**系统信息**  
arch 显示机器的处理器架构(1)   
uname -m 显示机器的处理器架构(2)   
uname -r 显示正在使用的内核版本   
dmidecode -q 显示硬件系统部件 - (SMBIOS / DMI)   
hdparm -i /dev/hda 罗列一个磁盘的架构特性   
hdparm -tT /dev/sda 在磁盘上执行测试性读取操作   
cat /proc/cpuinfo 显示CPU info的信息   
cat /proc/interrupts 显示中断   
cat /proc/meminfo 校验内存使用   
cat /proc/swaps 显示哪些swap被使用   
cat /proc/version 显示内核的版本   
cat /proc/net/dev 显示网络适配器及统计   
cat /proc/mounts 显示已加载的文件系统   
lspci -tv 罗列 PCI 设备   
lsusb -tv 显示 USB 设备   
date 显示系统日期   
cal 2007 显示2007年的日历表   
date 041217002007.00 设置日期和时间 - 月日时分年.秒   
clock -w 将时间修改保存到 BIOS   
  
  
  
**关机 (系统的关机、重启以及登出 )**  
shutdown -h now 关闭系统(1)   
init 0 关闭系统(2)   
telinit 0 关闭系统(3)   
shutdown -h hours:minutes & 按预定时间关闭系统   
shutdown -c 取消按预定时间关闭系统   
shutdown -r now 重启(1)   
reboot 重启(2)   
logout 注销   
  
  
  
**文件和目录**   
cd /home 进入 '/ home' 目录'   
cd .. 返回上一级目录   
cd ../.. 返回上两级目录   
cd 进入个人的主目录   
cd ~user1 进入个人的主目录   
cd - 返回上次所在的目录   
pwd 显示工作路径   
ls 查看目录中的文件   
ls -F 查看目录中的文件   
ls -l 显示文件和目录的详细资料   
ls -a 显示隐藏文件   
ls \*[0-9]\* 显示包含数字的文件名和目录名   
tree 显示文件和目录由根目录开始的树形结构(1)   
lstree 显示文件和目录由根目录开始的树形结构(2)   
mkdir dir1 创建一个叫做 'dir1' 的目录'   
mkdir dir1 dir2 同时创建两个目录   
mkdir -p /tmp/dir1/dir2 创建一个目录树   
rm -f file1 删除一个叫做 'file1' 的文件'   
rmdir dir1 删除一个叫做 'dir1' 的目录'   
rm -rf dir1 删除一个叫做 'dir1' 的目录并同时删除其内容   
rm -rf dir1 dir2 同时删除两个目录及它们的内容   
mv dir1 new\_dir 重命名/移动 一个目录   
cp file1 file2 复制一个文件   
cp dir/\* . 复制一个目录下的所有文件到当前工作目录   
cp -a /tmp/dir1 . 复制一个目录到当前工作目录   
cp -a dir1 dir2 复制一个目录   
ln -s file1 lnk1 创建一个指向文件或目录的软链接   
ln file1 lnk1 创建一个指向文件或目录的物理链接   
touch -t 0712250000 file1 修改一个文件或目录的时间戳 - (YYMMDDhhmm)   
file file1 outputs the mime type of the file as text   
iconv -l 列出已知的编码   
iconv -f fromEncoding -t toEncoding inputFile > outputFile creates a new from the given input file by assuming it is encoded in fromEncoding and converting it to toEncoding.   
find . -maxdepth 1 -name \*.jpg -print -exec convert "{}" -resize 80x60 "thumbs/{}" \; batch resize files in the current directory and send them to a thumbnails directory (requires convert from Imagemagick)   
  
  
  
**文件搜索**   
find / -name file1 从 '/' 开始进入根文件系统搜索文件和目录   
find / -user user1 搜索属于用户 'user1' 的文件和目录   
find /home/user1 -name \\*.bin 在目录 '/ home/user1' 中搜索带有'.bin' 结尾的文件   
find /usr/bin -type f -atime +100 搜索在过去100天内未被使用过的执行文件   
find /usr/bin -type f -mtime -10 搜索在10天内被创建或者修改过的文件   
find / -name \\*.rpm -exec chmod 755 '{}' \; 搜索以 '.rpm' 结尾的文件并定义其权限   
find / -xdev -name \\*.rpm 搜索以 '.rpm' 结尾的文件，忽略光驱、捷盘等可移动设备   
locate \\*.ps 寻找以 '.ps' 结尾的文件 - 先运行 'updatedb' 命令   
whereis halt 显示一个二进制文件、源码或man的位置   
which halt 显示一个二进制文件或可执行文件的完整路径   
  
