**Linux命令大全**

**一、系统管理与设置;**

**二、用户和用户组管理;**

**三、磁盘管理;**

**四、文件和目录管理;**

**五、备份与压缩;**

**六、网络管理与相关应用;**

**七、vi/vim编辑器；**

**八、Shell编程**

一、系统管理与设置

1、信息显示命令

# man & info //帮助手册

# man 命令 //显示相应命令的帮助内容

# arch //显示当前系统体系结构

# cal //显示当前月份

# cal 2012 //显示2012年的月历

# cal 10 2012 //显示2012年10月的月历

# cal -y //显示整年日历

# cat /etc/issue //看当前系统发行版本

# cat /etc/redhat-release //看操作系统版本（redhat和centos）

# cat /etc/shells //查看shell版本

# cat /etc/services | more //查看各种服务的port

# cat /proc/cpuinfo //显示CPU信息

# cat /proc/cpuinfo | grep flags | grep ' lm ' | wc -l //结果大于0, 说明支持64位计算。lm指long mode, 支持lm则是64位

# getconf LONG\_BIT //查看CPU位数(32 or 64)

# cat /proc/cpuinfo | grep name | cut -f2 -d: | uniq -c //查看CPU型号

# cat /proc/cpuinfo | grep physical | uniq -c //查看实际有几颗CPU

# getconf LONG\_BIT //显示当前CPU运行在什么模式下

# cat /proc/devices //列出字符和块设备的主设备号，以及分配到这些设备号的设备名称

# cat /proc/filesystems //看文件系统

# cat /proc/interrupts //显示中断

# cat /proc/ioports //看设备io端口

# cat /proc/loadavg // 看系统负载

# cat /proc/meminfo //看内存信息

# cat /proc/mounts //显示当前系统所安装的文件系统信息

# free //看内存信息

# dmidecode //查看内存型号

# /usr/platform/sun4u/sbin/prtdiag -v //查看内存信息（unix）

# cat /proc/modules //看当前系统模块

# cat /proc/net/dev 显示网络适配器及统计

# cat /proc/partitions //看当前系统分区

# cat /proc/scsi/scsi //查看scsi硬盘信息

# cat /proc/swaps //看所有swap分区

# cat /proc/version //查看Linux内核版本

# cat /etc/security/limits.conf //查看打开最大文件数等设置

# cd /proc/pid号;ls -l exe //查看进程的完整路径

# date //显示当前时间

# date +'%Y/%m/%d' //以yyyy/mm/dd格式显示日期

# date +'%Y-%m-%d' //以yyyy-mm-dd格式显示日期

# date +%H:%M //显示时、分

# date -r test //显示test文件最后一次的修改时间

# dmesg //看启动信息

# dmidecode | grep "Product Name" //查看机器型号

# dmidecode | more //查看硬件（如内存型号、生产厂家等）信息

# dmidecode |grep 'Serial Number' //查看主板的序列号

# dmidecode -q //显示硬件系统部件 （SMBIOS / DMI）

# dmidecode -s system-serial-number //查看系统序列号

# dmidecode -t 11 //查看OEM信息

# dmidecode -t memory //查看内存信息

# dmidecode -t processor //查看CPU详细信息

# echo //显示文本行

# echo $LANG //显示系统语言

# echo $PATH //显示系统的环境变量

# env //显示所有环境变量

# export //查看环境变量（在bash下）

# fgconsole //显示活动的虚拟终端数目

# file -s /dev/sd\* //查看设备信息

# file /sbin/init //查看系统是32位还是64位：

64位系统输出信息如下：

sbin/init: ELF 64-bit LSB executable, AMD x86-64, version 1 (SYSV), for GNU/Linux 2.6.9, dynamically linked (uses shared libs), for GNU/Linux 2.6.9, stripped

32位系统输出信息如下：

/sbin/init: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.2.5, dynamically linked (uses shared libs), stripped

# free -m //以M为单位显示内存状态

# free -m -s5 //以M为单位，每隔5秒刷新一次内存状态

# gcc -v //查看GCC版本

# getconf LONG\_BIT //查看系统是32位还是64位

# glxinfo //显示有关GXL扩展和OpenGL渲染器的信息

# grep -i '10.52' /etc/hosts //查找/etc/hosts文件中包含10.52的行，不区分大小写

# 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

# head -5 /etc/passwd //看文件passwd的前5行

# tail -10 /etc/passwd //看文件passwd的后10行

# sed -n '5,10p' /etc/passwd //查看文件passwd的第5行到第10行

# history //显示历史记录

# history 5 //显示最近执行的5个历史记录

# history -c //清除历史记录

# sysdef -h //查看主机ID（Unix）

# hostname //显示主机名

# hostname -a //显示主机别名

# hostname -d //显示主机域名

# hostname -i //显示主机IP地址

# hostname 主机名 //设置主机名称

# hwclock //获取当前硬件时间

# hwclock --show //查看硬件时间

# clock --show //查看硬件时间

# hwclock –version //显示hwclock命令的版本信息

# info 命令 //显示相应命令info内容

# iostat -d 2 3 //评估磁盘性能，每隔2s刷新一次信息，且刷新3次

# lastlog //最近登入的时间

# locale //显示系统当前的语言设置

# locale -a //显示所有可用字体

# /etc/sysconfig/i18n //语言配置文件

# lsb\_release -a //查看系统发行版本，该命令适合所有的linux发行版本

# lsmod //显示所有已加载的模块

# lspci | more //显示当前系统的硬件配置

# lspci -v //查看系统硬件配置详细信息

# lspci -vv ｜more //查看系统硬件配置更为详细的信息

# lspci| grep Ethernet //查看网卡信息

# lspci -tv //列出所有PCI设备

# lsusb //输出所有的usb设备

# ls /etc/rc3.d/S\* |cut -c 15- //显示运行3级别开启的服务

# ls -l /lib/modules/$(uname -r)/kernel/fs //查看Linux支持哪些文件系统

# cat /proc/filesystems //查看当前已加载到内存中支持的文件系统

# mount //列出系统的所有分区

# mpstat //多处理器使用率

# netstat -pan|grep 177 //查看177端口情况

# netstat -nlap //查询进程占用哪些端口

# ps aux //显示所有进程状态和进程的基本信息

# ps -e //查看当前所有进程

# ps -u root –N //显示所有不是以root身份运行的进程

# ps -U ow2003 //显示ow2003用户进程

# ps -ef |grep ow2003 //显示ow2003用户进程

# pstree -h //高亮显示当前正在执行的进程

# pstree -p //以树状图方式显示进程及进程号

# pwd //显示当前目录

# rpm -qa redhat-release //看操作系统版本（只适用RH系列）

# rpm -qa | xargs rpm -V //查看系统中所有被修改过的文件，这对新安装的系统比较有效

# rpm -qa vsftpd | xargs rpm -V //查看系统中某个rpm包中的文件是否被修改

# runlevel //显示系统运行级别

# set //显示所有本地定义的Shell变量

# stat -f filename //显示文件系统状态

# stat filename //显示文件详细内容

# tac /etc/hosts //与cat相反，是将hosts内容从最后一行到第一行反向显示在屏幕上

# time ls //查看执行ls命令所需的时间

# top n 2 //显示进程信息，刷新两次后退出

# top -d 2 -n 3 -b >test.txt //将top的结果输出到文件test.txt中（每隔2秒，打印3次）

# top -bn1 //top命令列出所有进程

# top -bn1 > 1.txt //将进程输出到文件1.txt里

# touch --help //显示touch帮助信息

# tty //显示当前终端的名称

# uname -a //显示全部信息

64位系统输出如下信息：

Linux server141.guodu.net 2.6.18-164.el5 #1 SMP Tue Aug 18 15:51:48 EDT 2009 x86\_64 x86\_64 x86\_64 GNU/Linux

32位系统输出如下信息：

Linux bank.guodu.net 2.6.9-5.ELsmp #1 SMP Wed Jan 5 19:30:39 EST 2005 i686 i686 i386 GNU/Linux

