



施威铭研究室 著

Linux

命令详解词典

- 精选381个命令。
- 详细列出1729个范例。
- 提供3种查询索引。
- 适合各种Linux发行版。

旗標



机械工业出版社
China Machine Press

详尽的说明

- 详细说明命令的功能、用法和使用场合以及操作命令。
- 大量列举操作范例，结合学习和实践。
- 完整介绍命令参数，详细列出各参数的作用。
- 列出相关命令，提供全面的学习指引。

Linux

命令详解词典

贴心的查询设计

- 按字母顺序a~z编排，方便查找已知命令。
- 分类命令索引，可以按照功能来查询可用的同类型命令。
- 按常用指数排列索引，引导渐进式学习。

多功能的学习词典

- 严格筛选、验证，适合各层次用户。
- 常用指数表示命令的常用程度。
- 列举相关命令，建立命令索引。



投稿热线: (010) 88379604

购书热线: (010) 68995259, 68995264

读者信箱: hzsj@hzbook.com

华章网站 <http://www.hzbook.com>

网上购书: www.china-pub.com

封面设计: 任超



本书不得在中国内地以外之地区销售 (Not For Sale Outside Mainland China, Especially HongKong SAR, Macao SAR And Taiwan Area)

上架指导: 计算机/程序设计

ISBN 978-7-111-23327-5



9 787111 233275

ISBN 978-7-111-23327-5

定价: 59.00元

TP316.81-61/2

2008

Linux

命令详解词典

施威铭研究室 著



机械工业出版社
China Machine Press



本书精选了381个Linux命令，详细罗列1729个范例，并提供3种查询索引方式。本书详细列出Linux命令的参数、功能说明、所属包、丰富的范例以及相关命令等信息，以统一的结构呈现，让Linux用户在短时间内找到需要的命令，了解命令的正确使用方式，并成功地将范例应用到实际中。

本书适合作为程序员和工程技术人员的工具手册。

本书中文简体字版由中国台湾旗标出版股份有限公司授权机械工业出版社出版，未经本书原版出版者和本书出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书原版版权旗标出版股份有限公司

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2008-0339

图书在版编目 (CIP) 数据

Linux命令详解词典 / 施威铭研究室著. —北京：机械工业出版社，2008.3

ISBN 978-7-111-23327-5

I. L… II. 施… III. Linux操作系统 IV. TP316.89

中国版本图书馆CIP数据核字 (2008) 第010091号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑：周茂辉

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2008年3月第1版第1次印刷

145mm × 210mm · 28.875印张

标准书号：ISBN 978-7-111-23327-5

定价：59.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书电话 (010) 68326294

序

Linux图形模式下的各种应用程序、工具程序日益完善，操作界面也更为友善。对一般仅需计算机完成日常文本编辑、上网、收发邮件等工作的用户来说，熟悉X Window的操作的确已经足够了。但对于系统管理员和用Linux架设网站的网络管理员而言，文本模式下的各项命令却是不可或缺的工具。

本书整理了近400个命令，详细列出其参数、功能、所属包等信息。并考虑大多数用户在面对各命令的长串参数时常会有不知如何下手的困扰，特别列举大量的常用操作范例以期帮助用户在实际操作上更容易上手。

在命令的分类、检索部分，除了一般按照字母顺序编排的方式外，另有“分类索引”及“常用指数索引”。前者供用户在不知该使用什么命令时根据需求查询；并方便在命令学习上通过此索引学会相关功能的命令及其差异。后者则方便初学者在茫茫命令的大海中挑选较常用的命令入手，提高学习效率。

计算机的命令词典与一般文字的词典一样，在整理及编纂上耗时费力，却仍难免有所缺漏，并且常有新命令、新参数出现，系统命令也根据需求变化而多有变动。因此，本书若有未竟之处，尚请您不吝指正，以作为我们日后修订的参考。

施威铭研究室

2007年12月

如何使用本书

结构说明

本书分为两大部分：一为目录索引，一为命令词典。分别描述如下。

目录索引

为方便查阅，本书共有3种索引。

- 目录：按照字母顺序A~Z列出所有命令，可用于检索特定命令的详细用法。
- 分类命令索引：按照命令功能分类列出相关命令及其功能说明。方便查询同类型命令，并可在不知用什么命令时按照目的查询。
- 常用指数索引：按照常用指数分类，提示学习重点，可作为命令学习进程的依据。

命令词典

此部分以字母A到Z的方式排列命令，每个命令包含常用指数、功能说明、语法、包名称、相关命令、补充说明、参数、范例及操作命令等。可由命令名称查得其功能、使用方法及实际操作范例。

有关各项的详细说明，请参考“本书编排体例”。

执行环境与注意事项

本书所列命令可在大多数 Linux 发行版下执行，仅有部分命令为特定发行版专有，此类命令所在包名称将会标示出来。

本书说明操作命令时，请注意下列事项：

- 执行时出现“command not found”的错误信息

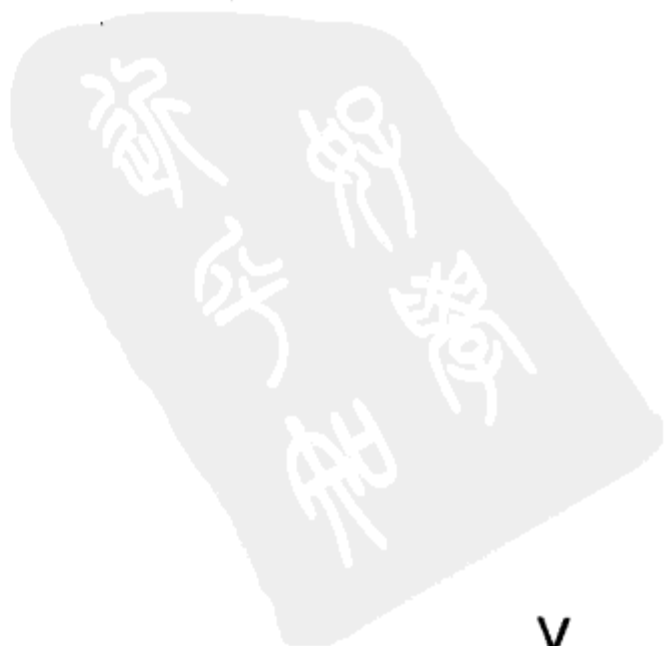
在命令提示字符后直接输入命令名称执行命令时，系统会到 \$PATH 环境变量设置的目录下（一般至少有 /sbin、/bin、/usr/sbin、

/usr/bin、/usr/X11R6/bin) 查找命令并执行，若所执行的命令没有在 \$PATH 设置的目录中，便会出现找不到命令的情况。另外，若没有安装该命令所属的包，或是所使用的账号没有执行该命令的权限，也会出现此错误信息。

如果您在执行命令时出现此错误信息，请先确定已安装指定所属的包，并确认命令所在目录设置在 \$PATH 环境变量中。

- 参数顺序

一般而言参数没有固定的顺序，即执行 `ls -al` 与 `ls -la` 的结果相同。只有参数后有指定实参者，该参数与实参间不可插入其他参数，或更改参数与实参的顺序。






本书编排体例

❶ 命令名称

命令名称及其原意。

❷ 常用指数

以1~3只企鹅标示该命令的使用频率：

- ：一般用户最常用的命令，应该要熟悉其用法。
- ：一般用户次常用的命令，应知道其基本用法，具体操作可待实际使用时再行查阅。
- ：较少使用的命令，可于需要时再行查阅。

❸ 功能说明

简述命令功能。

❹ 语 法

详列命令的语法、可用参数及实参。

❺ 包 名 称

命令所属的包。若在不同发行版中分属不同包，则会分别标示。若执行命令时出现“command not found”的错误信息，可能是尚未安装该命令所属的包，此时，可依此项所列内容检查确认。

❻ 相关命令

与该命令相关的其他命令。

❼ 补充说明

补充功能说明不足的部分，详细解说命令的功能、相关概念、使用时机以及它与相关命令间的异同。

❽ 参 数

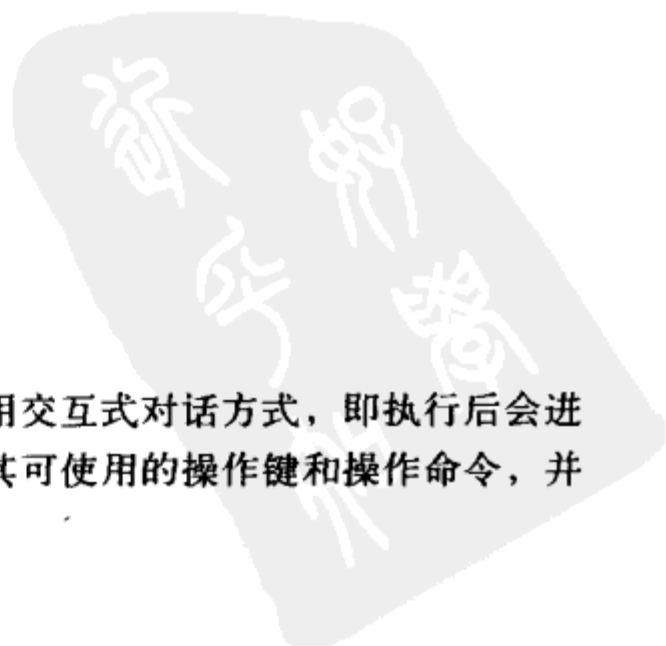
详解语法中所列参数的功能。

❾ 范 例

举实例说明命令的用法、使用时机与效果。

❿ 操作命令

部分命令（如文本编辑命令、文件传输命令等）采用交互式对话方式，即执行后会进入其操作环境，有其专有的操作命令。此处即列出其可使用的操作键和操作命令，并说明功能。



① ncftp (nc file transfer protocol) 常用指数: ②

③ 功能说明: 传输文件。

④ 语 法: ncftp [-u <账号名称>] [主机名称或 IP 地址]

⑤ 包 名 称: ncftp。

⑥ 相关命令: ftp, ncftpget, ncftpput。

⑦ 补充说明: ftp让用户得以下载存放于远程主机的文件,也能将文件上传至远程主机放置。NcFTP是文本模式ftp程序的佼佼者,它具备多样特色,包括显示传输速率、下载进度、自动续传、标注书签、可通过防火墙和代理服务器等等。

⑧ 参 数:

-u <账号名称>	设定登录ftp服务器的账号,不使用匿名登录
-----------	-----------------------

⑨ 范 例:

① 以匿名方式登录ftp.flag.com.tw服务器,可用稍后所介绍的操作命令来上传或下载文件:

```
# ncftp ftp.flag.com.tw
```

② 以指定的alkane账号登录ftp.flag.com.tw服务器:

```
# ncftp -u alkane ftp.flag.com.tw
```

⑩ 操作命令:

ascii	将传输数据的模式设成ASCII模式
bgget <文件名称>	以后台的方式下载远程的文件
bgput <文件名称>	以后台的方式上传本地端的文件
bgstart	以后台的方式传输当前正在下载或下传的文件

目 录

A

命令名称	功能说明	页码
adduser	建立用户账号	1
alias	设置命令的别名	3
apachectl	用来控制Apache HTTP服务器的程序	4
apmd	高级电源管理服务程序	6
ar	建立或修改档案文件 (archive), 或是从档案文件中提取文件	7
arp	管理系统中的ARP高速缓存	12
arpwatch	监听网络上ARP的记录	14
at	在指定的时间执行命令	16
atq	显示待执行的工作	18
atrm	删除待执行的工作	19
aumix	设置音效设备	20

B

命令名称	功能说明	页码
badblocks	检查磁盘设备中损坏的区块	25
batch	在系统负载许可时, 立即执行批处理命令	27
bg	将程序放到后台执行	29
bind	显示或设置键盘按键与其相关的功能	30
bunzip2	.bz2文件的解压缩程序	31
bzip2	.bz2文件的压缩程序	33
bzip2recover	用来修复损坏的.bz2文件	36

C

命令名称	功能说明	页码
cal	显示月历	38
cat	连接多个文件, 并将它们的内容输出到标准输出设备	40

cd	切换目录	44
cdp	播放音乐CD	45
cfdisk	硬盘分区	48
chattr	改变文件属性	52
chfn	改变finger命令显示的信息	55
chgrp	更改文件或目录的所属组	56
chkconfig	检查、设置系统的各种服务	59
chmod	更改文件或目录的权限	62
chown	更改文件或目录的拥有者或所属组	66
chroot	改变根目录	68
chsh	更换登录系统时使用的shell	69
cksum	检查文件的CRC (Cyclic Redundancy Checking)是否正确	70
clear	清除画面	71
clock	调整RTC (Real Time Clock)时间	71
cmp	比较两个文件是否有差异	74
cmuwmtopbm	转换位图文件	76
col	过滤控制字符	76
colrm	滤掉指定的列	77
comm	比较两个已排序的文件	78
compress	压缩或解压文件	81
convdate	转换日期时间	83
cp	复制文件或目录	85
cpio	备份文件	90
crontab	设置计时器	100
csplit	分割文件	103
ctlinnd	设置INN新闻组服务器	106
cu	连接另一个系统主机	107
cut	指定欲显示的文件内容, 输出到标准输出设备	109

D

命令名称	功能说明	页码
date	显示或设置系统时间与日期	112
dd	读取、转换并输出数据	119

declare	声明shell变量	122
depmod	分析可载入模块的相依性	124
df	显示磁盘的文件系统与使用情形	125
diff	比较文件的差异	130
diffstat	根据diff的比较结果, 显示统计数字	139
dip	IP拨号连接	140
dircolors	设置ls命令在显示目录或文件时所用的颜色	143
dirs	显示目录记录	146
dmesg	显示开机信息	148
du	显示目录或文件的大小	149
dump	备份文件系统	156

E

命令名称	功能说明	页码
e2fsck	检查ext2文件系统的正确性	160
echo	显示文本	165
ed	文本编辑器	167
edquota	编辑用户或组的quota	169
efax	收发传真	171
egrep	查找文件里符合条件的字符串	174
eject	退出抽取式设备	175
enable	启动或关闭shell内建命令	176
enscript	将文本文件转为PostScript	178
eval	重新计算求出参数的内容	187
ex	在Ex模式下启动vim文本编辑器	188
exec	shell执行指定的命令后即交出控制权	188
exit	退出当前的shell	189
export	设置或显示环境变量	190
ext2ed	ext2文件系统编辑程序	191

F

命令名称	功能说明	页码
fbset	设定帧缓冲 (Frame Buffer)	195
fc	修改命令, 且执行该命令	200
fdisk	磁盘分区	202
fetchmail	接收电子邮件	206
fg	将程序或命令切换至前台执行	214
fgrep	查找文件里符合条件的字符串	215
file	识别文件类型	215
find	查找文件或目录	219
finger	查找并显示用户信息	231
fmt	编排文本文件	234
fold	限制文件行宽	236
free	查看内存状态	237
fsck	检查文件系统并尝试修复错误	240
fsck.ext3	检查文件系统并尝试修复错误	242
fsck.minix	检查文件系统并尝试修复错误	248
ftp	传输文件	250
ftpcount	显示当前以FTP登录的用户人数	253
ftpsht	在指定的时间关闭FTP服务器	254
ftpwho	显示当前所有以FTP登录的用户信息	255

G

命令名称	功能说明	页码
gemtopbm	转换图像文件	257
getlist	下载新闻组列表	257
giftopnm	转换图像文件	259
gouldtoppm	转换扫描文件	260
grep	查找文件里符合条件的字符串	260
groupadd	建立组	266
groupdel	删除组	268

groupmod	更改组识别码或名称	268
grpconv	打开组的影子密码	269
grpunconv	关闭组的影子密码	270
grub-install	安装GRUB启动管理程序	271
gunzip	解压文件	272
gzexe	压缩执行文件	275
gzip	压缩文件	276

H

命令名称	功能说明	页码
halt	关闭系统	282
hdparm	显示与设置硬盘的参数	283
head	输出文件内容的最前面部分	291
help	显示shell内建命令的说明	293
history	列出之前用过的命令	293
host	DNS查询工具	295
hostname	查询及设置主机名称	297
httpd	Apache HTTP服务器程序	299
hwclock	显示与设置硬件时钟	302

I

命令名称	功能说明	页码
id	显示用户ID, 以及所属组ID	304
ifconfig	显示或设置网络设备	306
indent	调整C源代码文件的格式	309
info	显示说明	315
inncheck	检查inn相关设置文件的语法是否正确	318
insmod	载入模块	321
iptables	包过滤功能和NAT的管理工具	322
ispell	拼写检查程序	325

J

命令名称	功能说明	页码
jed	编辑文本文件	329
jobs	显示正在后台执行的工作	332
joe	编辑文本文件	334
join	将两个文件中, 指定列内容相同的行连接起来	341

K

命令名称	功能说明	页码
kill	删除执行中的程序或工作	345

L

命令名称	功能说明	页码
last	列出当前与过去登录系统的用户相关信息	347
less	显示文件内容	350
lha	压缩或解压文件	356
lilo	安装内核载入、启动管理程序	361
lispmtopgm	转换图像文件	364
ln	连接文件或目录	365
lnidir	连接目录内容	369
locate	查找文件	370
login	登录系统	372
logname	显示用户名称	372
logout	退出系统	373
logrotate	管理记录文件	374
look	查询单词	375
losetup	设置循环设备	376
lpc	控制打印机	378
lpd	提供打印机排队常驻服务	379
lpq	显示打印操作	380
lpr	打印文件	382

lprm	删除打印工作	383
ls	列出目录内容	384
lsattr	显示文件属性	394
lsb_release	显示Linux发行版本的相关信息	395
lsmod	显示已载入系统的模块	397
lynx	浏览互联网 (World Wide Web)	398

M

命令名称	功能说明	页码
mail	E-mail管理程序	408
mailq	显示待发送邮件的列表	411
make	编译系统内核或模块	412
man	在线查询命令	414
manpath	显示说明文件的查找路径	418
mattrib	更改或显示MS-DOS文件的属性	419
mbadblocks	检查MS-DOS文件系统的磁盘 是否有损坏的扇区	420
mc	菜单式的文件管理程序	421
mcd	在MS-DOS文件系统中切换工作目录	425
mcopy	复制MS-DOS文件	425
md5sum	计算与检查MD5函数值	428
mdel	删除MS-DOS文件	430
mdeltree	删除MS-DOS目录	430
mdir	显示MS-DOS目录	431
mdu	显示MS-DOS目录所占用的磁盘空间	433
mesg	设置终端的写入权限	434
metamail	负责处理非文本E-mail的程序	435
mformat	对MS-DOS文件系统的磁盘进行格式化	439
minfo	显示MS-DOS文件系统的各项参数	440
mingetty	精简版的getty	442
minicom	数据机通信程序	443
mkbootdisk	建立当前系统的启动盘	447
mkdir	建立目录	448

mkdosfs	建立MS-DOS文件系统	450
mke2fs	建立ext2文件系统	452
mkfs	建立各种文件系统	455
mkfs.ext3	与mke2fs相同	457
mkfs.minix	建立Minix文件系统	457
mkfs.msdos	与mkdosfs相同	458
mkinitrd	建立要载入ramdisk的映像文件	459
mkisofs	建立ISO 9660映像文件	460
mkraid	初始化磁盘阵列	466
mkswap	设置交换区 (swap area)	467
mktemp	建立临时文件	469
mlabel	显示或设置MS-DOS磁盘的标签名称	470
mmd	在MS-DOS文件系统中建立目录	471
mmount	加载MS-DOS文件系统	472
mmove	在MS-DOS文件系统中, 移动文件或目录, 或更改名称	473
modinfo	显示kernel模块的信息	473
modprobe	自动处理可载入模块	475
more	使文件能逐页地显示	477
mount	加载文件系统	479
mouseconfig	设置鼠标相关参数	484
mpage	在PostScript打印机上, 将许多页面 合并成一页来打印	485
mpartition	建立或删除MS-DOS的分区	491
mrd	删除MS-DOS文件系统中的目录	492
mren	更改MS-DOS文件或目录的名称, 或移动文件或目录	493
mshowfat	显示MS-DOS文件在FAT中的记录	494
mtools	显示mtools支持的命令	494
mtoolstest	测试并显示mtools的相关设置	496
mtype	显示MS-DOS文件的内容	498
mutt	E-mail管理程序	498
mv	移动或更名现有的文件或目录	503
mzip	Zip/Jaz磁盘控制命令	505

N

命令名称	功能说明	页码
nc	连接与监听 TCP/UDP 通信端口	506
ncftp	传输文件	507
ncftpget	下载文件	510
ncftpput	上传文件	513
netconfig	设置网络环境	517
netstat	显示网络状态	518
newgrp	登录另一个组	523
nice	设置优先级	523
nntpget	下载新闻组文章	525
nohup	可在退出后继续执行程序	526
ntsysv	设置系统的各种服务	527

O

命令名称	功能说明	页码
od	输出文件内容	530
open	打开虚拟终端执行命令	535

P

命令名称	功能说明	页码
passwd	设置密码	537
paste	合并文件的行	539
patch	修补文件	541
pcxto ppm	转换图像文件	545
pfbtops	转换字体文件	546
pico	编辑文本文件	547
pictto ppm	转换图像文件	552
pine	收发电子邮件、浏览新闻组	553
ping	侦测主机	558
pjto ppm	转换打印文件	561

playmidi	播放音乐文件	562
popd	删除目录堆栈中的记录	566
pppd	提供 PPP 连接常驻服务	567
pppstats	显示 PPP 连接状态	571
pr	将文件格式化编排, 以便打印	573
procinfo	显示系统状态	578
ps	报告程序状况	582
pstree	以树状图显示程序	590
pushd	加入目录堆栈中的记录	594
pwconv	打开用户的影子密码	596
pwd	显示工作目录	597
pwunconv	关闭用户的影子密码	598

Q

命令名称	功能说明	页码
qrttoppm	转换QRT输出文件	599
quota	显示磁盘已使用的空间与限制	599
quotacheck	检查磁盘的使用空间与限制	602
quotaoff	关闭磁盘空间限制	605
quotaon	打开磁盘空间限制	607

R

命令名称	功能说明	页码
raidstart	启动磁盘阵列	609
raidstop	停止运作中的磁盘阵列	610
rasttopnm	转换图像文件	611
rcp	远程复制文件或目录	612
rdate	显示其他主机的日期与时间	613
reboot	重新开机	614
renice	调整优先级	615
repquota	检查磁盘空间限制的状态	617

resize	设定X Window终端窗口大小	619
restore	还原 (Restore) 由转储 (Dump) 操作所备份 下来的文件或整个文件系统 (一个分区)	620
rgrep	递归查找文件里符合条件的字符串	621
rlogin	远程登录	627
rm	删除文件或目录	628
rmdir	删除目录	629
rmmod	删除模块	630
rmt	远程磁带传输协议模块	631
route	管理与显示路由表	632
rpm	管理包 (Package)	634
rsh	远程登录的shell	645
runlevel	显示当前系统的执行等级	646
rwho	查看系统用户	648

S

命令名称	功能说明	页码
samba	Samba 服务器控制	649
screen	多重窗口管理程序	651
sed	利用 script 来处理文本文件	657
set	设置 shell	660
setenv	查询或显示环境变量	665
setserial	设置或显示串行端口的相关信息	666
setup	设置公用程序	670
sfdisk	硬盘分区工具程序	671
shapecfg	限制网络设备的流量	675
shutdown	系统关机命令	676
sleep	暂停执行命令	678
slocate	查找文件或目录	679
slrn	新闻组阅读程序	682
smbclient	可存取 SMB/CIFS 服务器的客户端程序	685
smbd	Samba 服务器程序	692
sndconfig	设置声卡	693

sort	将文本文件内容加以排序	694
sox	音效文件转换程序	697
spell	拼字检查程序	700
split	切割文件	700
startx	启动X Window	703
stat	显示inode内容	704
statserial	显示串行端口状态	705
su	变更用户身份	706
sudo	以其他身份来执行命令	710
sum	计算文件的校验和与区块数	712
suspend	暂停执行shell	713
SVGATextMode	增强文本模式的显示画面	714
swapoff	关闭系统交换区 (swap area)	715
swapon	启动系统交换区 (swap area)	716
swatch	系统监控程序	718
symlinks	维护符号链接的工具程序	719
sync	将内存缓冲区内的数据写入磁盘	722

T

命令名称	功能说明	页码
tac	连接多个文件，并将其内容反序 输出到标准输出设备	723
tail	输出文件内容的最后部分	724
talk	与其他用户交谈	726
tar	备份文件	728
tcpdump	转储网络传输数据	738
tee	读取标准输入的数据，并将其内容输出成文件	741
telinit	切换系统的执行等级	742
telnet	远程登录	744
testparm	测试Samba的设置是否正确无误	746
tftp	传输文件	747
tgatoppm	转换图像文件	749
tifftopnm	转换图像文件	749

timeconfig	设置时区	750
tload	显示系统负载状况	752
tmpwatch	删除临时文件	753
top	显示、管理执行中的程序	754
touch	改变文件或目录时间	757
tr	转换字符	759
traceroute	显示包到主机间的路径	762
tree	以树状图列出目录的内容	765
ttmkfdir	建立TTF字体的索引文件	770
tty	显示终端连接标准输入设备的文件名称	772
tunelp	改变打印设备的参数	773

U

命令名称	功能说明	页码
ulimit	控制shell程序的资源	775
umask	指定在建立文件时默认的权限掩码	777
umount	卸除文件系统	778
unalias	删除别名	780
uname	显示系统信息	781
unarj	解压缩.arj文件	783
uncompress	解压缩.Z文件	785
uniq	检查及删除文本文件中重复出现的行列	786
unset	删除变量或函数	789
unzip	解压缩zip文件	790
updatedb	更新文件数据库	794
useradd	建立用户账号	796
userconf	用户账号设置程序	799
userdel	删除用户账号	800
usermod	修改用户账号	801
uucico	UUCP文件传输服务程序	803
uucp	在UNIX系统之间传送文件	805
uudecode	将uuencode所产生的编码文件转换回原来的格式	807

uuencode	将文件转换为ASCII编码的文件	808
uulog	显示UUCP记录文件	809
uuname	显示全部的UUCP远程主机	810
uupick	处理传送进来的文件	811
uustat	显示UUCP当前的状态	812
uuto	将文件传送至远程的UUCP主机	815
uux	在远程的UUCP主机上执行命令	815