  
  
**挂载一个文件系统**   
mount /dev/hda2 /mnt/hda2 挂载一个叫做hda2的盘 - 确定目录 '/ mnt/hda2' 已经存在   
umount /dev/hda2 卸载一个叫做hda2的盘 - 先从挂载点 '/ mnt/hda2' 退出   
fuser -km /mnt/hda2 当设备繁忙时强制卸载   
umount -n /mnt/hda2 运行卸载操作而不写入 /etc/mtab 文件- 当文件为只读或当磁盘写满时非常有用   
mount /dev/fd0 /mnt/floppy 挂载一个软盘   
mount /dev/cdrom /mnt/cdrom 挂载一个cdrom或dvdrom   
mount /dev/hdc /mnt/cdrecorder 挂载一个cdrw或dvdrom   
mount /dev/hdb /mnt/cdrecorder 挂载一个cdrw或dvdrom   
mount -o loop file.iso /mnt/cdrom 挂载一个文件或ISO镜像文件   
mount -t vfat /dev/hda5 /mnt/hda5 挂载一个Windows FAT32文件系统   
mount /dev/sda1 /mnt/usbdisk 挂载一个usb 捷盘或闪存设备   
mount -t smbfs -o username=user,password=pass //WinClient/share /mnt/share 挂载一个windows网络共享   
  
  
  
**磁盘空间**  
df -h 显示已经挂载的分区列表   
ls -lSr |more 以尺寸大小排列文件和目录   
du -sh dir1 估算目录 'dir1' 已经使用的磁盘空间'   
du -sk \* | sort -rn 以容量大小为依据依次显示文件和目录的大小   
rpm -q -a --qf '%10{SIZE}t%{NAME}n' | sort -k1,1n 以大小为依据依次显示已安装的rpm包所使用的空间 (fedora, redhat类系统)   
dpkg-query -W -f='${Installed-Size;10}t${Package}n' | sort -k1,1n 以大小为依据显示已安装的deb包所使用的空间 (ubuntu, debian类系统)   
  
  
  
**用户和群组**  
groupadd group\_name 创建一个新用户组   
groupdel group\_name 删除一个用户组   
groupmod -n new\_group\_name old\_group\_name 重命名一个用户组   
useradd -c "Name Surname " -g admin -d /home/user1 -s /bin/bash user1 创建一个属于 "admin" 用户组的用户   
useradd user1 创建一个新用户   
userdel -r user1 删除一个用户 ( '-r' 排除主目录)   
usermod -c "User FTP" -g system -d /ftp/user1 -s /bin/nologin user1 修改用户属性   
passwd 修改口令   
passwd user1 修改一个用户的口令 (只允许root执行)   
chage -E 2005-12-31 user1 设置用户口令的失效期限   
pwck 检查 '/etc/passwd' 的文件格式和语法修正以及存在的用户   
grpck 检查 '/etc/passwd' 的文件格式和语法修正以及存在的群组   
newgrp group\_name 登陆进一个新的群组以改变新创建文件的预设群组   
  
  
  
**文件的权限 - 使用 "+" 设置权限，使用 "-" 用于取消**  
ls -lh 显示权限   
ls /tmp | pr -T5 -W$COLUMNS 将终端划分成5栏显示   
chmod ugo+rwx directory1 设置目录的所有人(u)、群组(g)以及其他人(o)以读（r ）、写(w)和执行(x)的权限   
chmod go-rwx directory1 删除群组(g)与其他人(o)对目录的读写执行权限   
chown user1 file1 改变一个文件的所有人属性   
chown -R user1 directory1 改变一个目录的所有人属性并同时改变改目录下所有文件的属性   
chgrp group1 file1 改变文件的群组   
chown user1:group1 file1 改变一个文件的所有人和群组属性   
find / -perm -u+s 罗列一个系统中所有使用了SUID控制的文件   
chmod u+s /bin/file1 设置一个二进制文件的 SUID 位 - 运行该文件的用户也被赋予和所有者同样的权限   
chmod u-s /bin/file1 禁用一个二进制文件的 SUID位   
chmod g+s /home/public 设置一个目录的SGID 位 - 类似SUID ，不过这是针对目录的   
chmod g-s /home/public 禁用一个目录的 SGID 位   
chmod o+t /home/public 设置一个文件的 STIKY 位 - 只允许合法所有人删除文件   
chmod o-t /home/public 禁用一个目录的 STIKY 位   
  
