：文件或目录名**

**[root@xuegod60 ~]# mv rm.txt mk.txt**

删除文件和文件夹

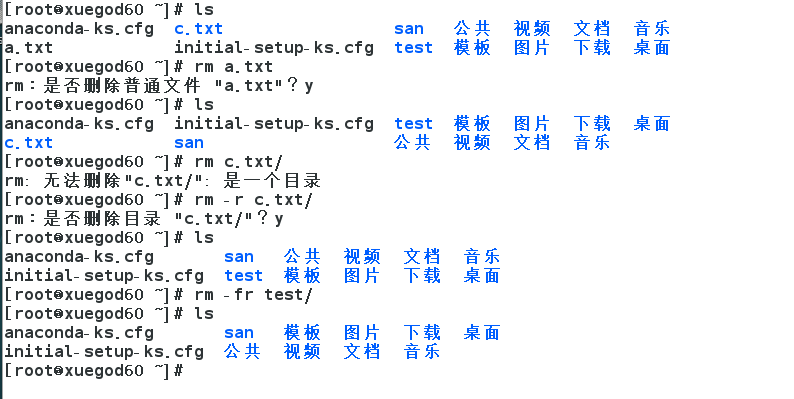
## rm

**作用：删除文件或目录**

**语法： rm -rf 文件或目录名**

**-r 递归删除（可以删除目录和目录里面的东西）**

**-f 强行删除**



## touch

Touch /data/oldboy.txt //在data目录下创建oldboy.txt文件

1. touch 1.txt 2.txt 3.txt 2.touch {1,2,3}.txt // 批量创建文件
2. touch /tmp/grup/{a..d}\_{1..3}.txt

a\_1.txt a\_2.txt a\_3.txt b\_1.txt b\_2.txt b\_3.txt c\_1.txt c\_2.txt c\_3.txt d\_1.txt d\_2.txt d\_3.txt

## grep

不显示a.txt中oldboy字符串

cat >a.txt<<EOF

Txt

Liyan

Oldboy

EOF // 先将字符写到文件中

过滤：

1. cat a.txt | grep -v “oldboy”// 过滤不包含这个字符的
2. grep -v oldboy a.txt

more 1.txt|grep -n "NullPointException" // 过滤相应的行，并显示行号

grep oldboy a.txt // 只打印过滤的内容

## sed

Sed删除方法， sed -e ‘/oldboy/d’test.txt // 从txt中删除oldboy内容，删除并没有删文件中的内容，只是输出的时候删除了，-e是编辑的意思，d是删除的意思

Sed -n /oldboy/p test.txt // 取消默认打印输出，-n是过滤出oldboy， p是打印出来

相当于 grep oldboy test.txt

sed -n /[^oldboy]/p test.txt // 处理不是oldboy的文件打印出来

sed -n ‘20,30p’test/txt // 只显示文件的20~30行内容

sed -n “$P” test.txt // 打印最后一行(注意是大字母P)

sed -n ‘2P’ test.txt // 打印第二行

sed -n ‘/ydds/’p text.txt // 打印匹配ydds的行

sed -n ‘4,/ydds/’p test.txt // 在第四行匹配模式ydds，若匹配，则打印

sed -n ‘/\$/’p test.txt // 匹配元字符$

sed -n ‘1,$’ p test.txt // 打印整个文件，只需将行范围设为第一行到最后一行1，$,$意为最后一行。

sed -n ‘/.\*ing’p test.txt // 任意字符，使用模式/.\*ing/ 查询以ing结尾的任意单词

sed -n ‘/movie/=’test.txt // 打印匹配行的号

sed -n -e ‘/movie/p’-e ‘/movie/=’test.txt // 打印匹配行的内容和行号

sed ‘=’test.txt | tail -2 显示行数和最大行号

sed –n ‘$=’ /etc/services 显示文件的有多少行

sed -i '3d' 1.txt // 如果是要删除第三行：

sed –I ‘2,4d’3.txt // 删除第二行到第四行的内容

sed -i '/^Love/d' 1.txt 如果删除以Love开头的行

sed -i '/Love/d' 1.txt 删除包含Love的行

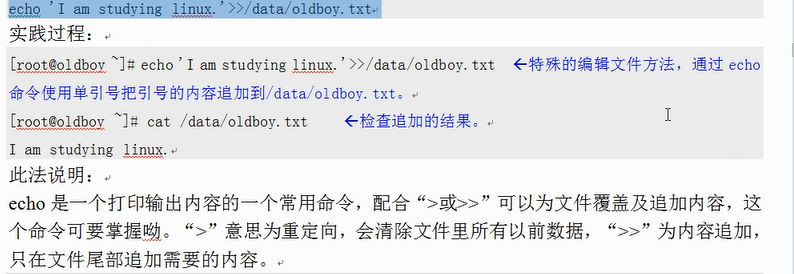
替换文件内容：

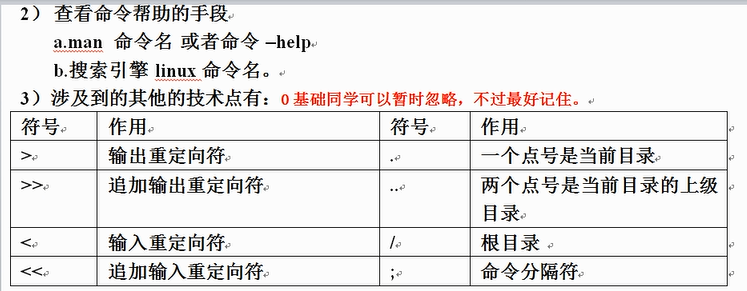
cat /etc/sysconfig/network

sed -i ‘s#要替换的内容#想替换的内容#g’文件的路径名

## echo

echo "" > 1.log // 清空日志用

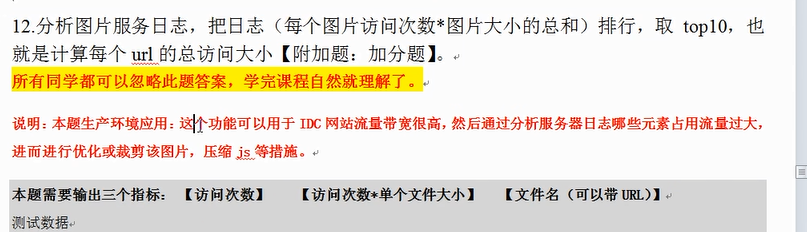


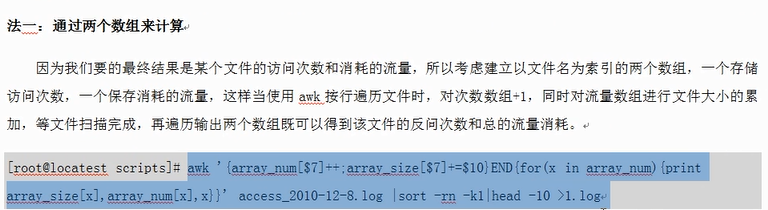


## awk

awk /[^oldboy]/ test.txt // 过滤不要oldboy，显示其余的字符

Awk ‘{if(NR>19 && NR<31) print $1}’ett.txt // 打印第20~30行的文件





awk ‘{print NR}’/etc/services |tail -1 显示文件有多少行

## tree

文件树型结果显示目录文件

## \

意思是将一个命令持续到下一行，

如cp -i \

>mylinux1.txt mylinu2.txt // 将命令分两行输入

## nslookup

查看域名解析过程

Dig [www.zhihuishu.com](http://www.zhihuishu.com) +trace 跟踪域名解析过程

Windows下是刷新缓存：ipconfig /flushdns

Linux 下：sudo /etc/init.d/nscd rstart

## which

which useradd // 查看useradd命令的路径,查询的是 echo $PATH //环境变量的路径

## tar

文件打包：

tar zcvf etc.tar.gz /etc

z是压缩，c是copy v是显示，f 是文件

tar zxvf 文件

排查其中的部分文件，给其余文件打包：

tar zcvf oldboy.tar.gz ./oldboy –exclude=oldboy/a.jpg

## wc

wc –l /etc/services 查看文件有多少行

vi nginx.conf

set nu 显示行号

ps –ef|grep jboss |wc –l 查看进程的数量，

## ps

ps –e –o “%C : %p : %z : %a”|sort –k5 –nr|head -10 查看进程按内存从大到小排列

ps –e –o “%C : %p : %z : %a”|sort –k5 –nr 查看cpu的利用率

## ln

我们安装的文件安装在/usr/local/appllication/apache2.2.17目录下面，想下次访问的时候直接访问/usr/local/application/apache可以直接这样做(实现链接地址的效果)

ln -s /usr/local/application/apache2.2.17 /usr/local/application/apache

问题：

已知apache服务的日志文件按天记录在服务器本地/app/logs 下面，由于磁盘空间紧张现在只需要保留7天的日志文件如何解决：

方案一：安装cronolog日志轮训软件，并在apache下配置如下形式按周轮训日志

CustomLog”|/usr/local/sbin/cronolog /app/logs/access\_%w.log” combined

/app/logs/access\_%1.log 每周一都会覆盖这个文件

方案二：

find ./ -type f -name “\*.log” -mtime +7|xargs rm -f 删除七天以前的文件

问题：

在一个窗口中操作，在另外一个窗口中跟踪文件的变化

for n in ‘seq 10000’; do echo 1111>> /var/log/message;usleep 500;done

tail -f /var/log/messages

# 1.6 实战：xfs文件系统的备份和恢复

**XFS提供了 xfsdump 和 xfsrestore 工具协助备份XFS文件系统中的数据。xfsdump 按inode顺序备份一个XFS文件系统。**

**与传统的UNIX文件系统不同，XFS不需要在dump前被卸载；对使用中的XFS文件系统做dump就可以保证镜像的一致性。这与XFS对快照的实现不同，XFS的dump和restore的过程是可以被中断然后继续的，无须冻结文件系统。xfsdump 甚至提供了高性能的多线程备份操作——它把一次dump拆分成多个数据流，每个数据流可以被发往不同的目的地。**