V

命令名称	功能说明	页码
vi	编辑文本文件	818
vim	编辑文本文件	820
vlock	锁住虚拟控制台 (Console)	826

W

命令名称	功能说明	页码
w	显示当前登录系统的用户信息	828
wait	等待程序回报其状态	830
wall	传送信息	831
wc	计算字数	832
wget	从互联网 (WWW) 下载文件	833
whereis	查找文件	839
which	查找文件	840
who	显示当前登录系统的用户信息	841
whoami	显示用户名称	844
whois	查找并显示指定网址的相关信息	845
write	传送信息	846

X

命令名称	功能说明	页码
xauth	编辑修改X server授权信息	848
xf86config	设置X Window	850
xfst	提供X Window字体服务器	851
xhost	控制存取X server的主机	853
xlsatoms	列出X Server定义的成分	855
xlsclients	列出显示器中的客户端应用程序	856
xlsfonts	列出X Server使用的字体	858
xplaymidi	播放音乐文件	861
xset	设置X Window的使用偏好	865

Y

命令名称	功能说明	页码
yes	响应字符串	868
ytalk	与其他用户对话	869
yuvsplittoppm	转换视频文件	871
yuvtoppm	转换视频文件	872

Z

命令名称	功能说明	页码
zcat	连接多个压缩文件，并将它们的内容 输出到标准输出设备	873
zip	压缩文件	874
zipinfo	列出压缩文件信息	879

分类命令索引

系统设置

命令名称	功能说明	页码
alias	设置命令的别名	3
apmd	高级电源管理服务程序	6
aumix	设置音效设备	20
bind	显示或设置键盘按键与其相关的功能	30
chkconfig	检查、设置系统的各种服务	59
chroot	改变根目录	68
clock	调整RTC (Real Time Clock) 时间	71
crontab	设置计时器	100
declare	声明shell变量	122
depmod	分析可载入模块的相依性	124
dircolors	设置ls命令在显示目录或文件时所用的颜色	143
dmesg	显示开机信息	148
enable	启动或关闭shell内建命令	176
eval	重新计算求出参数的内容	187
export	设置或显示环境变量	190
fbset	设置帧缓冲 (Frame Buffer)	195
grpconv	打开组的影子密码	269
grpunconv	关闭组的影子密码	270
grub-install	安装GRUB启动管理程序	271
hwclock	显示与设置硬件时钟	302
insmod	载入模块	321
lilo	安装内核载入、启动管理程序	361
lsmod	显示已载入系统的模块	397
minfo	显示MS-DOS文件系统的各项参数	440
modinfo	显示kernel模块的信息	473
modprobe	自动处理可载入模块	475
mouseconfig	设置鼠标相关参数	484
ntsysv	设置系统的各种服务	527
passwd	设置密码	537
pwconv	打开用户的影子密码	596

pwunconv	关闭用户的影子密码	598
rdate	显示其他主机的日期与时间	613
resize	设置X Window终端窗口大小	619
rmmod	删除模块	630
rpm	管理包 (Package)	634
set	设置shell	660
setenv	查询或显示环境变量	665
setup	设置公用程序	670
sndconfig	设置声卡	693
SVGATextMode	增强文本模式的显示画面	714
telinit	切换系统的执行等级	742
timeconfig	设置时区	750
ulimit	控制shell程序的资源	775
unalias	删除别名	780
unset	删除变量或函数	789

系统管理

命令名称	功能说明	页码
adduser	建立用户账号	1
chfn	改变finger命令显示的信息	55
chsh	更换登录系统时使用的shell	69
date	显示或设置系统时间与日期	112
exit	退出当前的shell	189
finger	查找并显示用户信息	231
free	查看内存状态	237
groupadd	建立组	266
groupdel	删除组	268
groupmod	更改组识别码或名称	268
halt	关闭系统	282
hostname	查询及设置主机名称	297
id	显示用户ID, 以及所属组ID	304
kill	删除执行中的程序或工作	345

last	列出当前与过去登录系统的用户相关信息	347
login	登录系统	372
logname	显示用户名称	372
logout	退出系统	373
logrotate	管理记录文件	374
lsb_release	显示Linux发行版本的相关信息	395
newgrp	登录另一个组	523
nice	设置优先级	523
procinfo	显示系统状态	578
ps	报告程序状况	582
pstree	以树状图显示程序	590
reboot	重新开机	614
renice	调整优先级	615
rlogin	远程登录	627
rsh	远程登录的shell	645
runlevel	显示当前系统的执行等级	646
rwho	查看系统用户	648
screen	多重窗口管理程序	651
shutdown	系统关机命令	676
su	变更用户身份	706
sudo	以其他身份来执行命令	710
suspend	暂停执行shell	713
swatch	系统监控程序	718
tload	显示系统负载状况	752
top	显示、管理执行中的程序	754
uname	显示系统信息	781
useradd	建立用户账号	796
userconf	用户账号设置程序	799
userdel	删除用户账号	800
usermod	修改用户账号	801
vlock	锁住虚拟控制台 (console)	826
w	显示当前登录系统的用户信息	828
who	显示当前登录系统的用户信息	841

whoami	显示用户名称	844
whois	查找并显示指定网址的相关信息	845

文件管理

命令名称	功能说明	页码
chattr	改变文件属性	52
chgrp	更改文件或目录的所属组	56
chmod	更改文件或目录的权限	62
chown	更改文件或目录的拥有者或所属组	66
cksum	检查文件的CRC (Cyclic Redundancy Checking) 是否正确	70
cmp	比较两个文件是否有差异	74
cp	复制文件或目录	85
diff	比较文件的差异	130
diffstat	根据diff的比较结果, 显示统计数字	139
file	识别文件类型	215
find	查找文件或目录	219
indent	调整C源代码文件的格式	309
ln	连接文件或目录	365
locate	查找文件	370
lsattr	显示文件属性	394
lsattr	更改或显示MS-DOS文件的属性	419
mc	菜单式的文件管理程序	421
mcopy	复制MS-DOS文件	425
mdel	删除MS-DOS文件	430
mdir	显示MS-DOS目录	431
mktemp	建立临时文件	469
mmove	在MS-DOS文件系统中, 移动文件或目录, 或更改名称	473
mren	更改MS-DOS文件或目录的名称, 或移动文件或目录	493
mshowfat	显示MS-DOS文件在FAT中的记录	494
mtools	显示mtools支持的命令	494
mtoolstest	测试并显示mtools的相关设置	496
mv	移动或更名现有的文件或目录	503

paste	合并文件的行	539
patch	修补文件	541
rcp	远程复制文件或目录	612
rm	删除文件或目录	628
slocate	查找文件或目录	679
split	切割文件	700
tee	读取标准输入的数据，并将其内容输出成文件	741
tmpwatch	删除临时文件	753
touch	改变文件或目录时间	757
umask	指定在建立文件时默认的权限掩码	777
updatedb	更新文件数据库	794
whereis	查找文件	839
which	查找文件	840

备份压缩

命令名称	功能说明	页码
ar	建立或修改档案文件 (archive)，或是 从档案文件中提取文件	7
bunzip2	.bz2文件的解压缩程序	31
bzip2	.bz2文件的压缩程序	33
bzip2recover	用来修复损坏的.bz2文件	36
compress	压缩或解压文件	81
cpio	备份文件	90
dump	备份文件系统	156
gunzip	解压文件	272
gzexe	压缩执行文件	275
gzip	压缩文件	276
lha	压缩或解压文件	356
restore	还原 (Restore) 由转储 (Dump) 操作备份 下来的文件或整个文件系统 (一个分区)	620
tar	备份文件	728
unarj	解压缩.arj文件	783
uncompress	解压缩.Z文件	785
unzip	解压缩zip文件	790

zip	压缩文件	874
zipinfo	列出压缩文件信息	879

磁盘管理

命令名称	功能说明	页码
cd	切换目录	44
df	显示磁盘的文件系统与使用情形	125
dirs	显示目录记录	146
du	显示目录或文件的大小	149
edquota	编辑用户或组的quota	169
eject	退出抽取式设备	175
ln	连接目录内容	369
ls	列出目录内容	384
mcd	在MS-DOS文件系统中切换工作目录	425
mdeltree	删除MS-DOS目录	430
mdu	显示MS-DOS目录所占用的磁盘空间	433
mkdir	建立目录	448
mkraid	初始化磁盘阵列	466
mlabel	显示或设置MS-DOS磁盘的标签名称	470
mmd	在MS-DOS文件系统中建立目录	471
mmount	加载MS-DOS文件系统	472
mount	加载文件系统	479
mrd	删除MS-DOS文件系统中的目录	492
mzip	Zip/Jaz磁盘控制命令	505
popd	删除目录堆栈中的记录	566
pushd	加入目录堆栈中的记录	594
pwd	显示工作目录	597
quota	显示磁盘已使用的空间与限制	599
quotacheck	检查磁盘的使用空间与限制	602
quotaoff	关闭磁盘空间限制	605
quotaon	打开磁盘空间限制	607
raidstart	启动磁盘阵列	609

raidstop	停止运作中的磁盘阵列	610
repquota	检查磁盘空间限制的状态	617
rmdir	删除目录	629
rmt	远程磁带传输协议模块	631
stat	显示 inode 内容	704
tree	以树状图列出目录的内容	765
umount	卸除文件系统	778

磁盘维护

命令名称	功能说明	页码
badblocks	检查磁盘设备中损坏的区块	25
cfdisk	硬盘分区	48
e2fsck	检查ext2文件系统的正确性	160
ext2ed	ext2文件系统编辑程序	191
fdisk	磁盘分区	202
fsck	检查文件系统并尝试修复错误	240
fsck.ext3	检查文件系统并尝试修复错误	242
fsck.minix	检查文件系统并尝试修复错误	248
hdparm	显示与设置硬盘的参数	283
losetup	设置循环设备	376
mbadblocks	检查MS-DOS文件系统的磁盘是否有损坏的扇区	420
mformat	对MS-DOS文件系统的磁盘进行格式化	439
mkbootdisk	建立当前系统的启动盘	447
mkdosfs	建立MS-DOS文件系统	450
mke2fs	建立ext2文件系统	452
mkfs	建立各种文件系统	455
mkfs.ext3	与mke2fs相同	457
mkfs.minix	建立Minix文件系统	457
mkfs.msdos	与mkdosfs相同	458
mkinitrd	建立要载入ramdisk的映像文件	459
mkisofs	建立ISO 9660映像文件	460
mkswap	设置交换区 (Swap area)	467

mpartition	建立或删除MS-DOS的分区	491
sfdisk	磁盘分区工具程序	671
swapoff	关闭系统交换区 (Swap area)	715
swapon	启动系统的交换区 (Swap area)	716
symlinks	维护符号链接的工具程序	719
sync	将内存缓冲区内的数据写入磁盘	722

网络通信

命令名称	功能说明	页码
apachectl	可用来控制Apache HTTP服务器的程序	4
arp	管理系统中的ARP高速缓存	12
arpwatch	监听网络上ARP的记录	14
cu	连接另一个系统主机	107
dip	IP拨号连接	140
efax	收发传真	171
host	DNS查询工具	295
httpd	Apache HTTP服务器程序	299
ifconfig	显示或设置网络设备	306
iptables	包过滤功能和NAT的管理工具	322
lynx	浏览互联网 (World Wide Web)	398
mesg	设置终端的写入权限	434
mingetty	精简版的getty	442
minicom	数据机通信程序	443
nc	连接与监听TCP/UDP通信端口	506
netconfig	设置网络环境	517
netstat	显示网络状态	518
ping	侦测主机	558
pppd	提供PPP连接常驻服务	567
pppstats	显示PPP连接状态	571
route	管理与显示路由表	632
samba	Samba 服务器控制	649
setserial	设置或显示串行端口的相关信息	666

shapecfg	限制网络设备的流量	675
smbclient	可存取SMB/CIFS服务器的客户端程序	685
smbd	Samba服务器程序	692
statserial	显示串行端口状态	705
talk	与其他用户交谈	726
tcpdump	转储网络传输数据	738
telnet	远程登录	744
testparm	测试Samba的设置是否正确无误	746
traceroute	显示包到主机间的路径	762
tty	显示终端连接标准输入设备的文件名称	772
uulog	显示UUCP记录文件	809
uuname	显示全部的UUCP远程主机	810
uustat	显示UUCP当前的状态	812
uux	在远程的UUCP主机上执行命令	815
wall	传送信息	831
wget	从互联网 (WWW) 下载文件	833
write	传送信息	846
ytalk	与其他用户对话	869

电子邮件新闻组

命令名称	功能说明	页码
ctlinnd	设置INN新闻组服务器	106
fetchmail	接收电子邮件	206
getlist	下载新闻组列表	257
inncheck	检查inn相关设置文件的语法是否正确	318
mail	E-mail管理程序	408
mailq	显示待发送邮件的列表	411
metamail	负责处理非文本E-mail的程序	435
mutt	E-mail管理程序	498
nntpget	下载新闻组文章	525
pine	收发电子邮件、浏览新闻组	553
slrn	新闻组阅读程序	682

文件传输

命令名称	功能说明	页码
ftp	传输文件	250
ftpcount	显示当前以FTP登录的用户人数	253
ftpsht	在指定的时间关闭FTP服务器	254
ftpwho	显示当前所有以FTP登录的用户信息	255
ncftp	传输文件	507
ncftpget	下载文件	510
ncftpput	上传文件	513
tftp	传输文件	747
uucico	UUCP文件传输服务程序	803
uucp	在UNIX系统之间传送文件	805
uupick	处理传送进来的文件	811
uuto	将文件传送至远程的 UUCP 主机	815

文本编辑

命令名称	功能说明	页码
col	过滤控制字符	76
colrm	滤掉指定的列	77
comm	比较两个已排序的文件	78
csplit	分割文件	103
dd	读取、转换并输出数据	119
ed	文本编辑器	167
egrep	查找文件里符合条件的字符串	174
ex	在 Ex 模式下启动 vim 文本编辑器	188
fgrep	查找文件里符合条件的字符串	215
fmt	编排文本文件	234
fold	限制文件行宽	236
grep	查找文件里符合条件的字符串	260
ispell	拼写检查程序	325
jed	编辑文本文件	329
joe	编辑文本文件	334

join	将两个文件中，指定列内容相同的行连接起来	341
look	查询单词	375
mtype	显示MS-DOS文件的内容	498
pico	编辑文本文件	547
rgrep	递归查找文件里符合条件的字符串	621
sed	利用script来处理文本文件	657
sort	将文本文件内容加以排序	694
spell	拼字检查程序	700
tr	转换字符	759
uniq	检查及删除文本文件中重复出现的行列	786
vi	编辑文本文件	818
vim	编辑文本文件	820
wc	计算字数	832

输出操作

命令名称	功能说明	页码
cat	连接多个文件，并将它们的内容输出到标准输出设备	40
cut	指定欲显示的文件内容，输出到标准输出设备	109
enscript	将文本文件转为 PostScript	178
head	输出文件内容的最前面部分	291
lpc	控制打印机	378
lpd	提供打印机排队常驻服务	379
lpq	显示打印操作	380
lpr	打印文件	382
lprm	删除打印工作	383
mpage	在PostScript打印机上，将许多页面合并成一页来打印	485
od	输出文件内容	530
pr	将文件格式化编排，以便打印	573
tac	连接多个文件，并将其内容反序输出到标准输出设备	723
tail	输出文件内容的最后部分	724
tunelp	改变打印设备的参数	773
zcat	连接多个压缩文件，并将它们的内容输出到标准输出设备	873

X Window

命令名称	功能说明	页码
startx	启动X Window	703
ttmkfdir	建立TTF字体的索引文件	770
xauth	编辑修改X server授权信息	848
xf86config	设置X Window	850
xfs	提供X Window字体服务器	851
xhost	控制存取X server的主机	853
xlsatoms	列出X Server定义的成分	855
xlsclients	列出显示器中的客户端应用程序	856
xlsfonts	列出X Server使用的字体	858
xset	设置X Window的使用偏好	865

格式转换

命令名称	功能说明	页码
cmuwmtopbm	转换位图文件	76
convdate	转换日期时间	83
gemptopbm	转换图像文件	257
giftopnm	转换图像文件	259
gouldtoppm	转换扫描文件	260
lispmtopgm	转换图像文件	364
pcxtoppm	转换图像文件	545
pfbtops	转换字体文件	546
picttoppm	转换图像文件	552
pjtoppm	转换打印文件	561
qrttoppm	转换QRT输出文件	599
rasttopnm	转换图像文件	611
sox	音效文件转换程序	697
tgatoppm	转换图像文件	749
tifftopnm	转换图像文件	749
uudecode	将uuencode所产生的编码文件转换回原来的格式	807
uuencode	将文件转换为ASCII编码的文件	808

yuvsplittoppm	转换视频文件	871
yuvtoppm	转换视频文件	872

其 他

命令名称	功能说明	页码
at	在指定的时间执行命令	16
atq	显示待执行的工作	18
atrm	删除待执行的工作	19
batch	在系统负载许可时, 立即执行批处理命令	27
bg	将程序放到后台执行	29
cal	显示月历	38
cdp	播放音乐CD	45
clear	清除画面	71
echo	显示文本	165
exec	shell执行指定的命令后即交出控制权	188
fc	修改命令, 且执行该命令	200
fg	将程序或命令切换至前台执行	214
help	显示shell内建命令的说明	293
history	列出之前用过的命令	293
info	显示说明	315
jobs	显示正在后台执行的工作	332
less	显示文件内容	350
make	编译系统内核或模块	412
man	在线查询命令	414
manpath	显示说明文件的查找路径	418
md5sum	计算与检查MD5函数值	428
more	使文件能逐页地显示	477
nohup	可在退出后继续执行程序	526
open	打开虚拟终端执行命令	535
playmidi	播放音乐文件	562
sleep	暂停执行命令	678
sum	计算文件的校验和与区块数	712

常用指数索引



alias3
bg29
cat40
cd44
chmod62
clear71
cp85
df125
ed167
exit189
fg214
find219
ftp.....250
grep260
gunzip272
gzip276
joe334
kill345
last347
less350
ln365
login.....372
logout373
ls384
mail408

man414
mkdir448
more477
mount479
mv503
ncftp.....507
nice523
passwd537
pico547
pine553
ping558
ps582
pwd597
quota599
renice615
rm628
rmdir629
route632
shutdown676
startx703
tar728
telnet744
umount.....778
unzip790
vi818
vim820
w828

whereis.....839
which840
write846
zip874



adduser.....1
ar7
arp12
batch27
bunzip231
bzip233
cal38
cfdisk48
chgrp56
chown.....66
cksum.....70
cmp.....74
comm78
compress81
cpio90
crontab100
cu107
cut109
date112

dd	119	lha	356	rlogin	627
diff	130	lilo	361	rpm	634
dmesg	148	locate	370	set	660
du	149	lpq	380	setenv	665
echo	165	lpr	382	setup	670
edquota	169	lprm	383	smbclient	685
export	190	lsmod	397	split	700
fdisk	202	make	412	su	706
file	215	mesg	434	swapoff	715
finger	231	mingetty	442	swapon	716
free	237	minicom	443	symlinks	719
fsck	240	mkbootdisk	447	talk	726
ftpcount	253	mkfs	455	top	754
ftpwho	255	mkinitrd	459	touch	757
groupadd	266	mkisofs	460	umask	777
groupdel	268	mkswap	467	unalias	780
groupmod	268	modprobe	475	uname	781
grup-install	271	netconfig	517	unarj	783
halt	282	netstat	518	uncompress	785
help	293	nohup	526	useradd	796
history	293	patch	541	userdel	800
host	295	procinfo	578	usermod	801
id	304	quotacheck	602	uudecode	807
ifconfig	306	quotaoff	605	uuencode	808
info	315	quotaon	607	wall	831
insmod	321	reboot	614	wget	833
iptables	322	repquota	617	who	841

whoami844
 whois845
 xf86config850



apachectl 4
 apmd.....6
 arpwatch14
 at.....16
 atq18
 atrm19
 aumix20
 badblocks25
 bind30
 bzip2recover36
 cdp45
 chattr52
 chfn55
 chkconfig59
 chroot.....68
 chsh69
 clock71
 cmuwmtopbm76
 col76
 colrm77
 convdate.....83

csplit103
 ctilnd106
 declare122
 depmod124
 diffstat139
 dip140
 dircolors143
 dirs146
 dump156
 e2fsck160
 efax171
 egrep174
 eject175
 enable176
 encrypt178
 eval187
 ex188
 exec188
 ext2ed191
 fbset195
 fc200
 fetchmail206
 fgrep.....215
 fmt234
 fold236
 fsck.ext3242
 fsck.minix248
 ftpshut254

gemtopbm257
 getlist257
 giftopnm259
 gouldtoppm260
 grpconv269
 grpunconv270
 gzexe275
 hdparm283
 head291
 hostname297
 httpd299
 hwclock302
 indent309
 inncheck318
 ispell325
 jed329
 jobs332
 join341
 lispmtopgm364
 lndir369
 logname372
 logrotate374
 look375
 losetup376
 lpc378
 lpd379
 lsattr394
 lsb_release395

lynx	398	mouseconfig	484	pr	573
mailq	411	mpage	485	pstree	590
manpath	418	mpartition	491	pushd	594
mattrib	419	mrd	492	pwconv.....	596
mbadblocks	420	mren	493	pwunconv	598
mc.....	421	mshowfat	494	qrttoppm	599
mcd	425	mtools	494	raidstart	609
mcopy	425	mtoolstest	496	raidstop	610
md5sum	428	mtype	498	rasttopnm	611
mdel	430	mutt	498	rcp	612
mdeltree	430	mzip	505	rdate	613
mdir	431	nc	506	resize	619
mdu	433	ncftpget	510	restore	620
metamail	435	ncftpput	513	rgrep.....	621
mformat	439	newgrp	523	rmmod	630
minfo	440	nntpget	525	rmt	631
mkdosfs	450	ntsysv	527	rsh	645
mke2fs	452	od	530	runlevel	646
mkfs.ext3	457	open	535	rwho	648
mkfs.minix	457	paste	539	samba	649
mkfs.msdos	458	pcxtoppm	545	screen	651
mkraid	466	pfbtops	546	sed	657
mktemp	469	picttoppm	552	setserial	666
mlabel	470	pjtoppm	561	sfdisk	671
mmd	471	playmidi	562	shapecfg	675
mmount	472	popd	566	sleep.....	678
mmove	473	pppd	567	slocate	679
modinfo	473	pppstats	571	slrn	682

smbd.....	692	tree	765	xset	865
sndconfig	693	ttmkfdir	770	yes	868
sort	694	tty	772	ytalk	869
sox	697	tunelp	773	yuvsplittoppm	871
spell	700	ulimit	775	yuvtoppm	872
stat	704	uniq	786	zcat	873
statserial	705	unset.....	789	zipinfo	879
sudo	710	updatedb	794		
sum	712	userconf	799		
suspend	713	uucico	803		
SVGATextMode	714	uucp	805		
swatch	718	uulog	809		
sync	722	uuname.....	810		
tac.....	723	uupick	811		
tail	724	uustat	812		
tcpdump	738	uuto	815		
tee.....	741	uux	815		
telinit	742	vlock	826		
testparm	746	wait	830		
tftp	747	wc.....	832		
tgatoppm	749	xauth	848		
tifftopnm	749	xfst	851		
timeconfig	750	xhost.....	853		
tload	752	xlsatoms	855		
tmpwatch	753	xlsclients	856		
tr	759	xlsfonts	858		
traceroute	762	xplaymidi.....	861		

adduser

常用指数:



A

功能说明: 建立用户账号。

语 法: adduser [-mMnr] [-c <备注>] [-d <登录目录>] [-e <有效期限>] [-f <缓冲天数>] [-g <组>] [-G <组>] [-s <shell>] [-u <uid>] [用户账号]或adduser -D [-b<用户目录>] [-e <有效期限>] [-f <缓冲天数>] [-g <组>] [-G 组>] [-s <shell>]

包名称: shadow-utils。

相关命令: groupadd, groupdel, groupmod, id, passwd, userdel, usermod。

补充说明: adduser可用来建立用户账号。账号建好之后,再用passwd设置账号的密码。可用userdel删除账号。此命令在Linux中,其实是 adduser的符号链接。使用adduser命令所建立的账号,实际上存储在/etc/passwd文本文件中。

参 数:

-c <备注>	加上备注文字。备注文字会存储在passwd的备注参数中
-d <登录目录>	指定用户登录时的起始目录
-D	<p>变更默认值。默认值的选项说明如下:</p> <p>-b <用户目录> 在指定的<用户目录>下,建立所有用户的<登录目录>。默认值为/home</p> <p>-e <有效期限> 指定默认账号的有效期限</p> <p>-f <缓冲天数> 指定默认在密码过期多少天后即关闭该账号</p> <p>-g <组> 指定用户默认所属的组</p> <p>-G <组> 指定用户默认所属的附加组</p> <p>-s <shell> 指定使用默认的shell</p>
-e <有效期限>	指定账号的有效期限。<有效期限>的格式为月份/日期/公元年后两位

-f <缓冲天数>	指定在密码过期后多少天即关闭该账号
-g <组>	指定用户所属的组
-G <组>	指定用户所属的附加组
-m	自动建立用户的登录目录
-M	不要自动建立用户的登录目录
-n <组>	不建立以用户名为名的组
-r	建立系统账号
-s <shell>	指定用户登录后所使用的shell
-u <uid>	指定用户ID

范 例:

① 建立名称为alkane的用户账号:

```
# adduser alkane
```

② 建立名称为alkane的用户账号, 并指定登录目录为/home/alkane1:

```
# adduser -d /home/alkane1 alkane
```

③ 建立名称为alkane的用户账号, 并指定账号的有效期限为2004年12月31日:

```
# adduser -e 12/31/04 alkane
```

④ 建立名称为alkane的用户账号, 并指定所属组为users:

```
# adduser -g users alkane
```

⑤ 建立名称为alkane的用户账号, 并指定其登录后所使用的shell为C shell:

```
# adduser -s /bin/csh alkane
```

alias

常用指数:



A

功能说明: 设置命令的别名。

语法: alias[-p][别名][=命令]

包名称: bash。

相关命令: unalias。

补充说明: 用户可利用alias自定义命令的别名。若仅输入alias, 则可列出当前所有的别名设置。alias仅作用于该次登录的作业。若要每次登录时都自动设好别名, 请在.profile (bash) 或.cshrc (tcsh) 中设置命令的别名。

参数:

-p	列出当前所有的别名设置
别名	列出指定的别名设置
别名=命令名称	设置某个命令的别名

范例:

① 列出当前所有的别名设置:

```
# alias
alias cp='cp -i'
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias mc='. /usr/share/mc/bin/mc-wrapper.sh'
...
```


A

② 将mkdir命令的别名设为md:

```
# alias md=mkdir
# alias md ← 显示md的别名设置
alias md='mkdir'
```

③ 将ls-al命令的别名设为dir:

```
# alias dir='ls-al'
↑
指定别名时可加上命令的参数
```

apachectl

常用指数:



(Apache HTTP server control interface)

功能说明: 用来控制Apache HTTP服务器的程序。

语法: apachectl [configtest] [fullstatus] [graceful] [help] [restart] [start] [status] [stop]

包名称: httpd (Red Hat Linux/Fedora Core), apache-conf (Mandrake Linux)。

相关命令: httpd。

补充说明: 各Linux发布版内附的Apache HTTP服务器可能有很小的差异, 因此不一定都会有这个script文件。



参 数:

configtest	检查设置文件中的语法是否正确
fullstatus	显示服务器完整的状态信息
graceful	重新启动Apache服务器。与restart参数的差异在于使用graceful参数时，不会中断原有的连接
help	显示说明信息
restart	重新启动Apache服务器
start	启动Apache服务器
status	显示服务器摘要的状态信息
stop	停止Apache服务器

范 例:

① 启动Apache服务器:

```
# apachectl start
```

```
strting http2 [ok] ← 若是Red Hat Linux或Fedora Core, 并不会出现此信息
```

② 重新启动 Apache 服务器:

```
# apachectl restart
```

③ 重新启动Apache服务器, 但不中断现有的连接:

```
# apachectl graceful
```

④ 停止执行Apache服务器：

```
# apachectl stop
```

```
Shutting down httpd2: [ok] ← Red Hat Linux或Fedora  
Core不会出现此信息
```

⑤ 检查Apache服务器设置文件的语法：

```
# apachectl configtest
```

```
Syntax OK ← 检查结果正常
```

apmd

常用指数：



(advanced power management (APM)daemon)

功能说明：高级电源管理服务程序。

语法：apmd [-quvVW] [-p <百分比变化量>] [-w <百分比值>]

包名称：apmd。

补充说明：apmd负责BIOS高级电源管理（APM）相关的记录、警告与管理工作。

参数：

-p<百分比变化量> 或--percentage<百分比变化量>	当电量变化的幅度超出设置的百分比变化量时，即记录事件。百分比变化量的默认值为5，若设置值超过100，则关闭此功能
-q	取消-w参数的功能
-u 或 --utc	将BIOS时钟设为UTC（GMT，格林威治标准时间），以便从挂起模式（suspend mode）恢复时，将-u参数传送到clock或hwclock（设置硬件时钟）程序

-v 或 --verbose	记录所有的APM事件
V 或 --version	显示版本信息
-w <百分比值>或 --warn<百分比值>	当电池不在充电状态时，且电池电量低于设置的百分比值，则在syslog(2)的ALERT层级记录警告信息。百分比值的默认值为10，若设置值为0，则关闭此功能
-W 或 --wall	发出警告信息给所有登录者

范 例：

① 当电池的电量低于20%时，警告所有的用户：

```
# apmd -w 20 -W
```

② 取消电量低于某个百分比就发出警告的功能：

```
# apmd -q
```

③ 当电量变化幅度超出10%时，记录事件：

```
# apmd -p 10
```

ar

常用指数：



功能说明： 建立或修改档案文件 (archive)，或是从档案文件中提取文件。

语 法： ar [-dmpqrtx] [cfosSuvV] [a <成员文件>] [b <成员文件>] [i <成员文件>][档案文件][成员文件]

包 名 称： binutils。

补充说明： ar可以集合许多文件（称为成员文件），成为单一的档案文件。在档案文件中，所有成员文件都保留原来的属性与权限。ar也可以为档案文件建立符号表 (Symbol table)，功能如同文件的索引，以便直接调用档案文件中的成员文件。

第1类参数（[-dmpqrtx]）为命令参数，指定ar所要执行的工作。第2类参数（[abcfiosSuvV]）为可选参数，用来指定工作的选项。

参 数：

命令参数	
-d	删除档案文件中的成员文件
-m	变更成员文件在档案文件中的次序。如果没指定第2个参数，则-m参数会将指定的成员文件移动到档案文件的最后面
-p	显示档案文件中的成员文件的内容。如果没指定[成员文件]参数，则显示档案文件中全部成员文件的内容
-q	将文件附加在档案文件末尾，不用检查档案文件中是否有重复的成员文件
-r	将文件插入档案文件中。若文件重复，则取代原有的成员文件
-t	显示档案文件中所包含的文件
-x	自档案文件中取出成员文件（档案文件中仍留有该成员文件）
可选参数	
a <成员文件>	将文件插入档案文件中指定的成员文件之后
b <成员文件>	将文件插入档案文件中指定的成员文件之前
c	建立档案文件
f	为避免过长的文件名不兼容于其他系统的ar命令，因此可利用此参数，截掉要放入档案文件中过长的成员文件名称
i <成员文件>	将文件插入档案文件中指定的成员文件之前（与b的作用相同）
o	保留档案文件中文件的日期。若不设置此参数，文件自档案文件中抽取出来时，文件日期将更改为现在的日期
s	若档案文件中包含了对象模块，可利用此参数建立档案文件的符号表。档案文件建好符号表后，其他程序便可调用档案文件中的对象
S	不产生符号表。若档案文件很庞大时，可利用此参数加快程序的执行速度
u	只将日期较新的文件插入档案文件中
v	程序执行时显示详细的信息
V	显示版本信息

范 例:

① 将file1、file2与file3文件档案为arfile，执行时显示详细信息；然后列出其所包含的文件：

```
# ar -rv arfile file1 file2 file3 ← 建立arfile档案文件
a - file1
a - file2
a - file3
# ar -t arfile ← 显示arfile中所包含的文件
file1
file2
file3
```

② 接上例，列出档案文件 arfile 内所有成员文件的内容：

```
# ar -p arfile
file1 content
line1 in file1
line2 in file1
file2 content
line1 in file2
line2 in file2
file3 content
line1 in file3
line2 in file3
```

③ 以较详细的方式列出档案文件arfile的内容：

```
# ar -pv arfile
```

```
<member file1>
file1 content
line1 in file1
line2 in file1
```



A

<member file2>

file2 content
line1 in file2
line2 in file2

<member file3>

file3 content
line1 in file3
line2 in file3

在文件内容之前，列出档案文件中成员文件的文件名

④ 列出档案文件arfile中file1的内容：

ar -p arfile file1

file1 content
line1 in file1
line2 in file1

⑤ 接上例，将newfile加入档案文件中file2之前：

ar -rbv file2 arfile newfile

a - newfile

ar -t arfile

file1

newfile ← newfile位于file2之前

file2

file3

⑥ 接上例，将档案文件中的newfile移动到档案文件的最后面：

ar -mv arfile newfile

m - newfile

ar -t arfile

```
file1
file2
file3
newfile          ← newfile 已经在档案文件的最后面
```

⑦ 接上例，将档案文件中的newfile移动到file2之前：

```
# ar -mbv file2 arfile newfile
m - newfile
# ar -t arfile
file1
newfile          ← newfile 已经移动到file2之前
file2
file3
```

⑧ 接上例，将档案文件中的newfile移动到file2之后：

```
# ar -mav file2 arfile newfile
m - newfile
# ar -t arfile
file1
file2
newfile          ← newfile已经移动到file2之后
file3
```

⑨ 接上例，取出档案文件中的newfile，但仍保留这个文件在档案文件中：

```
# ar -xv arfile newfile
x - newfile
# ar -t arfile
file1
file2
newfile          ← 取出的newfile仍保留在档案文件中
file3
# ls -l newfile
-rwxr-xr-x  1 root  root  44 5月 27 14:39 newfile
```

↑
文件的时间为取出的时间

A

⑩ 接上例，取出档案文件中的newfile，但保留取出文件的更改时间：

```
# ar -xov arfile newfile
x - newfile
# ls -l newfile
-rwxr-xr-x 1 root root 44 5月 7 14:32 newfile
```

↑
文件时间为原本在档案文件中的时间

⑪ 接上例，删除档案文件中的newfile：

```
# ar -dv arfile newfile
d - newfile
# ar -t arfile
file1
file2
file3
```

newfile已经从档案文件中删除

⑫ 接上例，将file2加入档案文件的末尾，不用检查是否有重复的成员文件：

```
# ar -qv arfile file2
a - file2
# ar -t arfile
file1
file2
file3
file2
```

arp

常用指数：



功能说明：管理系统中的ARP高速缓存。

语法：arp [-aenv] [-H <硬件类型>] [-i <网络接口>] [-s <IP地址>
<网卡物理地址>] [主机名称]

包名称: net-tools。

相关命令: arpwatch, ifconfig, netstat。

补充说明: ARP命令可以列出、修改、增加与删除kernel中的ARP高速缓存数据。

参 数:

-a	列出所有ARP高速缓存数据
-e	以默认的排版方式显示ARP高速缓存的数据
-H <硬件类型>	根据指定的硬件显示ARP高速缓存的数据，默认为ether
-i <网络接口>	指定要查询的网络接口
-n	以数字模式显示
-s <IP地址> <网卡物理地址>	设置ARP高速缓存IP地址（或主机名称）与网卡物理地址的对应
-v	程序执行时显示详细的信息

范 例:

① 列出arp缓存中所有的数据:

```
# arp -a
? (192.168.0.35) at 00:01:80:0E:64:C8 [ether] on eth0
? (192.168.0.3) at 00:10:B5:3A:92:37 [ether] on eth0
? (192.168.0.132) at 00:00:E8:97:6C:B2 [ether] on eth0
```



这里应该要出现主机名称，系统首先会参考/etc/hosts文件，若查不到对应的名称，则向DNS服务器查询。若仍查不到，则会出现“?”号。

② 接上例，笔者于/etc/hosts中加入下列数据:

```
192.168.0.3          gateway
```

A

接着再次查询：

```
# arp -a
? (192.168.0.35) at 00:01:80:0E:64:C8 [ether] on eth0
gateway (192.168.0.3) at 00:10:B5:3A:92:37 [ether] on eth0
```

已可查询到主机名称

```
? (192.168.0.132) at 00:00:E8:97:6C:B2 [ether] on eth0
```

③ 强迫以IP来显示与网卡物理地址的对应：

```
# arp -n
Address          HWtype HWaddress          Flags Mask Iface
192.168.0.35     ether  00:01:80:0E:64:C8  C           eth0
192.168.0.3      ether  00:10:B5:3A:92:37  C           eth0
```

又恢复以IP方式显示了

...

④ 以手动方式指定主机名称（或IP地址）与网卡物理地址的对应：

```
# arp -s 192.168.0.50 00:01:80:0E:64:C5
```

```
# arp
```

```
Address          HWtype HWaddress          Flags Mask Iface
192.168.0.35     ether  00:01:80:0E:64:C8  C           eth0
gateway          ether  00:10:B5:3A:92:37  C           eth0
192.168.0.50     ether  00:01:80:0E:64:C5  CM          eth0 ←
```

刚才所增加的对应记录

...

arpwatch (ARP watcher)

常用指数：



功能说明： 监听网络上ARP的记录。

语 法： arpwatch [-d] [-f <记录文件>] [-i <接口>] [-r <记录文件>]

包 名 称： arpwatch。

相关命令：arp。

补充说明：ARP (Address Resolution Protocol) 是用来解析IP与网络设备硬件地址的协议。arpwatch可监听区域网络中的ARP包并记录，同时将监听到的变化用E-mail通知。

参 数：

-d	启动排错模式。这样，程序将不会在后台中执行，也不会通过E-mail来报告。而是将变化直接显示在标准错误输出设备(屏幕)
-f <记录文件>	设置存储ARP记录的文件，默认为 /var/arpwatch/arp.dat
-i <接口>	指定监听ARP的接口，默认的接口为eth0 (也就是第1个网卡)
-r <记录文件>	从指定的文件中读取ARP记录，而不是从网络上监听

范 例：

① 执行arpwatch后，若在网络检测到新的ARP记录，即通过E-mail来报告：

```
# arpwatch
```

当检测到ARP记录时，管理员会收到下列内容的E-mail

```
Date: Mon, 26 May 2003 15:07:02 +0800
From: Arpwatch <root@free.flag.com.tw>
To: root@free.flag.com.tw
Subject: new station
```

```
hostname: <unknown>
ip address: 192.168.0.6
ethernet address: 0:80:c8:4a:d0:34
ethernet vendor: D-Link (also Solectek Pocket Adapters, and
...)
```

← 主机名

← IP地址

← 网卡的硬件地址

网卡制造商

← 发生的时间

a

A

- ② 利用排错模式执行arpwatch，若在网络上检测到新的ARP记录，将结果输出到屏幕：

```
# arpwatch -d
```

```
From: arpwatch (Arpwatch)
```

```
To: root
```

```
Subject: new station
```

```
hostname: <unknown>
```

```
ip address: 192.168.0.42
```


```
ethernet address: 0:1:80:d:a5:a5
```

```
ethernet vendor: AOpen, Inc.
```

```
timestamp: Friday, May 30, 2003 19:47:00 +0800
```

```
...
```

at

常用指数：

功能说明：在指定的时间执行命令。

语法：at [-lmV] [-d<工作编号>] [-f<文件>] [-q<队列>] [日期/时间]

包名称：at。

相关命令：atq, atrm, batch, crontab。

补充说明：at可从标准输入设备（键盘）或从文件读取命令，然后在指定的时间执行这些命令。

参数：

-d <工作编号>	将待执行的工作删除，与直接执行atrm命令作用相同
-f <文件>	从文件读取要执行的命令，而不是从标准输入设备中读取
-l	显示待执行的工作，与直接执行atq命令作用相同
-m	工作完成后，将结果以E-mail返回

-q <队列>	使用指定的队列。队列是以单一英文字母来表示，其顺序是从小写的a到z，再从大写的A到Z。顺序越后面，表示工作的优先级越高。at的默认队列为a队列
-V	显示版本信息

范 例：

① 在2004年8月28日20:25分提醒用户保存手边的工作，并在5分钟后关闭系统：

```
# at 20:25 08/28/04 ← 设置在2004/8/28 20:25
                        分要执行的计划工作

at> wall "Hello everybody, please store your jobs!" ←
                                                    传送广播信息

at> shutdown -h +5 ← 指定5分钟后关闭系统
at> <EOT> ← 输入完毕后按 Ctrl + D 键结束计划工作设置
job 3 at 2004-08-28 20:25 ← 已在2004/8/28 20:25分设置
                               了工作编号为3的计划工作
```

② 在20:50时使用队列d执行job_schedule文件中所指定的命令：

```
# at -f job_schedule -q d 20:50
job 5 at 2003-05-30 20:50
```

③ 显示队列中所有待执行的工作：

```
# at -l
3      2003-05-30 20:28  a root
5      2003-05-30 20:50  d root
7      2003-05-30 19:33  c root
↑      |      |      |
工作编号  预定执行的时间  所使用的队列
```

④ 删除队列中编号第5号工作：

```
# at -d 5
```

atq (at queue)

常用指数:

**功能说明:** 显示待执行的工作。**语 法:** atq [-V] [-q <队列>]**包 名 称:** at。**相关命令:** at, atrm, batch。**补充说明:** atq命令可显示当前待执行的工作，除非用户是系统管理员，才能显示所有人的工作，否则只能显示自己待执行的工作。实际上atq命令是指向at命令的符号链接。**参 数:**

-q <队列>	使用指定的队列。队列是以单一英文字母来表示，其顺序是从小写的a到z，再从大写的A到Z。顺序越后面，表示工作的优先级越高。at的默认队列为a队列
-V	显示版本信息

范 例:

① 显示所有队列中待执行的工作:

atq

4 2003-05-30 23:50 a root

7 2003-05-30 19:33 c root



工作编号



预定的执行时间




所使用的队列

② 显示a队列待执行的工作：

```
# atq -q a
4          2003-05-30 23:50 a root
```

atrm (at remove)

常用指数：

功能说明：删除待执行的工作。

语 法：atrm [-V] [工作编号]

包名称：at。

相关命令：at, atq, batch。

补充说明：atrm可删除待执行的工作。实际上atrm命令是指向at命令的符号链接。

参 数：

-V	显示版本信息
----	--------

范 例：

删除队列中的工作：

```
# atq
4          2003-05-30 23:50 a root
7          2003-05-30 19:33 c root
# atrm 7
# atq
4          2003-05-30 23:50 a root
```

← 显示队列中的所有的工作

← 删除第7号的工作

← 再次显示队列中的工作

← 第7号工作已经被成功地删除

aumix (audio mixer)

常用指数:

**功能说明:** 设置音效设备。**语法:** aumix [-123bcilmoprstvwWx] [(+/-)强度] [PqR] [-dfhILqS]**包名称:** aumix。**相关命令:** sndconfig。**补充说明:** 设置各项音效设备的信号强度，以及指定播放与录音的设备。

[-123bcilmoprstvwX]为频道参数，用来指定设备的频道，[PqR]可用来指定播放或录音设备；[dfhILqS]则为命令参数。若不加任何参数，aumix会显示简单的图形界面供调整设置。

参数:

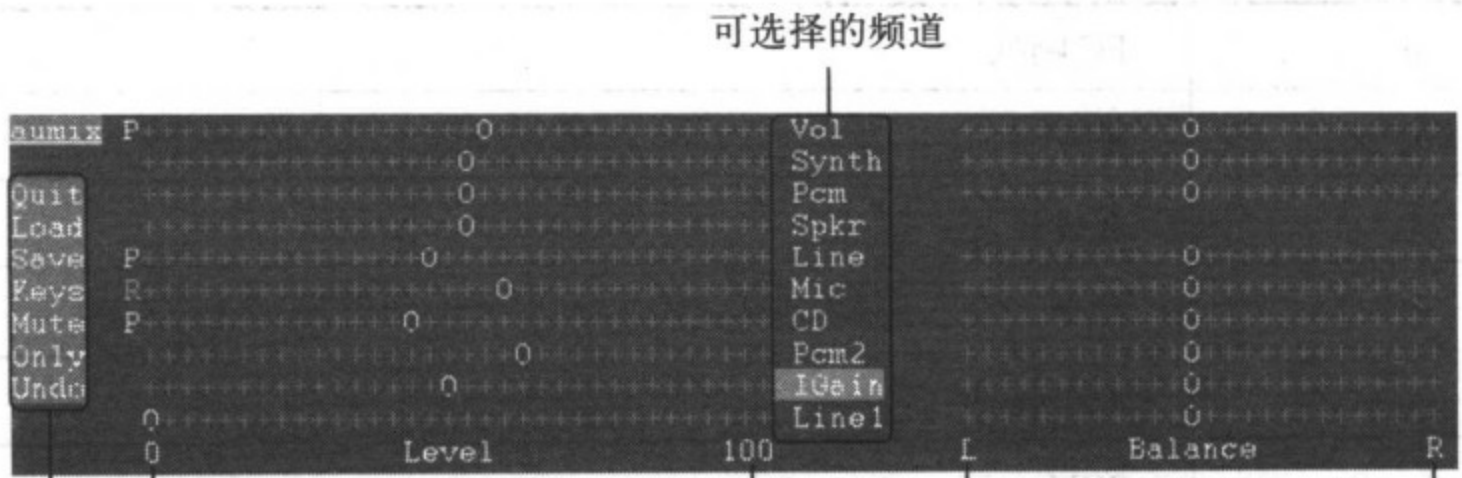
频道参数	
-1	输入信号线1
-2	输入信号线2
-3	输入信号线3
-b	低音
-c	CD
-i	输入信号强度
-l	输入信号线
-m	麦克风
-o	输出信号强度

-p	PC 喇叭
-r	录音
-s	合成器
-t	高音
-v	主音量
-w	PCM
-x	混音器
(+/-) 强度	出现 (+/-) 时, 表示在原有的强度上加减指定值。如果没使用 (+/-), 则直接将强度设为指定值
指定音效设备	
P	指定播放设备
q	显示频道设置
R	指定录音设备
命令参数	
-d	指定音效设备的名称
-f	指定存储或载入设置的文件
-h	在使用时显示信息
-I	以图形界面方式来执行aumix
-L	从\$HOME/.aumixrc载入设置文件, 如果没这个文件则从/etc/aumixrc载入
-q	显示所有音效设备的设置值
-S	将设置值存储到\$HOME/.aumixrc

范 例:

- ① 执行aumix会出现以下的画面, 可用稍后所介绍的操作命令设置音效设备 (执行aumix -I也有同样的效果):

```
# aumix
```



常用的功能，可按
每个功能第1个字
母执行

信号强度

左右喇叭
的平衡度

② 显示当前所有音效设备的设置值：

```
# aumix -q
vol 56, 56, P
```

表示左右喇叭的平衡度

```
synth 53, 53
```

```
pcm 51, 51
```

```
speaker 51, 51
```

```
line 45, 45, P
```

当前设为播放设备

```
mic 57, 57, R
```

当前设为录音设备

```
cd 42, 42, P
```

```
pcm2 62, 62
```

```
igain 49, 49
```

```
line1 0, 0
```



③ 将CD音量调整为50，并显示其设置值：

```
# aumix -c 50 -c q
```

```
cd 50, 50, P
```

显示CD频道的设置值

将CD音量调整为50

← 这是CD频道的设置值

④ 将主音量设为85，并显示其设置值：

```
# aumix -v 85 -q
```

```
vol 85, 85
```

⑤ 将主音量频道的强度加10，并显示其设置值：

```
# aumix -v +10 q
```

```
vol 95, 95
```

⑥ 减弱主音量频道的强度，并显示其设置值：

```
# aumix -v - q
```

```
vol 94, 94
```

⑦ 将输入信号线设成录音设备，并显示其设置值：

```
# aumix -l R -l q
```

```
line 75, 75, R
```

⑧ 将当前的设置值存储到音效设置文件中：

```
# aumix -S
```

⑨ 从\$HOME/.aumixrc或/etc/aumixrc载入音效设置文件：



A

```
# aumix -L
vol set to 83, 83, P
synth set to 53, 53
pcm set to 51, 51
speaker set to 51, 51
line set to 45, 45, R
mic set to 57, 57, P
cd set to 50, 50, P
pcm2 set to 62, 62
igain set to 49, 49
linel set to 0, 0
```

操作命令：执行aumix之后，常用的操作命令说明如下：

Page Up 或 ↑	往上选择频道
Page Down 或 ↓	往下选择频道
Tab、Enter、<、>、) 或 .	切换信号强度或左右喇叭平衡度
+ 或 →	以3%的比例加强信号强度或加强右边喇叭的强度
- 或 ←	以3%的比例减少信号强度或加强左边喇叭的强度
I	将信号强度或右边喇叭的强度调至最大
J	将信号强度调至最小或左边喇叭的强度调至最大
1 ~ 9	直接指定信号强度或左右喇叭的平衡度由10%到90%
I	将左右喇叭的平衡度置中
M	静音或取消静音
O	除了当前频道外，其余频道的信号强度均调至最小（静音）
U	取消静音功能
空白键	切换设备的播放或录音功能（如有部分音效设备支持此功能，此参数才有效）
L	载入用户根目录下的.aumixrc设置文件
S	存储当前设置到设置文件
K	显示可用的按键
Q	结束程序

badblocks

常用指数:



B

功能说明: 检查磁盘设备中损坏的区块。

语法: badblocks [-svw] [-b <区块大小>] [-c <磁盘区块数>]
[-i <输入文件>] [-o <输出文件>] [磁盘设备] [磁盘区块数]
[起始区块][结束区块]

包名称: e2fsprogs。

相关命令: e2fsck, fsck.ext2, mbadblocks, mke2fs, sfdisk。

补充说明: 执行命令时须指定所要检查的磁盘设备，以及此设备的磁盘区块数。

参数:

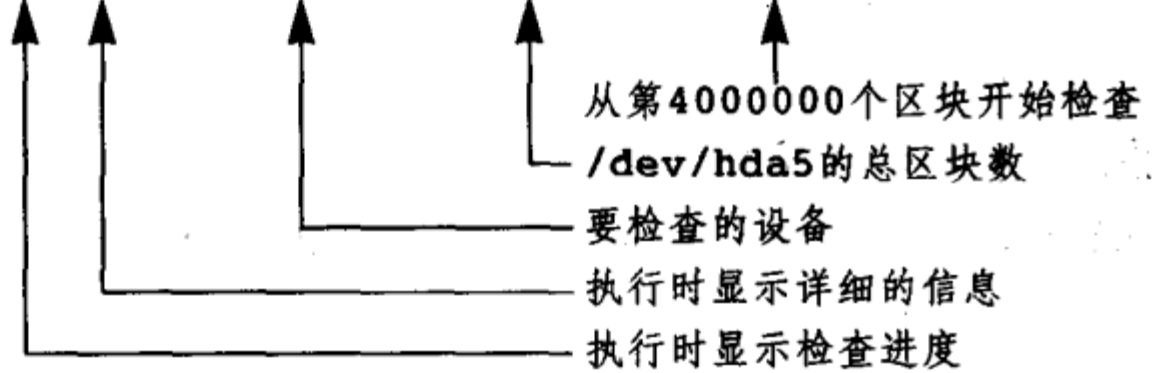
-b <区块大小>	指定磁盘的区块大小，单位为字节
-c <磁盘区块数>	一次检查几个区块，默认为16个
-i <输入文件>	由文件中读取已知的损坏区块，当检查时会忽略这些区块
-o <输出文件>	将检查的结果写入指定的输出文件。此输出文件可供 mke2fs 等程序在格式化磁盘时使用。若不加此参数，则默认会将检查结果显示在屏幕上
-s	在检查时显示进度
-v	执行时显示详细的信息
-w	在检查时，执行写入测试（也就是将一小段数据写入区块中，然后再读出作比较，看看是否一致）。请注意，执行此参数时，会破坏磁盘中原有的数据
[磁盘设备]	指定要检查的磁盘设备
[磁盘区块数]	指定磁盘设备的区块总数
[起始区块]	指定要从哪个区块开始检查。若不指定此参数，则默认会从第0个区块开始检查
[结束区块]	指定检查结束的区块。若不指定，默认会检查到最后一个区块

范 例:

- ① 检查/dev/hda5磁盘分区，并从第100000区块开始检查（先用sfdisk测出分区的总区块数）:

```
# sfdisk -s /dev/hda5
4305388
```

```
# badblocks -s -v /dev/hda5 4305388 4000000
```



```
Checking for bad blocks in read-only mode
From block 4000000 to 4305388
Checking for bad blocks (read-only test): done
Pass completed, 0 bad blocks found.
```