  
  
**文件的特殊属性 - 使用 "+" 设置权限，使用 "-" 用于取消**  
chattr +a file1 只允许以追加方式读写文件   
chattr +c file1 允许这个文件能被内核自动压缩/解压   
chattr +d file1 在进行文件系统备份时，dump程序将忽略这个文件   
chattr +i file1 设置成不可变的文件，不能被删除、修改、重命名或者链接   
chattr +s file1 允许一个文件被安全地删除   
chattr +S file1 一旦应用程序对这个文件执行了写操作，使系统立刻把修改的结果写到磁盘   
chattr +u file1 若文件被删除，系统会允许你在以后恢复这个被删除的文件   
lsattr 显示特殊的属性   
  
  
  
**打包和压缩文件**  
bunzip2 file1.bz2 解压一个叫做 'file1.bz2'的文件   
bzip2 file1 压缩一个叫做 'file1' 的文件   
gunzip file1.gz 解压一个叫做 'file1.gz'的文件   
gzip file1 压缩一个叫做 'file1'的文件   
gzip -9 file1 最大程度压缩   
rar a file1.rar test\_file 创建一个叫做 'file1.rar' 的包   
rar a file1.rar file1 file2 dir1 同时压缩 'file1', 'file2' 以及目录 'dir1'   
rar x file1.rar 解压rar包   
unrar x file1.rar 解压rar包   
tar -cvf archive.tar file1 创建一个非压缩的 tarball   
tar -cvf archive.tar file1 file2 dir1 创建一个包含了 'file1', 'file2' 以及 'dir1'的档案文件   
tar -tf archive.tar 显示一个包中的内容   
tar -xvf archive.tar 释放一个包   
tar -xvf archive.tar -C /tmp 将压缩包释放到 /tmp目录下   
tar -cvfj archive.tar.bz2 dir1 创建一个bzip2格式的压缩包   
tar -xvfj archive.tar.bz2 解压一个bzip2格式的压缩包   
tar -cvfz archive.tar.gz dir1 创建一个gzip格式的压缩包   
tar -xvfz archive.tar.gz 解压一个gzip格式的压缩包   
zip file1.zip file1 创建一个zip格式的压缩包   
zip -r file1.zip file1 file2 dir1 将几个文件和目录同时压缩成一个zip格式的压缩包   
unzip file1.zip 解压一个zip格式压缩包   
  
  
  