**首先了解一下xfsdump的备份级别有以下两种，默认为0（即完全备份）**

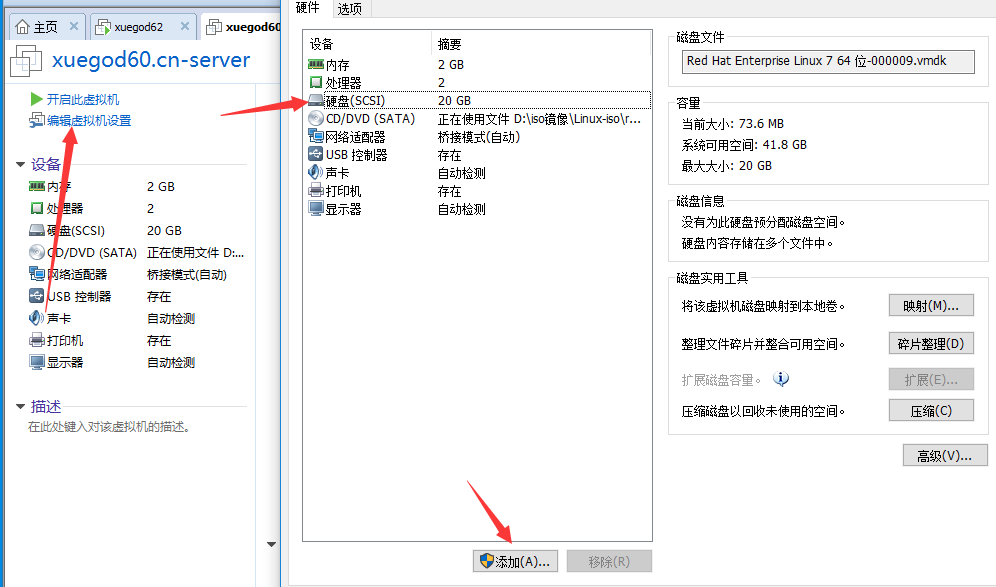
**0       完全备份**

**1       增量备份**

**实验环境 系统RHEL7.2 添加一块虚拟硬盘**

## 准备一个测试分区

**首先添加一块硬盘/dev/sdb，格式化为xfs文件系统**





**正常使用一块磁盘**

**添加磁盘🡪分区🡪格式化🡪挂载**

**[root@xuegod60 ~]# fdisk /dev/sdb #指定分区的设备**

**Welcome to fdisk (util-linux 2.23.2).**

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

**Be careful before using the write command.**

**Command (m for help): n #创建一个新的分区**

**Partition type:**

**p primary (2 primary, 0 extended, 2 free) #创建主分区**

**e extended #创建扩展分区**

**Select (default p):**

**Using default response p**

**Partition number (3,4, default 3): #指定分区号**

**First sector (25593856-41943039, default 25593856): #指定分区的开始柱面**

**Using default value 25593856**

**Last sector, +sectors or +size{K,M,G} (25593856-41943039, default 41943039): +1G #指定分区大小**

**Partition 3 of type Linux and of size 1 GiB is set**

**Command (m for help): p #打印分区表**

**Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 sectors**

**Units = sectors of 1 \* 512 = 512 bytes**

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

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

**Disk label type: dos**

**Disk identifier: 0x00083333**

**Device Boot Start End Blocks Id System**

**/dev/sdb1 25593856 27691007 1048576 83 Linux**

**Command (m for help): w #保存**

**The partition table has been altered!**

**Calling ioctl() to re-read partition table.**

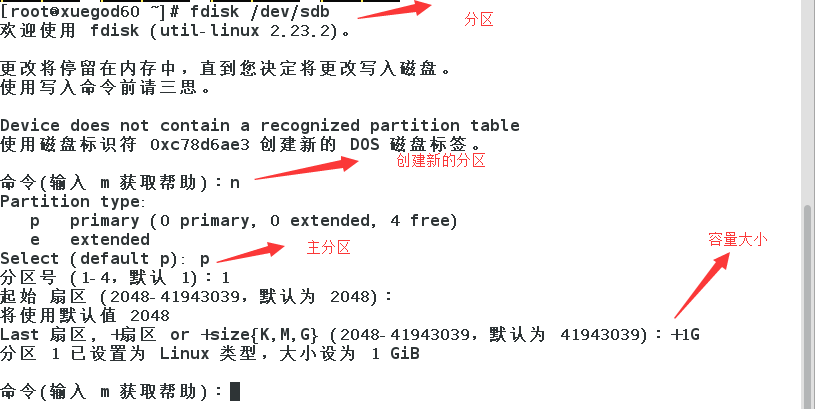
**WARNING: Re-reading the partition table failed with error 16: Device or resource busy.**

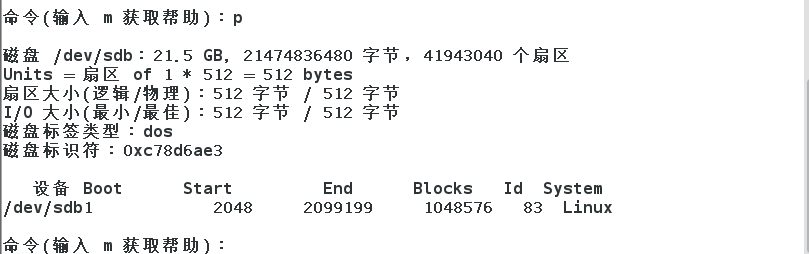
**The kernel still uses the old table. The new table will be used at**

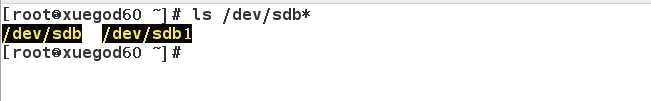
**the next reboot or after you run partprobe(8) or kpartx(8)**

**Syncing disks.**

**截图说明:**



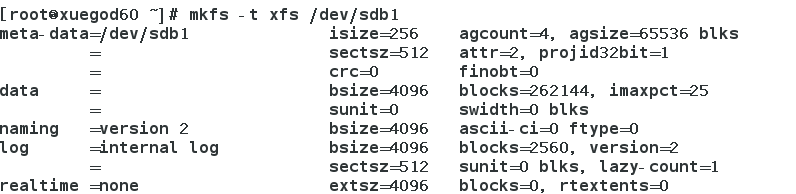




## 格式化分区，并进行挂载

**建立了一个名为sdb1的目录，把/dev/sdb1挂载在其目录下，并在下面创建两个文件做测试**

**[root@xuegod60 ~]# mkfs.xfs /dev/sdb1 #格式化分区**

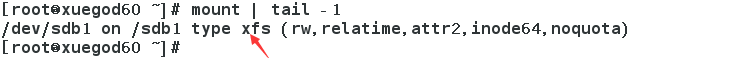


**[root@xuegod60 ~]# mkdir /sdb1 #创建挂载点**

**[root@xuegod60 ~]# mount /dev/sdb1 /sdb1/ #挂载**



**确认文件系统是xfs**



**准备备份测试文件**

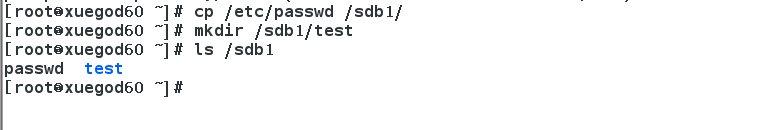
**[root@xuegod60 ~]# cp /etc/passwd /sdb1/**

**[root@xuegod60 ~]# mkdir /sdb1/test**

**[root@xuegod60 ~]# ls /sdb1/**

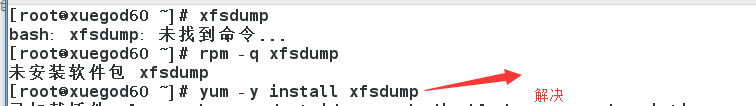
**passwd test**

**图解:**



## 对整个分区进行备份

1. **如果没有找到xfsdump 命令**



**[root@xuegod60 ~]# xfsdump -f /opt/dump\_sdb1 /sdb1**

**注意：备份的设备这里不能写成/sdb1/**

**xfsdump: using file dump (drive\_simple) strategy**

**xfsdump: version 3.1.4 (dump format 3.0) - type ^C for status and control**

**============================= dump label dialog ==============================**

**please enter label for this dump session (timeout in 300 sec)**

**-> dump\_sdb1**

**#指定备份会话标签**

**session label entered: "dump\_sdb1"**

**--------------------------------- end dialog ---------------------------------**

**xfsdump: level 0 dump of xuegod60.cn:/sdb1**

**xfsdump: dump date: Thu Jan 28 22:01:25 2016**

**xfsdump: session id: 29356f1f-8957-4fe0-ad70-2ae087f27abe**

**xfsdump: session label: "dump\_sdb1"**

**xfsdump: ino map phase 1: constructing initial dump list**

**xfsdump: ino map phase 2: skipping (no pruning necessary)**

**xfsdump: ino map phase 3: skipping (only one dump stream)**

**xfsdump: ino map construction complete**

**xfsdump: estimated dump size: 25536 bytes**

**xfsdump: /var/lib/xfsdump/inventory created**

**============================= media label dialog =============================**

**please enter label for media in drive 0 (timeout in 300 sec)**

**-> media0**

**#指定设备标签**

**media label entered: "media0"**

**--------------------------------- end dialog ---------------------------------**

**xfsdump: creating dump session media file 0 (media 0, file 0)**

**xfsdump: dumping ino map**

**xfsdump: dumping directories**

**xfsdump: dumping non-directory files**

**xfsdump: ending media file**

**xfsdump: media file size 24232 bytes**

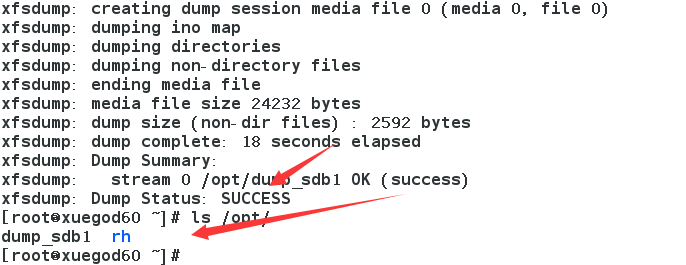
**xfsdump: dump size (non-dir files) : 2592 bytes**