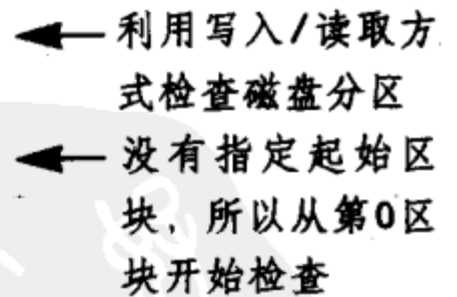
- ② 从第4000000区块开始检查/dev/hda5磁盘分区，并将结果输出到hda5_badblocks文件中:

```
# badblocks -o hda5_badblocks /dev/hda5 4305388 4000000
```

- ③ 利用写入测试的方式检查/dev/fd0磁盘分区，执行时显示详细的信息:

```
# badblocks -w -v /dev/fd0 1440
```

```
Checking for bad blocks in read-write mode
From block 0 to 1440
```



Writing pattern 0xaaaaaaaa: done ← 在磁盘中写入0xaaaaaaaa字符串

Reading and comparing: done ← 读取并比较数据的正确性

Writing pattern 0x55555555: done ← 在磁盘中写入0x55555555字符串

Reading and comparing: done

Writing pattern 0xffffffff: done ← 在磁盘中写入0xffffffff字符串

Reading and comparing: done

Writing pattern 0x00000000: done ← 在磁盘中写入0x00000000字符串

Reading and comparing: done

Pass completed, 0 bad blocks found. ← 检查结果, 没有发现损坏的区块

B

batch

常用指数:



功能说明: 在系统负载许可时, 立即执行批处理命令。

语法: batch [-mV] [-f <文件>] [-q <队列>]

包名称: at。

相关命令: at, atq, atrm, top。

补充说明: 执行batch后, 若系统平均负载低于0.8 (可用top命令显示系统平均负载), 即执行批处理命令。batch在Red Hat Linux、Fedora Core与Mandrake Linux中为script文件。

参数:

-f <文件>	从文件读取要执行的命令, 而不是从标准输入中读取
-m	工作完成后, 将结果以E-mail返回
-q <队列>	使用指定的队列。队列是以单一英文字母来表示, 其顺序是从小写的a到z, 再从大写的A到Z。越后面的顺序, 表示工作的优先权越高。batch的默认队列为a队列
-V	显示版本信息

范 例:

- ① 利用batch命令打印mydoc1.txt与mydoc2.txt两个文件，batch所要执行的命令由键盘输入:

```
# batch
warning: commands will be executed using (in order) a)
$SHELL b) login shell c) /bin/sh
at> lpr mydoc1.txt
at> lpr mydoc2.txt
at> <EOT>
job 8 at 2003-05-31 16:19
```

← 输入要执行的命令
← 按 [] + [] 键

- ② 利用batch命令打印mydoc1.txt与mydoc2.txt两个文件，batch所要执行的命令由文件execfile读入:

```
# cat execfile
lpr mydoc1.txt
lpr mydoc2.txt
# batch -f execfile
```

← 先看一下execfile文件的内容
← 由execfile文件读入命令

- ③ 利用batch命令执行execfile文件中指定的命令，工作完成后，用E-mail将结果返回:

```
# batch -f execfile -m
```

- ④ 利用batch命令执行execfile文件中指定的命令，并使用队列d:

```
# batch -f execfile -q d
```



bg (background)

常用指数: **功能说明:** 将程序放到后台执行。**语法:** bg [工作编号]**包名称:** bash。**相关命令:** fg, jobs, ps。**补充说明:** 将正在执行的工作移到后台执行, 功能与命令后加上“&”相同。如果没指定工作编号, 会将当前的工作移到后台。查询工作编号, 可用jobs命令。若要将工作移回到前台执行, 则用fg命令。**参数:** 无。**范例:**

① 将执行的程序program 1、program 2与program 3暂停, 然后将program 2移到后台继续执行:


```
# program1
[1]+ Stopped program1 ← 按 Ctrl + Z 键暂停程序, [1]为工作编号
```

```
# program2
[2]+ Stopped program2 ← 按 Ctrl + Z 键暂停程序, [2]为工作编号
```

```
# program3
[3]+ Stopped program3 ← 按 Ctrl + Z 键暂停程序, [3]为工作编号
```

```
# bg 2
[2]- program2 & ← 将编号2的工作移到后台执行
```

bind

常用指数：

B

功能说明：显示或设置键盘按键与其相关的功能。**语法：**bind [-dlv] [-f<按键配置文件>] [-m<按键配置>] [-q<功能>]**包名称：**bash。**相关命令：**kbdconfig。**补充说明：**用户可以利用bind命令了解有哪些按键组合及其功能，也可自行指定要用哪些按键组合。**参数：**

-d	显示按键配置的内容
-f <按键配置文件>	载入指定的按键配置文件
-l	列出所有的功能
-m <按键配置>	指定按键配置。系统提供了emacs、emacs-meta、emacs-ctlx、vi、vi-move 与 vi-insert等按键配置。默认的按键配置为emacs
-q <功能>	显示指定功能的按键
-v	列出当前的按键配置与其功能

范例：

① 查询当要显示shell版本时，所需按下的按键是什么：

```
# bind -l | grep version
display-shell-version
do-lowercase-version
```

← 显示与version相关的功能
 ← 此功能可显示shell的版本

```
# bind -q display-shell-version ←显示功能所对应的按键
display-shell-version can be invoked via "\C-x\C-v". ←
                就是 Ctrl + X、Ctrl + V 的意思
# ←根据顺序按 Ctrl + X、Ctrl + V 键
GNU bash, version 2.05b.0(1)-release (i386-redhat-linux-gnu) ←
    Red Hat Linux及Fedora Core 所出现的信息
GNU bash, version 2.05b.0(1)-release (i586-mandrake-linux-gnu) ←
    若是Mandrake Linux则会出现此信息
```


② 显示vi的按键配置与其相关功能:

```
# bind -m vi -v
set blink-matching-paren on
set byte-oriented off
set completion-ignore-case off
set convert-meta off
...
```

③ 将指定的按键配置文件载入, 此例为/etc/inputrc:

```
# bind -f /etc/inputrc
```

bunzip2

常用指数: 

功能说明: .bz2文件的解压缩程序。

语 法: bunzip2 [-fkLsvV] [.bz2压缩文件]

包 名 称: bzip2。



相关命令： bzip2, bzip2recover。

补充说明： bunzip2可解压缩.bz2格式的压缩文件。bunzip2实际上是bzip2的符号链接，执行bunzip2与bzip2 -d的效果相同。

参 数：

-f或--force	解压缩时，若输出的文件与现有文件同名时，默认不会覆盖现有的文件。若要覆盖，请使用此参数
-k或--keep	在解压缩后，默认会删除原来的压缩文件。若要保留压缩文件，请使用 -k 参数
-s或--small	以另一种算法解压缩文件，能降低程序执行时内存的使用量，但会增加解压缩时间
-v或--verbose	解压缩文件时，显示详细的信息
-L、--license、 -V或--version	显示版本信息

范 例：

① 将file.bz2压缩文件解开后，不要删除原来的压缩文件，同时显示详细的信息：

```
# ls
file.bz2
# bunzip2 -kv file.bz2
file.bz2: done
# ls
file    file.bz2    ← 原来的file.bz2并没有被删除
```

② 解压缩当前目录下的file2.bz2及file3.bz2等两个文件：

```
# ls
file2.bz2  file3.bz2
# bunzip2 file2.bz2 file3.bz2 ← 同时指定两个要解压缩的文件
# ls
file2  file3
```

③ 当前目录下已有相同文件名的文件，先以默认的方式解压缩，再用-f参数解压缩：

```
# ls
file file.bz2
# bunzip2 file.bz2
bunzip2: Output file file already exists.
```

输出文件名已经存在，因此不覆盖现有文件

```
# bunzip2 -f file.bz2
```

强制覆盖现有文件

④ 解压缩file.bz2文件时，使用较少的内存：

```
# bunzip2 -s file.bz2
```

bzip2

常用指数：

功能说明： .bz2文件的压缩程序。

语 法： bzip2 [-cdfhkLstvVz] [-压缩等级] [要压缩的文件...]

包 名 称： bzip2。

相关命令： bunzip2, bzip2recover。

补充说明： bzip2采用新的压缩算法，压缩效果比传统的LZ77/LZ78压缩算法（如gzip命令即是采用此算法）好。若没有加上任何参数，bzip2压缩完文件后会产生.bz2的压缩文件，并删除原始文件。



参 数:

-c或--stdout	将压缩与解压缩的结果送到标准输出。若使用此参数执行压缩操作，且标准输出为屏幕时，bzip2会拒绝将压缩的结果输出到屏幕，因为这样只会在屏幕上显示乱码
-d或--decompress	执行解压缩
-f或--force	bzip2在压缩或解压缩时，若输出文件与现有文件同名，默认不会覆盖现有的文件。若要覆盖，请使用此参数
-h或--help	显示说明
-k或--keep	bzip2在压缩或解压缩后，会删除原始文件，若要保留原始文件请使用此参数
-s或--small	以另一种算法压缩或解压缩文件，能降低程序执行时内存的使用量，但会增加解压缩时间
-t或--test	测试.bz2文件的完整性。测试时会实际解压缩文件，再将结果删除
-v或--verbose	压缩或解压缩文件时，显示详细的信息
-z或--compress	强制执行压缩
-L、--license、 -V或--version	显示版本信息
-压缩等级	压缩时的区块大小。设置值从1~9，分别表示区块的大小为100~900 KB，默认值为9。压缩等级越大，压缩效果越佳，但所需时间较长

范 例:

① 压缩、解压缩与测试file文件:

```
# ls
file          ← 要压缩的文件
# bzip2 file ← 压缩文件
# ls
file.bz2     ← 压缩后会产生.bz2文件，并删除原始文件
```

```
# bzip2 -t -v file.bz2 ← 测试压缩文件是否正常
file.bz2: ok ← 文件没问题
# bzip2 -d file.bz2 ← 解压
# ls
file ← 解压后会解开原始文件，
      并删除.bz2压缩文件
```

② 压缩目录下的file2及file3两个文件，然后再将其解压缩：

```
# ls
file2 file3
# bzip2 file2 file3 ← 同时指定两个要压缩的文件
# ls
file2.bz2 file3.bz2 ← file2及file3被压缩成
                       file2.bz2及 file3.bz2
# bzip2 -d file2.bz2 file3.bz2 ← 同时指定两个要解压的文件
# ls
file2 file3
```

③ 当前目录下已有同文件名的压缩文件，先以默认方式压缩，再用-f参数压缩：

```
# ls
file file.bz2
# bzip2 file
bzip2: Output file file.bz2 already exists. ← 输出文件名已经
                                                存在，因此不覆
                                                盖现有文件
# bzip2 -f file ← 当输出的文件名已经存在时，覆盖现有文件
```

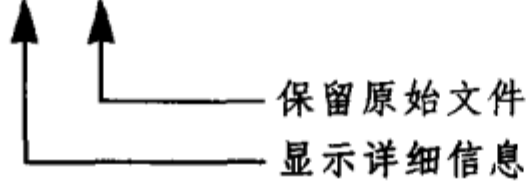
④ 压缩file文件时，使用较少的内存：

```
# bzip2 -s file
```

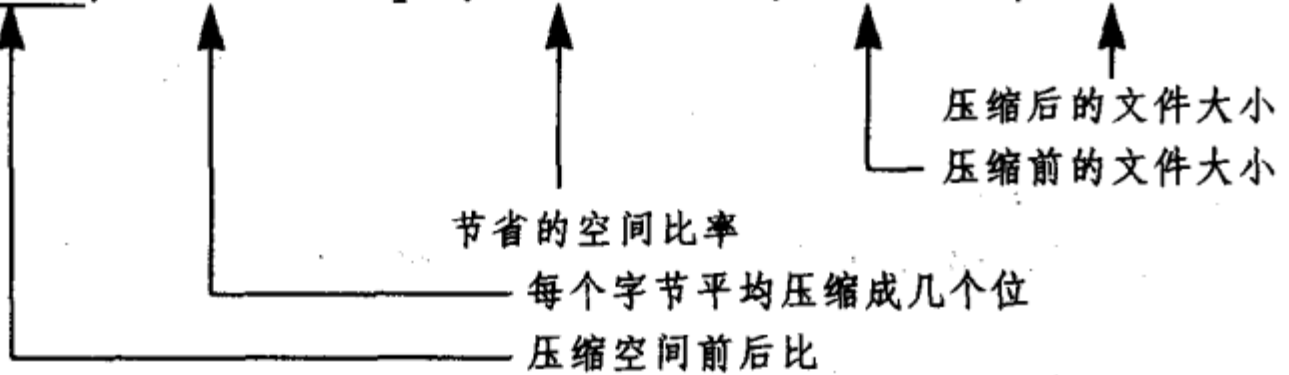
⑤ 先用默认方式压缩file文件，再以压缩等级1压缩同一个文件：

```
# bzip2 -v -k file
```

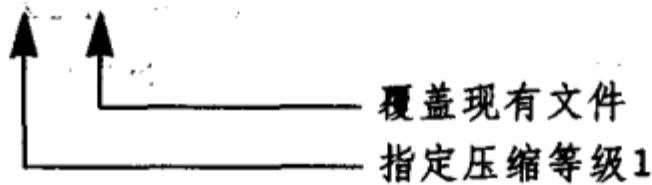
B



```
file: 5.083:1, 1.574 bits/byte, 80.33% saved, 271756 in, 53461 out.
```




```
# bzip2 -v -1 -f file
```



```
file: 2.971:1, 2.693 bits/byte, 66.34% saved, 271756 in, 91475 out.
```

因为指定了较小的压缩等级，所以压缩比例较小

bzip2recover

常用指数: 

功能说明: 用来修复损坏的 .bz2 文件。

语 法: bzip2recover [.bz2 压缩文件]

包 名 称: bzip2。

相关命令: bunzip2, bzip2。



补充说明: bzip2是以区块的方式来压缩文件, 每个区块视为独立的单位。因此, 当某一区块损坏时, 便可利用bzip2recover, 试着将文件中的区块分隔开来, 以便解压正常的区块。通常只适用于压缩文件很大(即包含许多区块)的情况。

B

参 数: 无。

范 例:

尝试解压file.bz2文件中的正常区块:

```
# bzip2recover file.bs2          ←—— 修复损坏的文件
bzip2recover 1.0.2: extracts blocks from damaged .bz2 files.
bzip2recover: searching for block boundaries ...
  block 1 runs from 80 to 132524
  block 2 runs from 132573 to 132616 (incomplete)
bzip2recover: splitting into blocks
  writing block 1 to 'rec0001file.bz2' ... ←—— 将block 1解压成
                                          为rec0001file.
                                          bz2
bzip2recover: finished
```



cal (calendar)

常用指数:

**功能说明:** 显示月历。**语法:** cal [-jmy] [<月份> 年份]**包名称:** util-linux。

相关命令: 执行cal命令会显示一份简单的月历，如果不加上任何参数，则会显示当前月份的月历。若仅指定一个数字，则该数字会当成年份而非月份，而年份必须用公元年来表示。

参数:

-j	用凯撒历 (Dates of Julius Caesar) 的形式来显示月历，即不分月份，以每年的1月1日为第1天，一直向后数到年底 (365或366) 为止
-m	显示月历时，把星期一定为一周的开始。若不加此参数，则默认以星期日为一周的开始
-y	显示今年所有月份的月历

范例:

① 显示当月的月历:

cal

5月 2003

```

日 一 二 三 四 五 六
      1  2  3
4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

```

← 不指定任何参数

← 现在是2003年5月

— 2003年5月的月历

↑ 以星期日为一周的开始

② 显示当月的月历，但显示时以星期一为一周的开始：

```
# cal -m
```

```
      5月2003
 一  二  三  四  五  六  日
           1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

↑
—— 以星期一为一周的开始

③ 显示今年整年的月历：

```
# cal -y
```

2003 ← 公元2003年月历

一月份月历							二月份月历							三月份月历						
一月							二月							三月						
日	一	二	三	四	五	六	日	一	二	三	四	五	六	日	一	二	三	四	五	六
			1	2	3	4							1							1
5	6	7	8	9	10	11	2	3	4	5	6	7	8	2	3	4	5	6	7	8
12	13	14	15	16	17	18	9	10	11	12	13	14	15	9	10	11	12	13	14	15
19	20	21	22	23	24	25	16	17	18	19	20	21	22	16	17	18	19	20	21	22
26	27	28	29	30	31	23	24	25	26	27	28	23	24	25	26	27	28	29		
														30	31					

...

④ 显示公元2003年6月的月历：

```
# cal 6 2003
```

↑
—— 若指定成 "01"，将会显示公元3年而非2003年的月历

六月 2003						
日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

2003年6月的月历

⑤ 用凯撒历的形式显示公元2003年6月的月历：

```
# cal -j 6 2003
```

六月 2003						
日	一	二	三	四	五	六
152	153	154	155	156	157	158
159	160	161	162	163	164	165
166	167	168	169	170	171	172
173	174	175	176	177	178	179
180	181					

这是凯撒历的显示状态

cat (concatenate)

常用指数: 

功能说明: 连接多个文件，并将它们的内容输出到标准输出设备。

语 法: `cat [-AbeEnstTuv] [--help] [--version] [文件...]`

包名称: coreutils。

相关命令: `csplit`, `cut`, `head`, `tac`, `tail`, `zcat`。

补充说明: `cat`命令会读取指定文件的内容，并输出到标准的输出设备上（例如显示器）。若不指定任何文件名称，或是指定的文件名为“-”，则`cat`命令会从标准输入设备读取数据（例如键盘），然后再把所得到的数据输出到输出设备。也可运用shell的特殊字符“>”和“>>”，把多个文件的内容合并成一个文件。

参 数:

-A或--show-all	此参数的效果和同时指定“-vET”参数相同
-b或--number-nonblank	列出文件内容时,在所有非空白行的开头标上编号,号码从1开始依次累加
-e	此参数的效果和同时指定“-vE”参数相同
-E或--show-ends	在每一行的最后标上“\$”符号
-n或--number	列出文件内容时,在每一行的开头标上编号,号码从1开始依次累加
-s或--squeeze-blank	当内容某部分的空白行超过一行以上时,则该处仅以一行空白行显示
-t	此参数的效果和同时指定“-vT”参数相同
-T或--show-tabs	将制表符(Tab)以“^I”表示
-u	此参数将忽略不予处理,仅负责解决UNIX的兼容性问题
-v或--show-nonprinting	除了换行符(LFD)及制表符之外,其他的控制字符皆以“^”符号表示,高位字符(十进制字码大于127以上者)则用“M-”表示
--help	显示帮助
--version	显示版本信息

范 例:

① 列出file文件的内容:

```
# cat file
line1 in file
line2 in file
line3 in file
```

file文件的内容

② 列出file文件的内容，并在每行的开头标上编号：

```
# cat -n file
 1 line1 in file
 2 line2 in file
 3
 4
 5 line3 in file
```

每列的开头均依次标上编号

③ 列出file文件的内容，除了空白行外，在每行的开头标上编号：

```
# cat -b file
 1 line1 in file
 2 line2 in file

```

空白行不标编号

```
 3 line3 in file
```

④ 列出file文件的内容，并在每行的结束处加上“\$”符号：

```
# cat -E file
line1 in file$
line2 in file$
$
$
line3 in file$
```

结束处均加上“\$”

⑤ 列出file文件的内容，若有连续的空白行，则只以1个空白行显示：

```
# cat -s file
line1 in file
line2 in file
line3 in file
```

← 虽然有2个空白行，但只以一行显示

⑥ 同时列出file1及file2文件的内容：

```
# cat file1 file2
line1 in file1
line2 in file1
line1 in file2
line2 in file2
```

file1的内容

file2的内容

⑦ 利用特殊字符“>”将名称为file1与file2的文件合并成一个文件file3：

```
# cat file1 file2 > file3
```

↑ 若文件file3已经存在，则其内容会被覆盖过去，要避免这种情况发生，可用“>>”代替“>”，新的内容就会附加在原有内容之后，而不会覆盖它

⑧ 将.bashrc文件的内容显示在显示器上，并将控制、高位字符、制表符分别以“^”、“M-”与“^I”替代，且在每行的最后面标上“\$”符号：

```
# cat -A ~/.bashrc
# .bashrc$
$
# User specific aliases and functions$
$
alias rm='rm -i'$
alias cp='cp -i'$
```

每一行的结尾都标上“\$”符号


```
alias mv='mv -i'$
$
# Source global definitions$
if [ -f /etc/bashrc ]; then$
^I. /etc/bashrc$
fi$
```

制表符用“^|”表示

⑨ 让cat命令从标准输入设备（如键盘）读取数据，转而输出到标准输出设备（如显示器）：

```
# cat
Linux is Powerful!!
Linux is Powerful!!
```

← 执行命令，不加任何参数
 ← 键入任何文字后，按下 **Enter** 键
 ← 系统响应一模一样的文字

cd (change directory)

常用指数：



功能说明：切换目录。

语法：cd [目标目录]

包名称：cd。

相关命令：mcd。

补充说明：cd命令可让用户在不同的目录间切换，但该用户必须拥有足够的权限才能进入目的目录。

参数：无。

范 例:

① 从当前目录切换到/home/jessica目录中:

```
# cd /home/jessica ← 必须拥有足够权限, 才能切换到/home/jessica
                        目录
```

② 将当前目录切换到用户根目录:

```
# cd ← 不加任何参数
```

③ 将当前目录切换到上一层目录下的tmp目录:

```
# cd ../tmp
```

④ 将当前目录切换到指定的目录:

```
# cd "/mnt/harddisk/Program Files"
```

因为“Program Files”目录名称中含有空格字符，
所以要用" "符号将目录名称包含起来

cdp (cd play)

常用指数:



功能说明: 播放音乐CD。

语 法: cdp [play<音轨编号>] [stop] [table] [-hlms] [-c<外围设备代号>]

包 名 称: cdp。

相关命令: aumix。

补充说明: cdp是一个CD播放程序，启动之后可利用数字键操作cdp。



参 数:

play <音轨编号>	直接指定从哪一首音乐开始播放
stop	停止播放音乐
table	从数据库中显示当前CD的数据，并播放音乐
-c <外围设备代号>	指定用来播放CD的外围设备代号
-h	显示帮助
-l	等待播放外围设备初始化后再播放音乐
-n	不要自动播放CD
-s	静音模式，此时播放面板上会显示“Silent”字符串

范 例:

① 执行cdp会出现以下的画面，可用稍后所介绍的操作命令播放音乐CD:

cdp

作者名称

此CD的总播放时间

可对照键盘右方的KeyPad操作

此CD的名称

此CD的总歌曲数

当前播放的状态

每首歌的流水号

每首歌在这片CD开始的时间

每首歌的时间长度

每首歌的名称

```

CDPlay 0.33
Artist      : Alkane
CD Name     : Yesterday
Tracks Number : 12
Total Play Time : 52:01

playing #1 0:16 0:18

1 0:02 3:08 Music 01
2 3:10 4:56 Music 02
3 8:06 4:19 Music 03
4 12:25 4:18 Music 04
5 16:43 4:36 Music 05
6 21:19 4:30 Music 06
7 25:49 4:38 Music 07
8 30:27 4:11 Music 08
9 34:38 4:21 Music 09
10 38:59 4:14 Music 10
11 43:13 3:52 Music 11
12 47:06 4:55 Music 12

STOP      ||      PLAY
<-      ||      >
<<      RESTART  >>
QUIT     EJECT   HELP
    
```

- ② 指定第2个光驱播放CD（假设系统中有1个IDE接口硬盘，还有至少2个以上的IDE/ATAPI光驱）：

```
# cdp -c /dev/hdc
```

↑
第2个光驱的外围设备代号。本范例中，硬盘代号为/dev/hda，第1个光驱是/dev/hdb，第2个光驱为/dev/hdc

- ③ 直接指定从第10首音乐开始播放：

```
# cdp play 10
```

- ④ 停止播放音乐：

```
# cdp stop
```

- ⑤ 从数据库中显示当前CD的数据，并播放音乐：

```
# cdp table
```

```
Artist : na
```

```
CD Name: cdname
```

```
Tracks Number: 11 Total Play Time: 0:51:53
```

```
Track Num Start Time Length Name
```

```
* 1      0:00:02    4:26
   2      0:04:28    4:20
   3      0:08:48    4:35
   4      0:13:23    4:35
   5      0:17:58    3:30
   6      0:21:28    4:40
   7      0:26:08    5:19
   8      0:31:27    4:43
   9      0:36:10    5:17
  10      0:41:27    4:58
  11      0:46:25    5:28
```

⑥ 执行cdp后，不要自动播放CD：

```
# cdp -n
```

⑦ 执行cdp后，等待外围设备初始化后再播放音乐：

```
# cdp -l
```

操作命令：执行cdp之后可利用数字键操作，常用的操作命令如下：

9	播放
8	暂停/重新播放
7	停止播放
6	播放下一首歌曲
5	重新播放当前播放的歌曲
4	播放上一首歌曲
3	前进15秒
2	退出CD
1	后退15秒
0	结束cdp程序，但仍继续播放CD
Enter	编辑当前歌曲的名称
A	编辑作者的名称
C	编辑此张CD的名称

cfdisk

常用指数：



功能说明：硬盘分区。

语法： cfdisk [-avz] [-c <柱面数目> -h <磁头数目> -s <盘区数目>]
[-P <r,s,t>] [外围设备代号]

包名称: util-linux (Mandrake Linux)。

相关命令: fdisk, sfdisk。

补充说明: 这是Mandrake Linux才有的命令。cfdisk是用来硬盘分区的命令。它十分类似DOS的fdisk, 具有互动式操作界面而非传统fdisk的问答式界面。可以轻易地利用方向键来操控分区操作。

参 数:

-a	加此参数, 会以箭头表示选取; 若不加, 则以反白表示选取
-c <柱面数目>	忽略BIOS的数值, 直接指定硬盘的柱面数目。警告: 除非控制卡上没有BIOS (如一些廉价的SCSI控制卡), 或 BIOS 所检测到的CHS数值是错误的, 而且你很清楚该硬盘设备的硬件设置值, 否则不应使用此参数
-h <磁头数目>	忽略BIOS的数值, 直接指定硬盘的磁头数目。警告: 除非控制卡上没有BIOS (如一些廉价的SCSI控制卡), 或 BIOS 所检测到的CHS数值是错误的, 而且你很清楚该硬盘设备的硬件设置值, 否则不应使用此参数
-P <r,s,t>	显示分区表的内容, 附加参数“r”会显示整个分区表的详细数据, 附加参数“s”会按照盘区 (Sector) 的顺序显示相关信息, 附加参数“t”则会以磁头 (Head)、盘区、柱面 (Cylinder) 的方式来显示数据
-s <盘区数目>	忽略BIOS的数值, 直接指定硬盘的盘区数目。警告: 除非控制卡上没有BIOS (如一些廉价的SCSI控制卡), 或 BIOS 所检测到的CHS数值是错误的, 而且你很清楚该硬盘设备的硬件设置值, 否则不应使用此参数
-v	显示版本信息
-z	不读取现有的分区表, 直接当作没有分区的新硬盘使用。当硬盘的分区表毁损、错误, 导致cfdisk无法读取而不能正常启动时, 用此参数可强制执行

范 例:

① 执行cfdisk会出现以下的画面, 可用稍后所介绍的操作命令操作:

cfdisk

反白表示当前作用中的分区

当前硬盘机的状态