**RPM 包 - （Fedora, Redhat及类似系统）**   
rpm -ivh package.rpm 安装一个rpm包   
rpm -ivh --nodeeps package.rpm 安装一个rpm包而忽略依赖关系警告   
rpm -U package.rpm 更新一个rpm包但不改变其配置文件   
rpm -F package.rpm 更新一个确定已经安装的rpm包   
rpm -e package\_name.rpm 删除一个rpm包   
rpm -qa 显示系统中所有已经安装的rpm包   
rpm -qa | grep httpd 显示所有名称中包含 "httpd" 字样的rpm包   
rpm -qi package\_name 获取一个已安装包的特殊信息   
rpm -qg "System Environment/Daemons" 显示一个组件的rpm包   
rpm -ql package\_name 显示一个已经安装的rpm包提供的文件列表   
rpm -qc package\_name 显示一个已经安装的rpm包提供的配置文件列表   
rpm -q package\_name --whatrequires 显示与一个rpm包存在依赖关系的列表   
rpm -q package\_name --whatprovides 显示一个rpm包所占的体积   
rpm -q package\_name --scripts 显示在安装/删除期间所执行的脚本l   
rpm -q package\_name --changelog 显示一个rpm包的修改历史   
rpm -qf /etc/httpd/conf/httpd.conf 确认所给的文件由哪个rpm包所提供   
rpm -qp package.rpm -l 显示由一个尚未安装的rpm包提供的文件列表   
rpm --import /media/cdrom/RPM-GPG-KEY 导入公钥数字证书   
rpm --checksig package.rpm 确认一个rpm包的完整性   
rpm -qa gpg-pubkey 确认已安装的所有rpm包的完整性   
rpm -V package\_name 检查文件尺寸、 许可、类型、所有者、群组、MD5检查以及最后修改时间   
rpm -Va 检查系统中所有已安装的rpm包- 小心使用   
rpm -Vp package.rpm 确认一个rpm包还未安装   
rpm2cpio package.rpm | cpio --extract --make-directories \*bin\* 从一个rpm包运行可执行文件   
rpm -ivh /usr/src/redhat/RPMS/`arch`/package.rpm 从一个rpm源码安装一个构建好的包   
rpmbuild --rebuild package\_name.src.rpm 从一个rpm源码构建一个 rpm 包   
  
  
  
**YUM 软件包升级器 - （Fedora, RedHat及类似系统）**  
yum install package\_name 下载并安装一个rpm包   
yum localinstall package\_name.rpm 将安装一个rpm包，使用你自己的软件仓库为你解决所有依赖关系   
yum update package\_name.rpm 更新当前系统中所有安装的rpm包   
yum update package\_name 更新一个rpm包   
yum remove package\_name 删除一个rpm包   
yum list 列出当前系统中安装的所有包   
yum search package\_name 在rpm仓库中搜寻软件包   
yum clean packages 清理rpm缓存删除下载的包   
yum clean headers 删除所有头文件   
yum clean all 删除所有缓存的包和头文件   
  
  
  
**DEB 包 (Debian, Ubuntu 以及类似系统)**  
dpkg -i package.deb 安装/更新一个 deb 包   
dpkg -r package\_name 从系统删除一个 deb 包   
dpkg -l 显示系统中所有已经安装的 deb 包   
dpkg -l | grep httpd 显示所有名称中包含 "httpd" 字样的deb包   
dpkg -s package\_name 获得已经安装在系统中一个特殊包的信息   
dpkg -L package\_name 显示系统中已经安装的一个deb包所提供的文件列表   
dpkg --contents package.deb 显示尚未安装的一个包所提供的文件列表   
dpkg -S /bin/ping 确认所给的文件由哪个deb包提供   
  
  
  
**APT 软件工具 (Debian, Ubuntu 以及类似系统)**  
apt-get install package\_name 安装/更新一个 deb 包   
apt-cdrom install package\_name 从光盘安装/更新一个 deb 包   
apt-get update 升级列表中的软件包   
apt-get upgrade 升级所有已安装的软件   
apt-get remove package\_name 从系统删除一个deb包   
apt-get check 确认依赖的软件仓库正确   
apt-get clean 从下载的软件包中清理缓存   
apt-cache search searched-package 返回包含所要搜索字符串的软件包名称   
  
  
  
**查看文件内容**   
cat file1 从第一个字节开始正向查看文件的内容   
tac file1 从最后一行开始反向查看一个文件的内容   
more file1 查看一个长文件的内容   
less file1 类似于 'more' 命令，但是它允许在文件中和正向操作一样的反向操作   
head -2 file1 查看一个文件的前两行   
tail -2 file1 查看一个文件的最后两行   
tail -f /var/log/messages 实时查看被添加到一个文件中的内容   
  
  
  