**xfsdump: dump complete: 48 seconds elapsed**

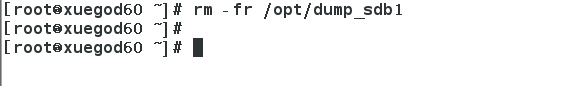
**xfsdump: Dump Summary:**

**xfsdump: stream 0 /opt/dump\_sdb1 OK (success)**

**xfsdump: Dump Status: SUCCESS**



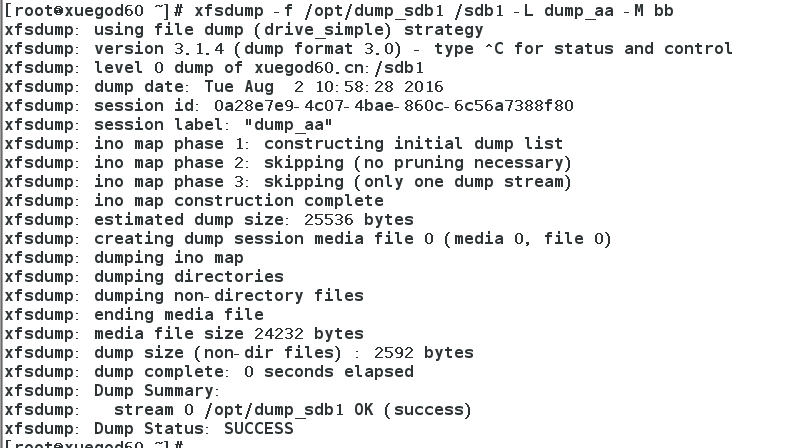
**删除刚刚备份的文件**

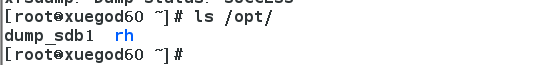


## 非交互式进行备份

**[root@xuegod60 ~]# xfsdump -f /opt/dump\_sdb1 /sdb1 -L dump\_sdb1 -M media0**

**注意：passwd /sdb1中间有空格，前后都不能加“/”**





**针对指定文件或目录进行备份**

**[root@xuegod60 ~]# xfsdump -f /opt/dump\_passwd -s passwd /sdb1 -L dump\_sdb1\_passwd -M media1**

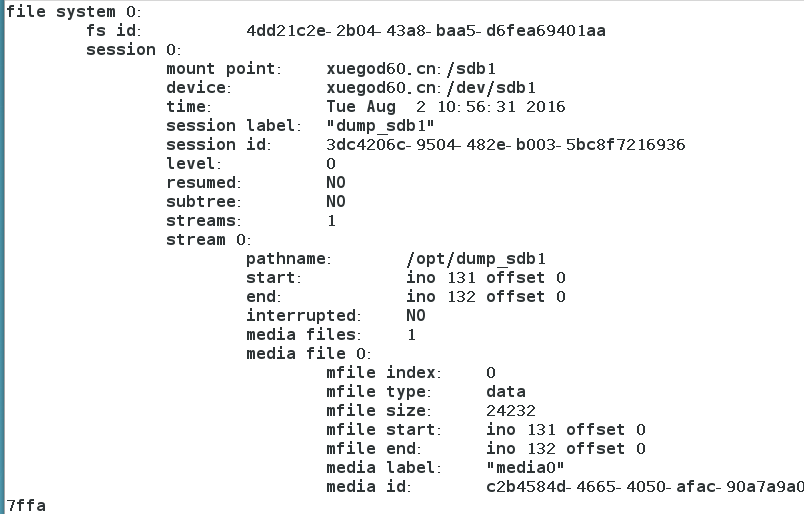
**查看备份文件是否生成**

**[root@xuegod60 opt]# ls**



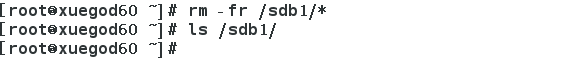
**查看备份信息**

**[root@xuegod60 ~]# xfsdump -I (大写)**



**文件系统恢复**

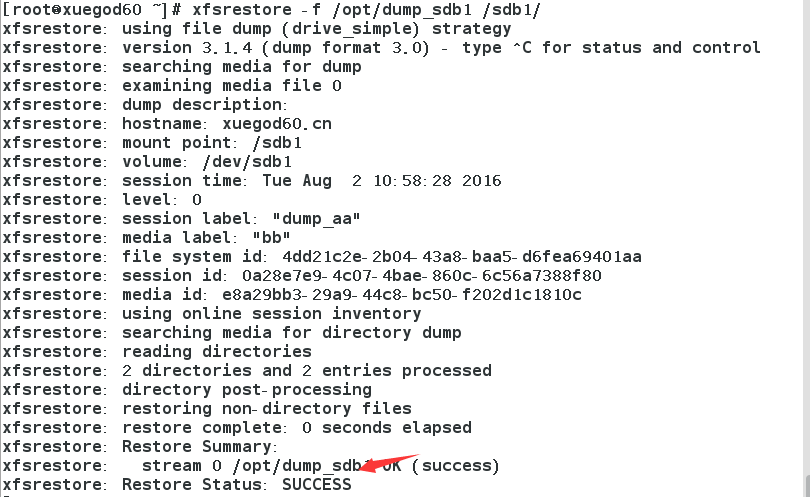
**首先，模拟数据被删除**



**[root@xuegod60 ~]# rm -rf /sdb1/\***

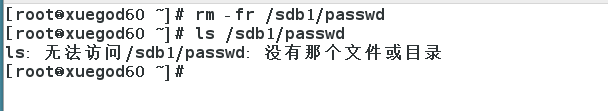
**恢复**

**[root@xuegod60 ~]# xfsrestore -f /opt/dump\_sdb1 /sdb1/**

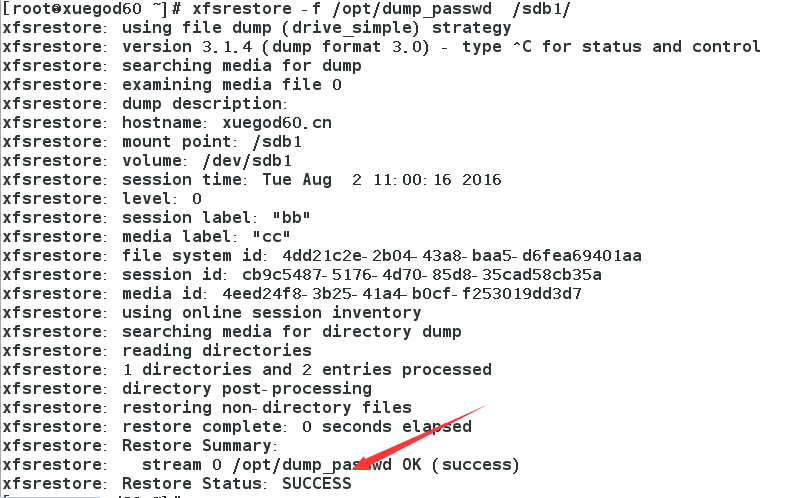




**只恢复单个的目录或文件**



**[root@xuegod60 ~]# xfsrestore -f /opt/dump\_sdb1\_passwd /sdb1/**





**1-5-vim编辑的使用和Xmanager远程工具的使用**

* 1. **vim主要模式介绍，vim命令模式。**
  2. **实战-解决vim打开之后乱码乱行的问题**
  3. **xmanager远程工具的使用**