# uname -i //显示硬件平台

# uname -m //显示机器硬件名

# uname -n //显示网络节点主机名

# uname -o //显示操作系统

# uname -p //显示处理器类型

# uname -r //显示内核版次

# uname -s //显示内核名

Linux内核版本信息：

Redhat 9.0———————————————2.4.20-8

RHEL 3 Update 8————————————2.4.21-47

RHEL 4 ————————————————2.6.9-5

RHEL 4 Update 1————————————2.6.9-11

RHEL 4 Update 2————————————2.6.9-22

RHEL 4 Update 3————————————2.6.9-34

RHEL 4 Update 4————————————2.6.9-42

RHEL 4 Update 5————————————2.6.9-55

RHEL 4 Update 6————————————2.6.9-67

RHEL 4 Update 7————————————2.6.9-78

CENTOS 5/RHEL 5 ———————————2.6.18-8

CENTOS 5.1/RHEL 5 Update 1——————2.6.18-53

CENTOS 5.2/RHEL 5 Update 2——————2.6.18-92

CENTOS 5.3/RHEL 5 Update 3——————2.6.18-128

CENTOS 5.4/RHEL 5 Update 4——————2.6.18-164

CENTOS 5.5/RHEL 5 Update 5——————2.6.18-194

CENTOS 5.6/RHEL 5 Update 6——————2.6.18-238

# uptime //查询系统自启动到现在总的运行时间及负载情况

# vmstat //显示虚拟内存的使用信息

# vmstat 2 3 //显示虚拟内存每隔2s刷新一次信息，且刷新3次

# whatis [命令或数据] //相当于man -f [命令或数据]

# yes string //设定重复显示的字符串

2、系统管理命令

# bc //调用计算器（如果要输出小数，则要执行 scale=number，这个number就是小数点后的位数），输入quit退出bc

# chkconfig rlogin on //开启rlogin服务

# chkconfig rsh on //开启远端服务器rsh

# chsh //shell设置

# chsh -l //列出当前所有可用的shell

# chsh -s /bin/bash //改变当前的shell设置为/bin/bash

# /bin/sh //切换shell

# exit //退出shell

# clear //清除终端屏幕

# cp /dev/cdrom mycd.iso & //后台运行制作镜像(加&)

# Ctrl + L //清除终端屏幕

# 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.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

# 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

# echo $shell //显示当前用户的默认shell

# fg 2178 //将后台进程编号为2178调到前台

# ftp 10.52.19.189 & //让ftp进程在后台执行

# init 0 //关闭系统（停止）

# telinit 0 //关闭系统(3)

# init 1 //单用户模式（root）

# init 2 //多用户文本模式（不能使用NFS）

# init 3 //多用户文本模式（能使用网络共享）

# init 5 //图形登录模式

# init 6 //重新启动系统

# shutdown -h now //关闭系统(1)

# 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

# jobs 查看后台运行的进程

# kill -9 2178 //强制终止ID为2178的进程（强制法可能导致系统资源无法正常释放，一般不推荐使用，除非其他办法都无效）

# mc //对当前目录文件进行可视化管理

# ntsysv //设置系统开机时启动的各种服务

# system-config-services //图形界面的设置系统开机时启动的各种服务

# pgrep nslookup //显示与指定字符串相关的进程

# pgrep -u root -l //查找由root用户创建的进程

# pgrep -u root -l -v //查找不是由root用户创建的进程

# ps -aux //显示当前运行的进程（静态）

# ps -ef // 查看所有进程

# pstree //查看当前进程树

# pwck /etc/passwd //检测passwd文件的正确性

# pwck /etc/shadow //检测shadow文件的正确性

# rlogin -l ow2003 10.52.19.200 //以用户ow2003登录远端主机

# rmmod 未使用的模块名 //卸载unused模块

# rsh 10.52.19.189 //登录远端主机10.52.19.189

# rsh -l ow2003 10.52.19.189 //以用户ow2003登录远端主机

# runlevel //显示系统当前运行级别

# sleep 2m //让系统休眠2分钟

# ssh 10.52.19.189 //登录远端主机10.52.19.189

# ssh -l ow2003 10.52.19.189 //以用户ow2003登录远端主机

# sudo -L //显示sudo命令可以使用的参数及相关描述信息

# sudo -u ow2003 ls -l /home //以用户ow2003身份执行命令

# suspend //暂停正在使用的shell

# suspend -f //强制暂停正在使用的shell

# symlinks -v / //显示根目录下的所有符号链接

# testparm //测试samba配置，回车后继续

# top //显示当前运行的进程（动态）

# top (Ctrl+Z) // 将当前top进程调到后台并停止

# vlock //锁定虚拟终端

# watch -n 10 --difference=cumulative who //以高亮字符显示累加差异

# watch -n 10 who //每隔10秒执行一次who命令

# whereis //找到指定文件的源、二进制文件和手册等各部分

# xkill //鼠标点死掉的图形即可终止，如果想终止xkill ，按右键取消

bash 快捷键:

常用：

Ctrl+a 开始位置

Ctrl+e 最后位置

Ctrl+k 删除此处至末尾所有内容

Ctrl+u 删除此处至开始所有内容

Ctrl-C 杀死当前任务

Ctrl-L 刷新屏幕

Ctrl-shift-c 复制

Ctrl-shift-v 粘贴

tab 补全

terminal窗口操作：

Alt+1 切换到标签页1

Alt+2 切换到标签页2

Alt+3 切换到标签页3

Ctrl+PageDown 后一标签页

Ctrl+PageUp 前一标签页

Shift+Ctrl+N 新建窗口

Shift+Ctrl+PageDown 标签页右移

Shift+Ctrl+PageUp 标签页左移

Shift+Ctrl+Q 关闭终端

Shift+Ctrl+T 新建标签页

Shift+Ctrl+W 关闭标签页

terminal窗口中的复制／粘贴:

Shift+Ctrl+C 复制

Shift+Ctrl+V 粘贴

改变terminal窗口大小：

!! 执行上一条命令

!?string? 执行含有string字符串的最新命令

!num 执行命令历史列表的第num条命令

↑(Ctrl+p) 显示上一条命令

↓(Ctrl+n) 显示下一条命令

Alt+< 历史列表第一项

Alt+> 历史列表最后一项

Ctrl - 减小

Ctrl+Shift + 放大

Ctrl 0 原始大小

Ctrl+r 然后输入若干字符，开始向上搜索包含该字符的命令，继续按Ctrl+r，搜索上一条匹配的命令

Ctrl+s 与Ctrl+r类似,只是正向检索

F11：全屏

history 显示命令历史列表

ls !$ 执行命令ls，并以上一条命令的参数为其参数

光标移动：

Alt+b 光标向后移动一个单词

Alt+c 把当前词汇变成首字符大写

Alt+d 剪切光标之后的词

Alt+f 光标向前移动一个单词

Alt+l 把当前词转化为小写

Alt+t 交换当前与以前单词的位置

Alt+u 把当前词转化为大写

Ctrl+(x u) 按住Ctrl的同时再先后按x和u，撤销刚才的操作

Ctrl+a 光标移到行首。

Ctrl+b 光标左移一个字母

Ctrl+c 删除整行

Ctrl+d 删除光标所在字母（注意与backspace以及Ctrl+h的区别，这两个是删除光标前的字符）

Ctrl+d 退出当前 Shell（当前行无字符时）

Ctrl+e 光标移到行尾。

Ctrl+f 光标右移。

Ctrl+h 删除光标前一个字符，相当于按 backspace 键。

Ctrl+k 清除光标后至行尾的内容。

Ctrl+l 清屏，相当于clear

Ctrl+q 重新启用挂起的shell

Ctrl+r 搜索以前输入过的命令。将有提示，根据输入的关键字搜索bash的history

Ctrl+s 挂起当前shell

Ctrl+t 交换光标位置前的两个字符。

Ctrl+u 清除光标前至行首间的所有内容。

Ctrl+v 插入特殊字符,如Ctrl+v+Tab加入Tab字符键

Ctrl+w 清除光标所在处之前的一个词（以空格、标点等为分隔符）

Ctrl+y 粘贴或恢复上次的删除。

Ctrl+z 把当前进程转到后台运行，使用 fg 命令恢复。

Esc+b 移动到当前单词的开头

Esc+f 移动到当前单词的结尾

Esc+t 颠倒光标所在处及其相邻单词的位置

Esc+w 删除光标所在处之前的字符至其单词尾（以空格、标点等为分隔符）

3、系统维护命令

# alias //列出已设置的别名

# alias rm 'rm -i' //设置别名

# authconfig //设置系统的认证信息

# bind -l //显示按键组合的内容

# bind -q abort //查询abort上绑定的键

# chattr +i filename //禁止删除

# chattr -i filename //取消禁止

# chattr +i /etc/fstab //禁止修改fstab文件

# chattr -i /etc/fstab //取消禁止修改fstab文件

# chmod 4755 test //使test文件具有SUID权限（4为SUID）。SUID对目录是无效的

# chmod 2755 test //使test文件具有SGID权限（2为SGID）。SGID可用于文件或目录

# chmod 1755 /test //使/test目录具有SBIT权限（1为Sticky bit即SBIT）。SBIT只对当前目录有效

# chmod 7666 test //test文件拥有者无执行权限，设置后的大写S、T表示“空的”，即没有执行权限

# lsattr -a test //显示文件test的隐藏属性

# lsattr -a //显示当前目录下所有的目录和文件，包括隐藏文件或目录

# lsattr -R //递归显示指定目录及子目录的内容

# lsattr //显示当前目录下的文件属性

# chkconfig --list //列出chkconfig所知道的所有服务

# chkconfig telnet off //关闭telnet服务

# chkconfig telnet on //开启telnet服务

# cat /etc/sysconfig/clock //查看所属时区和是否使用UTC时间

# date -s 2007-08-03 //只修改日期

# date -s "2007-08-03 14:15:00" //同时修改日期和时间，加双引号

# date -s 02/16/2012 //修改日期（按月日年格式）

# date -s 13:56:00 //修改时间（按时分秒格式）

# clock -r //查询BIOS时间

# cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime //修改时区

修改/etc/sysconfig/clock文件的内容为：

ZONE=”Asia/Shanghai”