**文本处理**  
cat file1 file2 ... | command <> file1\_in.txt\_or\_file1\_out.txt general syntax for text manipulation using PIPE, STDIN and STDOUT   
cat file1 | command( sed, grep, awk, grep, etc...) > result.txt 合并一个文件的详细说明文本，并将简介写入一个新文件中   
cat file1 | command( sed, grep, awk, grep, etc...) >> result.txt 合并一个文件的详细说明文本，并将简介写入一个已有的文件中   
grep Aug /var/log/messages 在文件 '/var/log/messages'中查找关键词"Aug"   
grep ^Aug /var/log/messages 在文件 '/var/log/messages'中查找以"Aug"开始的词汇   
grep [0-9] /var/log/messages 选择 '/var/log/messages' 文件中所有包含数字的行   
grep Aug -R /var/log/\* 在目录 '/var/log' 及随后的目录中搜索字符串"Aug"   
sed 's/stringa1/stringa2/g' example.txt 将example.txt文件中的 "string1" 替换成 "string2"   
sed '/^$/d' example.txt 从example.txt文件中删除所有空白行   
sed '/ \*#/d; /^$/d' example.txt 从example.txt文件中删除所有注释和空白行   
echo 'esempio' | tr '[:lower:]' '[:upper:]' 合并上下单元格内容   
sed -e '1d' result.txt 从文件example.txt 中排除第一行   
sed -n '/stringa1/p' 查看只包含词汇 "string1"的行   
sed -e 's/ \*$//' example.txt 删除每一行最后的空白字符   
sed -e 's/stringa1//g' example.txt 从文档中只删除词汇 "string1" 并保留剩余全部   
sed -n '1,5p;5q' example.txt 查看从第一行到第5行内容   
sed -n '5p;5q' example.txt 查看第5行   
sed -e 's/00\*/0/g' example.txt 用单个零替换多个零   
cat -n file1 标示文件的行数   
cat example.txt | awk 'NR%2==1' 删除example.txt文件中的所有偶数行   
echo a b c | awk '{print $1}' 查看一行第一栏   
echo a b c | awk '{print $1,$3}' 查看一行的第一和第三栏   
paste file1 file2 合并两个文件或两栏的内容   
paste -d '+' file1 file2 合并两个文件或两栏的内容，中间用"+"区分   
sort file1 file2 排序两个文件的内容   
sort file1 file2 | uniq 取出两个文件的并集(重复的行只保留一份)   
sort file1 file2 | uniq -u 删除交集，留下其他的行   
sort file1 file2 | uniq -d 取出两个文件的交集(只留下同时存在于两个文件中的文件)   
comm -1 file1 file2 比较两个文件的内容只删除 'file1' 所包含的内容   
comm -2 file1 file2 比较两个文件的内容只删除 'file2' 所包含的内容   
comm -3 file1 file2 比较两个文件的内容只删除两个文件共有的部分   
  
  
  
  
**字符设置和文件格式转换**  
dos2unix filedos.txt fileunix.txt 将一个文本文件的格式从MSDOS转换成UNIX   
unix2dos fileunix.txt filedos.txt 将一个文本文件的格式从UNIX转换成MSDOS   
recode ..HTML < page.txt > page.html 将一个文本文件转换成html   
recode -l | more 显示所有允许的转换格式   
  
  
  
**文件系统分析**  
badblocks -v /dev/hda1 检查磁盘hda1上的坏磁块   
fsck /dev/hda1 修复/检查hda1磁盘上linux文件系统的完整性   
fsck.ext2 /dev/hda1 修复/检查hda1磁盘上ext2文件系统的完整性   
e2fsck /dev/hda1 修复/检查hda1磁盘上ext2文件系统的完整性   
e2fsck -j /dev/hda1 修复/检查hda1磁盘上ext3文件系统的完整性   
fsck.ext3 /dev/hda1 修复/检查hda1磁盘上ext3文件系统的完整性   
fsck.vfat /dev/hda1 修复/检查hda1磁盘上fat文件系统的完整性   
fsck.msdos /dev/hda1 修复/检查hda1磁盘上dos文件系统的完整性   
dosfsck /dev/hda1 修复/检查hda1磁盘上dos文件系统的完整性   
  
  
  
**初始化一个文件系统**   
mkfs /dev/hda1 在hda1分区创建一个文件系统   
mke2fs /dev/hda1 在hda1分区创建一个linux ext2的文件系统   
mke2fs -j /dev/hda1 在hda1分区创建一个linux ext3(日志型)的文件系统   
mkfs -t vfat 32 -F /dev/hda1 创建一个 FAT32 文件系统   
fdformat -n /dev/fd0 格式化一个软盘   
mkswap /dev/hda3 创建一个swap文件系统   
  