```

cfdisk 2.11x

Disk Drive: /dev/hda
Size: 20547641536 bytes, 20.5 GB
Heads: 255 Sectors per Track: 63 Cylinders: 2498

Name      Flags      Part Type  FS Type    [Label]      Size (MB)
-----
hda1     Boot      Primary   Linux ext3  [/]          6004.46
hda2     Primary   Primary   Linux ext3  [/1]         6004.46
hda3     Primary   Primary   Linux ext3  [/123]       4005.72
hda5     Logical   Logical   Linux ext3  [ ]          4408.76
hda6     Logical   Logical   Linux swap  [ ]          123.38

[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ]   [ Type ]  [ Units ] [ Write ]

Toggle bootable flag of the current partition
    
```

可使用的操作命令

当前操作命令的简要说明

可切换的分区

② 切割系统上第2个SCSI接口硬盘的分区:

cfdisk /dev/sdb

第2个SCSI硬盘的外围设备代号

③ 以盘区排序，观看第1个IDE硬盘的分区状态：

```
# cfdisk -P s /dev/hda
```

Partition Table for /dev/hda

#	Type	First Sector	Last Sector	Offset	Length	Filesystem Type (ID)	Flags
1	Primary	0	11727449	63	11727450	Linux(83)	Boot(80)
2	Primary	11727450	23454899		0	11727450 Linux(83)	None(00)
3	Primary	23454900	31278554		0	7823655 Linux(83)	None(00)
4	Primary	31278555	40130369		0	8851815 0Extended(05)	None(00)
5	Logical	31278555	39889394	63		8610840 Linux(83)	None(00)
6	Logical	39889395	40130369	63		240975 Linux swap(82)	None(00)

④ 执行cfdisk时，以箭头代替反白表示当前选取的分区：

```
# cfdisk -a
```

⑤ 执行cfdisk时，不读取现有的分区：

```
# cfdisk -z
```

操作命令：执行cfdisk之后，常见的操作命令说明如下：

↑ 或 ↓	向上或向下切换分区
B	将分区设成可开机
D	删除当前分区
H 或 ?	显示帮助
G	改变柱面、磁头及盘区数
M	将分区的容量尽可能地最大化，但可能不兼容于DOS或OS/2操作系统

■	若还有未分配的分区，则建立新的分区
■	输出当前分区的状态
■	不存储分区更改后的状态，直接结束程序
■	改变分区文件系统的类型
■	改变显示分区容量时的单位，可显示的单位有百万字节、柱面数及盘区数
■	写入分区更改后的状态，由于此操作影响非常大，所以会紧接要求确定，可输入“yes”确认，或“no”取消

chattr (change attribute)

常用指数：



功能说明：改变文件属性。

语法：chattr [-RV] [-v<版本编号>] [+/-/=<属性模式>] [文件或目录...]

包名称：e2fsprogs。

相关命令：lsattr, mattrib。

补充说明：这项命令可改变存放在ext2文件系统上的文件或目录属性，这些属性共有8种模式，内容如下。

- a: 让文件或目录仅供附加用途。此项属性必须管理员才能设置或清除，它让文件或目录只能从附加模式中修改，而不能直接编辑写入。
- A: 不更新文件或目录的最后访问时间 (Access Time)。设置这项属性，可减少硬盘访问的次数，节省笔记本电脑等的电源。
- c: 将文件或目录压缩后存放。遇到设有此项属性的文件或目录时，系统内核会自动在写入数据时，先把数据压缩再存放至硬盘上，读取数据时则会自动解压缩。
- d: 将文件或目录排除在转储 (Dump) 操作之外。转储操作是一种针对文件系统的全面性备份，若文件或目录具备这个属性，转储操作进行时，就不会备份该文件或目录。

- i: 不得任意更改文件或目录。此项属性必须管理员才能设置或清除, 文件或目录具有这项属性, 将无法删除、更改名称、编辑修改, 甚至无法连接这类文件或目录。
- s: 保密性删除文件或目录。在删除设有此类属性的文件或目录时, 系统会用“00”(十六进制字码) 字符填满原本存放这些文件或目录的块, 彻底破坏数据, 达到保密目的。
- S: 即时更新文件或目录。对硬盘写入数据时, 许多操作系统都会先把数据填入缓冲区(Buffer)内, 避免硬盘读写频繁, 提升整体性能。如果对文件或目录设置这项属性, 则在其改变时, 系统将不会通过缓冲区, 而会直接把数据写入文件或目录。
- u: 预防意外删除。当此类文件或目录删除时, 其内容会另行保存, 让用户得以救回不小心删除的数据。

参 数:

-R	递归处理, 将指定目录下的所有文件及子目录一并处理
-v <版本编号>	设置文件或目录版本
-V	显示命令执行过程
+<属性模式>	打开文件或目录的该项属性
-<属性模式>	关闭文件或目录的该项属性
=<属性模式>	指定文件或目录的该项属性

范 例:

- ① 将file文件的属性设成压缩后存放:


```
# chattr -R +c file
# lsattr file ← 利用lsattr命令检查文件的属性
--c----- file
↑
|
└── file的压缩属性已经打开
```

- ② 接上例，将file文件的属性设成保密性删除和即时更新，并删除其他的属性：

```
# chattr =sS file
# lsattr file
s-S----- file
↑ ↑
└── 文件的属性已经设成s及S
```

- ③ 接上例，取消file文件的s属性：

```
# chattr -s file
# lsattr file
--S----- file
↑
└── s属性已经取消
```

- ④ 将file1及file2文件设成不得任意更改：

```
# chattr +i file1 file2
# lsattr file1 file2
----i--- file1
----i--- file2
```

- ⑤ 把killme目录下的所有文件和子目录，统统设置为不得任意更改：

```
# chattr -R +i killme
```

chfn (change finger information)

常用指数:



功能说明: 改变finger命令显示的信息。

语 法: chfn [-uv] [-f <真实姓名>] [-h <家庭电话>] [-o <办公地址>]
[-p <办公电话>] [账号名称]

包 名 称: util-linux。

相关命令: finger, passwd。

补充说明: chfn命令可用来更改执行finger命令时所显示的信息, 这些信息都存放在/etc目录里的passwd文件里。若不指定任何参数, 则chfn命令会进入问答式界面。

参 数:

-f<真实姓名>或--full-name<真实姓名>	设置真实姓名
-h <家庭电话>或--home-phone<家庭电话>	设置家庭的电话号码
-o <办公地址>或--office<办公地址>	设置办公室的地址
-p <办公电话>或--office-phone<办公电话>	设置办公室的电话号码
-u或--help	显示帮助
-v或--version	显示版本信息

范 例:

① 设置用户alkane的电话为123-456-789:

```
# chfn -h 123-456-789 alkane
```

```
Changing finger information for alkane.
```

```
Changing finger information changed.
```

← 已经成功地改变用户alkane的数据

```
# finger alkane          ← 用finger查询用户alkane
Login: alkane           Name:
Directory: /home/alkane Shell: /bin/bash
Home Phone: 123-456-789 ← 用户alkane的电话已经改成刚刚所设
                          的电话
Never logged in.
No mail.
No Plan.
```

② 设置用户alkane的真实姓名为 ShineWin:

```
# chfn -f ShineWin alkane
```

③ 设置用户alkane办公室地址为 “Beijing city”，办公室电话为234-567-890:

```
# chfn -o "Beijing city" -p 234-567-890 alkane
```

因为指定的地址有空格，故在前后加上引号

chgrp (change group)

常用指数:



功能说明: 更改文件或目录的所属组。

语法: chgrp [-cfhRv] [--help] [--version] [所属组] [文件或目录...] 或 chgrp [-cfhRv] [--help] [--reference=<参考文件或目录>] [--version] [文件或目录...]

包名称: coreutils。

相关命令: chown。

补充说明: 在UNIX系统家族里，文件或目录权限的掌控以拥有者及所属组来管理。可以使用chgrp命令去更改文件与目录的所属组，设置方式采用组名称或组识别码（GID, Group ID）都可以。

参 数:

-c 或 --changes	效果类似“-v”参数，但仅报告改变的部分
-f 或 --quiet 或 --silent	不显示错误信息
-h 或 --no-dereference	只对符号链接 (Symbolic Link) 的文件作修改，而不更改原始文件
-R 或 --recursive	递归处理，将指定目录下的所有文件及子目录一并处理
-v 或 --verbose	显示命令执行过程
--help	显示帮助
--reference=<参考文件或目录>	把指定文件或目录的所属组，统统设成和参考文件或目录的所属组相同
--version	显示版本信息

范 例:

① 更改file文件的所属组为cirgroup:

```
# ls -l file
-rwxr-xr-x  1 root  root      44 5月 27 11:53 file
                ↑
                原来所属组为root
# chgrp cirgroup file ← 更改所属组为 cirgroup
# ls -l file
-rwxr-xr-x  1 root  cirgroup  44 5月 27 11:53 file
                ↑
                所属组已经改成 cirgroup
```

② 把tmp目录下所有文件和子目录的所有组，统统设为cirgroup，并显示执行的过程:

```
# chgrp -R cirgroup tmp
```

- ③ symbolic_file是一个符号链接文件，而file是被连接的原始文件，现在更改symbolic_file文件所属的组：

```
# ls -l          ← 查看目录下的文件
总计 8
-rw-r--r--    1 root root   5060   5月   26 09:58 file
lrwxrwxrwx    1 root root     4     5月   26 09:58 symbolic_
                                     file→file ←
```

symbolic_file 是 file 文件的符号链接

```
# chgrp lstgroup symbolic_file ← 以默认的方式更改
                                     symbolic_file所属的组
```

```
# ls -l
```

实际上会更改原始文件file所属的组，而不是更改symbolic_file所属的组

```
总计8
-rw-r--r--    1 root lstgroup 5060 5月26 09:58 file
lrwxrwxrwx    1 root root     4 5月26 09:58 symbolic_file
→ file
```

```
# chgrp -h 2ndgroup symbolic_file ← 加上“-h”参数
```

```
# ls -l
```

```
total 8
```

```
-rw-r--r--    1 root lstgroup 5060 5月 26 09:58 file
lrwxrwxrwx    1 root 2ndgroup   4 5月 26 09:58 symbolic_file→file
```

更改了symbolic_file符号链接所属的组

- ④ 文件alpha、bravo、charlie、delta分别隶属于不同的组，现将bravo、charlie及delta文件的所属组改成与alpha文件一致：

```
# chgrp --reference=alpha bravo charlie delta
```

以文件alpha作标准，将其余文件的所属组设成与它相同

chkconfig (check configuration) 常用指数:

功能说明: 检查、设置系统的各种服务。

语法: `chkconfig [--add] [--del] [--list] [系统服务] 或`
`chkconfig [--level<运行级代号>] [系统服务] [on/off/reset]`

包名称: chkconfig。

相关命令: ntsysv, runlevel, telinit。

补充说明: 这是Red Hat公司遵循GPL (GNU Public License) 规则所开发的程序, 它可查询操作系统在每一个运行级 (Runlevel) 中, 会执行哪些系统服务 (System Service), 其中包括各类守护进程 (Daemon)。

Linux操作系统将操作环境分为以下7个等级:

运行级	执行的状态
0	关机 (请不要切换到此运行级)
1	单人用户模式的文字界面
2	多人用户模式的文字界面, 但不具有网络文件系统 (NFS) 功能
3	多人用户模式的文字界面, 具有全部的网络功能
4	某些发行版的Linux使用此等级进入X Window
5	某些发行版的Linux则使用此等级进入X Window
6	重新开机

参 数:

<code>--add</code>	新增所指定的系统服务, 让chkconfig命令得以管理它, 并同时在系统启动的脚本内新增相关数据
<code>--del</code>	删除所指定的系统服务, 不再由chkconfig命令管理, 并同时在系统启动的脚本内删除相关数据
<code>--level<运行级代号></code>	指定该系统服务要在哪一个运行级中打开或关闭, 运行级代号的范围在0~6之间

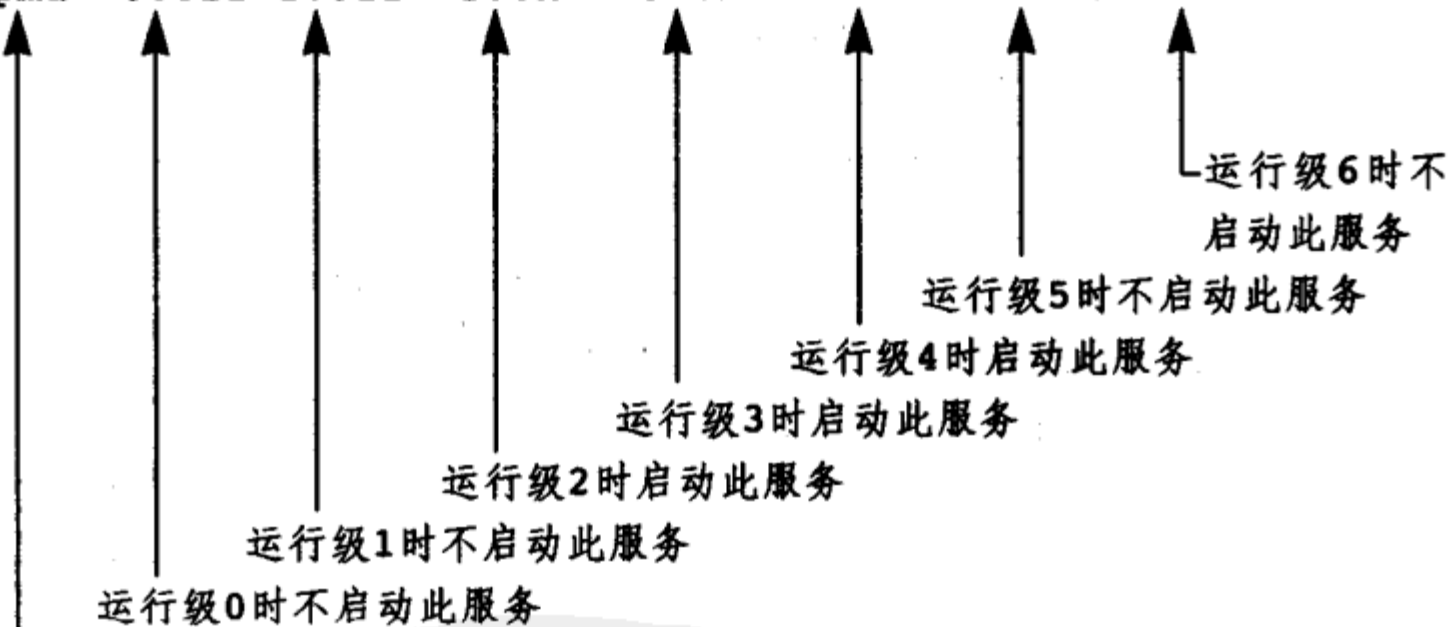
<code>--list</code>	列出当前可从chkconfig命令管理的所有系统服务，并标出每项服务在运行级代号为0~6的状态
<code>on</code>	在指定的运行级，打开该项系统服务。如不指定运行级，默认会在运行级3、4、5打开该服务
<code>off</code>	在指定的运行级，关闭该项系统服务。如不指定运行级，默认会在运行级3、4、5打开该服务
<code>reset</code>	在指定的运行级，重置该项系统服务，使它回到操作系统启动时，其脚本（script）内的默认值

范 例：

① 查看在各个不同的运行级中，各项系统服务的情况：

```
# chkconfig --list
```

```
amd    0:off 1:off 2:off 3:off 4:on 5:off 6:off
httpd  0:off 1:off 2:off 3:off 4:on 5:off 6:off
apmd   0:off 1:off 2:on  3:on  4:on 5:off 6:off
```



系统服务的名称

...

② 列出系统服务apmd在各个运行级的启动情况：

```
# chkconfig --list apmd
```

```
apmd 0:off 1:off 2:on 3:on 4:on 5:off 6:off
```

③ 在运行级2、3及5时，关闭apmd系统服务：

```
# chkconfig --level 235 apmd off
```

↑
关闭此服务

↑
运行级没有235，此处是指运行级2、运行级3与运行级5

```
# chkconfig --list apmd ← 查看apmd系统服务的状态
```

```
apmd 0:off 1:off 2:off 3:off 4:on 5:off 6:off
```

↑ ↑ ↑

运行级2、3及5已经
关闭此服务

④ 在运行级2、3及5时，打开apmd系统服务：

```
# chkconfig --level 235 apmd on
```

↑
打开此服务

↑
指定运行级2、3及5

```
# chkconfig --list apmd ← 查看apmd系统服务的状态
```

```
apmd 0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

↑ ↑ ↑

运行级2、3及5已经
打开此服务

⑤ 在运行级2、3及5时，重设apmd系统服务成操作系统启动时的默认值：

```
# chkconfig --level 235 apmd reset
```

```
# chkconfig --list apmd
```

```
apmd 0:off 1:off 2:on 3:on 4:on 5:off 6:off
```

↑ ↑ ↑

已恢复成操作系统
启动时的默认值

⑥ 关闭sendmail系统服务，由于执行chkconfig时没有指定运行级，所以默认会在运行级3、4及5更改此服务的设置值：

```
# chkconfig --list sendmail ← 查看sendmail系统服务的状态
sendmail 0:on 1:on 2:on 3:on 4:on 5:on 6:on ← 在每个等级
# chkconfig sendmail off ← 执行时不指定运行级 下都是启动
# chkconfig --list sendmail 的状态
sendmail 0:on 1:on 2:on 3:off 4:off 5:off 6:on
```

默认会更改运行级
3、4及5的状态

chmod (change mode)

常用指数:



功能说明: 更改文件或目录的权限。

语法: chmod [-cfRv] [--help] [--version] [<权限范围>+/-/=<权限设置>,...] [文件或目录...]或chmod [-cfRv] [--help] [--version] [数字代号] [文件或目录...]或chmod [-cfRv] [--help] [--reference=<参考文件或目录>] [--version] [文件或目录...]

包名称: coreutils。

相关命令: umask。

补充说明: 在UNIX家族里，文件或目录权限的控制分别以读取、写入、执行（或切换）3种一般权限来区分，另有3种特殊权限可供使用，再搭配拥有者与所属组，来管理每个文件及目录的权限范围。你可以使用chmod命令去更改文件与目录的权限，设置方式采用文字或数字代号皆可。符号链接的权限无法更改，如果你对符号链接有修改权限，其改变会作用在被连接的原始文件。权限范围的表示法如下所述。

- u: User, 即文件或目录的拥有者。
 - g: Group, 即文件或目录的所属组。
 - o: Other, 除了文件或目录的拥有者或所属组之外, 其他用户皆属于这个范围。
 - a: All, 即全部的用户, 包含拥有者、所属组, 以及其他用户。
- 有关权限代号的部分, 参见如下。
- r: 读取权限, 数字代号为“4”。
 - w: 写入权限, 数字代号为“2”。
 - x: 执行或切换权限, 数字代号为“1”。
 - -: 不具任何权限, 数字代号为“0”。
 - s: 特殊权限SUID或SGID, 各自表示拥有者和所属组的特殊权限, 数字代号分别为“4”和“2”。对文件而言, 应用SUID或SGID之后, 便能任意访问该文件的拥有者或所属组权限所及的全部资源。对目录而言, 将文件复制到一个套用SGID权限的目录里, 文件的所属组会重设为该目录的所属组。
 - t: 特殊权限Sticky, 数字代号为“1”。对一个目录设置这项权限, 则存放于此目录中的所有文件或子目录, 都只能被该文件或目录的拥有者删除。

参 数:

-c或--changes	效果类似“-v”参数, 但仅返回改变的部分
-f、--quiet或--silent	不显示错误信息
-R或--recursive	递归处理, 将指定目录下的所有文件及子目录一并处理
-v或--verbose	显示命令执行过程

-help	显示帮助
--reference= <参考文件或目录>	把指定文件或目录的权限，统统设成和参考文件或目录的权限相同
--version	显示版本信息
<权限范围>+<权限设置>	打开权限范围的文件或目录的该项权限设置
<权限范围>-<权限设置>	关闭权限范围的文件或目录的该项权限设置
<权限范围>=<权限设置>	指定权限范围的文件或目录的该项权限设置

范 例:

① 查询文件或目录的权限状态:

```
# ls -l                                     ← 用ls命令加上“-l”参数，以详细格式查询
总计6405
drwxr-xr-x  2 root  root   1024   5月 28 16:02 News
drwxr-xr-x  2 root  root   1024   5月  5 16:02 SETI
.....
-rw-r--r--  1 root  root  15427   5月  6 17:03 sigma
drwxr-xr-x  2 root  root   1024   5月  6 17:17 zeta
```

第2~10字节就是权限的标示，其中每3个为一组，各自表示拥有者、所属组、其他用户的权限范围

② 将文件sigma和目录zeta的权限设置为拥有者能够读取和写入，同一组的用户可以读取，其他用户则不准读写:

```
# chmod u=rw,g=r,o= sigma zeta
```

若用数字来表示，其代号为“0640”，第1个数字表示特殊权限，其余3个数字依次表示拥有者、所属组、其他用户的权限范围

③ 用文字代号把名称为tango的文件开放给所有用户读取和写入:

```
# chmod ugo=rw tango
```

用“a=rw”也有相同效果

④ 取消其他人执行file文件的权限:

```
# chmod o-x file
```

↑
取消执行权限

⑤ 增加所属组写入file文件的权限:

```
# chmod g+w file
```

↑
加入写入权限

⑥ 将tmp目录下所有文件和子目录, 增加所属组写入与执行的权限:

```
# chmod -R g+wx tmp
```

⑦ 以file文件做标准, 将file1、file2及file3文件的权限设成与它相同:

```
# chmod --reference=file file1 file2 file3
```

↑
以file文件做标准, 将其余文件的权限设成与它相同

⑧ 用数字代号把名称为tango的文件开放给所有用户读取和写入:

```
# chmod 0666 tango
```

↑
数字“6”等于“4+2”, 也就是读取和写入权限的数字代号的总和



chown (change owner)

常用指数:



功能说明: 更改文件或目录的拥有者或所属组。

语 法: `chown [-cfhRv] [--dereference] [--help] [--version] [拥有者.<所属组>] [文件或目录...]` 或 `chown [-cfhRv] [--dereference] [--help] [--version] [.所属组][文件或目录...]` 或 `chown [-cfhRv] [--dereference] [--help] [--reference=<参考文件或目录>] [--version] [文件或目录...]`

包 名 称: coreutils。

相关命令: chgrp。

补充说明: 在UNIX系统家族里，文件或目录权限的掌控以拥有者及所属组来管理。你可以使用chown命令更改文件与目录的拥有者或所属组，设置方式采用用户名称或用户识别码 (UID, User ID) 都可以，设置组则可用组名称或组识别码。

参 数:

-c或--changes	效果类似“-v”参数，但仅返回改变的部分
-f、--quiet或--silent	不显示错误信息
-h或--no-dereference	只对符号链接的文件作修改，而不更改其他任何相关文件
-R或--recursive	递归处理，将指定目录下的所有文件及子目录一并处理
-v或--verbose	显示命令执行过程
--dereference	效果和“-h”参数相反
--help	显示帮助
--reference=<参考文件或目录>	把指定文件或目录的拥有者与所属组，统统设成和参考文件或目录的拥有者与所属组相同
--version	显示版本信息

范 例：

① 更改file1文件的拥有者为alkane：

```
# chown alkane file1
```

↑
只更改拥有者

② 更改file1文件的所属组为sphenoid：

```
# chown .sphenoid file1
```

↑
只更改所属组

③ 更改rebel文件的拥有者为eagle，及所属组为users：

```
# chown eagle.users rebel
```

|
指定用户名称及组名称

由于eagle的用户识别码为505，user的组织识别码为100，你也可以用指定识别码的方式，来更改文件的拥有者及所属组：

```
# chown 505.100 rebel
```

|
指定用户识别码，及组织识别码

④ 更改tmp目录下所有文件和子目录的拥有者跟所属组：

```
# chown -R eagle.users tmp
```



- ⑤ 以file文件为标准，将file1、file2及file3文件的拥有者跟所属组设成与file一致：

```
# chown --reference=file file1 file2 file3
```

↑
以file文件做标准，更改其余文件的拥有者与所属组

chroot (change root)

常用指数：



功能说明：改变根目录。

语法：chroot [--help] [--version] [目的目录] [执行命令...]

包名称：coreutils。

补充说明：把根目录换成指定的目的目录。

参数：

--help	显示帮助
--version	显示版本信息


范 例：

当系统发生错误无法正常开机启动时，我们可借助救援硬盘或安装光盘的救援模式进入Linux系统，将有问题的外围设备挂入后，加以更正修复。例如先把分区编号/dev/hda1的系统根目录 (/) 载入成为/mnt目录，再利用chroot命令把/mnt目录切换成/目录，便于执行各种系统命令与程序：

```
# chroot /mnt
```

↑
把/mnt目录换成/目录（绝对路径而非相对路径）

chsh (change shell)

常用指数: 

功能说明: 更换登录系统时使用的shell。

语法: chsh [-luv] [-s <shell 名称>] [用户名称]

包名称: util-linux。

相关命令: login。

补充说明: 每位用户在登录系统时，都会拥有默认的shell环境，这个命令可更改其默认值。若不指定任何参数与用户名称，则chsh会以问答的方式进行设置。

参数:

-s <shell名称>或--shell <shell名称>	更改系统默认的shell环境
-l或--list-shells	列出当前系统可用的shell列表
-u或--help	显示帮助
-v或--version	显示版本信息

范 例:

① 列出当前可用的shell:

```
# chsh -l
/bin/sh
/bin/bash
/sbin/nologin
...
```

— 当前所有可用的shell

② 采用问答的方式设置shell环境:

```
# chsh ← 不加参数和用户名称，以问答法设置 shell
Changing shell for root.
New shell [/bin/bash]: /bin/csh ← 输入欲使用的shell
Shell changed.
```


③ 指定shadow用户要使用的shell:

```
# chsh -s /bin/bash shadow
Changing shell for shadow.
Shell changed.
```

cksum (check sum)

常用指数: 

功能说明: 检查文件的CRC (Cyclic Redundancy Checking) 是否正确。

语法: cksum [--help] [--version] [文件...]