UTC=false

ARC=false

# hwclock –w //同步BIOS时间

# hwclock -set -date="07/07/06 10:19:00'' //设置硬件时间（月/日/年 时:分:秒）

# clock -set -date="07/07/06 10:19:00'' //设置硬件时间（月/日/年 时:分:秒）

# hwclock -hctosys //硬件时钟与系统时钟同步（hc代表硬件时间，sys代表系统时间）

# clock -hctosys //硬件时钟与系统时钟同步（hc代表硬件时间，sys代表系统时间）

# hwclock -systohc //系统时钟与硬件时钟同步

# clock -systohc //系统时钟与硬件时钟同步

# tzselect //时区设置

# 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

# declare BASH=/bin/csh //修改变量BASH的值为/bin/csh

# declare -p //显示shell的所有变量及值

# declare -x //显示所有环境变量的值

# dircolors -b //显示Bourne Shell颜色代码设置

# dircolors -c //显示C Shell颜色代码设置

# dircolors -p //显示ls命令默认颜色值

# enable -a //显示shell的所有内置指令

# enable alias //加载内部命令alias

# enable -n alias //关闭alias

# eval pwd;df -h;ls //连接多个命令pwd，df -h和ls

# export MYENVIRON //自定义一个环境变量MYENVIRON

# export MYNAME= " Wang Xuri" //定义环境变量并赋值

# export -p //列出当前的环境变量值

# unset MYNAME //清除环境变量MYNAME

（注：以上命令使用的shell是bash）

# fdisk /mbr //删除GRUB

# hostid //显示当前主机的数字标识

# ldd /bin/ls //显示/bin/ls命令所使用的共享函数库

# ldd -v /bin/ls //以冗余模式显示的/bin/ls所使用的共享函数库

# losetup /dev/loop0 //显示设备/dev/loop0的状态

# depmod //分析可载入模块的相依性

# insmod //载入模块

# lsmod //显示所有已加载的模块

# lsmod -v usb-uhci //使用insmod命令安装模块"usb-uhci"

# modinfo //显示kernel模块的信息

# modprobe //自动处理可载入模块

# modprobe -c //显示当前自动处理可载入模块的默认配置

# rmmod //删除模块

# modprobe -v 8139too //安装网卡8139too

# mouseconfig --device psaux //指定鼠标端口为PS/2，鼠标为psaux

# mouseconfig --noui genericusb //以命令形式指定鼠标类型为genericusb

# mouseconfig --text //在图形界面方式下配置鼠标

# resize -c //使用C shell表示当前终端窗口大小

# resize -s 30 100 //设置虚拟终端大小，高为30列，长为100个字符

# resize -u //使用Bourne shell表示当前终端窗口大小

# rpm --checksig package.rpm //确认一个rpm包的完整性

# rpm -e package-name //卸载具体的软件包

# rpm -ev RealPlayer //卸载软件RealPlay

# rpm -hiv RealPlayer10GOLD.rpm //安装RealPlayer10GOLD.rpm包

# rpm -i mplay.rpm //安装mplay软件包（不显示信息）

# rpm -ivh --force mplay.rpm //强制安装mplay软件包（显示信息）

# rpm -ivh package-name //安装软件包并显示过程

# rpm -oldpackge mplay.rpm //降级mplay软件包

# rpm -q http //查询指定的包http是否安装

# rpm -q installed-package-name //查看是否安装

# rpm -q telnet-server //查看telnet服务器包

# rpm -qa //查看系统中已安装的所有rpm包（不分屏）

# rpm -qa|grep ftp //查询指定RPM

# rpm -qa|less //查询已安装RPM（分屏）

# rpm -qf luo //查询文件luo所属的软件包

# rpm -qf package-name //查询某个文件所属的软件包

# rpm -qi http //查询指定的包http详细信息（已安装）

# rpm -qi package-name //查看软件的描述信息

# rpm -qi Realplayer //查询RealPlay的安装信息

# rpm -ql http //查询指定的包http文件列表（已安装）

# rpm -ql package-name //查询软件包的文件列表

# rpm -qp package-name //查询未安装的软件包信息

# rpm -qpi http //查询指定的包http详细信息（未安装）

# rpm -qpl http //查询指定的包http文件列表（未安装）

# rpm -U mplay.rpm //升级mplay软件包

# rpm -Uvh package-name //升级软件包并显示过程

# rpm -V package-name //验证软件包的大小，类型等等

# which mount //获得mount所属包可执行文件路径

# for i in ‘rpm -qa |grep -i php’

>do rpm -e $i -nodeps

>done //删除系统中与软件php相关的所有已安装的软件包

# alias td //显示别名信息

# alias td=tcpdump //tcpdump命令设置别名td

# runlevel //显示当前系统运行等级

# runlevel 2 //设置执行级别

# set //显示当前的环境变量设置

# set SHELL “/bin/sh” //设置环境变更SHELL的值为/bin/sh

# setserial -a /dev/ttyS1 //显示串口/dev/ttyS1的详细配置信息

# setserial -v /dev/ttyS1 //显示串口/dev/ttyS1的配置信息

# setup //用来设置认证、防火墙、鼠标、网络配置、系统服务等公用程序

# fdisk -l //显示分区信息

# startx & init 5 //进入图形界面

# swapoff /dev/sda3 //卸载交换区/dev/sda3

# swapon /dev/sda3 //加载交换区/dev/sda3

# sync //将内存信息同步写入磁盘（在一些“危险”操作前使用）

# system-config- //连续按两下tab键，调出配制命令列表，以备选用

# tmpwatch -t 100 /tmp/ //查看100小时内未被使用的文件（不删除文件，仅进行测试）

# tmpwatch 100 /tmp/ //删除/tmp目录下100小时内未被使用的文件

# tmpwathc -s 100 /tmp/ //在删除文件之前，使用fuser命令检测该文件是否正被使用

# ulimit -a //显示系统资源配置

# ulimit -n 1000 //设置同一时间可打开的文件数目

# ulimit -u 500 //设置单用户创建进程上限为500

# ulimit -v 1024 //设置最大虚拟内存数

# unalias td //删除别名

# unset pwd //删除环境变量

# up2date //升级Red Hat Linux系统

# up2date -l //列出所有可用的升级包

释放linux系统的内存（在RHEL4.0及以上版本测试通过。注：在应用程序未退出时不能使用该命令）:

# free -m

# sync

# echo 3 > /proc/sys/vm/drop\_caches

# cat /proc/sys/vm/drop\_caches

# free -m

# ./configure;make;make install //编译安装过程（cd到安装目录执行）

# sh ./file\_name //安装.run后缀文件

# system-config-packages //启动图形界面包管理程序

# nvidia-settings //设置nvidia显卡

为内核打补丁

# bunzip2 patch-2.6.0-rmk2.bz2 //将包解压成patch-2.6.0-rmk2

# mv patch-2.6.0-rmk2 ./linux-2.6.0 //将补丁移到大内核目录

# cd linux-2.6.0 //进入内核目录

# patch -p1 < patch-2.6.0-rmk2 //“<”前后各有一个空格，-p1中的1是数字1。若补丁文件非bz2格式，如gz，则可用以下命令为内核打补丁：

# zcat ./patch-2.6.0-rmk2.gz | patch -p1

# 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

# iconv -f big5 -t utf8 vi.big5 -o vi.utf8 //把用big5编码的文件vi.big5转成utf8编码文件vi.utf8

# iconv -f utf8 -t big5 vi.utf8 | \

> iconv -f big5 -t gb2312 | iconv -f gb2312 -t utf8 -o vi.gb.utf8 //把用繁体编码的utf8文件vi.utf8转成简体的utf8文件vi.gb.utf8