# 1.7 vim主要模式介绍，vim命令模式。

**1. 确保系统已经安装了VIM工具**

**[root@xuegod60 ~]# rpm -qf `which vim` // 看是否安装了vim**

**vim-enhanced-7.4.160-1.el7.x86\_64**

**[root@xuegod60 ~]# rpm -qf `which vi` // 看是否安装了vi**

**vim-minimal-7.4.160-1.el7.x86\_64**

**[root@xuegod60 ~]#**

**vi和vim是同一个软件包安装的吗?**

**NO**

**注意:没有安装的话，安装方式**

**[root@xuegod61~]# rpm -ivh /mnt/Packages/vim-enhanced-7.4.160-1.el7.x86\_64.rpm**

**1、Vi /data/oldboy.txt 进入vi编辑器，点击i进入编辑模式，敲入编辑内容，然后esc退出编辑模式然后：wq保存并退出、**

**2、echo “i am studying linux.” > oldboy.txt, // 一个>是清空后面的文件并追加内容到文件，两个>>是直接追加内容到文件**

**３、cat >> /data/oldboy.txt<<EOR**

**I am studying linxt**

**I am studying linxt**

**I am studying linxt**

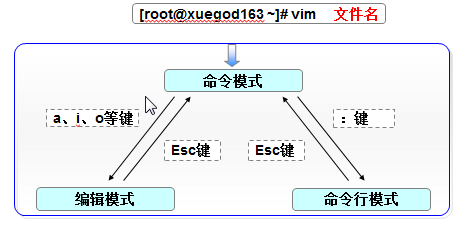
**EOF**

**// 批量追加多行**

**Ctrl +L清空当前屏幕**

### 1.1-2vim 编辑器模式切换:

**sed vim**



**命令模式 、命令行模式、编辑模式**

**首次进入文件: 命令模式**

**出现 Insert:编辑模式**

**输入: 表示 命令行模式**

**从编辑模式到命令行模式怎样切换?**

**编辑模式->esc->命令模式->: ->命令行模式**

**字符操作(怎样进入编辑模式?)**

**进入编辑模式 a io A I O**

**i 当前字符之前插入 (光标前)**

**I 行首插入 (行首)**

**a 当前字符之后插入 (光标后)**

**A 行尾插入(行尾)**

**o 下一行插入 (另起一行)**

**O 上一行插入(上一行插入)**

**x 向后删除一个字符 等同于delete**

**X 向前删除一个字符**

**u 撤销一步 每按一次就撤销一次**

### 1.2-1行操作

**home键或^ (shift 6) 行首**

**$ (shift 4)end键 行尾**

**删除**

**dd**

**dd 删除一行 数字+dd 删除多行**

**删除5行,使用5dd**

**删除光标后全部内容 d+G**

**复制粘贴**

**yy 复制一行 数字+yy 复制N行**

**p 将复制行粘贴**

**扩展：剪切**

**先删除，再粘贴**

**删除到行首 d + HOME 或^ (shift+6)**

**删除到行尾 d + END 或$ (shift+4)**

### 1.2-4 V模式

**进入v模式 移动光标选择区域、**

**编程的时候需要进行多行注释：  
1、ctrl+v 进入列编辑模式**

**2向下或向上移动光标**

**3把需要注释的行的开头标记起来**

**4然后按大写的I**

**5再插入注释符,比如"#"。**

**6再按Esc,就会全部注释了。**

**删除：再按ctrl+v 进入列编辑模式；向下或向上移动光标 ；选中注释部分,然后按d, 就会删除注释符号。**

### 1.2-5 VIM命令行模式操作

**:w 保存 save  
:q 没有进行任何修改，退出 quit  
:q! 修改了，不保存，强制退出  
:wq 保存并退出**

**：wq! 强制保存并退出。**

**:!ifconfig 调用系统命令**

**!+命令**

### 1.3-1替换

**格式: %所有内容 s替换 旧的内容 新的内容**

**默认是每一行的第一个符合要求的词 (/g全部)**

**:% s/a/b 每一行的第一个a被替换成b**

**:% s/a/b/g 将文本中所有的1替换成5**

**:2,3 s/a/b**

**/g 全部**

**#替换2到3行 的5**

**我们怎么知道是第二行还是第三行呢?**

**:set nu #显示行号**

**set nonu #不显示行号**

**/ 正向查找 ：/san n 往下查找，N 往上查找**

**我想要编辑写入我的网卡地址,但是我忘记了,我要查看一下,当前在vim编辑文档**

**怎么办呢? 要退出文档吗? 这样好麻烦.**

**:!ifconfig 调用系统命令**

**!+命令**

**定位:**

**vim中定位到某行：**

**gg 定位到行首**

**G 定位到最后一行，行首**

**#G 定位到某一行**

**:# 定位到某一行**

**#gg 定位到某一行**

**#代表行号**

**读取其他文件**

**:r /etc/passwd**

**保存退出 :wq**

### 1.3-3vim打开多个文件：

**[root@xuegod60 ~]# vim -o /etc/passwd /etc/hosts**

**[root@xuegod60 ~]# vim -O /etc/passwd /etc/hosts**

**ctrl+ww 在文件之间进行切换**

**大写O左右分屏，小写的o上下分屏**

### 1.4 解决文件上传到Linux下打开乱码

在windows中编辑好的汉字文本文档。

xshell为例

rz 上传

sz下载

**上传的文件打开乱码，使用乱码解决工具 iconv**

**1.安装**

**[root@xuegod60 ~]# rpm -qf `which iconv` // 判断是否安装iconv**

**glibc-common-2.17-105.el7.x86\_64**

**// 如果没有安装，安装命令**

**[root@xuegod60 ~]# rpm -ihv /mnt/Packages/glibc-common-2.17-105.el7.x86\_64.rpm**

**通过iconv命令转码**

**输入/输出格式规范：**

**-f, --from-code=名称 原始文本编码  
-o, --output=FILE 输出文件  
-l, --list 列举所有已知的字符集**

**GB2312编码适用于汉字处理、**

**格式: Iconv -f 编码 源文件 -o 转换后文件**

**cat 文件出现乱码时候，字符转码解决方案**

**[root@xuegod60 ~]# iconv -f gb2312 san.txt -o new.txt**

**本节所讲内容：**

**恢复Linux下误删除的文件**

### 1.5 恢复删除文件和文件夹

**rm -rf 文件名或目录名**

**-r 递归删除（可以删除目录和目录里面的东西）**

**-f 强行删除**

**ext4文件系统上删除文件，可以恢复: extundelete**

**windows恢复误删除的文件： final data v2.0 汉化版 和 easyrecovery**

**Linux文件系统由三部分组成：文件名，inode，block**

**a.txt -->inode -> block**

**文件名 存放文件元数据信息 真正存放数据**

**查看inode号：**

**常识： 每个文件，最少有一个inode号。**

**[root@xuegod63 ~]# ls -i a.txt**

**393788 a.txt**

**查看inode中的文件属性; 通过stat命令查看inode中包含的内容**

**[root@xuegod63 ~]# stat a.txt #查看inode信息：**

**File: `a.txt'**

**Size: 1720 Blocks: 8 IO Block: 4096 regular file**

**Device: 802h/2050d Inode: 393788 Links: 1**

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

**Access: 2015-10-25 10:21:07.368552471 +0800**

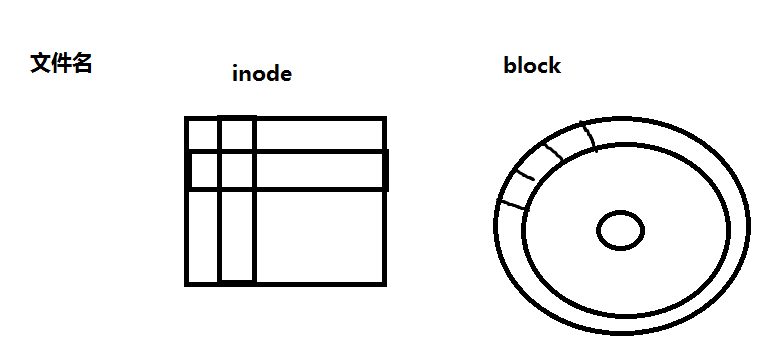
**Modify: 2015-10-25 10:21:02.773527478 +0800**

**Change: 2015-10-25 10:21:02.773527478 +0800**

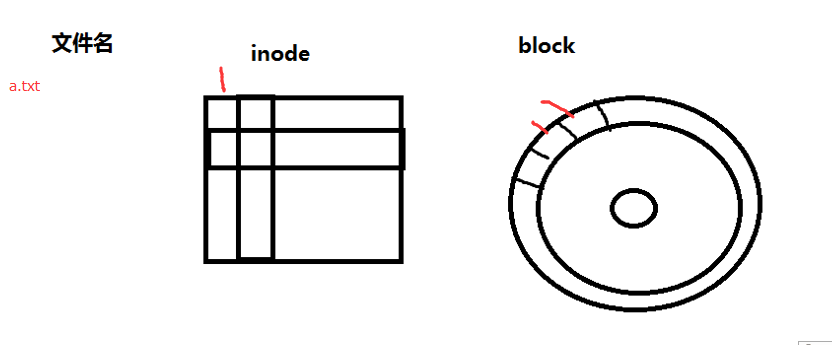
**block块：真正存储数据的地方**

**块 2M 数据 101M**

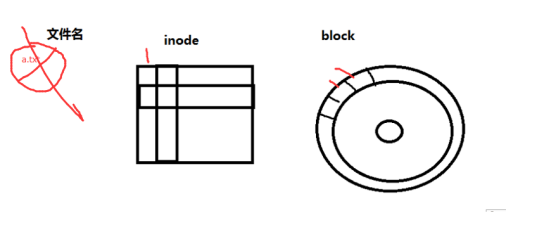
**为什么删除比复制快？**

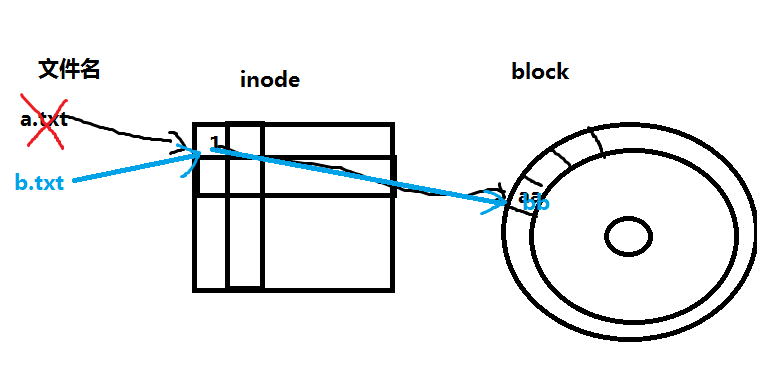
****

**创建**



**删除**





**如何避免误删除的文件内容被覆盖？？？**

**卸载需要恢复文件的分区：或以只读的方式挂载**

**实战：在RHEL6上恢复被误删除的文件**

**下载extundelete**

**http://sourceforge.net/ 开源软件发布中心**

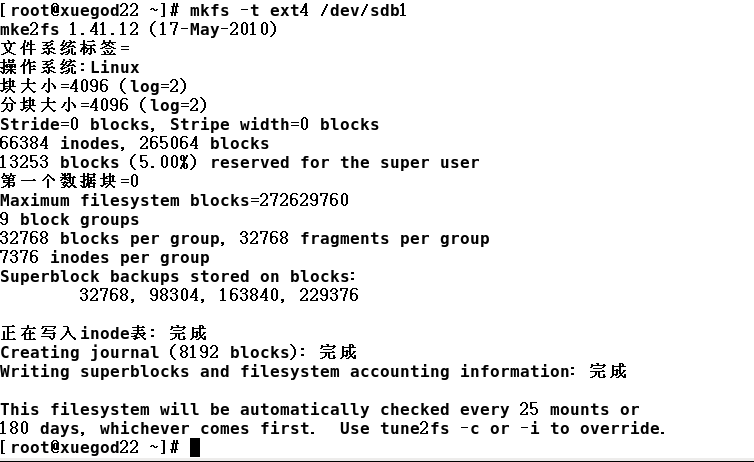
**准备测试分区：**

**[root@xuegod63 /]# fdisk /dev/sdb1 #创建一个sdb1分区**

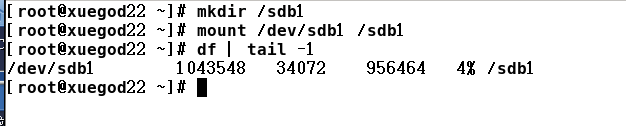
**root@xuegod63 ~]# partx -a /dev/sda #获得新分区表**