包名称: coreutils。

相关命令: md5sum, sum。

补充说明: CRC是一种排错检查方式, 该算法的标准由CCITT所制定, 至少可检测到99.998%的已知错误。指定文件交由cksum计算, 它会返回计算结果, 供用户核对文件是否正确无误。若不指定任何文件名称, 或是所给予的文件名为“-”, 则cksum命令会从标准输入设备读取数据。

参数:

--help	显示帮助
--version	显示版本信息

范例:

下载setiathome-3.08.i686-pc-linux-gnu.tar文件之后, 用cksum命令算出该文件的checksum, 然后与网站或FTP站点所提供的checksum比较:

```
# cksum setiathome-3.08.i686-pc-linux-gnu.tar
2866721805 247808 setiathome-3.08.i686-pc-linux-gnu.tar
```



cksum的计算结果, 和网站或FTP站点上所提供的计算结果相互比较后就能得知文件正确与否



文件的大小, 以字节为单位

clear

常用指数: 

功能说明: 清除画面。

语 法: clear

包 名 称: ncurses。

补充说明: 它的作用类似DOS的cls命令，用来清除显示画面。不论在纯文本模式的全屏幕下，或图形界面中的文字模式窗口都可执行。


参 数: 无。

范 例:

将画面清除干净:

```
# clear
```

clock

常用指数: 

功能说明: 调整RTC (Real Time Clock) 时间。

语 法: clock [--adjust] [--debug] [--directisa] [--getepoch] [--hctosys] [--set --date="<日期时间>"] [--setepoch --epoch=<年份>] [--show] [--systohc] [--test] [--utc] [--version]

包 名 称: util-linux。

相关命令: date, hwclock, rdate, timeconfig。

补充说明：RTC是电脑内建的硬件时钟，执行这项命令可以显示现在时刻、调整硬件时钟的时间、将系统时间设成与硬件时钟的时间一致，或是把系统时间回存到硬件时钟。假设你没有指定任何参数给clock命令，则效果和指定参数“--show”相同。

参 数：

--adjust	第一次使用“--set --date”或“--systohc”参数设置硬件时钟，会在/etc目录下产生一个名称为adjtime的文件。当再次使用这两个参数调整硬件时钟，此文件便会记录两次调整之间的差异，日后执行clock命令加上“--adjust”参数时，程序会自动根据记录文件的数值差异，计算出平均值，自动调整硬件时钟的时间
--debug	详细显示命令执行过程，便于debug或了解程序执行的情形
--directisa	告诉clock命令不要通过/dev/rtc设备文件，直接对硬件时钟进行访问。这个参数适用于仅有ISA总线结构的旧电脑
--getepoch	把系统内核内的硬件时钟新时代（Epoch）数值，呈现到标准输出设备（如显示器）。现在只剩下Alpha硬件架构的版本，系统内核才会保留这项数值，如果你不是使用Digital的Alpha硬件平台，此参数将没有任何作用
--hctosys	Hardware Clock to System Time，把系统时间设成和硬件时钟一致。由于这个操作将会造成系统全面更新文件的访问时间，因此最好在系统启动时就先执行它
--set --date= “<日期时间>”	设置硬件时钟的日期和时间。这边的日期与时间格式以硬件时钟为准，年份用两位数字表示，即“00”表示公元2000年，时间则采用24小时制
--setepoch --epoch=<年份>	设置系统核心的硬件时钟的新时代数值，年份以四位数字表示。现在只剩下Alpha硬件架构的版本，系统核心才会保留这项数值，如果不是使用Digital的Alpha硬件平台，此参数将没有任何作用
--show	读取硬件时钟的时间，并将其呈现至标准输出设备（如显示）
--systohc	System Time to Hardware Clock，将系统时间存回到硬件时钟
--test	仅作测试，并不真的将时间写入硬件时钟或系统时间
--utc	把硬件时钟上的时间视为CUT（Coordinated Universal Time），有时也称为UTC或UCT。CUT的前身就是GMT（Greenwich Mean Time），即格林威治标准时间
--version	显示版本信息

范 例:

- ① 把硬件时钟的日期与时间，调整为公元2003年5月29日，下午1时50分25秒：

```
# clock --set --date= "5/29/03 13:50:25"
```

此处是指公元2003年，而非03年

若不指定时间，一律当成0时0分0秒

- ② 查阅硬件时钟的时间：

```
# clock --show
```

← 若无特别标明，则硬件时钟的时间视同本地时间
公元2003年05月29日（周四）14时32分13秒 -0.009090 seconds

```
# clock --show --utc
```

← 把硬件时钟上的时间视为UTC时间，并显示本地时间

公元2003年05月29日（周四）22时32分17秒 -0.904055 seconds

- ③ 调整硬件时钟的时间，执行时显示执行的过程：

```
# clock --adjust --debug
```

```
hwclock from util-linux-2.11x
```

```
Using /dev/rtc interface to clock.
```

```
Last drift adjustment done at 1054190640 seconds after 1969
```

```
Last calibration done at 1054190640 seconds after 1969
```

```
Hardware clock is on UTC time
```

```
Assuming hardware clock is kept in UTC time.
```

```
Waiting for clock tick...
```

```
...got clock tick
```

```
Time read from Hardware Clock: 2003/05/29 06:44:16
```

```
Hw clock time : 2003/05/29 06:44:16 = 1054190656 seconds since 1969
```

```
Time since last adjustment is 16 seconds
```

```
Need to insert -1 seconds and refer time back 0.999895 seconds ago
```

```
Needed adjustment is less than one second, so not setting clock.
```

不需调整时钟


- ④ 调整系统时间跟硬件时钟一致：

```
# clock --hctosys
```

⑤ 将系统时间存回到硬件时钟内：

```
# clock --systemc
```

cmp (compare)

常用指数：

功能说明：比较两个文件是否有差异。

语法：cmp [-clsv] [-i <字符数目>] [--help] [第1个文件] [第2个文件]

包名称：diffutils。

相关命令：diff。

补充说明：当相互比较的两个文件完全一样时，则该命令不会显示任何信息。若发现有所差异，默认会标示出第1个差异处的字节数和行数编号。若不指定任何文件名称，或是所给予的文件名为“-”，则cmp命令会从标准输入设备读取数据。

参 数：

-c 或 --print-chars	除了标明差异处的十进制码之外，一并显示该十进制码所对应的字符
-i <字节数目> 或 --ignore-initial=<字节数目>	指定一个数目，例如该数目是10，则cmp指令会从两个文件的第11个字节起开始比较，而忽略前面10个字节的异同
-l 或 --verbose	标示出所有不一样的地方
-s、--quiet 或 --silent	不显示错误信息
-v 或 --version	显示版本信息
--help	显示帮助

范 例:

① 比较file1与file2两个文件的差异:

cmp file1 file2

← 因为2个文件一致, 所以执行后不显示任何信息

② 对名称为omega和landa的两个文件进行比较:

cmp omega landa

← 以默认的方式比较 2 个文件的差异

omega landa differ: byte 5, line 1

2个文件第一个不同处在第1列的第5个字节

cmp -c omega landa ← 比较2个文件的差异, 若有差异的地方, 列出该字节omega landa differ: byte 5, line 1 is 63 3 143 clanda文件的该字符为
“c”, 十进制码为“143”omega文件的该字符为“3”, 十
进制码为“63”# **cmp -l omega landa**

← 列出2个文件所有差异的地方

```

5  63 143
6  63 143
13 67 147
14 67 147

```

↑ ↑ ↑
 第2个文件和第1个文件的差异字节的列表, 以十进制码标示
 第1个文件和第2个文件的差异字节的列表, 以十进制码标示
 差异处所在编号, 这里表示两个文件的第5、6、13、14字节有差异



cmuwmtopbm

常用指数:



(cmu to portable bitmap)

功能说明: 转换位图文件。**语 法:** cmuwmtopbm [CMU 位图文件]**包 名 称:** netpbm-progs (Red Hat Linux/Fedora Core), netpbm (Mandrake Linux)。**补充说明:** 执行cmuwmtopbm命令，读取指定的CMU Windows管理程序位图文件，并把该文件转成Portable Bitmap（默认最后的扩展名为“.pbm”）格式后，输出到标准输出设备。**参 数:** 无。**范 例:**

将mypicture位图文件转成Portable Bitmap格式:

```
# cmuwmtopbm mypicture > mypicture.pbm
```

col

常用指数:

**功能说明:** 过滤控制字符。**语 法:** col [-bfx] [-l <缓冲区行数>]**包 名 称:** util-linux。**补充说明:** 在许多UNIX说明文件里，都有RLF (Reverse Line Feed) 控制字符。当我们运用shell特殊字符“>”和“>>”，把说明文件的内容输出成为纯文本文件时，控制字符会变成乱码，col命令则能有效滤除这些控制字符。

参 数:

-b	过滤掉所有的控制字符，包括RLF和HRLF (Half Reverse Line Feed)
-f	滤除RLF字符，但允许将HRLF字符呈现出来
-x	以多个空格字符来表示制表符
-l <缓冲区行数>	默认的内存缓冲区有128行，用户可以自行指定缓冲区的大小

范 例:

将命令fsck的帮助存成文本文件:

```
# man fsck | col -b > fsck_manual.txt
```

↑ 建立管道，把man命令的输出结果转为col命令的输入数据

colrm (column remove)

常用指数:



功能说明: 滤掉指定的列。

语 法: colrm [开始列数编号 <结束列数编号>]

包 名 称: util-linux。

相关命令: paste。

补充说明: colrm命令从标准输入设备（如键盘）读取数据，转而输出至标准输出设备（如显示器）。如果不加任何参数，则该命令不会过滤任何一列。若仅指定一个数字，则该数字会当成开始列数编号，从该列数编号起，之后所输入的任何字符都会滤除。

参 数: 无。

范 例:

① 让colrm命令从标准输入设备读取数据，然后显示到标准输出设备:

```
# colrm                                     ← 执行命令，不加任何参数
Linux is Powerful!!                         ← 键入任何文字后，按下 Enter 键
Linux is Powerful!!                         ← 系统响应一模一样的文字
                                           ← 按 Ctrl + C 键退出
# colrm 15                                  ← 从第15列起统统删除，即删除该列第15
                                           列以后的字符
Linux is Powerful!!                         ← 键入相同的文字（“f”字节是该列的第15
                                           个字符），
Linux is Power                               ← 再按下 Enter 键
```

只剩下原先输入的前14列

② 删除numbers文件编号第3及第7列后，再存储为新的文件:

```
# cat numbers                               ← 列出名称为numbers的文件内容
1234567890
1234567890
1234567890
# cat numbers | colrm 3 3 | colrm 6 6 > neonumbers ←
```

因为前面已滤掉一列，故欲滤除原来文件中的第7列，必须指定“6”而非“7”

去掉第3、第7列后，以名称为neonumbers的文件存储结果

comm (common)

常用指数:



功能说明: 比较两个已排序的文件。

语 法: comm [-123] [--help] [--version] [第1个文件] [第2个文件]

包 名 称: coreutils。

相关命令: sort。

补充说明：这项命令会一行行地比较两个已排序（sort）文件的差异，并将其结果显示出来。用户可以使用sort命令将文件内容排序。如果没有指定任何参数，则会把结果分成3列显示：第1列是仅在第1个文件中出现过的行，第2列是仅在第2个文件中出现过的行，第3列则是在第1个与第2个文件里都出现过的行。若给予的文件名称为“-”，则comm命令会从标准输入设备读取数据。

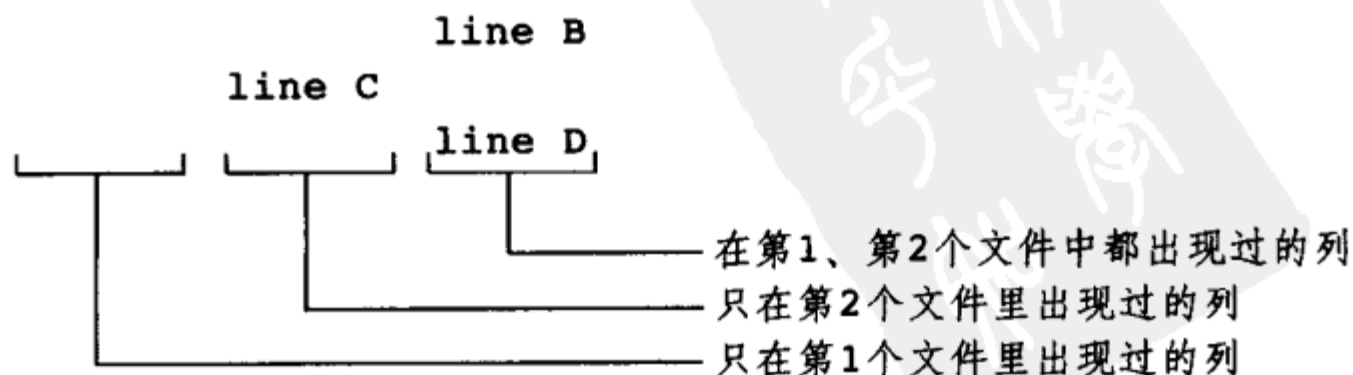
参 数：

-1	不显示只在第1个文件里出现过的行
-2	不显示只在第2个文件里出现过的行
-3	不显示在第1、第2个文件中都出现过的行
--help	显示帮助
--version	显示版本信息

范 例：

① 文件file1和file2都已经排序过，我们可用comm命令得知两文件之间的异同：

```
# cat file1          ← 列出file1的内容
line A
line B
line D
# cat file2          ← 列出file2的内容
line B
line C
line D
# comm file1 file2   ← 比较file1 file2的异同
line A
```

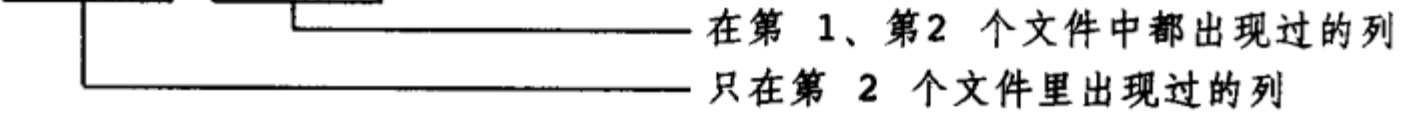


② 接上例，比较file1与file2文件的异同，但不显示只在第1个文件出现过的列：

```
# comm -1 file1 file2
```

```
line B
line C
```

```
line D
```



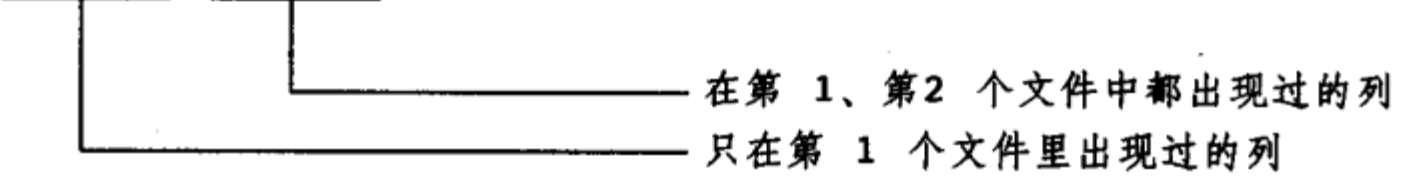
③ 接第1例，比较file1与file2文件的异同，但不显示只在第2个文件出现过的列：

```
# comm -2 file1 file2
```

```
line A
```

```
line B
```

```
line D
```



④ 接第1例，比较file1与file2文件的异同，但不显示在2个文件都出现过的列：


```
# comm -3 file1 file2
```

```
line A
```

```
line C
```



compress

常用指数: 

功能说明: 压缩或解压文件。

语法: `compress [-cdfrvV] [-b <压缩比>] [文件或目录...]`

包名称: ncompress (Red Hat Linux/Fedora Core)。

相关命令: uncompress。

补充说明: 这是Red Hat Linux/Fedora Core才有的命令。compress是个历史悠久的压缩程序，文件经它压缩过后，名称后面会多出“.Z”的扩展名。要解压缩时，可执行uncompress命令。事实上uncompress是compress的符号链接，因此不论压缩或解压缩，都可借助compress命令完成。

参 数:

-b <压缩比>	压缩比是一个介于9~16的数值，默认值为16，指定越大的数值，就会有越大的压缩比。压缩比高，产生的压缩文件就小，但所需的压缩时间也就愈多
-c	把压缩后的文件输出到标准输出设备，不去更改原始文件。默认compress命令会将文件直接压缩后，再加上“.Z”扩展名，而不保留原来的文件。用户可用“>”等特殊字符将压缩结果导向一个新的文件，存储压缩文件却不变动原始文件。假设不把输出导向到某个存储设备，将会在显示器上看见压缩文件的内容，即一堆乱码
-d	对文件进行解压缩。设置此参数后，compress将对具有“.Z”扩展名的文件进行解压缩
-f	强制压缩文件。compress默认不允许压缩具有硬链接的文件。使用此参数可强制压缩此类文件。但需注意的是，解压缩具有连接的文件之后，就失去了原有的连接关系。而当已经存在所要压缩后的文件名称，compress默认会询问用户是否要覆盖文件。用户也可以使用此参数，使得compress不询问意见，即强制覆盖文件
-r	递归处理，将指定目录下的所有文件及子目录一并处理
-v	显示命令执行过程
-V	显示版本信息，及压缩/解压缩文件时的默认值

范 例:

① 压缩file1文件:

```
# ls
```

← 列出当前目录下的文件

```
file1
```

```
# compress file1
```

← 压缩file1文件

```
# ls
```

```
file1.Z
```

← 压缩完后, 将会删除原有file1文件

② 解压缩file1.Z文件:

```
# compress -d file1.Z
```

③ 将文件setiathome压缩后另存新文件, 不要更改原始文件:

```
# compress -c setiathome > setiathome.Z
```

← 使用这种方法只能一个个文件依次压缩, 不能指定目录名称

指定要存储的文件名称

④ 以较省时间 (但压缩比较差) 的方式, 压缩file1文件:

```
# compress -b 9 file1
```

↑
指定较小的压缩比

⑤ 压缩test目录中的文件:

```
# ls test
```

← 列出test目录下的文件有哪些

```
file1 file2
```

```
# compress -r test
```

← 压缩test目录下的文件

```
# ls test
```

```
file1.Z file2.Z
```

← 每个文件分别压缩成.Z的压缩文件

⑥ symbolic_file是一符号链接文件，压缩此文件，然后再解压缩：

```
# ls -l
```

← 查看当前目录下的文件

```
总计8
-rw-r--r--  1 root  root  5060  5月 26 11:56 file
lrwxrwxrwx  1 root  root    4  5月 26 12:04
symbolic_file→file
```

symbolic_file 是指向 file 文件的符号链接

```
# compress -f symbolic_file
```

← 压缩该符号链接文件

```
# compress -d symbolic_file.z
```

← 解压缩该符号链接

```
# ls -l
```

← 再一次查看目录下的文件

```
总计16
-rw-r--r--  1 root  root  5060  5月 26 11:56 file
-rw-r--r--  1 root  root  5060  5月 26 11:56 symbolic_file
```

已经失去原有的连接关系

⑦ hardlink_file是具有硬链接的文件，现强制压缩此文件：

```
# compress hardlink_file
```

← 以默认的方式压缩文件

```
file has 1 other links: unchanged
```

← 此文件已经有其他的连接，强制压缩此文件

```
# compress -f hardlink_file
```

← 不能压缩此文件

convdate (convert date)

常用指数：



功能说明：转换日期时间。

语法：convdate [-cns] [日期时间...]

包名称：inn。

补充说明： convdate 命令可转换普通日期时间和UNIX时间。所谓的UNIX时间是以UNIX操作系统出现的格林威治标准时间公元1970年1月1日开始算起，单位以秒计算。

参 数：

-c	把 UNIX 时间转成普通日期时间
-n	把普通日期时间转成 UNIX 时间
-s	显示完整日期，这是默认值

范 例：

① 把UNIX时间1054263600秒转换成普通的日期时间：

```
# /usr/lib/news/bin/convdate -c 1054263600
Fri May 30 11:00:00 2003
```

② 把普通时间公元2003年1月1日转换成UNIX时间：

```
# /usr/lib/news/bin/convdate -n '00:00:00 01/01/2003' '12am 01/01/2003'
946656000
946656000
```

无论使用12小时或24小时格式都可以。日期的顺序为月/日/年，日期时间的前后要用'号标示起来

③ 以完整的格式显示今年1月1日：

```
# /usr/lib/news/bin/convdate -s '1/1'
Wed Jan 1 00:00:00 2003
```

↑
由于没有指定年份，会以当年（如2003年）为准

cp (copy)

常用指数: **功能说明:** 复制文件或目录。**语 法:** cp [-abdfilprRsuvx] [-S <备份字尾字符串>] [--help] [--sparse=<使用时机>] [--version] [来源文件或目录] [目的文件或目录]或cp [-abdfilprRsuvx] [-S<备份字尾字符串>] [--help] [--sparse=<使用时机>] [--version] [来源文件或目录...] [目的目录]**包名称:** coreutils。**相关命令:** mcopy。**补充说明:** cp命令用在复制文件或目录,如同时指定两个以上的文件或目录,且最后的目的目录是一个已经存在的目录,则它会把前面指定的所有文件或目录复制到该目录中。若同时指定多个文件或目录,而最后的目的地并非是一个已存在的目录,则会出现错误信息。**参 数:**

a或--archive	此参数的效果和同时指定“-dpR”参数相同
-b或--backup	删除、覆盖目的文件之前先备份,备份的文件会在字尾加上一个备份字符串
-d或--no-dereference	复制符号链接时,把目的文件或目录建立为符号链接,并指向来源文件或目录连接的原始文件或目录。假如没有加上这个参数,在复制过程中若遇到符号链接,则会直接复制该连接所指向的原始文件或目录,而非重新建立一个指向该原始文件或目录的符号链接
-f 或 --force	强行复制文件或目录,不论目的文件或目录是否已经存在
-i或--interactive	覆盖有文件之前先询问用户

-l或--link	对来源文件建立硬链接，而非复制文件
-p或--preserve	保留来源文件或目录的属性，包括拥有者、所属组、权限与时间
-r	递归处理，将指定目录下的文件与子目录一并处理。若来源文件或目录的类型，不属于目录或符号链接，则一律视为普通文件处理
-R或--recursive	递归处理，将指定目录下的所有文件及子目录一并处理
-s或--symbolic-link	对来源文件建立符号链接，而非复制文件
-S <备份字尾字符串> 或--suffix=<备份字尾字符串>	用“-b”参数备份目的文件后，备份文件的字尾会加上一个备份字符串。默认的备份字尾字符串是符号“~”，可通过“-S”参数来改变它
-u或--update	使用这项参数之后，只会在来源文件的改变时间 (Modification Time) 比目的文件更新的时候，或是名称相互对应的目的文件不存在时，才复制文件
-v或--verbose	显示命令执行过程
-x或--one-file-system	复制的文件或目录存放的文件系统，必须与cp命令执行时所处的文件系统相同，否则不予以复制。即不处理在其他分区的文件
--help	显示帮助
--sparse=<使用时机>	设置存储稀疏文件 (Sparse File) 的时机。稀疏文件是一种内含大量连续0字节的文件，这种现象称之为坑洞 (Holes)，许多的二进制文件都具有这种特性。假使文件系统有支持这种特性，这些坑洞将不会占用大量的存储块，则对节省存放空间和提高系统性能都有助益。使用时机设为“auto”，则来源文件若是稀疏文件，目的文件也会是稀疏文件，这是cp命令的默认值。使用时机设为“always”，则目的文件将一概存储成稀疏文件。使用时机设为“never”，则目的文件将不会存储成稀疏文件
--version	显示版本信息

范 例:

① 将文件file1复制成文件file2:

```
# cp file1 file2
```

② 采用互动式方式将文件file1复制成文件file2:

```
# cp -i file1 file2
```

cp: 是否覆盖 'file2'? ← 因为目的文件file2已经存在, 所以询问是否要覆盖文件, 可输入“y”表示确定, 输入“no”则取消复制操作

③ 将文件file1复制成文件file2, 因为目的文件已经存在, 所以指定使用强行复制的模式:

```
# cp -f file1 file2
```

↑
强行复制

④ 将目录dir1复制成目录dir2, 目录之间的复制将因目的目录的存在与否而有所差异:

```
# cp -R dir1 dir2
```

如果此目录不存在, 则会建立这个目录, 并把dir1目录下的文件与子目录都复制到dir2中。若该目录已经存在, 则会把整个dir1目录都复制到dir2里, 即将dir1目录下的所有文件与子目录都复制到dir2/dir1中

因为来源是目录, 故须指定这项参数, 否则无法复制目录

⑤ 同时将文件file1、file2、file3与目录dir1复制到dir2:

```
# cp -R file1 file2 file3 dir1 dir2
```

↑
因为来源文件或目录超过一个以上，故此目录必须已经存在，否则会出现错误信息

⑥ 将文件file1复制成文件file2，复制时保留来源文件的属性:

```
# cp file1 file2
```

← 先用默认的方式复制文件 (root账号)

```
# ls -l
```

```
总计 32
```

```
-rwxr-xr-x 1 alkane sphenoid 13980 5月 15 14:43 file1 ←
```

```
-rwxr-xr-x 1 root root 13980 5月 15 16:23 file2 ←
```

```
# cp -p -f file1 file2
```