4、自动作业处理

# /sbin/service anacron restart //重启anacron服务

# /sbin/service anacron start //启动anacron服务

# /sbin/service anacron stop //停止anacron服务

# /usr/sbin/atd //启动atd守护进程

# at -c 6 //显示已经设置的任务6的内容

# at -l //查询已设置的任务

# atd -d //以输出调试信息的方式运行队列中的任务

# atq //查询当前已设置的任务

# atrm 3 //删除当前任务队列中的第3个任务

# atrun //执行已排队的任务

# cat /var/spool/cron/root //查看任务内容

# chkconfig --level 35 crond off //关闭crond服务（在3，5级别）

# chkconfig --level 35 crond on //启动crond服务（在3，5级别）

# chkconfig --level级别列表 服务名称 [on|off|reset]

# chkconfig --level 24 syslog off //设置syslog 2,4级别的启动状为off

# chkconfig --list 服务名称

# chkconfig --list //显示已运行所有服务的启动状态

# chkconfig --list syslog //显示指定syslog服务的启动状态

# chkconfig服务名 [on,off,reset]

# chkconfig rsync on //设置rsync服务状态为启动

# chkconfig --list crond //查询crond在各运行级别的启动状态

# chkconfig --list|grep iptables //查看防火墙状态

# crontab -e //编辑cron任务（编完后，系统默认保存位置是/tmp）

# crontab -l //查看当前的cron任务列表

# crontab -r //删除cron任务

# rpm -qa anacron //查看系统中是否已安装anacron服务

# rpm -qa crontabs //检查系统中是否已安装crond服务

# service crond restart //重启cron服务

# service crond start //启动crond服务

# service crond status //查看cron服务启动状态

# service crond stop //停止crond服务

# pgrep crond //判断crond 是否在运行

# pkill crond //杀掉正在运行中的crond

# pgrep crond //查看是否被杀掉了

# /usr/sbin/crond //运行crond

# /etc/init.d/crond stop //停止cron服务

# /etc/init.d/crond restart //重启cron服务

5、系统日志

# /sbin/syslogd //启动syslog日志守护进程

# /sbin/syslogd -r //守护进程可以接收来自网络的syslog信息

# killall -HUP syslogd //重新修改配置文件后，重启syslog进程使其生效

# echo > /user/local/apache/logs/error\_log //清空Apache服务日志

二、用户和用户组管理

# ac //查看系统总的连接时间

# ac -d //按天对连接进行汇总

# ac -p //列出所有用户的连接时间

# cat /etc/shadow //查看/etc/shadow文件

# chage //改变用户变更密码的期限，要求用户必须在几天之内变更密码

# chage -l ow2003 //查看ow2003用户密码的有效期

# chfn //用来改变一个用户的完整用户名和其他信息

# chpasswd //一次性更新一组现有用户的密码

# chroot /tmp/empty /ls –Rl / //以/tmp/empty为根运行ls命令

# chsh //改变用户的缺省登陆shell

# cut -d: -f 1 /etc/group // 查看系统所有组

# cut -d: -f 1 /etc/passwd // 查看系统所有用户

# dpasswd //删除或者更新用户登录shell的拨号密码

# expiry //检查并强制执行密码失效策略

# faillog //检查登录失败日志/var/log/faillog，设置允许登录失败的次数或重置失败次数

# finger -l //列出当前登录用户的相关信息

# finger ow2003 //查看ow2003用户描述信息

# gpasswd //用来管理/etc/group文件

# grep ow2003 /etc/shadow //查看ow2003是否禁用（用户名后紧跟!号）

# grep ow2003 /etc/shadow //查看ow2003是否启用（用户名后无!号）

# groupadd //建立新的用户组

# groupadd benet //添加benet组

# groupadd -g 502 student //添加用户组student，GID为502

# groupadd -g benet st03 //添加st03用户并指定属于benet组

# groupdel //删除用户组

# groupdel benet //删除benet组

# groupmod //修改用户组

# groupmod -n new\_group\_name old\_group\_name //重命名一个用户组

# groupmod -g 503 teacher //修改teacher用户的组标识为503

# groupmod -g 550 -n dirctor teacher //将teacher用户组标识号改为550，用户组名改为director

# groups //显示当前用户所属的用户组

# grpck //校验用户组文件的完整性，/etc/group 和 /etc/gshadow

# grpconv //根据/etc/group文件建立/etc/gshadow文件

# grpunconv //根据/etc/group和/etc/gshadow文件建立新的/etc/group文件，并删除/etc/gshadow文件

# id //显示当前用户的UID、GID和用户所属的组列表

# id 用户名 // 查看指定用户信息

# id -g //显示当前用户的用户组GID

# id -G //显示所有的用户组GID

# id -G -n //显示所有的群组名称

# id ow2003 //显示指定用户ow2003的用户信息

# id root //显示root用户的用户信息

# last //显示最近用户的登录信息（last命令查找/var/log/wtmp文件）

# last -n 10 //仅显示10行记录

# last -x //查询最近用户登录情况，同时显示系统关机及系统运行等级变化

# lastb //显示最近登录失败用户信息（lastb命令查找/var/log/btmp文件）

# lastlog //格式化并输出最后一个登录日志的内容，或者是某个用户的最近一次登陆内容

# ln /usr/sbin/groupadd /usr/sbin/addgroup //创建名为addgroup的链接命令，链接到groupadd命令

# login //系统用它来允许用户登陆

# logname //显示登录用户的用户名

# logoutd //用/etc/porttime中的设置强制限制登录时间和端口

# mkpasswd //读取参数设定的某格式文件并转化至相应的数据库文件格式

# newgrp //不指定转换的用户组时，系统默认转换为登录时的用户组

# newgrp [-] [groupname] //选项“-”用于重新加载用户工作环境。如果不带“-”选项，则在切换用户组时，用户的工作环境（包括当前工作目录等）不会改变

# newgrp bin //改变当前用户的主用户组为bin

# newusers //批量加入新的用户

# passwd //修改当前用户的密码

# passwd -d ow2003 //快速删除ow2003用户的密码

# passwd -l ow2003 //锁定ow2003用户使其不能登录

# passwd ow2003 //根用户修改普通用户ow2003的密码

# passwd -u ow2003 //解锁ow2003用户使其可以登录

# pwck //校验密码文件的完整性，/etc/passwd 和 /etc/shadow

# pwconv //根据/etc/passwd文件建立/etc/shadow文件（解决两者不匹配时的问题，如用户ow203在passwd中有记录而在shadow中无记录的现象；解决无法进入图形界面的用户和组管理器现象）

# pwunconv //根据/etc/shadow和/etc/passwd文件建立新的/etc/passwd文件，并删除/etc/shadow文件

# rwho -a //显示局域网内所有用户

# sg //设置用户的GID到指定组,或者以指定组的身分执行一个命令

# su - //切换为根用户

# touch /etc/nologin //在系统维护期间禁止用户登录（但不限于SSH登录）

# useradd //添加新的用户或者改变新用户的默认信息

# useradd -D //显示当前的默认值

# useradd -D -s /bin/csh //修改该命令所用shell的默认值为/bin/csh

# useradd wxr2 -u 502 -d /home/wxr -s /bin/bash -e 10/30/11 -g 100 //添加一新用户wxr2，UID为502，用户组ID为100，用户目录为/home/wxr，用户的默认shell为/bin/bash，账号的失效期为2011年10月30日

# userdel /删除用户

# userdel ow2003 //删除 ow2003用户(仅删帐号)

# userdel -r ow2003 //删除 ow2003用户（连同主目录删除）

# usermod //修改用户信息

# usermod -d /home/wxr2 -s /bin/ksh -g users wxr //用户登录目录改为/home/wxr2，用户登录shell改为ksh，用户所在组改为users和wxr

# usermod -g benet st03 //设置st03用户新的组名为benet

# usermod -G szxs st03 //添加st03用户到其它组szxs(多组)

# usermod -L ow2003 //禁用ow2003用户

# usermod -U ow2003 //启用ow2003用户

# usermod -u0 -o ow2003 //提升ow2003用户管理员权限

# users //显示所有登录的用户

# vigr //能用来编辑 /etc/group 或 /etc/gshadow文件

# vigr /etc/group //编辑group时自动对该文件加锁，编辑结束时自动解锁

# vipw //编辑passwd时将自动对该文件加锁，编辑结束时自动解锁

# vipw //能用来编辑 /etc/passwd 或 /etc/shadow文件

# w -f //显示登录用户信息，但不显示登录位置（即from字段）

# w -h //在显示登录用户信息时，不显示标题栏

# w root //只查询指定用户root的信息

# w -s //以精简模式显示登录用户信息

# w // 查看当前登陆用户及所进行的操作

# who -H //显示登录系统的用户信息时，显示标题栏

# who -H -m //只显示使用当前标准输入设备的用户

# who -H -p //显示由init进程创建的活动进程

# who -H -u //显示空闲时间段

# who -q //显示所有登录的用户名以及登录用户的数目

# who -T -H //显示用户状态信息

# who // 显示当前登陆系统的用户

# whoami //显示当前登录用户名称

# skill -9 pts/2 //杀掉从pts/2虚拟终端登录的用户

# write wang

EOF //向登录用户wang发送即将关机的信息

System will shutdown soon!

三、磁盘管理

# blkid //列出当前系统中所有已挂载文件系统的类型

# blkid -s LABEL //仅显示每个（指定）设备相匹配的标签

# blkid /dev/sda1 //查看/dev/sda1设备所采用的文件系统类型

# e2label device [newlabel] //改变本地设备文件系统的label名称

# dd if=/dev/zero of=f1 bs=10MB count=2 //用虚拟文件f1检查配额情况

# dd if=/dev/sda of=/home/backup/boot\_MBR bs=446 count=1 //备份MBR。解决Linux和Windows双系统的引导问题（IED硬盘为hda）

# dd if=/home/backup/boot\_MBR of=/dev/sda bs=446 count=1 //恢复MBR

# dd if=/dev/zero of=/home/loopdev bs=1M count=512 //创建一个空的文件在/home/loopdev

# mkfs -t ext3 /home/loopdev //对/home/loopdev进行格式化

# mount -o loop /home/loopdev /media/cdrom //挂载/home/loopdev这个loop设备到/media/cdrom目录

# df //用于报告文件系统的总容量，使用量，剩余容量

# df -ah //查询所有分区使用量，以可被识别的方式展现

# df -h //以用户容易识别的方式显示磁盘空间使用情况

# df -i //以inode模式显示磁盘的使用情况

# df -T //查看磁盘格式挂载后的信息

# df -t ext3 //只显示ext3类型磁盘的信息

# df -x ext3 //不显示指定磁盘类型的磁盘使用信息

# dmesg | grep IDE // 查看启动时IDE设备检测状况

# dmesg | grep DMA //查看硬盘是否运行在DMA模式

# du -a DirPath //显示所有文件及其子目录占用的数据块数

# du -a | sort -n > /home/disk\_used & //将显示结果储存到/home/disk\_used文件中

# du -b /home //查看目前/HOME目录的容量(k)及子目录的容量(k)

# du -sh DirPath //以用户易读格式显示目录大小

# du -sk DirPath //查看目录的容量(k)

# du -sm DirPath //查看目录的容量(m)

# fdisk /dev/sda //对磁盘sda进行分区

# fdisk -l //查看磁盘及分区信息

# sfdisk -l //查看磁盘及分区信息

# sfdisk -s //查看系统中所有磁盘的大小

# fdisk -s /dev/sdb //显示/dev/sdb设备中所有分区大小总和

# fdisk -s /dev/sdb2 //显示/dev/sdb2分区大小

# fdisk -v //显示fdisk版本号

# hdparm -i /dev/hda // 查看磁盘参数(仅适用于IDE设备)

# mkfs.ext2 /dev/sda1 //格式化sda1为ext2格式

# mkfs.ext3 /dev/sda2 //格式化sda2为ext3格式

# mount //显示当前已挂载的文件系统

# mount -a //搜索/etc/fstab文件中满足条件的文件系统，进行挂载操作。只有root用户可以操作。

# mount -l //列出所有已经挂载的文件系统列表

# mount /dev/sda1 /bbb //挂载sda1到/bbb目录下

# mount | column -t // 查看挂接的分区状态

# mount -t 文件系统类型 设备路经 访问路经

# mount -t iso9600 /dev/cdrom /mnt/cdrom //在/mnt/cdrom目录下挂载光驱（iso9600为光盘的标准文件系统类型）

# mount /dev/cdrom /aaa/ //挂载设备CDROM到/aaa/目录下

# mount -t ntfs-3g /dev/sdb\* /mnt/aaa //不编译内核，mount ntfs分区

# cp /dev/cdrom 123.iso //将cdrom内容制作成123.iso镜像

# mount -t iso9660 -o loop xxx.iso /path //挂载ISO文件

# mount -o loop 123.iso /bbb/ //将123.iso镜像文件挂载到/bbb/下

# mount -t isoDVD /dev/cdrom /mnt/cdrom //挂载光驱

# mount -t iso9660 /dev/cdrom /mnt/cdrom //挂载光盘

# mount -t vfat -o iocharset=utf8,umask=000 /dev/hda2 /mnt/hda2 //挂载fat32分区

# mount -t ntfs -o nls=utf8,umask=000 /dev/hda3 /mnt/hda3 //挂载ntfs分区

# mount -t vfat /dev/hda6 /mnt/cdrom //挂第一个ide的第六个逻辑分区

# mount -no remount, ro / //以只读模式重新挂载/分区

# mount -n -o remount, rw / //以读写模式重新挂载/分区（单用户维护模式时有用）

# umount /aaa/ //卸载目录/aaa

# umount /dev/cdrom //卸载设备 cdrom

# showmount -e 10.52.19.4 //显示对方服务器nfs共享目录

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

参考：

mount远程目录并让本地非root用户可读可写

http://blog.chinaunix.net/uid-20680966-id-4153081.html

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

# fuser -km /mnt //当设备繁忙时强制卸载

# eject -n //查看系统默认的弹出设备

# eject //弹出默认的设备

# quotastats //显示系统磁盘空间限制的当前状态

# repquota -a //显示文件系统的磁盘使用情况

# edquota -u ccnp //对组ccnp设置配额

# edquota -u st26 //对用户设置配额

# quota -g ccnp //查看组ccnp配额情况

# quota -u st26 //查看用户st26配额情况

# quotacheck -cmug / //创建配额文件

# quotacheck -mfvug / //对文件系统的配额进行一致性检测

# quotaon -a //启用所有的磁盘配额限制

# quotaon /home //启用/home的磁盘空间配额

# quotaon -aguv //不指定分区，使用自动搜索启用磁盘配额

# quotaoff -a //关闭所有配额限制

# quotaoff /home //关闭/home的磁盘配额

# quotaoff -aguv //不指定分区，使用自动搜索关闭磁盘配额

# quotaoff -aguv //不指定分区，使用自动搜索关闭磁盘配额

# reboot //重启系统

# swapon /dev/sdb2 //开启交换分区/dev/sdb2

# swapoff /dev/sdb2 //关闭交换分区/dev/sdb2

# swapon -s // 查看所有交换分区

# grep SwapTotal /proc/meminfo //查看swap空间大小

# swapoff -a //关闭所有交换区

# swapon -a //开启所有交换区（与上面成对使用后可使swap还原到初始状态）

# fdisk /dev/sdb //由设备分区(sdb1)来创建SWAP分区,注意分区类型82

# mkswap /dev/sdb1 //格式化分区并创建文件系统

# vi /etc/fstab //添加：/dev/sdb1 swap swap defaults 0 0

# dd if=/dev/zero of=/data1/image/swap bs=1024 count=2048000 //添加交换文件并设置其大小为2G

# mkswap /data1/image/swap //创建交换空间

# swapon /data1/image/swap //启动新增加的2G的交换空间

# vi /etc/fstab //在文件最后加入：/data1/image/swap swap swap defaults 0 0 使得新加的2G交换空间在系统重新启动后自动生效

# vi /etc/fstab //自动挂载磁盘分区,添加: /dev/sdb5 dd ext3 defauls 0 0

# vi /etc/fstab //修改fstab文件中分区装载设置在defaults后加入usrquota,grpquota 重新挂载文件系统（对根分区设置配额重启）

# sync --version //显示sync命令版本信息

# sync --help //显示sync命令的帮助信息

# sync //将内存数据写入磁盘

# rsync -av -e ssh /home/share/\*.doc root@teacher.example.com:/usr/local/share //将一台名为student.example.com主机上的/home/share目录中的所有doc文件备份到主机teacher.example.com的/usr/local/share目录下