格式化 ext4 文件系统



挂载



**复制一些测试文件，然后把这些文件再删除：**

**[root@xuegod63 ~]# cp /etc/passwd /sdb1**

**[root@xuegod63 ~]# cp /etc/hosts /sdb1**

**[root@xuegod63 ~]# echo aaa > a.txt**

**[root@xuegod63 ~]# mkdir -p /sdb1/a/b/c**

**[root@xuegod63 ~]# cp a.txt /sdb1/a/**

**[root@xuegod63 ~]# cp a.txt /sdb1/a/b/**

**[root@xuegod63 ~]# touch /sdb1/a/b/kong.txt**

**安装tree命令：**

**[root@xuegod63~]#** **rpm -ivh /mnt/Packages/tree-1.5.3-2.el6.x86\_64.rpm**

**[root@xuegod63 ~]# tree /sdb1**



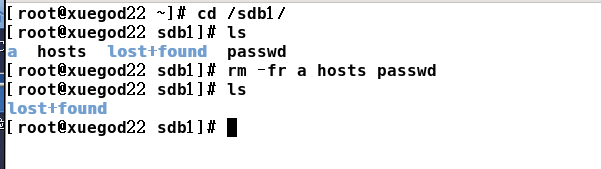
**删除文件：**

**[root@xuegod63 ~]# cd /sdb1**

**[root@xuegod63 sdb1]# ls**

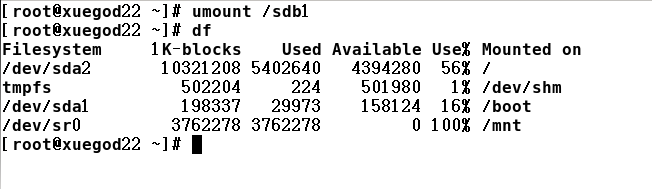
**a hosts lost+found passwd**

**[root@xuegod63 sdb1]# rm -rf a hosts passwd**



**如何避免误删除的文件内容被覆盖？？？**

**卸载需要恢复文件的分区：或以只读的方式挂载**



**上传extundelete到linux中：**

**从windows上传extundelete文件到linux，安装xmanager v4 或者ＣＲＴ**

**[root@xuegod63 ~]# rpm -ivh /mnt/Packages/lrzsz-0.12.20-27.1.el6.x86\_64.rpm**

**安装后，就有了ｒｚ命令和ｓｚ命令**

**ｒｚ　：　上传windows中的文件到linux**

**sz ：将linux中的文件传到windows**

**解压并安装extundelet**

**[root@xuegod63 extundelete-0.2.4]# tar jxvf extundelete-0.2.4.tar.bz2**

**[root@xuegod63 ~]# cd extundelete-0.2.4**

**[root@xuegod63 extundelete-0.2.4]# ls**

**acinclude.m4 autogen.sh configure depcomp LICENSE Makefile.in README**

**aclocal.m4 config.h.in configure.ac install-sh Makefile.am missing src**

**[root@xuegod63 extundelete-0.2.4]# ./configure #检查系统安装环境**

**Configuring extundelete 0.2.4**

**configure: error: Can't find ext2fs library**

**解决：**

**[root@xuegod63]# rpm -ivh /mnt/Packages/e2fsprogs-devel-1.41.12-11.el6.x86\_64.rpm**

**warning: /mnt/Packages/e2fsprogs-devel-1.41.12-11.el6.x86\_64.rpm: Header V3 RSA/SHA256 Signature, key ID fd431d51: NOKEY**

**Preparing... ########################################### [100%]**

**1:e2fsprogs-devel ########################################### [100%]**

**[root@xuegod63 extundelete-0.2.4]# make #编译，把源代码编译成可执行的二进制文件。**

**-j 4 使用4进程同时编译，提升编译速度 或 使用4核CPU同时编译。**

**[root@xuegod63 extundelete-0.2.4]# make install #安装**

**开始恢复：**

**方法1：通过inode结点恢复**

**方法二：通过文件名恢复**

**方法三：恢复某个目录，如目录a下的所有文件：**

**方法四：恢复所有的文件**

**[root@xuegod63 ~]# mkdir test #创建一个目录使用于存放恢复的数据**

**[root@xuegod63 ~]# cd test/**

**通过inode结点查看被删除的文件名字：**

**[root@xuegod63 test]# extundelete /dev/sdb1 --inode 2**

**方法1：通过inode结点恢复**

**[root@xuegod63 test]# extundelete /dev/sdb1 --restore-inode 12**

**[root@xuegod63 test]# diff /etc/passwd RECOVERED\_FILES/file.12 # 没有任何输出，说明一样**

**方法二，通过文件名恢复**

**[root@xuegod63 test]# extundelete /dev/sdb1 --restore-file passwd**



**方法三：恢复某个目录，如目录a下的所有文件：**

**[root@xuegod63 test]# extundelete /dev/sdb1 --restore-directory a**

**[root@xuegod63 test]# tree RECOVERED\_FILES/a/**

**├── a.txt**

**└── b**

**└── a.txt**

**方法四：恢复所有的文件**

**[root@xuegod63 test]# extundelete /dev/sdb1 --restore-all**

**[root@xuegod63 test]# tree RECOVERED\_FILES/**

**RECOVERED\_FILES/**

**├── a**

**│   ├── a.txt**

**│   └── b**

**│   └── a.txt**

**├── hosts**

**└── passwd**

**这是删除前的数据：**

**[root@xuegod63 ~]# tree /tmp/sda4/**

**/tmp/sda4/**

**├── a**

**│   ├── a.txt**

**│   └── b**

**│   ├── a.txt**

**│   ├── c #空目录**

**│   └── kong.txt #空文件**

**├── hosts**

**├── lost+found**

**└── passwd**

**extundelete在恢复文件的时候能不能自动创建空文件和目录？**

**答：不能。**

**REHL7 xfs系统 可以的**

**总结：**

**方法1：通过inode结点恢复**

**方法二：通过文件名恢复**

**方法三：恢复某个目录，如目录a下的所有文件：**

**方法四：恢复所有的文件**

# 1.8 优化开机服务及设置

## 1.8.1 设置selinux

vi /etc/selinux/config // 编辑selinux配置文件，防止一些软件因为selinux安全防护导致安装失败，设置SELINUX=disabled，然后想让系统配置生效用命令：setenforce 0 ,

查看用命令：getenforce,

## 1.8.2 设置开机默认运行级别

查看运行级别：runlevel

cat /etc/inittab 文件中id：5：initdefault // 显示及为默认的运行级别

工作中常用3级别，及文本模式

## 1.8.3 安装必要的软件包

yum grouplist

Yum groupinstall “Develompent Tools”

## 1.8.4 精简开机启动的服务

Windows > cmd > msconfig

linux手动设置Setup > system Service > 按Tab键设置确定或者退出

## 1.8.5 权限管理

先在root下操作命令：visudo



按yy粘贴一行。P复制一行

代表，oldboy 运行在所有机器上，切换到所有角色，只允许操作添加和删除用户

添加用户的命令:

>useradd leader

Passwd 123

Su - leader

Sudo su - root

Visudo编辑如下后，从oldboy切换到root角色下，不需要再输入密码了



Sudo -l // 查询自己有哪些权限

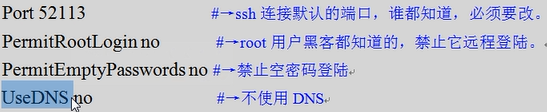
echo $PATH // 查询用户的环境变量的权限

编辑用户的权限：

vi .bash\_profile 给用户的环境变量添加进去，用：分割

然后让其生效，用命令：source .bash\_profile

## 1.8.6 更改ssh登陆配置

1. 更改前先进行备份：cp /etc/ssh/ssd\_config /etc/ssh/sshd\_config.ori
2. 编辑：vi /etc/ssh/sshd\_config
3. 