←
来源文件与复制后
文件属性不同

↑ ↑
强行复制文件

↑
复制文件时，保留来源文件的属性

```
# ls -l
```

```
总计 32
```

←
来源文件与复制后
文件属性一致

```
-rwxr-xr-x 1 alkane sphenoid 13980 5月 15 14:43 file1 ←
```

```
-rwxr-xr-x 1 alkane sphenoid 13980 5月 15 14:43 file2 ←
```

⑦ 复制文件时，只有当来源文件比目的文件的改变时间更新时，才复制文件:

```
# ls -l
```

← 查看目录下的文件

```
总计 48
```

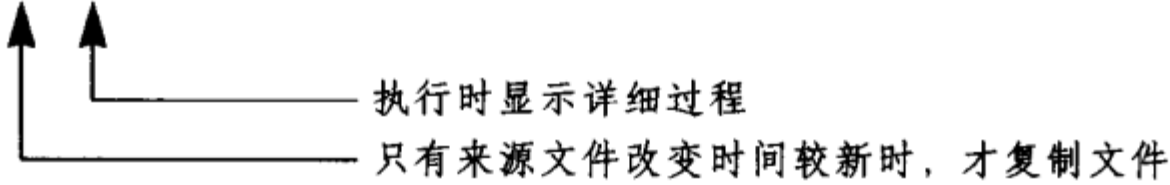
```
-rwxr-xr-x 1 alkane sphenoid 13666 5月 15 16:52 file1
```

```
-rwxr-xr-x 1 alkane sphenoid 13675 5月 15 16:49 file2
```

```
-rwxr-xr-x 1 alkane sphenoid 13627 5月 15 16:53 file3
```

↑
文件更新顺序为: file3、file1、file2

```
# cp -u -v file1 file2
```



```
cp: 是否覆盖 'file2' ? y
```

```
file1 -> file2 ← 由于file1较file2改变时间来的新, 所以会执行复制操作
```

```
# cp -u -v file1 file3 ← 由于file1比file3改变时间来的晚, 所以并不会执行复制操作
```

⑧ 假设文件theda是来源文件, tango是已存在的文件, 在将theda复制成tango时, 想生成备份文件:

```
# cp -b theda tango ← 产生名称为tango~的备份文件
```

```
# cp -b -s _backup theda tango ← 产生名称为tango_backup的备份文件
```

⑨ 来源文件file1为一个符号链接, 当复制文件时, 把目的文件建立为符号链接:

```
# cp -d file1 file2
```

⑩ 复制文件时, 是建立来源文件的硬链接, 而非真的复制文件:


```
# cp -l file1 file2
```

⑪ 复制文件时, 是建立来源文件的符号链接, 而非真的复制文件:

```
# cp -s file1 file2
```



cpio (copy in/out)

常用指数: 

功能说明: 备份文件。

语 法: cpio [-0aABckLovV] [-C <输入/输出大小>] [-F <备份文件>] [-H<备份格式>] [-O <备份文件>] [--block-size=<块大小>] [--force-local] [--help] [--quiet] [--version] 或

cpio [-bBcdfikmnrSuvV] [-C <输入/输出大小>] [-E <范本文件>] [-F <备份文件>] [-H <备份格式>] [-I <备份文件>] [-M <返回信息>] [-R <拥有者><:/.><所属组>] [--block-size=<块大小>] [--force-local] [--help] [--no-absolute-filenames] [--no-preserve-owner] [--only-verify-crc] [--quiet] [--sparse] [--version] [范本样式...] 或

cpio [-0adklLmpuvV] [-R <拥有者><:/.><所属组>] [--help] [--no-preserve-owner] [--quiet] [--sparse] [--version] [目的目录]

相关命令: tar。

包名称: cpio。

补充说明: cpio是用来建立、还原备份文件的工具程序，它可以加入、解开cpio或tar备份文件内的文件。备份的文件可存放在硬盘里、软盘上或是磁带机里。cpio有3种运作模式，分别是copy-out、copy-in和copy-pass。

- 在copy-out模式里，cpio将从标准输入依次读取文件名称列表，然后按照列表所列把文件加入到备份文件内，用户可以利用find命令来列出文件名称列表供cpio命令使用。

- 在copy-in模式里，cpio将从标准输入读取备份文件，然后解开备份文件内的文件，任何与参数无关的字符串都会视为范本样式，用户可指定一个或多个范本样式，以便解开符合范本条件的文件，如果没有指定任何范本样式，则会解开所有文件。
- 在copy-pass模式里，cpio结合了copy-out与copy-in两种模式，用户只需直接指定目的目录，它将从标准输入循序读取文件名称列表，然后按照列表所列把文件复制到目的目录，略过了备份文件的步骤。当前cpio支持的备份格式有8种，分别是：二进制、旧ASCII (American Standard Code for Information Interchange)、新ASCII、CRC、HPUX二进制、HPUX旧ASCII、旧TAR (Tape Archive)，以及POSIX.1 (Portable Operating System Interface) TAR。

参 数:

-0 或 --null	接受换行 (Newline, 即LF+CR) 控制字符, 通常配合find命令的“-print0”参数使用。“-print0”参数会把换行控制字符置换成符号“@”, 搭配cpio的“-0”参数后, 可让某些含有换行控制字符的文件名称能为cpio所接受, 以便备份。此参数在copy-out和copy-pass模式下有效
-a或--reset-access-time	重新设置文件的访问时间
-A或--append	附加到已存在的备份文件中, 且这个备份文件必须存放在硬盘上, 不得放置于磁带机里。此参数在copy-out模式下有效
-b或--swap	此参数的效果和同时指定“-sS”参数相同
-B	将输入/输出的块大小改成5120字节, 默认值是512字节
-c	使用旧ASCII备份格式
-C <块大小>或 --io-size=<块大小>	设置输入/输出的块大小, 单位是字节

-d或--make-directories	如有需要, cpio会自行建立目录
-E <范本文件>或 --pattern-file=<范本文件>	指定范本文件, 其内含有一个或多个范本样式, 让cpio解开符合范本条件的文件, 格式为每行一个范本样式。所谓的范本样式, 就是作为解开备份文件内的文件的名称依据, 例如样式指定为*.txt, 则会还原所有文件名称最后面是“.txt”字符串的文件。此参数在copy-in模式下有效
-f或--nonmatching	让cpio解开所有不符合范本条件的文件。此参数在copy-in模式下有效
-F <备份文件>或 --file=<备份文件>	指定备份文件的名称, 用来取代标准输入或输出, 也能借此通过网络使用另一台主机的存储设备访问备份文件。例如远端主机名称为dark, 你在该主机的账号名称为leader, 欲使用的备份文件名称为seed, 则你只需将备份文件的名称指定成ark:seed或leader@dark:seed, cpio就会去远端主机上存放或读取备份的文件, 前提是你必须在该主机上拥有足够的权限得以访问该存储设备。此参数在copy-out和copy-in模式下有效
-H <备份格式>	指定备份时欲使用的文件格式。在copy-in模式下, 默认会自动检测备份文件的格式; copy-out模式下, 默认的备份格式为二进制。用户可以经由“-H”参数来指定备份格式, 这些格式共有8种, 分别是: bin (二进制)、odc (旧ASCII)、newc (新ASCII)、crc (CRC)、hpbinary (HPUX二进制)、hpodc (HPUX旧ASCII)、tar (旧TAR) 和ustar (POSIX.1 TAR)
-i或--extract	执行copy-in模式, 还原备份文件
-I <备份文件>	指定备份文件的名称, 用来取代标准输入, 也能借此通过网络使用另一台主机的存储设备读取备份文件
-k	此参数将忽略不予处理, 仅负责解决cpio不同版本间的相容性问题
-l或--link	以硬链接的方式取代复制文件, 可在copy-pass模式下运用
-L或--dereference	不建立符号链接, 直接复制该连接所指向的原始文件

-m 或--preserve-modification-time	不去更改文件的改变时间
-M <返回信息> 或 --message=<返回信息>	设置更换存储媒体（如磁带机的磁带，或软驱的软盘）的信息。用户可以自行指定换卷、换盘时所列出的信息，如果内含“%d”字符串，它将由卷册编号（Volume Number）所取代，编号将从1开始依次递增
-n 或--numeric-uid-gid	使用“-tv”参数列出备份文件的内容时，若再加上参数“-n”，则会以用户识别码（User ID）和组织识别码（Group ID）替代拥有者和组名称，列出文件列表。此参数在copy-in模式下有效
-o 或--create	执行copy-out模式，建立备份文件
-O <备份文件>	指定备份文件的名称，用来取代标准输出，也能借此通过网络使用另一台主机的存储设档案放备份文件
-p 或--pass-through	执行copy-pass模式，略过备份步骤，直接将文件复制到目的目录
-r 或--rename	当有文件名称需要更改时，采用互动模式
-R <拥有者><:/.><所属组> 或--owner <拥有者><:/.><所属组>	在copy-in模式还原备份文件，或copy-pass模式复制文件时，可指定这些被备份、复制的文件的拥有者与所属组。用户可以仅设置拥有者、所属组，或两者都指定，拥有者和所属组之间以“:”或“.”号隔开。需注意无论是拥有者或所属组，其名称必须是该系统所认可的名称，也就是已存储在/etc/passwd 与/etc/group设置文件内的合法账号和组
-s 或--swap-bytes	一个字（word）含有两个半字，而每个半字由2字节组成。此参数用来互换每个半字内的2个字节。此参数在copy-in模式下有效
-S 或--swap-halfwords	一个字（word）含有两个半字，而每个半字由2字节组成。此参数用来互换每个字内的2个半字。此参数在copy-in模式有效

-t或--list	将输入的内容呈现出来。利用这项参数，我们可以列出备份文件的内容列表
-u或--unconditional	置换所有文件，不论日期时间的新旧与否，都不予询问而直接覆盖
-v或--verbose	详细显示命令的执行过程。配合参数“-t”使用，可列出更详细的文件列表，其效果类似ls命令的“-l”参数
-V或--dot	执行命令时，在每个文件的执行程序前面加上“.”号
--block-size=<块大小>	设置输入/输出的块大小，以512字节为单位。例如设置为5，则块大小为2560字节，若设置成10，则块大小为5120字节，依此类推
--force-local	强制将备份文件存放在本地主机。在使用“-F”、“-I”或“-O”参数指定备份文件名时，如果文件名称包含有“:”号，cpio默认会使用远端主机访问备份文件，加上这项参数后，即使本地主机文件的名称中有“:”号存在，仍然可强制cpio去使用它
--help	显示帮助
--no-absolute-filenames	使用相对路径建立文件名称。在copy-in模式下，就算备份文件内的文件具有完整的绝对路径文件名，依旧使用相对路径文件名。举例而言，备份文件里存有一个绝对路径文件名为/root/cheka的文件，若用绝对路径文件名去还原它，不论你现在身处哪个目录中，它都会还原至/root目录内。如果是以相对路径文件名去还原它，假设你现在位于/home目录中，则文件cheka会还原到/home/root目录内
--no-preserve-owner	不保留文件的拥有者，谁解开了备份文件，那些文件就归谁所有。此参数在copy-in和copy-pass模式下有效
--only-verify-crc	当备份文件采用CRC备份格式时，可使用这项参数检查备份文件内的每个文件是否正确无误，但不实际解开文件

<code>--quiet</code>	不显示复制了多少块
<code>--sparse</code>	如果一个文件内含大量的连续0字符，则将此文件存成稀疏文件。此参数在copy-out和copy-pass模式下有效
<code>--version</code>	显示版本信息

范 例：

① 从键盘中输入所要备份的文件，并备份到backupfile文件中：

```
# cpio -o -O backupfile
```

↑ ↑
 指定备份文件名称
 执行copy-out模式

file1 ← 输入第1个要备份的文件名称

```
cpio: file1: truncating inode number
```

file2 ← 输入第2个要备份的文件名称

```
cpio: file2: truncating inode number
```

1 block ← 按 **Ctrl + d** 结束

```
#
```

② 列出backfile备份文件内的文件列表：

```
# cpio -t -I backupfile   ← 列出备份文件内的文件列表
```

```
file1
```

```
file2
```

← 此备份文件内有2个文件：file1及file2

```
54 blocks
```

← 此备份文件总共使用了54个块。

```
# cpio -t -v -I backupfile ← 以较详细的方式列出备份文件的文件列表
```

```
-rwxr-xr-x 1 root root 13666 May 17 11:56 file1
```

```
-rwxr-xr-x 1 root root 13666 May 17 11:56 file2
```

```
54 blocks
```

```
# cpio -t -v -n -I backupfile   ← 以较详细的方式列出备份文件的文件列表，列出时以用户识别码和组识别码，替代拥有者和组名称
```

```

-rwxr-xr-x  1 0      0      13666 May 17 11:56 file1
-rwxr-xr-x  1 0      0      13666 May 17 11:56 file2
54 blocks

```

↑ ↑
 用户识别码 组织识别码

③ 解开backfile备份文件:

cpio -i -I backfile

↑ ↑
 执行copy-in模式 指定要解开的备份文件名称

④ 将当前目录中的所有文件，但不包括子目录，备份到backupfile文件中:

ls -a | cpio -o -O backupfile

↑
 不可搭配“--color”参数设置文件列表的颜色，因为指定颜色将造成输入cpio命令的文件名称混杂ANSI控制码，发生错误

⑤ 将当前目录中的所有文件，但不包括子目录，备份到backfile文件中，若当前目录下有符号链接，则备份时不建立符号链接，直接复制该连接所指向的原始文件:

ls -a | cpio -L -o -O backfile

↑
 不建立符号链接，直接复制原始文件

⑥ 将当前目录中的所有文件，但不包括子目录，备份到软盘中:

ls -a | cpio -o -O /dev/fd0

↑
 指定软驱的代号



⑦ 将软盘 /dev/fd0 的备份文件解开到当前目录下:

```
# cpio -i -I /dev/fd0
```

⑧ 以旧ASCII的格式备份当前目录下的所有文件,但不包括子目录:

```
# ls -a | cpio -o -c -O backfile
```

↑
指定使用旧ASCII格式

⑨ 以CRC的格式备份当前目录下的所有文件,但不包括子目录;然后检查该备份文件的正确性:

```
# ls -a | cpio -o -H crc -O backfile
```

↑
指定使用CRC格式

```
55 blocks
```

```
# cpio --only-verify-crc -i -I backfile
```

```
55 blocks
```

↑
检查CRC格式备份文件的正确性

⑩ 备份当前目录下的全部文件,包括子目录,备份到backfile文件中:

```
# find . -depth -print | cpio -o > backupfile
```

↑
不使用-O参数,改用特殊字符">"将标准输出导向backfile文件



- ⑩ 备份当前目录下文件名称以file开头的文件，备份时利用-B参数指定一个块大小由默认的512字节改成5120字节：

```
# ls file* | cpio -o -B -O backfile
```

↑
指定块大小为5120 Bytes

- ⑪ 备份当前目录下文件名称以file开头的文件，备份时利用-C参数指定一个块大小为2560字节：

```
# ls file* | cpio -o -C 2560 -O backfile
```

↑
指定块大小为2560字节

- ⑫ 备份当前目录下文件名称以file开头的文件，备份时利用--block-size参数指定一个块大小为4096字节：

```
# ls file* | cpio -o -O backfile --block-size=8
```

↑
指定块大小为512字节 * 8

- ⑬ 从键盘输入要备份的文件，并将输入文件附加到原有的 backfile 备份文件中：

```
# cpio -A -o -O backfile
```

↑
将要备份的文件附加在已存在的备份文件中

- ⑭ 解开backfile备份文件时，保留原有文件的改变时间：

```
# cpio -i -I backfile ← 先以默认的方式解开备份文件
```

```
54 blocks
```

```
# ls -l
```

```
total 60
```

```
-rw-rw-rw- 1 root root 27648 5月 17 11:04 backfile
-rwxr-xr-x 1 root root 13666 5月 17 11:13 file1
-rwxr-xr-x 1 root root 13666 5月 17 11:13 file2
```

↑
解开文件file1及file2的改变时间为当时的系统时间

```
# rm file1 file2          ← 删除解开的文件
# cpio -i -m -I backfile ← 解开文件时保留原有文件的改变时间
54 blocks
```

```
# ls -l
```

总计 60

```
-rw-rw-rw- 1 root root 27648 5月 17 11:04 backfile
-rwxr-xr-x 1 root root 13666 5月 17 10:59 file1
-rwxr-xr-x 1 root root 13666 5月 17 10:59 file2
```

↑
解开文件的改变时间为原来文件的改变时间

⑥ 解开备份文件时，若当前目录下已有重复的文件，直接置换文件：

```
# ls          ← 列出当前目录下的文件
```

```
backfile file1 file2
```

```
# cpio -i -I backfile ← 以默认的方式解开文件
```

```
cpio: file1 not created: newer or same age version exists ←
```

```
cpio: file2 not created: newer or same age version exists ←
```

因为当前目录下已有相同的文件，故不覆盖文件

```
54 blocks
```

```
# cpio -i -u -I backfile ← 若当前目录下已有重复的文件，
                        则直接置换文件
```

```
54 blocks ← 成功地覆盖文件
```

⑦ 解开备份文件时，设置文件的拥有者与所属组：

```
# cpio -i -d -R shadow.users < backupfile
```

↑
使用特殊字符“<”从
标准输入引进文件

↑
只想指定拥有者，用“shadow”即可，只想指定
所属组，则用“.users”或“:users”都可以

↑
解开文件时，如果需建立目录则由 cpio 自行建立

⑧ 将名称为 SETI 的目录复制到 /var/tmp 目录下：

```
# find /SETI | cpio -p /var/tmp
```

↑ 直接指定目的目录即可
↑ 执行 copy-pass 模式


C

⑨ 以硬链接的方式将名称为 TEST 的目录复制到 /var/tmp 目录下：

```
# find /TEST | cpio -p -l /var/tmp
```

↑ 以硬链接的方式复制文件

crontab

常用指数：

功能说明： 设置计时器。

语 法： crontab [-u <用户名称>] [设置文件] 或

crontab [-u <用户名称>] [-elr]

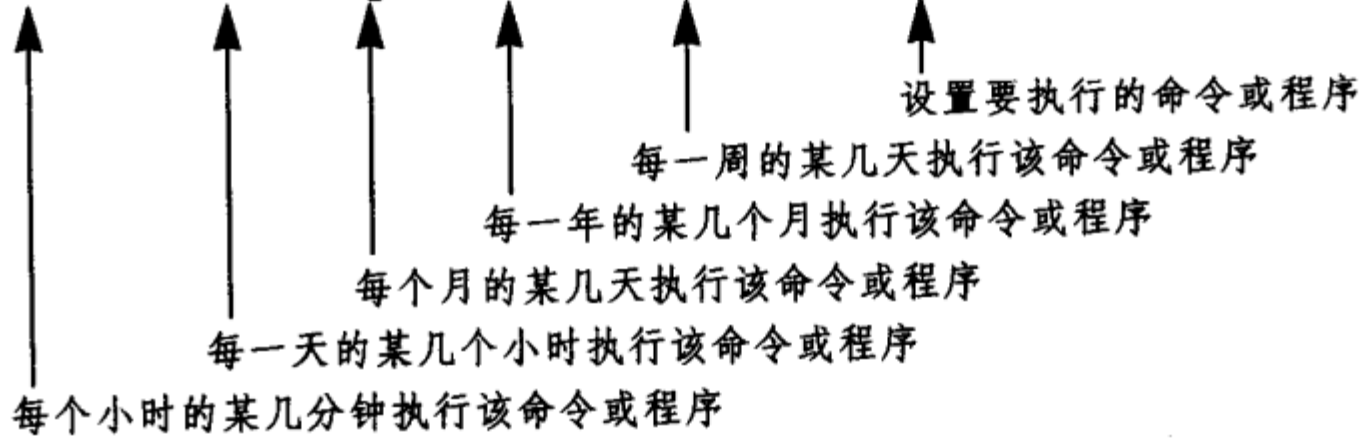
包 名 称： vixie-cron。

相关命令： at。

补充说明： cron 是一个常驻服务，它提供计时器的功能，让用户在特定的时间得以执行默认的命令或程序。只要用户会编辑计时器的设置文件，就可以使用计时器的功能。用户可以用 crontab 命令来设置所执行的工作。其设置文件的格式如下。



Minute Hour Day Month DayOfWeek Command



参 数:

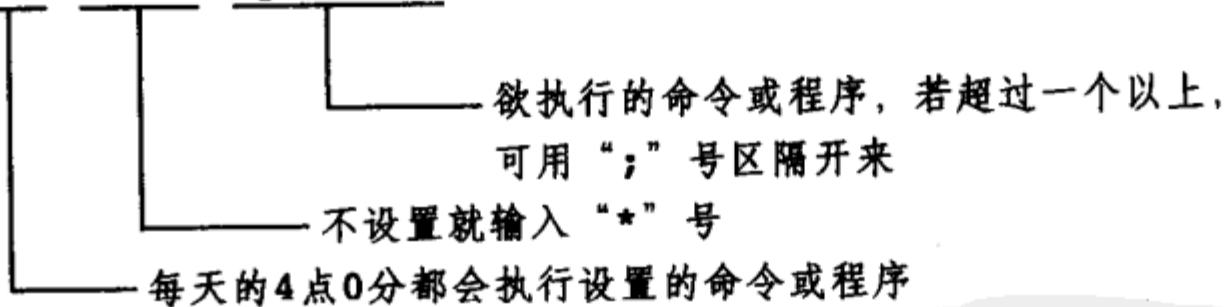
-e	编辑该用户的计时器设置
-l	列出该用户的计时器设置
-r	删除该用户的计时器设置
-u <用户名称>	指定要设置计时器的用户名称

范 例:

- ① 设置计时器的设置文件，文件名称为cronset（文件名可自定义），使得每天凌晨4点自动执行updatedb命令：

cat cronset ← 观看设置文件内容

0 4 * * * updatedb



crontab cronset ← 套用cronset设置文件

② 列出当前计时器的设置:

```
# crontab -l
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (cronset installed on Fri May 26 22:14:04 2000) ←启动设置文件的时间
# (Cron version -- #Id: crontab.c,v 2.13 1994/01/17
# 03:20:37 vixie Exp $) ←crontab的版本
0 4 * * * updatedb ←设置文件的内容
```

③ 编辑用户计时器的设置文件:

```
# crontab -e ←会启动vi编辑器来编辑设置文件
```

④ 删除使用计时器的设置:

```
# crontab -r ←删除计时器设置
# crontab -l ←列出当前的计时器设置
no crontab for root ←当前计时器已经没有任何设置值
```


⑤ 设置计时器的设置文件, 让系统在特定时间自动发送电子邮件, 最后并指定执行此命令的用户为 deathlord:

```
# cat cronset
00 18-23,0-7 * * Mon,Tue,Wed,Thu,Fri sendmail -q ←
30 * * * * Sat,Sun sendmail -q ←
    每周六、日的00时30分开始传送邮件, 每个小时的30分就会传送一次
    每周一至周五的18时整开始传送邮件, 每个小时整点就会传送一次,
    到第二天早上7时为止, 这样重复执行到周五的23时为止
```

```
# crontab -u deathlord cronset
```

↑
指定用户为deathlord

csplit (context split)

常用指数: 

功能说明: 分割文件。

语法: `csplit [-kqsz] [-b <输出格式>] [-f <输出字首字符串>] [-n <输出文件名位数>] [--help] [--version] [文件] [范本样式...]`

包名称: coreutils。

相关命令: cat, split。

补充说明: 将文件按照指定的范本样式予以切割后, 分别存储成名称为xx00、xx01、xx02...的文件。若给予的文件名称为“-”, 则csplit命令会从标准输入设备读取数据。csplit命令能够使用的范本样式条列如下。

- **<分割行数>:** 指定一个行数, csplit将以该行为分割点, 在该行之前的所有内容将存储成一个文件, 该行及该行之后的内容将存成另一个文件。搭配执行次数的范本样式使用, 就会重复进行分割操作。例如设置的分割行数为100, 则第100行前面的99行会存储为第1个分割文件, 重复进行分割时, csplit会把第100行再当作第1行, 然后重复前述的分割操作, 直到达成执行次数或文件的剩余部分已不足指定的分割行数为止。
- **/<分界字符串>/<偏移量>:** 指定一个分界字符串, 在该字符串所在列数之前的所有内容将存储成一个文件, 该字符串所在的行及该行之后的内容将存成另一个文件。至于偏移量是由一个“+”或“-”号搭配一个数字, 例如设置偏移量为“+3”, 即表示由该字符串所在行数往下3行, 才是分割点, 而“-3”则是往上3行。

- %<分界字符串>%<偏移量>: 和上一个功能大致类似, 差别在于此范本样式不会产生文件, 也就是该段内容将忽略不予处理。
- {<执行次数>}: 设置反复进行分割操作的次数, csplit会重复指定的范本样式的分割操作, 直到达成执行次数或文件的剩余部分已不足指定的分割行数为止。假如使用“*”号当成执行次数, 则会不断分割文件, 直到文件的剩余部分已不够分割为止。

参 数:

-b <输出格式>或 --suffix-format=<输出格式>	默认的输出格式的文件名称为xx00、xx01...等, 如果指定输出格式为“%d”(或“%i”、“%o”、“%u”、“%x”、“%X”都行), 则输出的文件名称会变成xx0、xx1..., 若设置为“%dmyfile”, 则输出文件名变成xx0myfile、xx1myfile等, 依此类推。如果使用了这项参数, 则参数“-n”将忽略
	前述的%d与%i表示十进制数, 而%u表示无负值的十进制数, %o表示八进制数, %x与%X表示十六进制数
-f <输出字首字符串>或 --prefix=<输出字首字符串>	默认的输出字首字符串的文件名称为xx00、xx01等, 如果指定输出字首字符串为“hello”, 则输出的文件名称会变成hello00、hello01等, 依此类推
-k或--keep-files	保留文件, 就算发生错误或中断执行, 也不删除已经输出存储的文件
-n <输出文件名位数>或 --digits=<输出文件名位数>	默认的输出文件名位数的文件名称为xx00、xx01等, 如果指定输出文件名位数为“3”, 则输出的文件名称会变成xx000、xx001等, 依此类推
-q、-s、--quiet或--silent	不显示命令执行过程
-z或--elide-empty-files	删除长度为0字节的文件
--help	显示帮助
--version	显示版本信息

范 例:

① 将文字文件textfile以120行为分界点切割成2份:

```
# csplit textfile 120
```

↑
以120行为分界点

7301 ← 显示切割后第1个文件的大小, 以字节为单位

6365 ← 显示切割后第2个文件的大小, 以字节为单位

```
# ls
```

```
textfile xx00 xx01
```

↑ ↑
切割后的文件名称

② 将文字文件textfile以120行为分割点切割成2份, 并指定输出文件名位数为3位数:

```
# csplit -n 3 textfile 120
```

```
# ls
```

```
textfile xx000 xx001
```

↑ ↑
输出文件名位数已经变成3位数

③ 将文字文件textfile以120行为分割点切割成2份, 并指定输出文件名的字首为“FileName”:

```
# csplit -f FileName textfile 120
```

```
# ls
```

```
FileName00 FileName01 textfile
```

↑ ↑
输出的文件均以“FileName”为文件名的字首

④ 将文字文件textfile以10行为分割点切割8次, 并自行指定输出文件名的格式:

```
# csplit -b "myfile%o" textfile 10 {8}
```

↑
%o表示八进制
文件名的字首以默认的xx加上myfile
指定文件名的输出格式

⑤ 将文字文件textfile以每20行为单位切割：

```
# csplit -k textfile 20 {*}
```

↑
重复执行分割操作，直到文件剩余部分不足 20 行为止

⑥ 文件report是一份纯文本格式的报告文件，我们要以第X章（假设其内容为“Chapter X”）为分界，将报告分成前后两个部分：

```
# csplit report /"Chapter X"/
```

↓
设置分界字符串，因为字符串中含有空格，所以前后要加上"

⑦ 接上例，但分割文件时以“Chapter X”字符串往下4行才是分割点：

```
# csplit report /"Chapter X"/+4
```

↑
由分界字符串所在的列数往下数4行

ctlinnd

常用指数：



(control the internet news daemon)

功能说明：设置INN新闻组服务器。

语法：ctlinnd [-hs] [-t <超时秒数>] [控制命令 <设置数值>...]