# pvdisplay //查看组成LVM卷的物理卷（PV）

# pvcreate /dev/sdd //在二级SCSI控制器的从属驱动器上创建一个新物理卷

# vgcreate myvolume /dev/sdd1 /dev/sdc2 //创建一个名为myvolume的VG

# lvextend -L 2000M /dev/myvolume/mylogical //对dev/myvolume/mylogical的容量进行添加

# smartctl -H /dev/sdb //对sdb盘做一下健康自检（PASSED为合格）

# tune2fs -l /dev/sdc1 |grep "Block size" //查看ext3文件系统的block size（bytes）

Block size: 4096

四、文件和目录管理

# cat /home/333 //查看/home下文件333的内容（不停顿）

# cd //进入当前帐户所在目录

# cd 配合通配符\*会更方便些

# cd /home/ccc //进入/home/ccc目录

# cd - //可以回到你之前所在的那个目录

# cd ../.. //返回上两级目录

cp [-adfilprsu] 源文件（source） 目标文件（destination）

cp [options] source1 source2 source3 … directory

参数：

-a：相当于-pdr的意思，见后文（常用）；

-d：若源文件为连接文件属性，则复制连接文件属性而非文件本身；

-f：若目标文件已存在且无法开启，则删除后再尝试一次；

-i：若目标文件已存在时，在覆盖时会先询问操作的进行（常用）；

-l：进行硬连接的连接文件创建，而非复制文件本身；

-p：连同文件的属性一起复制过去，而非使用默认属性（备份常用）；

-r：递归持续复制，用于目录的复制行为（常用）；

-s: 复制成为符号链接文件，即“快捷方式”文件；

-u：若destination比source旧才更新destination。最后需注意的是，如果源文件有两个以上，则最后一个目的文件一定要是“目录”才行。

# cp 333 /home/ccc //复制文件333到/home/ccc下

# cp 333 444 //复制文件333为444(同目录下)

# cp -fr test/ /data1/wxr/ //复制当前路径下的目录test到指定路径/data1/wxr/

# cp .bashrc bashrc //将 .bashrc 拷贝成 bashrc 这个文件

# cp -a /etc /tmp //将目录/etc以完整权限复制到/tmp目录

# dos2unix a.txt b.txt //将dos格式的a.txt转换成unix格式的b.txt

# dos2unix a.txt //直接将a.txt中的^M符号删除（也可用vi命令（:%s/^m//g）删除）

# scp -rp /path/filename username@remoteIP:/path //从A机复制到B机

# scp -rp username@remoteIP:/path/filename /path //从B机复制到A机

# cp /dev/cdrom mycdrom.iso //将光盘中的内容制成镜像文件，文件名为mycdrom.iso

# cp /dev/cdrom mycdrom.iso & //将光盘中的内容制成镜像文件，文件名为mycdrom.iso（后台运行，加&)

# dd if=/dev/cdrom of=file.iso //把 CD/DVD 作成 ISO 文件

# mkisofs -r -o mybackup.iso /home/backup //将/home/backup目录下的所有文件制成镜像文件，文件名为mybackup.iso

# mount -o loop mybackup.iso /mnt //将镜像文件mybackup.iso挂接到/mnt目录下

# file 222 //显示文件222的文件类型

# file \* //显示当前目录下所有文件的文件类型

# find /etc -name "host\*" -print //在/etc目录中查找文件名以host开头的文件

# find //显示当前目录下所有内容（无参数）

# find $HOME -print //查找当前用户主目录下的所有文件

# find . -group root -exec ls -l { } \; //查找系统中所有属于root组的文件

# find . -maxdepth 1 -size +1000000c //查找当前目录下的大于1M的文件

# find . -name "[A-Z]\*" -print //在当前目录及子目录中查找文件名以大写字母开头的文件

# find . -perm -7 -print | xargs chmod o-w //在当前目录下查找所有用户具有读、写和执行权限的文件，并取消others用户的写权限

# find . -perm 755 -print //当前目录下查找文件权限位为755的文件，即文件属主可以读、写、执行，其他用户可以读、执行的文件

# find . -type d | sort //查找当前文件系统中的所有目录并排序

# find . -type d -exec tree {} \; | more //显示当前目录的目录结构并以树形结构分页显示

# find . -type f -exec ls -l {} \; | more //查当前目录下的所有普通文件，并用ls -l命令将它们分页列出

# find . -type f -perm 644 -exec ls -l { } \; //让当前目录中文件属主具有读、写权限，且文件所属组的用户和其他用户具有读权限的文件

# find . -type f -print | xargs file //查找当前目录下的文件属于哪类文件

# find / -name "file\*" -print | xargs echo "" > /temp/core.log //在整个系统中查找内存信息转储文件(core dump) ，然后把结果保存到/tmp/core.log 文件中

# find / -type f -size 0 -exec ls -l { } \; //查找系统中所有文件长度为0的普通文件，并列出它们的完整路径

# find /apps -path "/apps/bin" -prune -o -print //在/apps目录下（排除/apps/bin目录）查找文件

# find /home -name “\*.sgy” –print //查找/home下所有sgy文件

# find /usr/sam -path "/usr/sam/dir1" -prune -o -print //在/usr/sam目录下查找不在dir1子目录之内的所有文件

# find /var/log -type f -mtime +7 -ok rm { } \; //查找/var/log目录中更改时间在7日以前的普通文件，并在删除之前询问它们

# find /data/pa/xyz/ -name "\*.cgm" -atime +8 -exec rm -f {} \; //查找并删除8昼夜（不含8昼夜本身）前被访问过的cgm文件

# find ~ -print //查找当前用户主目录下的所有文件

# find DirPath -type f | wc -l //查看某目录下有多少个文件

# find -name 11\* //查找文件名包含11的所有文件

# find -user luo //查找属于用户luo的所有文件

# mv 333 /home/ccc //移动文件333到/home/ccc下

# mv 333 444 //移动文件333为444（同目录下改名）

# mv 目录名 目录名 //目标目录已存在，源目录移到目标目录，目标目录不存在，改名

# rm 333 //删除文件333

# touch 333 //创建333空文件

chmod ---=0 -w-=2 r--=4 rw-=6 --x=1 -wx=3 r-x=5 rwx=7 s,S,t,T=特殊权限

# chmod [ugoa][+-=][rwx] file

u=属主 g=属组 o=其它用户 a=所有用户 rwx=读，写，执行

+-=：加权限，减权限，赋权限

# chown -R ow2003 wxr //更改目录wxr属于用户ow2003

# chgrp -R ow2003 wxr //更改目录wxr属于组ow2003

# chgrp root \* //把当前目录中所有文件的组属性设置成root

# chmod 765 111 //为111文件的属主设为完全权限，属组设成读写权，其它用户具有读和执行权限

# chmod a-r 111 //将111文件所有用户撤消读取权限

# chmod g+x 111 //将111文件属组增加执行权限

# chmod o+r 111 //将111文件其它用户增加读取权限

# chmod u+w 111 //将111文件属主增加可写权限

# chmod g-r,o-r 111 //去掉文件111的同组和其他用户的读权限

# chmod o=rwx 111 //重设文件111的其他用户权限为读、写执行

# chown :benet 222 //更改文件222属于组benet

# chown st03 222 //更改文件222属于用户st03

# chown st03:benet 222 //更改文件222属于用户st03组benet

# 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

# cut -d ":" -f 1,7 /etc/passwd //只显示第一列(user ID)和第 7 列(user shell) 内容

# file //查看文件类型

# grep 字符 文件名 //根据字符匹配来查看文件部分内容

# head 10 /home/333 //查看/home下文件333的头部10行内容

# less用法：

空格键：向下翻动一页；

[PageDown]：向下翻动一页；

[PageUp]：向上翻动一页；

/字符串：向下查询“字符串”的功能；

?字符串：向上查询“字符串”的功能；

n：重复前一个查询（与/或?有关）；

N：反向重复前一个查询（与/或?有关）；

q：离开less这个程序。

# less /home/333 //查看/home下文件333的内容（分屏,可以往前翻页）

# ln -s source\_dir destination\_dir //将目标目录链接到源目录。用绝对路径。在与源目录的同级目录处创建。文件链接同理。用ls -l查看，会看到destination\_dir -> source\_dir 如：