用yum或者rpm包安装的软件，一般都在/etc/init.d/sshd 下面

修改好后，用CRT如果连接不上，排查思路：

ping 192.168.129.129 ping服务端的ip地址看是不是通的

telnet 192.168.129.129 52113 如果一直显示常时间连接中，表示连接不通，怀疑服务端的服务开没开， 是不是防火墙的原有

lsof -i :52113 // 查看对应的端口的服务有没有开

/etc/init.d/iptables status // 看防火墙有没有开

/etc/init.d/iptables stop // 关闭防火墙

netstat -an|grep -i est // -i是不区分大小写，查看tcp连接

w // 查看哪些用户在连

## 1.8.7 设置字符集

1. 备份：cp /etc/sysconfig/i18n /etc/sysconfig/i18n.ori
2. echo ‘LANG=”ZH\_CH.GB18030”’ > /etc/sysconfig/i18n // 输入重定向
3. cat /etc/sysconfig/i18n // 查看修改
4. source /etc/sysconfig/i18n // 让配置生效
5. echo $LANG // 查看lang

## 1.8.8 配置163的yum源

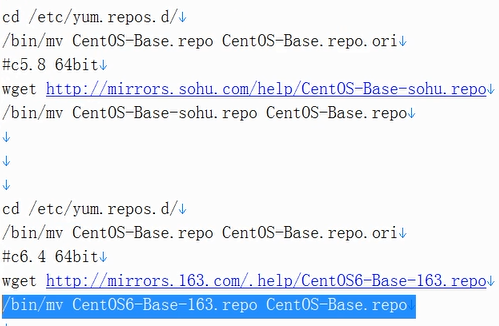
1. 进入yum源路径

cd /etc/yum.repos.d

1. 备份yum源

mv CentOS-Base.repo CentOS-Base.repo.Bak

3.设置镜像



wget http://mirrors.163.com/.help/CentOS6-Base-163.repo // 下载163的yum源

mv CentOS6-Base-163.repo CentOS-Base.repo.Bak // 替换原来的yum源

mv CentOS-Base.repo.Bak CentOS-Base.repo

// 用如下命令更新系统到最新

rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY\* // 导入key

yum upgrade

安装上传下载的工具和性能调优的工具：



安装包组：



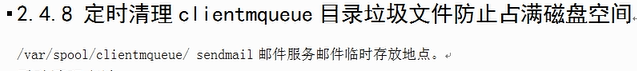
## 1.8.9 配置更新日期

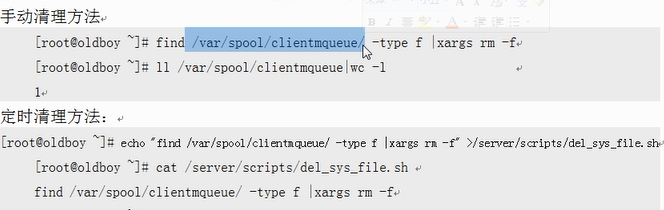
1. 手动更新

/usr/sbin/ntpdate time.nist.gov

2.加入到定时任务中更新，/var/spool/cron/root是定时任务的配置文件







## 1.8.10 配置网络

route -n 查看网络路由

route add default gw 10.0.0.254 配置默认网关

cat /etc/sysconfig/network-scripts/ifcfg-eth0 查看网卡配置

# 1.9 系统文件目录结构

## 1.9.1 一些常用命令

df -h 查看文件系统的挂载情况

tree -L 1显示一级目录

free -m 查看内存使用情况 cat /proc/meminfo

## 1.9.2 必须熟记的目录文件：

/etc/sysconfig/network-scripts/incfg-eth0 // 配置网络地址以及GW等使得网卡生效

通过setup-network configuration 修改网卡的配置就是在改这个网卡的配置文件

/etc/init.d/network restart, ifdown eth0&&ifup eth0

/etc/resolv.conf 设置本机的客户端DNS文件， DNS可以实现域名和ip相互解析

/etc/hosts 设定用户ip与名字（或域名）的对应表。相当于本地LAN内DNS

修改计算机名字：

>hostname test

>vi /etc/sysconfig/network 修改配置文件HOSTNAME=test

/etc/fstab 记录开机要mount（挂载）的文件系统的一个文件

tail -3 /etc/fstab // 查看系统挂载文件

/etc/inittab 设置系统启动是init进程将把系统设置成什么样的runlevel以及加载相关的配置文件

/ect/exports 设定NFS网络文件系统用的配置文件

/ect/inint.d 这个目录是用来存放系统或服务器以SystemV模式启动的脚本，

/etc/xinit.d 如果系统是通过xinetd模式运行的，他的脚本要放在这个目录下，有些系统没有这个目录，

/etc/profile 系统全局环境变量配置路径

定义环境变量：

vi /etc/profile

oldboy=’am I old boy’

. /etc/profile 让配置生效

echo $oldboy 查看配置的环境变量、有系统配置的全局环境变量$UID

/etc/profile.d 加载系统登录程序一个目录，命令或文件独立存在

/etc目录下的重要目录和文件

/etc/issue 记录用户登录前显示的信息

cat –n /etc/issue

/etc/motd 查看登录提醒

echo “Welcome to oldboy training Compute Service \!”>/etc/motd

cat /etc/motd

/etc/group 设定用户组的信息

/etc/passwd

/etc/shadow 密码信息文件

/etc/sudoers 可以执行使用sudo命令等的配置文件

/etc/securetty 设置哪些终端可以让root登录

LANG=EN

/etc/login.defs所有用户登录时的缺省配置

/etc/modprobe.conf 内核模块额外参数设定

/etc/rsyslog.conf 日志设置文件

其他目录

/lib 系统会使用到的函数库

/lib/modules kernel的相关模块

/var/lib/rpm rpm套件安装处

/usr/bin 这个目录是可执行程序的目录，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，相似的目录是/usr/local/sbin

/lib64 库文件存放目录

/usr/sbin库文件存放目录

/usr/sbin 这个目录是可执行程序的目录，但大多数存放涉及系统管理的命令，相似目录是/sbin 或/usr/local/sbin 或/usr/X11R6/sbin等

/usr/local 这个目录一般是用来存放用户自编译安装软甲的存放目录，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，一般是按照在这个目录中 相当于c:\Program files

/usr/local/bin 用户按照的小应用程序，和一些/usr/local目录下应用程序的符号链接

/usr/local/sbin 系统全局目录，课方式一些不需要加路径执行的脚本等

/usr/share 系统公用的东西存放地，比如/usr/share/doc 和/usr/share/man 等帮助文件、

/usr/src内核源码目录存放目录，

/usr/bin 使用者课执行的 binary file的目录

/usr/local/bin 使用者者课执行的binary file的目录

/usr/lib 系统会使用到的函数库

/usr/local/lib 系统会使用到的函数库

/var目录下的路径知识

/var 日志文件

/var/log 系统日志存放地点

/var/log/meeages 系统信息默认日志文件，非常重要，按周自动轮循

/var/spool/mail 系统用户邮件存放目录

/var/spool/clientmqueue sendmail临时邮件文件目录，有很多原因会导致这个目录碎文件很多，比如crontab 定时任务命令不加>/dev/mull 等，工作总偶尔会因为该目录文件太多，导致/var 所在的分区inode 数量被消耗尽，无法吸入文件的情况

/var/log/secure 记录登入系统存取信息的文件，按周自动轮询，例如pop3,ssh, telnet.ftp 等都会记录在此，系统安全的日志文件

/var/spool/cron/root 定时任务crontab默认，按用户名命令的文件

/proc 虚拟目录，是内存的映射

/proc/version 内核版本

/proc/cpuinfo 关于处理器的信息，如：类型，型号等

/proc/meminfo 系统每次信息

/proc/filesystems 当前运行内核锁配置的文件系统

/proc/interrupts 正在使用的中断，和曾经有多少个中断

/proc/loadavg 系统负载信息（系统的繁忙情况，比较准确，但是不够细致系统性能指标），uptime的结果

vi /etc/profile

export=PATH=/opt:$PATH 然后保存退出

tail -1 /etc/profile //查看文件

source /etc/profile 让生效

echo $PATH //查看path

然后就可以在任何目录下执行自己添加的/opt文件下的文件了

linux中以点.开头的文件是隐藏文件，查看隐藏文件使用 ls –al

也可以在 .bash\_profile文件中定义用户的环境变量，

/etc/profile是所有用户的

# 2.1 Linux中文件类型

>man find 查找

>/ -type 搜索

## 2.1.1普通文件

以-开头的文件是普通文件，在linux中以touch,cp,tar,echo,cat,等工具命令生成的文件都是普通文件，普通文件的删除方式可以使用rm命令

windows下的doc,ppt,zip,jpg等格式的文档在linux中都是普通文件，linux中的各种二进制文件， 各种服务器配置文件，如：/etc/exports, /etc/rc.local等都是纯文本文件

1.纯文本文件：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的

2.二进制文件：linxu中的可执行文件，比如命令文件

3.数据格式文件：有些程序在运行过程中会读取某些特定的格式的文件，那些特定格式的 文件被称数据文件，例如linux在用户登录时都会讲登录的数据记录在/var/log/wtmp那个文件内，文件是一个数据文件，通过last命令读取出来，cat会读取乱码，应为他属于一种特殊的格式文件，

如何通过命令判断普通文件的类型呢？？

file /etc/hosts

## 2.1.2 目录文件

mkdir a/b –p 创建多级目录

rmdir 删除 目录

find /dev –type b|xargs ls –l

## 2.1.3 设备文件

ll –al /etc/tty 我们看/etc/tty的属性是crw-rw-rw- 第一个字符是c这个表示字符设备文件

字符设备文件就是串行端口的接口设备，如猫等串口设备

## 2.1.4 块设备

ll –al /dev/sda 属性是brw-r-----,注意第一个字符是b这个表示是块设备，块设备技术储存数据供系统及吃过哪些访问的接口设备，如硬盘，光驱等都属于块设备

## 2.1.5 链接文件

[onlineExam@exam-yf-mem102 ~]$ ls -ld /etc/init.d

lrwxrwxrwx. 1 root root 11 Aug 14 2014 /etc/init.d -> rc.d/init.d

以l开头的文件表示链接文件是软链接文件

可以通过ln –s 源文件名 新文件名 来创建

默认情况下ln不带参数的情况下创建的是硬链接

# 2.2 Linux下的扩展名

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

② .sh表示shell的脚本文件，通过shell语言开发的程序

③ .pl表示perl语言文件，通过perl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表示系统的配置文件