  
  
**SWAP文件系统**   
mkswap /dev/hda3 创建一个swap文件系统   
swapon /dev/hda3 启用一个新的swap文件系统   
swapon /dev/hda2 /dev/hdb3 启用两个swap分区   
  
  
  
**备份**   
dump -0aj -f /tmp/home0.bak /home 制作一个 '/home' 目录的完整备份   
dump -1aj -f /tmp/home0.bak /home 制作一个 '/home' 目录的交互式备份   
restore -if /tmp/home0.bak 还原一个交互式备份   
rsync -rogpav --delete /home /tmp 同步两边的目录   
rsync -rogpav -e ssh --delete /home ip\_address:/tmp 通过SSH通道rsync   
rsync -az -e ssh --delete ip\_addr:/home/public /home/local 通过ssh和压缩将一个远程目录同步到本地目录   
rsync -az -e ssh --delete /home/local ip\_addr:/home/public 通过ssh和压缩将本地目录同步到远程目录   
dd bs=1M if=/dev/hda | gzip | ssh user@ip\_addr 'dd of=hda.gz' 通过ssh在远程主机上执行一次备份本地磁盘的操作   
dd if=/dev/sda of=/tmp/file1 备份磁盘内容到一个文件   
tar -Puf backup.tar /home/user 执行一次对 '/home/user' 目录的交互式备份操作   
( cd /tmp/local/ && tar c . ) | ssh -C user@ip\_addr 'cd /home/share/ && tar x -p' 通过ssh在远程目录中复制一个目录内容   
( tar c /home ) | ssh -C user@ip\_addr 'cd /home/backup-home && tar x -p' 通过ssh在远程目录中复制一个本地目录   
tar cf - . | (cd /tmp/backup ; tar xf - ) 本地将一个目录复制到另一个地方，保留原有权限及链接   
find /home/user1 -name '\*.txt' | xargs cp -av --target-directory=/home/backup/ --parents 从一个目录查找并复制所有以 '.txt' 结尾的文件到另一个目录   
find /var/log -name '\*.log' | tar cv --files-from=- | bzip2 > log.tar.bz2 查找所有以 '.log' 结尾的文件并做成一个bzip包   
dd if=/dev/hda of=/dev/fd0 bs=512 count=1 做一个将 MBR (Master Boot Record)内容复制到软盘的动作   
dd if=/dev/fd0 of=/dev/hda bs=512 count=1 从已经保存到软盘的备份中恢复MBR内容   
  
  
  
**光盘**  
cdrecord -v gracetime=2 dev=/dev/cdrom -eject blank=fast -force 清空一个可复写的光盘内容   
mkisofs /dev/cdrom > cd.iso 在磁盘上创建一个光盘的iso镜像文件   
mkisofs /dev/cdrom | gzip > cd\_iso.gz 在磁盘上创建一个压缩了的光盘iso镜像文件   
mkisofs -J -allow-leading-dots -R -V "Label CD" -iso-level 4 -o ./cd.iso data\_cd 创建一个目录的iso镜像文件   
cdrecord -v dev=/dev/cdrom cd.iso 刻录一个ISO镜像文件   
gzip -dc cd\_iso.gz | cdrecord dev=/dev/cdrom - 刻录一个压缩了的ISO镜像文件   
mount -o loop cd.iso /mnt/iso 挂载一个ISO镜像文件   
cd-paranoia -B 从一个CD光盘转录音轨到 wav 文件中   
cd-paranoia -- "-3" 从一个CD光盘转录音轨到 wav 文件中（参数-3）   
cdrecord --scanbus 扫描总线以识别scsi通道   
dd if=/dev/hdc | md5sum 校验一个设备的md5sum编码，例如一张 CD   
  
  
  