# ln -s /home /disk1

# ls -l

lrwxrwxrwx 1 root root 4 Apr 7 2010 disk1 -> home/

# unlink destination\_dir //取消目标目录与源目录的链接

# ll -h //以用户容易识别的方式的长格式显示当前目录的内容

# lsattr -a //显示所有文件和内容，包括现行目录“.”与上层目录“..”

# lsattr -R //递归显示目录下的所有文件和子目录

# lsattr test //显示文件test有属性

# ls //以短格式显示当前目录的内容

# ls \* //显示当前目录下的所有文件

# ls -l //以长格式显示当前目录的内容

# ls -l /home/bbb/ //显示指定目录/home/bbb的内容

# ls -l 111 //只显示当前目录下文件111的信息

# ls c\* //列出当前目录下以字母“c”开头的文件

# ls ??n\* //列出当前目录下所有第3个字母是“n”的文件

# ls [b,c]\* //显示所有以字母b或c开头的文件名

# ls -F | grep /$ //只列子目录（短格式）

# ls -l | grep "^d" //只列子目录（长格式）

# ls -l |grep "^-"|wc -l //计算当前目录下的文件数

# ls -l |grep "^d"|wc -l //计算当前目录下的目录数

# ls -lSr |more //以尺寸大小排列文件和目录

# ls --color=never //不要根据文件特性给予颜色

# ls --color=always //显示颜色

# ls --color=auto //让系统自行依据设置来判断是否给予颜色

# ls --full-time //以完整时间模式（包含年、月、日、时、分）输出

# lsof |grep 文件系统的挂接点 //查看是否有任何程序正在使用挂接点的文件系统

# man -t ls | ps2pdf - ls.pdf //创建ls命令的pdf格式文件

# man -t manpage | ps2pdf - filename.pdf //创建man的pdf格式文件

# mkdir ddd //在当前目录下创建ddd目录

# mkdir -p /a/b/c //创建/a/b/c目录（多层目录）

# mkdir -m a=r wxr //创建一个目录wxr且指定该目录的属性为对所有用户可读

# mkdir -m 700 /data1/wxr/test //创建新目录/data1/wxr/test，且指定权限为700

# more /home/333 //查看/home下文件333的内容（仅分屏）

# nl filename //显示文件内容时，同时输出行号

# od filename //以二进制方式读取文件内容

# pwd //显示当前所在目录

# rmdir ddd //删除ddd目录（空）

# rm -r ddd //删除ddd目录（非空，先询问）

# rm -rf ddd //直接删除ddd目录（非空，不询问）

# rm -f /data/\*.cgm //超级用户下删除/data目录下所有.cgm文件（不再询问）

# rm -i -r ddd //交互模式删除ddd目录（删除前先询问）

# \rm -r ddd/\* //删除ddd目录下文件（不询问）

# sed -n '5,10p' /etc/passwd //只查看passwd的第5行到第10行内容

# setfacl -m u:ow2003:rwx /home/hoework //对用户ow2003设置对/homework目录的读、写和执行权限

# setfacl -m u:ow2003:rw /home/hoework //对用户ow2003设置对/homework目录的读、写权限

# setfacl -m u:dba:r /home/hoework //对用户组dba设置对/homework目录的读权限

# setfacl l -x u:dba /home/hoework //删除用户组dba的权限

# setfacl -d g:rw /home/ftp //对/home/ftp目录的用户组设定默认的权限为读、写（若组中用户重新指定了权限，则默认的组权限将被覆盖，即具体指定的ACL权限优先于默认ACL权限）

# sort /home/ow2003/dk //显示文档dk内容并排序

# split -b 900m file file\_part //把文件file以900m增量来分割

# cat file\_part\* >file //将分割后的文件重新组装

# tac /home/ow2003/dk //以逆序显示文档dk内容（注：与cat相反）

# tail 10 /home/333 //查看/home下文件333的尾部10行内

# touch aa //使用touch命令创建aa文件（aa文件不存在）

# touch aa //使用touch命令更新文件aa的时间为当前时间（aa文件存在）

# touch aa -t 200808081500.00 //更新文件aa的存取时间为指定时间2008年8月8日15点

# tree //显示文件和目录由根目录开始的树形结构

# wc /home/ow2003/dk //显示文档dk的行数、词数和字数

# which ssh //通过命令which，查找文件ssh（其查找范围由环境变量$PATH设置）

# yes |rm \*.cgm //自动回答rm提出的问题，无需用户再连续输入yes

五、备份与压缩

# dd if=/dev/sda | gzip -c | ssh user@ip 'dd of=/mnt/backups/sda.dd' //通过ssh复制整个硬盘到远程目录(带压缩)

# tar -c 创建包；-x 释放包；-v 显示命令过程；-z 代表压缩包；-t 列出包文件（打包时应尽量采用相对路径，而不用绝对路径）

# tar -cvf wxr.tar wxr //把当前wxr目录打包

# tar -cvf wxr.tar wxr --exclude '\*.3dv' --exclude '\*.3dh' --exclude '\*.cgm' --exclude '\*.inp' //把当前wxr目录打包，并排除.3dv、.3dh、.cgm、.inp文件

# tar -zcvf apache.tar.gz --exclude=c --exclude=d apache //若/opt/apache/目录下有 a，b，c，d目录，只打包a 、b目录，不打包c 、d目录

# tar -zcvf wxr.tar.gz wxr //把当前wxr目录打包压缩

# tar -hcvf wxr.tar wxr //对目录wxr中链接文件也一并打包

# tar -rvf wxr.tar file1 //向wxr.tar包添加文件file1

# tar -tf wxr.tar //查看包wxr.tar内容

# tar -tzf wxr.tar.gz //查看压缩包wxr.tar.gz的内容

# tar -jtv -f wxr.tar.bz2 //查看压缩包wxr.tar.bz2的内容

# tar -jcv -f wxr.tar.bz2 bar/ //将目录bar打包并压缩为wxr.tar.bz2

# tar -jxv -f wxr.tar.bz2 //对压缩包wxr.tar.bz2解压恢复

# tar -jxv -f wxr.tar.bz2 -C bar/ //将压缩包解压到目录bar/

# tar -xvf wxr.tar -C /home //恢复包wxr.tar到指定目录/home

# tar -xvf wxr.tar //解压缩包wxr.tar

# tar -xzf wxr.tar.gz -C /home //恢复压缩包wxr.tar.gz到指定目录/home(不带-C则恢复到原目录)

# tar -xzf wxr.tar.gz //解压缩包wxr.tar.gz

# tar -xzf wxr.tar.gz blah.txt //从压缩包wxr.tar.bz2解压出文件blah.txt

# tar -zcvf wxr.tar.gz /project //把目录project打包并压缩

# tar -zxvf wxr.tar.gz //压缩包的文件解压恢复

# tar cvf - dir1 dir2 | (cd /dir3; tar xvf - ) //将当前目录dir1 dir2边打包边释放到/dir3目录下

# for ARK in ./\*.tar.gz; do tar xvf $ARK; done //解压当前目录下的所有 .tar.gz 结尾的文件（把当前目录下所有的 .tar.gz 文件的文件名逐一赋给变量 ARK，让 tar 来引用 ARK 变量）

# dump -0f home.dump /home //将/home目录备份到文件home.dump

# dump -0j -f /root/etc.dump.bz2 /etc //备份/etc目录并含压缩功能

# restore -rvf home.dump //恢复home目录

# restore -tf home.dump //查看备份文件home.dump的文件列表

# dump -w //查看有无任何文件系统被dump过的数据

# zip -r data.zip data //将data文件夹压缩成了data.zip格式

# unzip data.zip //将data.zip文件解压到当前文件夹

# zip -r yasuo.zip abc.txt dir1 //把一个文件abc.txt和一个目录dir1压缩成为yasuo.zip

# unzip yasuo.zip //解压缩yasuo.zip文件

# unzip abc\?.zip //将当前目录下的abc1.zip，abc2.zip和abc3.zip一起解压缩（?表示一个字符，如果用\*表示任意多个字符）

# unzip -v large.zip //不解压，只想查看压缩文件large.zip的内容

# unzip -t large.zip //检验压缩文件large.zip是否下载完全了

# unzip -j music.zip //用-v选项发现music.zip里有多级目录，且子目录中均为mp3文件，若想把这些文件都下载到第一级目录，而非一层一层建目录

六、网络管理与相关应用

# arp -a | awk '{print $4}' //得到网卡的 MAC地址

# dd if=/dev/zero bs=4096 count=1048576 | ssh user@ip 'cat > /dev/null' //不浪费磁盘的情况下测试网络速度（通过 ssh 发送 4 GB 数据到远程主机，但不会占用任何磁盘空间）