七 .rpm表示rpm安装包文件

系统文件

.conf 配置文件，.lock锁文件，用来判断程序或设备使用正常被使用，.rpm

使用find的方式删除

find /app/logs/ -type f –mtime +7| xargs rm –f

#删除当前目录下的所有修改时间在7天以前的文件，#生产环境删目录下我呢间用这个好

为什么用-mtime修改时间

find /app/logs/ -type d –mtime +30|xargs rm –rf # 删除30天以前的目录，慎用

xargs是个不错的命令，

使用find的-exec

find /app/logs/ -type f –mtime +7 –exec rm { } \;#在/app/logs/ 目录中查找更改时间在7日以前的文件，并删除， -exec …\;为固定格式{}表示前面find 的内容

find –tyep / -name /-size /-mtime 查找文件

find ./-type f –name “\*.log”-mtime +7|xargs rm –f

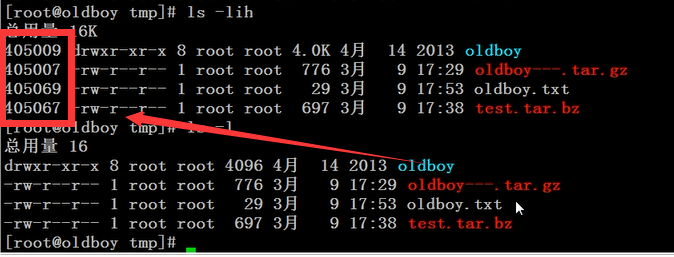
#意义在n天以前被修改过的文档

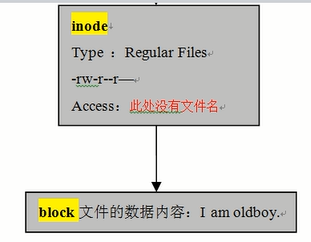
chkconfig –list sshd查看启动级别

LANG=EN

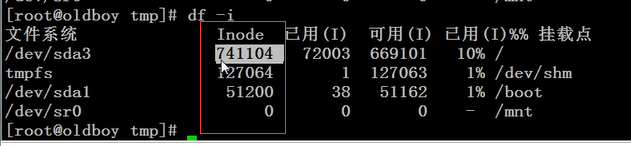
# 2.3 inode与block

inode除了记录文件属性的信息外，还会为每个文件进行信息的索引，所以有了inode的数值，操作系统根据指令，既可通过inode的值最快的找到相对应的文件实体，文件，inode,block之间的关系相当于书的目录和内容的关系



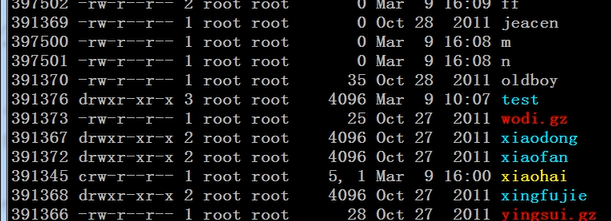


查看磁盘inode的数量



# 2.4 文件的链接

**ls -li**



inode 第一个字符是文件类型，权限和组 硬链接数

ln -s 创建的软链接的文件

硬链接是指通过索引节点（Inode)来进行链接，在Llinux(ext2,ext3)文件系统中，保存磁盘分区中的文件不管是什么类型都会给它们分配一个编号，这个编号被称为索引节点编号（Inode Index）简称为Inode即在文件系统文件中的编号

在文件系统中，多个文件名指向同一个索引节点inode是正常且被允许的这种情况的文件称硬链接

软连接 软连接文件实际上就是一个文本文件，这个文本文件包含有软件链接指向另一个文件的位置信息的内容，相当于“快捷方式”

创建一个目录后默认的链接数为2，ls -al 会显示两个，一个是 . 当前目录，一个是 。。 上一级目录

ls -lid . 显示目录结构

目录不能创建硬链接

结论：

1. 删除硬链接文件对源文件以及软链接文件无任何影响
2. 删除源文件，对硬链接文件没有影响，但是会导致软链接文件失效，白字红底闪烁
3. 同时删除源文件和硬链接文件，整个文件会真正的被删除。
4. 源文件和硬链接文件的索引节点好相同，可以认为是同一个文件，或者同一个文件的多个入口。
5. 源文件和软链接文件的索引节点好不同，是不同的文件，软链接文件可以认为是源文件的快捷方式。

问题：文件被删除的原理：

1. 源文件和者硬链接文件被删除
2. 没有线程读取这个文件了，这个文件，每当有线程读取，有一个计数器给数量+1，没有时候是0
3. 文件被覆盖了

安装Apache服务：

yum install httpd -y

/etc/init.d/httpd start

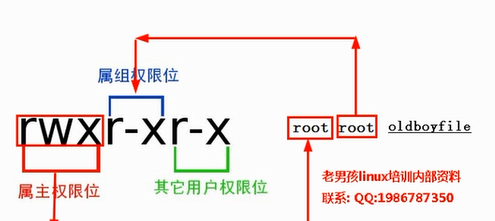
lsof -i :80 或者 netstat -lnt|grep 80 查看hppt服务

/etc/init.d/iptables stop

cd /etc/httpd/conf 编辑配置文件，让日志记录到/app/log下面

# 2.5 文件的权限

文件系统的用户是通过UID和GID来划分的，root的UID是0



对文件进行添加可执行权限：

chmod +x test.txt

对文件添加写权限：

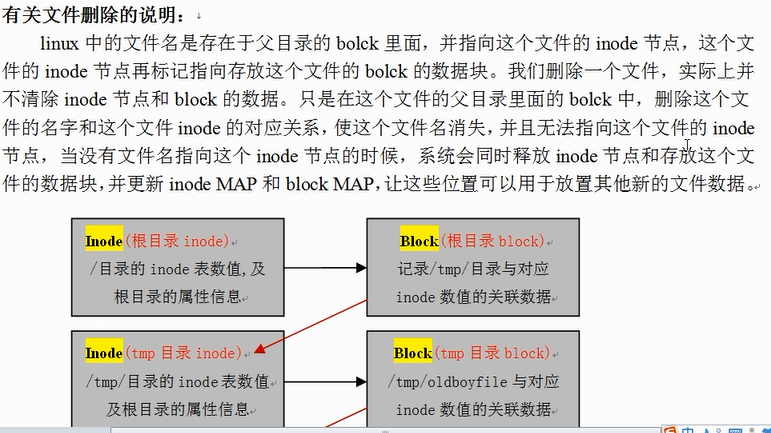
chmod 757 test.txt

结论：一个用户能不能对文件进行删除，要看文件的上级目录文件文件的权限，而不是文件本身

w:表示新增，修改文件的内容的权限；

x:表示具有执行文件的权限，普通用户还同时需要r的权限，root用户不用r的权限也能执行，本件本身也需要能执行。

Linux下的可执行文件：\*.sh , \*.py \*.perl等

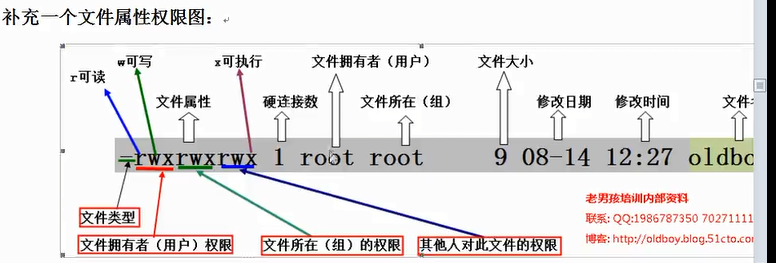


文件目录的权限：、

r:表示浏览目录的权限，

w:表示具有删除或修改目录内文件的权限

x:表示可进入目录的权限



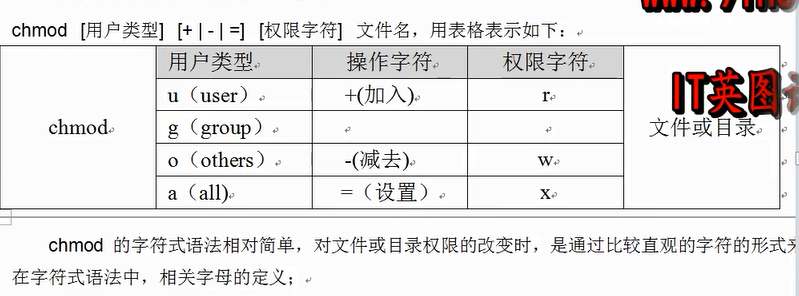
## 2.5.1 chmod 数字 文件

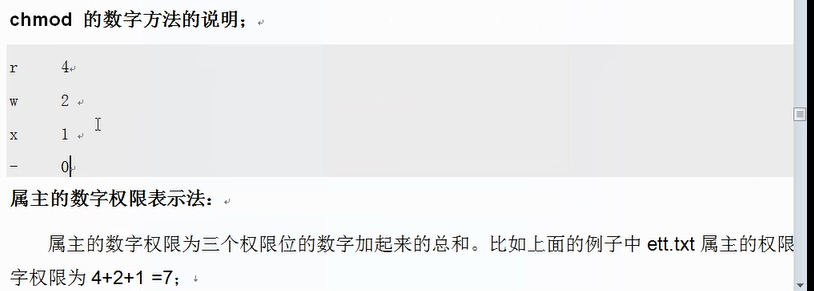
修改文件的权限，数字方式的计算方式：

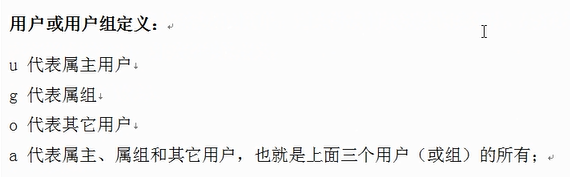
-r---w---x 421 test.txt

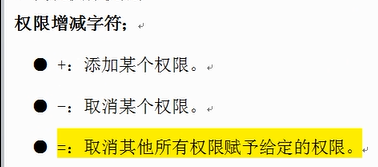
如果我们想改变的仅仅是打开目录的权限，使用chmod命令时不用加任何参数，如果想把所有文件或子目录也同时改变，需要使用-R参数