**网络 - （以太网和WIFI无线**）   
ifconfig eth0 显示一个以太网卡的配置   
ifup eth0 启用一个 'eth0' 网络设备   
ifdown eth0 禁用一个 'eth0' 网络设备   
ifconfig eth0 192.168.1.1 netmask 255.255.255.0 控制IP地址   
ifconfig eth0 promisc 设置 'eth0' 成混杂模式以嗅探数据包 (sniffing)   
dhclient eth0 以dhcp模式启用 'eth0'   
route -n show routing table   
route add -net 0/0 gw IP\_Gateway configura default gateway   
route add -net 192.168.0.0 netmask 255.255.0.0 gw 192.168.1.1 configure static route to reach network '192.168.0.0/16'   
route del 0/0 gw IP\_gateway remove static route   
echo "1" > /proc/sys/net/ipv4/ip\_forward activate ip routing   
hostname show hostname of system   
host www.example.com lookup hostname to resolve name to ip address and viceversa(1)   
nslookup www.example.com lookup hostname to resolve name to ip address and viceversa(2)   
ip link show show link status of all interfaces   
mii-tool eth0 show link status of 'eth0'   
ethtool eth0 show statistics of network card 'eth0'   
netstat -tup show all active network connections and their PID   
netstat -tupl show all network services listening on the system and their PID   
tcpdump tcp port 80 show all HTTP traffic   
iwlist scan show wireless networks   
iwconfig eth1 show configuration of a wireless network card   
hostname show hostname   
host www.example.com lookup hostname to resolve name to ip address and viceversa   
nslookup www.example.com lookup hostname to resolve name to ip address and viceversa   
whois www.example.com lookup on Whois database

**JPS工具**

jps(Java Virtual Machine Process Status Tool)是JDK 1.5提供的一个显示当前所有java进程pid的命令，简单实用，非常适合在linux/unix平台上简单察看当前java进程的一些简单情况。

    我想很多人都是用过unix系统里的ps命令，这个命令主要是用来显示当前系统的进程情况，有哪些进程，及其 id。 jps 也是一样，它的作用是显示当前系统的java进程情况，及其id号。我们可以通过它来查看我们到底启动了几个java进程（因为每一个java程序都会独占一个java虚拟机实例），和他们的进程号（为下面几个程序做准备），并可通过opt来查看这些进程的详细启动参数。

**使用方法：在当前命令行下打 jps(需要JAVA\_HOME，没有的话，到改程序的目录下打) 。**

**jps存放在JAVA\_HOME/bin/jps，使用时为了方便请将JAVA\_HOME/bin/加入到Path.**

$>**jps**  
23991 Jps  
23789 BossMain  
23651 Resin

比较常用的参数：

**-q 只显示pid，不显示class名称,jar文件名和传递给main 方法的参数**  
$>  **jps -q**  
28680  
23789  
23651

**-m 输出传递给main 方法的参数，在嵌入式jvm上可能是null**

$> **jps -m**  
28715 Jps -m  
23789 BossMain  
23651 Resin -socketwait 32768 -stdout /data/aoxj/resin/log/stdout.log -stderr /data/aoxj/resin/log/stderr.log

**-l 输出应用程序main class的完整package名 或者 应用程序的jar文件完整路径名**

$> **jps -l**  
28729 sun.tools.jps.Jps  
23789 com.asiainfo.aimc.bossbi.BossMain  
23651 com.caucho.server.resin.Resin

**-v 输出传递给JVM的参数**

$> **jps -v**  
23789 BossMain  
28802 Jps -Denv.class.path=/data/aoxj/bossbi/twsecurity/java/trustwork140.jar:/data/aoxj/bossbi/twsecurity/java/:/data/aoxj/bossbi/twsecurity/java/twcmcc.jar:/data/aoxj/jdk15/lib/rt.jar:/data/aoxj/jd

k15/lib/tools.jar -Dapplication.home=/data/aoxj/jdk15 -Xms8m  
23651 Resin -Xss1m -Dresin.home=/data/aoxj/resin -Dserver.root=/data/aoxj/resin -Djava.util.logging.manager=com.caucho.log.LogManagerImpl -

Djavax.management.builder.initial=com.caucho.jmx.MBeanServerBuilderImpl

**sudo jps看到的进程数量最全**

**jps 192.168.0.77**

**列出远程服务器192.168.0.77机器所有的jvm实例，采用rmi协议，默认连接端口为1099**

**（前提是远程服务器提供jstatd服务）**

**注：jps命令有个地方很不好，似乎只能显示当前用户的java进程，要显示其他用户的还是只能用unix/linux的ps命令。**