# dmesg | grep eth //显示每个网卡的硬件配置信息

# echo 1 >/proc/sys/net/ipv4/icmp\_echo\_ignore\_all //关闭ping

# echo 0 >/proc/sys/net/ipv4/icmp\_echo\_ignore\_all //重启ping

# ethtool eth0 //显示或改变网卡设置（参考：man ethtool）

# ifconfig //查看当前有效网络接口信息

# ifconfig -a //查看当前所有网络接口信息

# ifconfig eth0 //查看指定网络接口eth0信息

# ifconfig eth0 192.168.1.11 //设置eth0接口IP地址（标准掩码）（相当于动态修改IP）

# ifconfig eth0 192.168.1.11 netmask 255.255.255.128 //设置eth0接口IP地址及子网掩码

# ifconfig eth0 hw ether 000C1254D321 //更改网卡物理地址

# ifconfig lo //查看lo状态

# ifdown eth0 //禁用eth0网络接口

# ifdown lo //关闭本地回环网络接口

# ifup eth0 //启用eth0网络接口

# ifup lo //启动本地回环网络接口

# ip ad sh //显示IP地址等信息

# iptables -L // 查看防火墙设置

# iptraf //可交互式IP网络监控工具

# mii-tool -v //查看网卡有关信息

# netconfig //设置网络环境

# netstat -antp // 查看所有已经建立的连接

# netstat -apt //显示所有TCP应用程序所使用的端口号，并显示使用此端口号的进程

# netstat -lntp // 查看所有监听端口

# netstat -s // 查看网络统计信息

# netstat -tulnp //目前系统上已在监听的网络连接及其PID

# ping 10.52.19.187 //检测到10.52.19.187之间是否连通

# ping -c 5 10.52.19.187 //向10.52.19.187发送5个分组报文

# ping -s 6553 -c 5 10.52.19.187 //向10.52.19.187发送5个分组、每组大小为6553字节的报文（注：最大分组报文不能超过65507字节）

# ping -c 8 -i 3 -s 1024 -t 25 10.52.19.187 //向10.52.19.187以3秒间隔、包大小1024字节、IP生存期25、8个包发送报文

# route -n // 查看路由表

# tracepath 10.52.19.189 //显示连接到10.52.19.189的路由

# traceroute -n 10.52.19.190 //对到达10.52.19.190的路由进行跟踪，且使用IP地址表示每一跳主机

# ss //显示网络套接字信息，它允许显示类似netstat一样的信息

# wget 192.168.20.148 //从网站192.168.20.148下载资料

动态增加IP

# traceroute 211.148.192.136 //检测到目的IP经过的路由信息

# hostname //查看主机名称

# hostname luo //设置主机名称为 luo

# ping 192.168.1.10 //Ping主机192.168.1.10

# ifconfig eth0 add 10.52.19.147

# ifconfig eth0:0 broadcast 10.52.19.255

DNS域名

# nslookup 待解析域名（或IP）

# nslookup 211.148.192.136 //反向解析IP---域名

# nslookup www.163.com //解析指定的域名---IP

# route add default gw 192.168.1.1 //添加默认网关为192.168.1.1

# route del default gw 192.168.1.1 //删除默认网关为192.168.1.1

NFS服务

# exportfs -auv //停止输出所有共享目录

# exportfs -av //输出启用所有共享目录

# exportfs -rv //重新输出共享目录

# mount -t nfs 192.168.2.10:/luo /wei //在客户机上挂载共享目录到/wei

# service iptables stop //关闭防火墙

# service nfs start //启动nfs服务

# service portmap start //启动portmap服务

# showmount -a //显示NFS服务器的客户与被挂载目录

# showmount -d //显示NFS服务器被挂载的目录

# vi /etc/exports //修改共享配置文件（加共享目录）

showmount -e 192.168.2.10 6 //在客户上查看服务器共享目录

配置文件

vi /etc/hosts vi /etc/network //修改主机名称配置文件

vi /etc/rc.local //修改mac地址配置文件

vi /etc/resolv.conf //修改DNS配置文件

vi /etc/sysconfig/network-scripts/ifcfg-eth0 //修改网卡配置文件

vi /var/spool/cron/username //修改cron任务配置文件

vi /var/spool/mail //修改邮件配置文件

七、vi/vim编辑器

也可建立vi到vim的符号连接（即实际调用的是vim）

# mv /bin/vi /bin/vi.bak

# ln -s /usr/bin/vim /bin/vi

进入vi的命令：

vi 直接进入编辑并创建新文件

vi filename 编辑文件(不存在则创建文件)，并将光标置于第一行首

vi +n filename 打开文件，并将光标置于第n行首

vi + filename 打开文件，并将光标置于最后一行首

vi +/pattern filename 打开文件，并将光标置于第一个与pattern匹配的串处

vi -r filename 在上次正用vi编辑时发生系统崩溃，恢复文件

vi file1....filen 打开多个文件，依次进行编辑

vi -o aaa bbb 水平窗口打开aaa，bbb两个文件

vi -O aaa bbb 垂直窗口打开aaa，bbb两个文件(Ctrl+W两次可在多个文件间切换)

多文本编辑的按键：

:n 编辑下一个文件

:N 编辑上一个文件

:files 列出目前这个vi的打开的所有文件

多窗口情况下的按键功能：

:sp [filename] 打开一个新窗口，如果有加filename，表示在新窗口打开一个新文件，否则表示两个窗口为同一个文件内容（同步显示）

[ctrl]+w+j

[ctrl]+w+↓ 按键的按法是：先按下[ctrl]不放，再按下w后放开所有的按键，然后再按下j（或向下箭头键），则光标可移动到下方的窗口

[ctrl]+w+k

[ctrl]+w+↑ 同上，不过光标移动到上面的窗口

[ctrl]+w+q 其实就是:q结束离开。例如，想要结束下方的窗口，那么利用[ctrl]+w+↓移动到下方窗口后，按下:q即可离开，也可以按下[ctrl]+w+q

块选择的按键意义：

v 字符选择，将光标经过的地方反白选择

V 行选择，将光标经过的行反白选择

Ctrl + v 块选择，可以用长方形的方式选择数据

y 将反白的地方复制起来

d 将反白的地方删除

恢复命令：

u 恢复最后一个指令之前的结果

. 继续最后一个指令的执行结果

ctrl+r 撤销恢复

U 恢复光标该行之所有改变

移动光标命令：

h 光标左移一个字符

l 光标右移一个字符

space 光标右移一个字符

Backspace 光标左移一个字符

k或Ctrl+p 光标上移一行

j或Ctrl+n 光标下移一行

Enter 光标下移一行

w或W 光标右移一个字至字首

b或B 光标左移一个字至字首

e或E 光标右移一个字至字尾

) 光标移至句尾

( 光标移至句首

} 光标移至段落开头

{ 光标移至段落结尾

nG 光标移至第n行首

n+ 光标下移n行

n- 光标上移n行

n$ 光标移至第n行尾

H 光标移至屏幕顶行

M 光标移至屏幕中间行

L 光标移至屏幕最后行

0 （注意是数字零）光标移至当前行首

$ 光标移至当前行尾

翻屏命令：

Ctrl+u 向文件首翻半屏

Ctrl+d 向文件尾翻半屏

Ctrl+f 向文件尾翻一屏

Ctrl＋b 向文件首翻一屏

nz 将第n行滚至屏幕顶部，不指定n时将当前行滚至屏幕顶部

插入文本、复制与粘贴：

i 在光标前插入

I 在当前行首插入

a 在光标后插入

A 在当前行尾插入

o 在当前行之下新开一行

O 在当前行之上新开一行

r 替换当前字符

R 替换当前字符及其后的字符，直至按ESC键

s 从当前光标位置处开始，以输入的文本替代指定数目的字符

S 删除指定数目的行，并以所输入文本代替之

ncw或nCW 修改指定数目的字

nCC 修改指定数目的行

yy p 复制当前行，并粘贴到下一行

nyy n为数字。复制光标所在的向下n行，例如20yy，则是复制20行

y1G 复制光标所在行到第一行的所有数据

yG 复制光标所在行到最后一行的所有数据

y0 复制光标所在的那个字符到该行行首的所有数据

y$ 复制光标所在的那个字符到该行行尾的所有数据

shift + i 在该行之首插入字符

shift + a 在该行末尾输入字符

shift + r 改写自光标位置及后面的所有内容

shift +o 在光标之上新增一行输入字符

删除命令：

ndw或ndW 删除光标处开始及其后的n-1个字

d0 删至行首（为数字0）

d$ 删至行尾

shift + d 剪切至行尾

dd 删除一行

ndd 删除自光标开始的n行

s 删除光标所在之字符，并进入输入模式直到《ESC》

S 删除光标所在之该行资料，并进入输入模式直到《ESC》

x或X 删除一个字符，x删除光标后的，而X删除光标前的

nx n为数字，连续向后删除n个字符。如，要连续删除10个字符，则：“10x”

Ctrl+u 删除输入方式下所输入的文本

搜索及替换命令：

/pattern 从光标开始处向文件尾搜索pattern

?pattern 从光标开始处向文件首搜索pattern

n 在同一方向重复上一次搜索命令

N 在反方向上重复上一次搜索命令

：s/p1/p2/ 替换当前行第一个p1为p2

：s/p1/p2/g 将当前行中所有p1均用p2替代

：n1,n2s/p1/p2/g 将第n1至n2行中所有p1均用p2替代

：g/p1/s//p2/g 将文件中所有p1均用p2替换

: 1,$s/old/new/g 将文件中所有的『old』改成『new』

: 1,$s/old/new/gc 将文件中所有的『old』改成『new』，并在改前提示用户确认

: 10,20s/^/ / 将第10行至第20行资料的最前面插入5个空格

: %s/^m//g 删除文件中所有的^M符号

选项设置：

all 列出所有选项设置情况

term 设置终端类型

ignorance 在搜索中忽略大小写

list 显示制表位(Ctrl+I)和行尾标志（$)

number 显示行号

report 显示由面向行的命令修改过的数目

terse 显示简短的警告信息

warn 在转到别的文件时若没保存当前文件则显示NO write信息

nomagic 允许在搜索模式中，使用前面不带“\”的特殊字符

nowrapscan 禁止vi在搜索到达文件两端时，又从另一端开始

mesg 允许vi显示其他用户用write写到自己终端上的信息

最后行方式命令：

：n1,n2 co n3 将n1行到n2行之间的内容拷贝到第n3行下

：n1,n2 m n3 将n1行到n2行之间的内容移至到第n3行下

：n1,n2 d 将n1行到n2行之间的内容删除

：e filename 打开文件filename进行编辑

：e！ dd 强制关闭当前文件（不保存）打开新文件dd

: e! 放弃修改重新编辑

：q! 不保存文件并退出vi

：r fff 在当前位置读入fff文件

：set nonu 不显示行号

：set nu 显示行号

：w 保存输入的内容到当前文件

：w eee 另存为eee文件

：wq 保存并退出

：x 保存当前文件并退出

：n1,n2 w eee 将文件中n1行至n2行的内容保存成eee文件

：!command 暂时离开vi到命令行模式下执行command的显示结果。例如，“:! ls /home”即可在vi当中查看/home下面以ls输出的文件信息

寄存器操作：

"?nyy 将当前行及其下n行的内容保存到寄存器？中，其中?为一个字母，n为一个数字

"?nyw 将当前行及其下n个字保存到寄存器？中，其中?为一个字母，n为一个数字

"?nyl 将当前行及其下n个字符保存到寄存器？中，其中?为一个字母，n为一个数字

"?p 取出寄存器？中的内容并将其放到光标位置处。这里？可以是一个字母，也可以是一个数字

ndd 将当前行及其下共n行文本删除，并将所删内容放到1号删除寄存器中

vim操作环境（不建议修改/etc/vimrc文件；建议手动修改~/.vimrc文件）

vim常用指令示意图

八、shell编程

# grep $LOGNAME /etc/passwd //显示当前登录用户使用的shell

# sh //启动bsh（为Bourne Shell的简称---伯恩shell）

# bash //启动bash（为Bourne Again Shell的简称）

# csh //启动C Shell

# type [-tpa] name //bash shell的内置命令。

type：不加任何参数时，type会显示出name是外部命令还是bash内置命令

-t ：当加入-t参数时，type会将name以下面这些字眼显示出它的意义：

file：表示为外部命令

alias：表示该命令为命令别名所设置的名称；

builtin：表示该命令为bash内置的命令功能。

-p ：如果后面接的name为外部命令时，才会显示完整文件名；

-a ：会由PATH变量定义的路径中，将所有含name的命令都列出来，包含alias

# cp /var/spool/mail/root /etc/crontab \

> /etc/fstab /root //将三个文件复制到root目录下（如果命令串太长的话，可以用两行来输出）。“\[Enter]”为转义符，“>”为系统自动出现的，不需要输入。