## 2.5.2 chmod 字符 文件









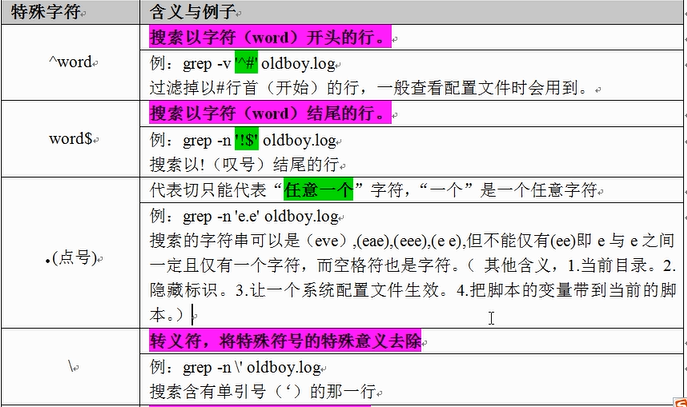
## 2.5.3 chown 用户名 文件

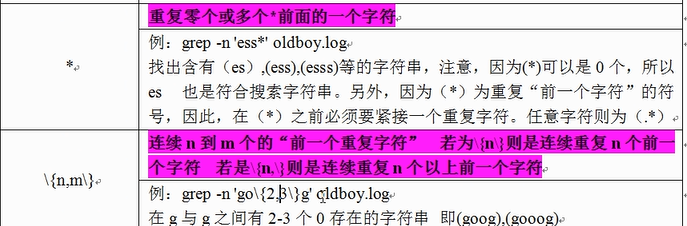
修改文件的的拥有着

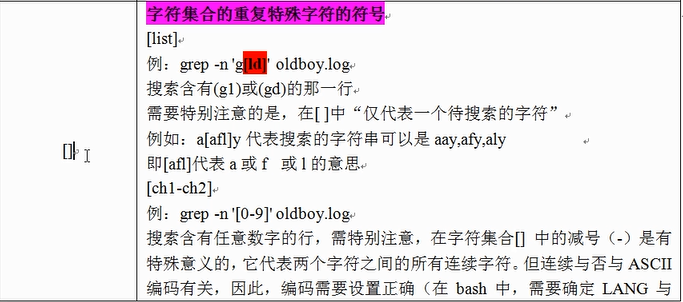
chown oldboy test

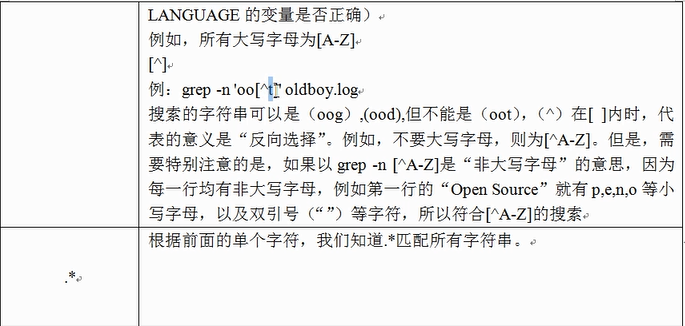
如果想给子目录也修改：chown -R oldboy test

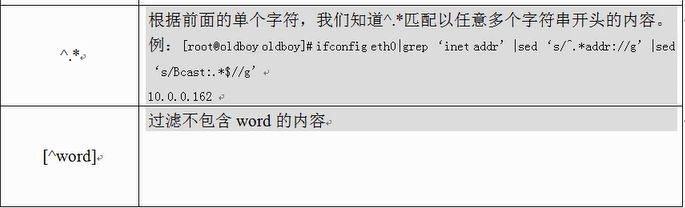
也可以，chown root:root test 使用：前是修改用户，后是修改组，的格式进行修改



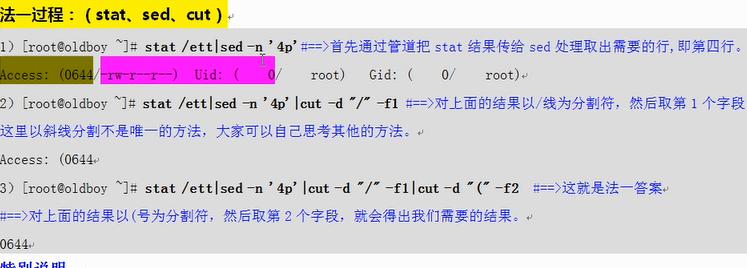








把相应的文件的权限转为数字：



# 2.6 定时任务 crond

定时任务分为两种，

一．linux系统自身执行的定时任务，如轮询系统日志，备份系统数据，清理系统缓存

ls -l /var/log/messages\*

ll /etc/ |grep cron 查看定时任务

ll /etc/cron.daily/ 查看每天的定时任务

1. 用户自定义的定时任务

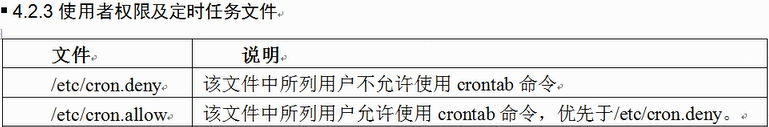
如时间同步等

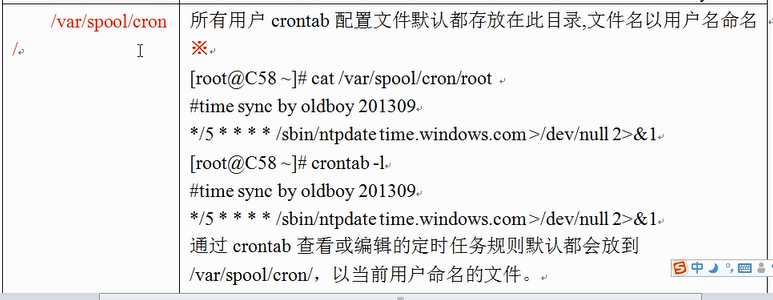
定时任务命令：crontab

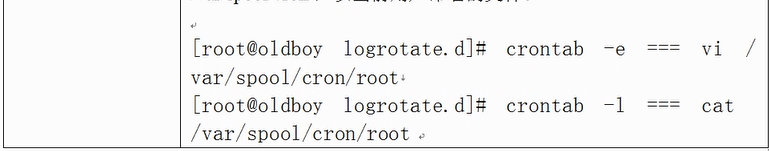
这个命令是周期性的执行任务，需要启动一个服务crond才行，

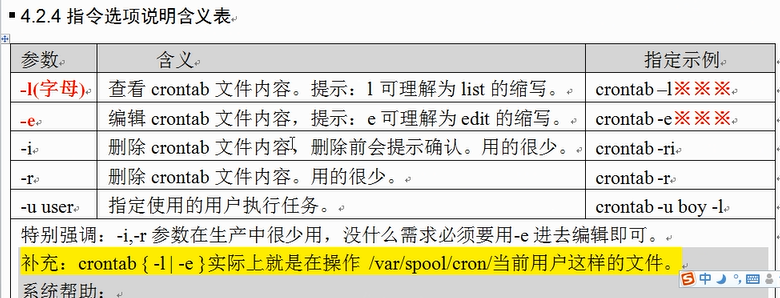
chkconfig --list crond

crond 的守护进程是0

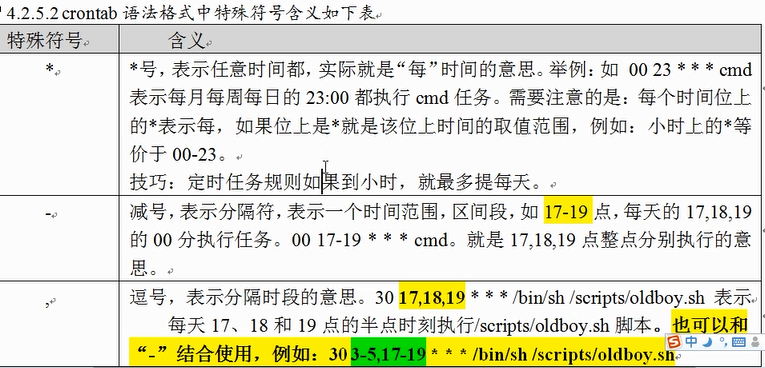








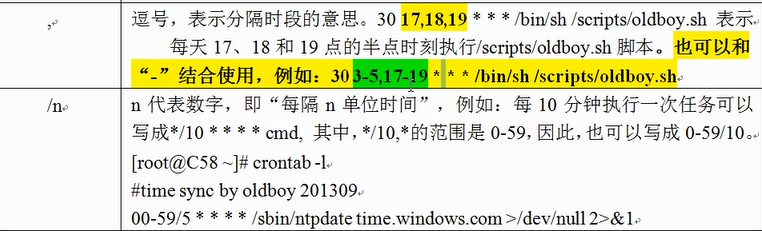




chkconfig --list crond 查看服务自启动

ps -ef|grep cron 查看服务进程是否在运行

/etc/init.d/crond restart 如果服务没有启动，使用命令进行启动



# 2.7 权限的集中管理

给用户组授权管理：

2.7.1 visudo命令自动编辑/etc/sudoers 文件

2.7.2 echo “oldboy ALL=(ALL) ALL”>>/etc/sudoers 追加一条命令让oldboy可以切换所有的主机，这种不会有语法检查，可以批量操作管理，

tail -1 /etc/sudoers

visudo -c 以上两种方式执行完后检查语法，

给组授权管理：

visudo 在添加组权限地方添加配置：注意给用户组授权前面有%

%sa ALL=(ALL) ALL

id sa // 判断组是否存在

groupadd -g 2001 sa 添加组

useradd xiaoli -g sa 给小李添加到sa组

passwd xiaoli 给小李添加密码

id xiaoli //查看小李组情况

# su - xiaoli 切换到小李权限

$ sudo su - 小李切换到root权限