# source ~/.bashrc //将主文件夹的~/.bashrc的设置读入目前的bash环境中

# . ~/.bashrc //将主文件夹的~/.bashrc的设置读入目前的bash环境中

# ll -d /etc/cron\* //找出/etc/下面以cron为开头的文件名

# ll -d /etc/????? //找出/etc/下面文件名刚好是五个字母的文件名

# ll -d /etc/\*[0-9]\* //找出/etc/下面文件名含有数字的文件名

# ll -d /etc/[^a-z]\* //找出/etc/下面文件名开头非为小写字母的文件名

# cp -a /etc/[^a-z]\* /tmp //将上例找到的文件复制到 /tmp中

# last | cut -d ' ' -f1 | sort //将输出的数据仅取账号，并加以排序

# last | cut -d ' ' -f1 | sort | uniq //将输出的数据仅取账号，排序后仅取出一位

# last | cut -d ' ' -f1 | sort | uniq -c //将输出的数据仅取账号，排序后仅取出一位并显示每个账号登录的总次数

# last | grep [a-zA-Z] | grep -v ‘wtmp’| wc -l //显示登录系统的总人数

# last | tee last.list | cut -d “ “ -f1 //将last的输出存一份到last.list文件中

# ls －l /home | tee ~/homefile | more //将ls的数据存一份到~/homefile，同时屏幕也有输出信息

# ls －l / | tee -a ~/homefile | more //加上-a参数则能将信息累加

# cp /etc/passwd /root/passwd && unix2dos /root/passwd

# file /etc/passwd /root/passwd

# cat /root/passwd | tr -d ‘\r’ > /root/passwd.linux

# ll /etc/passwd /root/passwd\* //将/etc/passwd转存成dos断行到/root/passwd中，再将^M 符号删除。\r指的是dos的断行字符 ^M

# cat -A /etc/man.config

# cat /etc/man.config | col -x | cat -A | more //利用cat -A 显示所有特殊按键，最后以 col将[tab]转成空白。上面看到的^I符号就是tab，如此一来，[tab]按键会被替换成为空格键，输出就美观多了

# man col > /root/col.man

# vi /root/col.man

# man col | col -b > /root/col.man //将col的man page转存成为/root/col.man的纯文本文件

#############################################

通配符与特殊符号

符号 意义

\* 代表『 0 个到无穷多个』任意字符

? 代表『一定有一个』任意字符

[ ] 同样代表『一定有一个在括号内』的字符(非任意字符)。例如 [abcd] 代表『一定有一个字符， 可能是 a, b, c, d 这四个任何一个』

[ - ] 若有减号在中括号内时，代表『在编码顺序内的所有字符』。例如 [0-9] 代表 0 到 9 之间的所有数字，因为数字的语系编码是连续的！

[^ ] 若中括号内的第一个字符为指数符号 (^) ，那表示『反向选择』，例如 [^abc] 代表 一定有一个字符，只要是非 a, b, c 的其他字符就接受的意思。

bash 环境中的特殊符号

符号 内容

# 批注符号：这个最常被使用在 script 当中，视为说明！在后的数据均不运行

\ 跳脱符号：将『特殊字符或通配符』还原成一般字符

| 管线 (pipe)：分隔两个管线命令的界定(后两节介绍)；

; 连续命令下达分隔符：连续性命令的界定 (注意！与管线命令并不相同)

~ 用户的家目录

$ 取用变量前导符：亦即是变量之前需要加的变量取代值

& 工作控制 (job control)：将命令变成背景下工作

! 逻辑运算意义上的『非』 not 的意思！

/ 目录符号：路径分隔的符号

>, >> 数据流重导向：输出导向，分别是『取代』与『累加』

<, << 数据流重导向：输入导向

' ' 单引号，不具有变量置换的功能

" " 具有变量置换的功能！

` ` 两个『 ` 』中间为可以先运行的命令，亦可使用 $( )

( ) 在中间为子 shell 的起始与结束

{ } 在中间为命令区块的组合！

#############################################

# dmesg | grep -n -color=auto ‘eth’ //用dmesg列出内核信息，再以grep找出内含eth的那些行，并显色、加上行号

# dmesg | grep -n -A3 -B2 -color=auto ‘eth’ //承上，在关键字所在行的前两行与后三行也一起找出来显示

#############################################

基础正则表达式字符 (characters)

^word

意义：待查找的字符串(word)在行首

范例：查找行首为 # 开始的那一行，并列出行号

grep -n '^#' regular\_express.txt

word$

意义：待查找的字符串(word)在行尾

范例：将行尾为 ! 的那一行列印出来，并列出行号

grep -n '!$' regular\_express.txt

.

意义：代表一定有一个任意字符的字符

范例：查找的字符串可以是 (eve) (eae) (eee) (e e)， 但不能仅有 (ee)，亦即 e 与 e 中间“一定”仅有一个字符，而空格符也是字符

grep -n 'e.e' regular\_express.txt

\

意义：转义字符，将特殊符号的特殊意义去除

范例：查找含有单引号 ' 的那一行

grep -n \' regular\_express.txt

\*

意义：重复零个到无穷多个的前一个 RE（regular）字符

范例：找出含有 (es) (ess) (esss) 等等的字符串，注意，因为 \* 可以是 0 个，所以 es 也是符合带查找字符串。另外，因为 \* 为重复“前一个 RE 字符”的符号， 因此，在 \* 之前必须要紧接著一个 RE 字符喔。例如任意字符则为 “.\*”

grep -n 'ess\*' regular\_express.txt

[list]

意义：从字符集合的 RE 字符里面找出想要选取的字符

范例：查找含有 (gl) 或 (gd) 的那一行，需要特别留意的是，在 [] 当中代表一个待查找的字符， 例如“ a[afl]y ”代表查找的字符串可以是 aay, afy, aly 即 [afl] 代表 a 或 f 或 l 的意思

grep -n 'g[ld]' regular\_express.txt

[n1-n2]

意义：从字符集合的 RE 字符里面找出想要选取的字符范围

范例：查找含有任意数字的那一行。需特别留意，在字符集合 [] 中的减号 - 是有特殊意义的，他代表两个字符之间的所有连续字符。但这个连续与否与 ASCII 编码有关，因此，你的编码需要配置正确(在 bash 当中，需要确定 LANG 与 LANGUAGE 的变量是否正确)。例如所有大写字符则为 [A-Z]

grep -n '[A-Z]' regular\_express.txt

[^list]

意义：从字符集合的 RE 字符里面找出不要的字符串或范围

范例：查找的字符串可以是 (oog) (ood) 但不能是 (oot) ，那个 ^ 在 [] 内时，代表的意义是“反向选择”的意思。例如，我不要大写字符，则为 [^A-Z]。但是，需要特别注意的是，如果以 grep -n [^A-Z] regular\_express.txt 来查找，却发现该文件内的所有行都被列出，为什么？因为这个 [^A-Z] 是“非大写字符”的意思， 因为每一行均有非大写字符，例如第一行的 "Open Source" 就有 p,e,n,o等小写字符

grep -n 'oo[^t]' regular\_express.txt

\{n,m\}

意义：连续 n 到 m 个的前一个 RE 字符，若为 \{n\} 则是连续 n 个的前一个 RE 字符，若是 \{n,\} 则是连续 n 个以上的前一个 RE 字符

范例：在 g 与 g 之间有2个到3个的o存在的字符串，亦即 (goog)(gooog)

grep -n 'go\{2,3\}g' regular\_express.txt

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

# test -e /pa/lamprj && echo "exist" || echo "Not exist" //检查/pa/lamprj目录是否存在

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

文件测试的标志与意义：

关於某个文件名的“文件类型”判断，如 test -e filename 表示存在否

-e 该文件名是否存在？(常用)

-f 该文件名是否存在且为文件(file)(常用)

-d 该文件名是否存在且为目录(directory)(常用)

-b 该文件名是否存在且为一个 block device 设备

-c 该文件名是否存在且为一个 character device 设备

-S 该文件名是否存在且为一个 Socket 文件

-p 该文件名是否存在且为一个 FIFO (pipe) 文件

-L 该文件名是否存在且为一个连结文件

关於文件的权限检测，如 test -r filename 表示可读否 (但 root 权限常有例外)

-r 检测该文件名是否存在且具有“可读”的权限

-w 检测该文件名是否存在且具有“可写”的权限

-x 检测该文件名是否存在且具有“可执行”的权限

-u 检测该文件名是否存在且具有“SUID”的属性

-g 检测该文件名是否存在且具有“SGID”的属性

-k 检测该文件名是否存在且具有“Sticky bit”的属性

-s 检测该文件名是否存在且为“非空白文件”

两个文件之间的比较，如：test file1 -nt file2

-nt (newer than)判断 file1 是否比 file2 新

-ot (older than)判断 file1 是否比 file2 旧

-ef 判断 file1 与 file2 是否为同一文件，可用在判断 hard link 的判定上。 主要意义在判定两个文件是否均指向同一个 inode

关於两个整数之间的判定，例如 test n1 -eq n2

-eq 两数值相等 (equal)

-ne 两数值不等 (not equal)

-gt n1 大於 n2 (greater than)

-lt n1 小於 n2 (less than)

-ge n1 大於等於 n2 (greater than or equal)

-le n1 小於等於 n2 (less than or equal)

判定字串的数据

test -z string 判定字串是否为 0，若 string 为空字串，则为 true

test -n string 判定字串是否非为0，若 string 为空字串，则为 false

注：-n 亦可省略

test str1 = str2 判定 str1 是否等於 str2 ，若相等，则回传 true

test str1 != str2 判定 str1 是否不等於 str2 ，若相等，则回传 false

多重条件判定，例如：test -r filename -a -x filename

-a 两个条件同时成立！例如 test -r file -a -x file，则 file 同时具有 r 与 x 权限时，才回传 true

-o 任何一个条件成立！例如 test -r file -o -x file，则 file 具有 r 或 x 权限时，就可回传 true

! 反相状态，如 test ! -x file ，当 file 不具有 x 时，回传 true

＝＝＝＝＝＝＝

常用shell命令组合

# kudzu --probe --class=network //kudzu查看网卡型号

# ps -e -o "%C : %p : %z : %a"|sort -k5 -nr //查看进程，按内存从大到小排列

# ps -e -o "%C : %p : %z : %a"|sort -nr //按cpu利用率从大到小排列

# ls /etc/rc3.d/S\* |cut -c 15- //显示运行3级别开启的服务(从中了解到cut的用途，截取数据)

# ifconfig eth0 |grep "inet addr:" |awk '{print $2}'|cut -c 6- //显示IP地址

# free -m |grep "Mem" | awk '{print $2}' //显示内存的大小

# cat /proc/cpuinfo |grep -c processor //显示CPU的数量

# mpstat 1 1 //显示CPU负载

# du -cks \* | sort -rn | head -n 10 //显存当前路径下从大到小排序的前10个文件或目录大小

# iostat -x 1 2 //磁盘I/O负载

# sar -n DEV //网络负载

# ps aux | wc -l //进程总数