包名称：inn。

相关命令：archive, inncheck。

补充说明：通过ctlinnd命令，可以对新闻组服务器这项常驻服务下达控制命令，直接设置相关数值。

参 数:

-h	显示帮助
-s	不显示命令执行过程
-t <超时秒数>	设置等待服务器回报的时间，单位以秒计算，超过此时间则为超时。当指定数值为“0”时，表示永远等待，数值小于“0”表示无需等待。如有设置超时秒数，则ctlinnd将会每2分钟检查一次，确认该服务器是否仍在执行

范 例:

检查INN新闻组服务器newsfeed设置文件语法的正确性:

```
# ctlinnd checkfile
```

```
Ok ← 语法正确无误
```

CU (call up)

常用指数:



功能说明: 连接另一个系统主机。

语 法: cu [-dehnotv] [-a <通信端口>] [-c <电话号码>] [-E <脱离字符>] [-I <设置文件>] [-l <外围设备代号>] [-s <连接速率>] [-x <排错模式>] [-z <系统主机>] [--help] [--nostop] [--parity=none] [<系统主机>/<电话号码>]

包 名 称: uucp。

相关命令: uucp。

补充说明: 本命令可连接另一台主机，并采用类似拨号连接终端的界面工作，也可以执行简易的文件传输操作（没有查错排错的通信协议）。假如给予其电话号码，则该命令将通过电话拨号连接，以终端类型连接远端主机。

参 数:

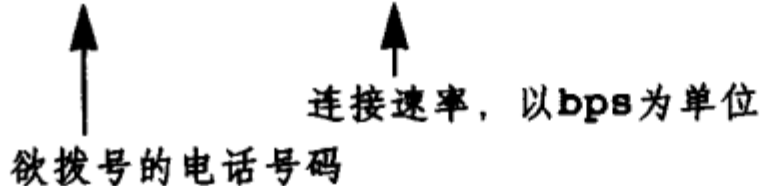
-a <通信端口>、 -p <通信端口>或 --port <通信端口>	使用指定的通信端口进行连接
-c <电话号码>或 --phone <电话号码>	拨号连接该电话号码
-d	进入排错模式, 功用相当于参数 "--debug all"
-e或--parity=even	使用双同位检查
-E <脱离字符>或 --escape <脱离字符>	设置脱离字符 (Escape)。默认值为 "~", 可通过此项参数更改。欲清除脱离字符, 将其指定成 "" 即可
-h 或 --halfduplex	使用半双工模式
-I <设置文件>或 --config <设置文件>	指定要使用的设置文件。默认的设置文件位于/etc/uucp目录里, 文件名称为config
-l <外围设备代号>或 --line <外围设备代号>	指定某项外围设备, 作为连接的设备。假如该设备不存在于UUCP (Unix to Unix Copy Protocol) 的设置文件中, 但用户对该外围设备拥有访问权, 便可用这个参数设置它
-n或--prompt	拨号时等待用户输入电话号码
-o或--parity=odd	使用单同位检查
-s <连接速率>、 --speed<连接速率>、 --baud<连接速率>或 <连接速率>	设置连接的速率, 单位以鲍率 (Baud Rate) 计算。鲍率原指每秒电子状态变换的次数, 后来因bit/s更能表示传输速率的含义, 所以就改用bit/s表示传输速率。现在一般所称的鲍率, 其实是指bit/s
-t或--mapcr	把CR (Carriage Return) 字符置换成LF+CR字符
-v或--version	显示版本信息
-x <排错模式>或 --debug <排错模式>	排错模式分为数种: "abnormal"、"chat"、"handshake"、"uucp-proto"、"proto"、"port"、"config"、"spooldir"、"execute"、"incoming" 以及 "outgoing"。cu命令仅用到 "abnormal"、"chat"、"handshake"、"port"、"config"、"incoming" 和 "outgoing"。你可以同时启动多个排错模式, 例如设置排错模式为 "chat, port" 就是同时使用两项排错。也能采用数字类型来指定, 比如排错模式设置为 "2", 即表示同时使用前2项排错 "abnormal" 和 "chat"

-z <系统主机> 或 --system <系统主机>	连接该系统主机
--help	显示帮助
--nosp	关闭Xon/Xoff软件流量控制，这是默认值
--parity=none	不使用同位检查，同时设置参数“-e”和“-o”亦有相同效果

范 例：

① 使用电话拨号与远端主机连接，并设置连接速率为 14400：

```
# cu -c 12345678 -s 14400
```



 欲拨号的电话号码 连接速率，以bps为单位

② 连接名称为shadow的远端主机：

```
# cu shadow
```

cut

常用指数：

功能说明：指定欲显示的文件内容，输出到标准输出设备。

语 法：cut [-b <输出范围>] [-n] [--help] [--version] [文件...] 或
 cut [-c <输出范围>] [--help] [--version] [文件...] 或
 cut [-d <分界字符>] [-f <输出范围>] [-s] [--help] [--version] [文件...]

包名称：coreutils。

相关命令：cat, head, tac, tail。

补充说明：cut命令一行行地读入文件内容，然后把符合指定条件的内容输出至标准设备（如显示器）上。若不指定任何文件名称，或是所给予的文件名为“-”，则cut命令会从标准输入设备读取数据。

参 数：

<p>-b <输出范围> 或 --bytes=<输出范围></p>	<p>设置欲输出至标准输出设备的范围，以Byte为单位，范围则以每行的列为准。例如指定的输出范围为“7”，就会显示（以显示器为标准输出设备）每行第7列的字符；指定的输出范围为“7-”，则会显示每行第7列起至最后一列的所有字符；指定的输出范围为“-7”，就显示每行第1列起至第7列为止的所有字符；指定的输出范围为“3-7”，则显示每行第3列起至第7列为止的所有字符；指定的输出范围为“3,7”，将显示每行第3列和第7列的字符。其中制表符和回退字符（Back Space），都视同为1个Byte</p>
<p>-c <输出范围> 或 --characters=<输出范围></p>	<p>设置欲输出至标准输出设备的范围，以字节为单位，范围则以每行的列为准。当前这项参数的效果类似“-b”参数，预留给未来操作系统国际化时使用。其中制表符和回退字符看作同为1个字符</p>
<p>-d <分界字符> 或 --delimiter=<分界字符></p>	<p>指定列的分界字符。默认值以制表符当作列分界字符，可以通过此参数来指定它。这项参数仅在配合参数“-f”时方能使用</p>
<p>-f <输出范围> 或 --fields=<输出范围></p>	<p>设置欲输出至标准输出设备的范围，以列为单位，范围则以每行的列为准。默认值以制表符当作列分界字符</p>
<p>-n</p>	<p>不分割多字符（Multi-Byte）的字符，此参数当前尚无作用</p>
<p>-s 或 --only-delimited</p>	<p>如果该列没有分界字符存在，则不输出该列的内容。这项参数仅在配合参数“-f”时才能使用</p>
<p>--help</p>	<p>显示帮助</p>
<p>--version</p>	<p>显示版本信息</p>

范 例:

文件sample里面包含了制表符与一般空格字符,运用各种参数将有不一样的效果:

```
# cat sample                                ←列出文件sample的内容
12345:67890:abcde:fg hij                    ←用“:”号作为分界字符
12345    67890    abcde    fg hij          ←用制表符当分界
ABCDE    FGHIJ    KLMNO    PQRST         ←普通的空格字符
12345    67890    abcde    fg hij          ←用制表符当分界
# cut -b 3-5,7,9 sample                       ←列出第3~5、第7、第9个字符
34568
34568                                        ←制表符当作1个字节处理
CDEF                                        ←每一个空格字符都当作1个字节处理
34568
# cut -b 7- sample                             ←列出每行第7个以后的字符
67890:abcde:fg hij
67890    abcde    fg hij
FGHIJ    KLMNO    PQRST
67890    abcde    fg hij
# cut -f 1,3 sample                             ←列出第1、第3列
12345:67890:abcde:fg hij                    ←没有制表符,因此全部视为1个列
12345    abcde                                       ←默认的分界字符为制表符,因此只
                                                    显示出指定的列
ABCDE    FGHIJ    KLMNO    PQRST           ←没有制表符,因此全部视为1个列
12345    abcde
# cut -f 1,3 -s sample                         ←列出第1、第3列,且不列出不含分
                                                    界字符者
12345    abcde                                       ←只显示了以制表符为分界字符的内容
12345    abcde
# cut -f 1,3 -d : sample                       ←指定分界字符为“:”号
12345:abcde                                       ←含有“:”号者,仅显示出第1、第3列
12345    67890    abcde    fg hij
ABCDE    FGHIJ    KLMNO    PQRST           } 没有指定的分界字符,
12345    67890    abcde    fg hij           } 因此全部视为1个列
```


date

常用指数: **功能说明:** 显示或设置系统时间与日期。**语法:** date [-u] [-d <字符串>] [+ %a %A %b %B %c %d %D %e %H %I %j %k %l %m %M %n %P %r %s %S %t %T %U %w %x %X %y %Y %Z] 或

date [-u] [-s <字符串>] [MMDDhhmmCCYYss] 或 date [--help] [--version]

包名称: coreutils。**相关命令:** clock, hwclock, rdate, timeconfig。**补充说明:** 第一种语法可用来显示系统日期或时间, 以%为开头的参数为格式参数, 可指定日期或时间的显示格式。第二种语法可用来设置系统日期与时间。只有管理员才有设置日期与时间的权限。

若不加任何参数, date会以%c格式参数显示当前的日期与时间。

参数:

%a	星期的简要名称
%A	星期的完整名称
%b	月份的简要名称
%B	月份的完整名称
%c	显示系统的日期与时间, 其格式为%a %b %e %T %Z。若只输入date命令也会显示同样的结果
%d	日期 (以01~31来表示)
%D	日期 (含年月日), 其格式为%m/%D/%Y
%e	该月中的第几天
%H	小时 (以00~23来表示)

%I	小时 (以01~12来表示)
%j	该年中的第几天
%k	小时 (以0~23来表示)
%l	小时 (以0~12来表示)
%m	月份 (以01~12来表示)
%M	分钟 (以00~59来表示)
%n	在显示时, 插入新的一行
%P	上午或下午 (AM或PM)
%r	时间 (含时分秒, 小时以12小时, 上午/下午来表示)
%s	总秒数。起始时间为1970-01-01 00:00:00 UTC
%S	秒 (以00~59来表示)
%t	在显示时, 插入tab
%T	时间 (含时分秒, 小时以24小时制来表示)
%U	该年中的第几周
%w	该周的第几天。0表示周日, 1表示周一, 依此类推
%x	日期 (以本地的惯用法来表示)
%X	时间 (以本地的惯用法来表示)
%y	年份 (以00~99来表示)
%Y	年份 (以四位数来表示)
%Z	时区
MM	月份 (必要)
DD	日期 (必要)
hh	小时 (必要)
mm	分钟 (必要)
CC	年份的前两位数 (选择性)
YY	年份的后两位数 (选择性)
ss	秒 (选择性)
-d <字符串> 或 --date=<字符串>	显示字符串所指的日期与时间。字符串前后必须加上双引号

d

-s <字符串> 或 --set=<字符串>	根据字符串来设置日期与时间。字符串前后必须加上双引号。使用此-s参数时，后面便不用再加上[MMDDhhmm CCYYss]的参数
-u	显示GMT（格林威治标准时间）
--help	显示在线帮助
--version	显示版本信息

D

范 例：

① 显示当前系统的日期与时间：

```
# date
```

```
— 6月 2 13:42:48 CST 2003
```

② 以格林威治标准时间显示当前系统的日期与时间：

```
# date -u
```

```
— 6月 2 05:44:05 UTC 2003
```

③ 显示当前系统的日期：

```
# date +%D
```

```
06/02/03 ← 2003年6月02日
```

④ 以本地的惯用法来显示当前系统的日期：

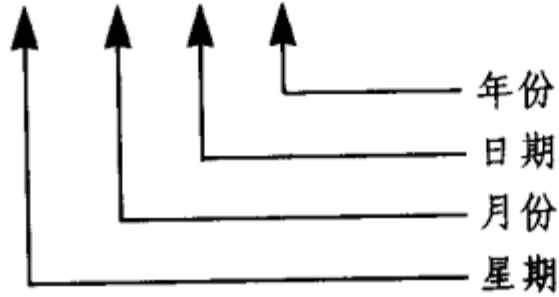
```
# date +%x
```

```
公元2003年06月02日
```



- ⑤ 以自定义的格式显示当前系统的星期、月份（以简要名称显示）、日期及2位数的年份：

```
# date "+%a %b %d %y"
```

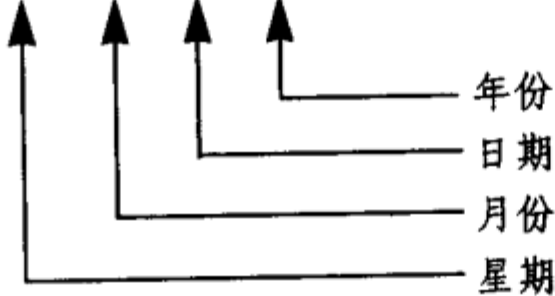


← 由于日期参数中包括了空白，所以前后要加上双引号

— 6月 02 03

- ⑥ 以自定义的格式显示当前系统的星期、月份（以完整名称显示）、日期及4位数的年份：

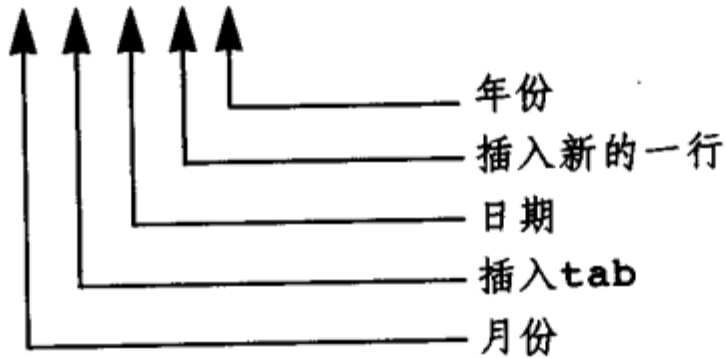
```
# date "+%A %B %d %Y"
```



周一 六月 02 2003

- ⑦ 以自定义的格式来显示当前系统的日期，以数字来显示月份，接着插入一个tab，随后加上日期，紧接着再插入新的一行，最后则以2位数显示当前年份：

```
# date +%m\t%d\n%y
```



06

02

03

⑧ 以12小时制来显示当前系统的时间：

date +%r

03:05:54下午 ← 下午3点05分54秒

⑨ 以24小时制来显示当前系统的时间：

date +%T

15:06:58

D

⑩ 以本地的惯用法来显示当前系统的时间：

date +%X

15时18分21秒

⑪ 以自定义的格式来显示当前系统的时间，小时以24小时制显示，时、分、秒以“:”隔开：

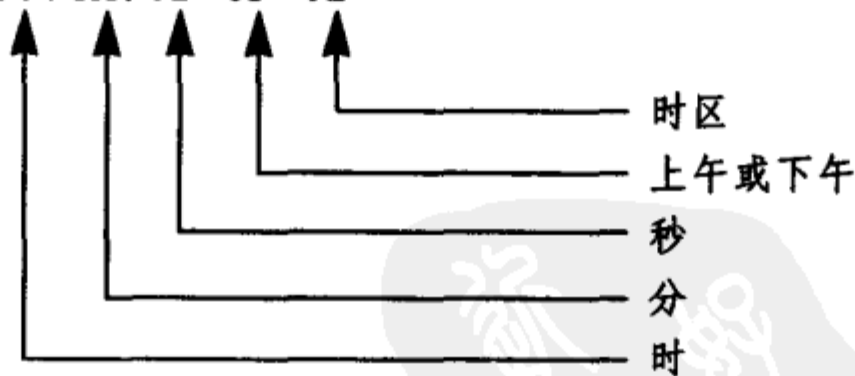
date +%H:%M:%S



16:28:24

⑫ 以自定义的格式来显示当前系统的时间，小时以12小时制显示，时、分、秒以“:”隔开，最后则加上上午或下午，及所在的时区：

date "+%I:%M:%S %P %Z"



04:33:11下午CST

⑬ 显示本周是今年的第几周：

```
# date +%U
22          ← 今年的第22周
```

⑭ 显示今天是本周的第几天：

```
# date +%w
2          ← 本周的第2天，即星期二
```

⑮ 显示今天是今年的第几天：

```
# date +%j
154       ← 今天是今年的第154天
```

⑯ 显示从1970/1/1开始到当前的总秒数：

```
# date +%s
1054624973
```

⑰ 显示7天前的日期与时间：

```
# date -d "7 days ago"
二  5月 27 15:23:40 CST 2003
# date -d "7 days ago" +%j%t%U
```

← 7天前的日期时间
← 显示7天前本年度的总日数与总周数
↑ 显示总周数
↑ 插入tab
↑ 显示总日数

```
147    21
```

↑ ↑
7天前为今年的第21周
7天前为今年的第147天

D



d

⑧ 显示4个月又2天后的日期与时间：

```
# date -d "4 months 2 day"
```

```
日 10月5 16:45:59 CST 2003
```

4个月又2天后是10月5日

⑨ 显示今年6月7日的日期与时间：

```
# date -d "7 Jun"
```

```
六 6月 7 00:00:00 CST 2003
```

⑩ 显示系统当前的日期与时间：

```
# date -d now
```

```
二 6月 3 16:54:04 CST 2003
```

⑪ 将系统时间设为2003年12月31号23点59分30秒：

```
# date 123123592003.30
```

```
三 12月 31 23:59:30 CST 2003
```

← 必须是管理员才有权限
设置时间与日期

← 设置后的时间

⑫ 将系统日期设为1年后的日期：

```
# date
```

```
二 6月3 17:03:03 CST 2003
```

```
# date -s "1 year"
```

```
四 6月3 17:03:15 CST 2004
```

← 现在的日期时间

← 设为1年后的日期

⑬ 将系统时间设为5分钟后的时间：

```
# date
```

```
二 6月3 17:08:44 CST 2003
```

```
# date -s "+5 minutes"
```

```
二 6月3 18 17:13:46 CST 2003
```

← 现在的日期时间


← 设为5分钟后的时间

D

④ 将系统日期设成2003年6月7日，时区为CST：

```
# date -s "2003-06-07 CST"
六 6月 7 00:00:00 CST 2003
```

dd

常用指数：

功能说明：读取、转换并输出数据。

语 法：dd [bs=<字节数>] [cbs=<字节数>] [conv=<关键字>] [count=<块数>] [ibs=<字节数>] [if=<文件>] [obs=<字节数>] [of=<文件>] [seek=<块数>] [skip=<块数>][--help] [--version]

包名称：coreutils。

补充说明：dd可从标准输入或文件读取数据，根据指定的格式来转换数据，再输出到文件、设备或标准输出。

参 数：

bs=<字节数>	将ibs（输入）与obs（输出）设成指定的字节数
cbs=<字节数>	每次只转换指定的字节数
conv=<关键字>	指定文件转换的方式。可用的选项如下： ascii 将EBCDIC转成ASCII ebclic 将ASCII转成EBCDIC ibm 将ASCII转成IBM EBCDIC block 将换行符号取代为cbs数目的空白字符 unblock 若读取到cbs数目的空白字符，则以换行符号来取代 lcase 将大写字母改为小写 notrunc 不要截断输出文件 ucase 将小写字母改为大写 swab 输入的字节每两个一组，彼此对调 noerror 即使读取时有误，仍然继续执行 sync 以NUL填满输入的块，使块与ibs的大小相同

count=<块数>	仅读取指定的块数
ibs=<字节数>	每次读取的字节数
if=<文件>	从文件读取。若未指定此参数，则从标准输入设备读取数据
obs=<字节数>	每次输出的字节数
of=<文件>	输出至文件。若未指定此参数，则输出至屏幕
seek=<块数>	一开始输出时，跳过指定的块数。块与obs的大小相同
skip=<块数>	一开始读取时，跳过指定的块数。块与ibs的大小相同
--help	显示帮助
--version	显示版本信息

范 例：

① 在Linux下，制作启动盘：

```
# dd if=boot.img of=/dev/fd0 bs=1440k
```

② 将lofile文件中的所有英文字母转换为大写，然后存储成upfile：

```
# cat lofile ← 显示lofile的内容
```

```
all those moments would be lost in times,
like tears in the rain.
time to die.
```

lofile中都是小写字母

```
# dd if=lofile of=upfile conv=ucase
```

↑
将所有字母转换为大写

读入了0+1个区段

输出了0+1个区段

```
# cat upfile
```

```
ALL THOSE MOMENTS WOULD BE LOST IN TIMES,
LIKE TEARS IN THE RAIN.
TIME TO DIE.
```

upfile中都是大写的字母

③ 接上例，将upfile文件中所有的英文字转换为小写，然后存储为lowfile:

```
# dd if=upfile of=lowfile conv=lcase
```

↑
将所有字母转换为小写

④ 由标准输入设备（键盘）读入字符串，将字符串转成大写后，再输出到标准输出设备（屏幕）:

```
# dd conv=ucase ← 执行后可按下 [ ] + [ ] 结束所要输入的字符串
```



⑤ 将ASCII编码的infile文件转换成EBCDIC编码，并输出到outfile文件:

```
# dd if=infile of=outfile conv=ebcdic
```

↑
将ASCII转成EBCDIC

⑥ 将EBCDIC编码的outfile文件转换成ASCII编码，并输出到asciifile文件:

```
# dd if=outfile of=asciifile conv=ascii
```

↑
需将EBCDIC转成ASCII

⑦ 将infile文件中第20字节以后的英文字母转换为大写，然后存储成outfile:

```
# dd if=infile of=outfile ibs=10 skip=2
```

↓
读取时，跳过10*2个字节



declare

常用指数: **功能说明:** 声明shell变量。**语法:** declare [+/-] [rxi] [变量名称=设置值] 或 declare [-f]**包名称:** bash。**补充说明:** declare 命令在第一种语法中可用来声明变量并设置变量的属性 ([rxi]即为变量的属性), 在第二种语法中可用来显示shell函数。

若不加上任何参数, 则会显示全部的shell变量与函数 (与执行set命令的效果相同)。

参数:

+/-	“-”可用来指定变量的属性, “+”则是取消变量所设的属性
-f	仅显示函数
r	将变量设置为只读
x	指定的变量会成为环境变量, 可供shell以外的程序来使用
i	[设置值]可以是数值、字符串或运算式。当[设置值]为运算式时, 若指定-i属性, 则会先求出运算式的结果, 再将结果指定给变量

范 例:

① 显示shell全部的变量及函数:

```
# declare
BASH=/bin/bash
BASH_VERSINFO=( [0]="2" [1]="03" [2]="0" [3]="1"
[4]="release" [5]="i386-redhat-linux-gnu" )
BASH_VERSION='2.05b.0(1)-release'
...
```

② 仅显示shell的函数 (Red Hat Linux/Fedora Core没有默认函数, 以下为Mandrake Linux中的执行结果):

```
# declare -f
screen ()
{
    if [ -z "$SCREENDIR" ]; then
        export SCREENDIR=$HOME/tmp;
    fi;
    /usr/bin/screen $@
}
```

D

③ 使用-i属性来指定变量A1:

```
# declare A1=200+300      ← 直接将 "200+300" 指定给A1
# declare | grep A1
A1=200+300                ← 此时A1变量的设置值为 "200+300"
_ =A1=200+300A
# declare -i A1=200+300  ← 将 "200+300" 的运算结果指定给A1
# declare | grep A1
A1=500                    ← 此时A1变量的设置值为 "500"
_ =A1=2"00+300A
```

④ 设置temp变量, 并将其设成只读属性:

```
# declare -r temp=/tmp    ← 设成只读属性
# declare temp=/usr       ← 想要重新设置temp变量
-bash: declare: temp: readonly variable ← 由于temp变量是只读, 所以不能更改变量值
```


depmod (depend module)

常用指数:

**功能说明:** 分析可载入模块的相依性。**语法:** depmod [-adeisvV] [--help] [模块名称]**包名称:** modutils。**相关命令:** insmod, lsmod, modinfo, modprobe, rmmod。**补充说明:** depmod可检测模块的相依性, 供modprobe在安装模块时使用。**参数:**

-a或--all	分析所有可用的模块。一般的用法会将depmod -a命令加入 /etc/rc.d/rc.S (Slackware) 或/etc/rc.d/rc.sysinit (Red Hat Linux/Fedora Core) 中。如此, 开机后即可自动分析模块的相依关系
-d或--debug	执行排错模式
-e	输出无法参照的符号 (unresolved symbol)
-i	不检查符号表的版本
-s或--system-log	在系统记录中记录错误
-v或--verbose	执行时显示详细的信息
-V或--version	显示版本信息
--help	显示帮助

范例:

① 检测并列出现所有可用的模块 (以Red Hat Linux, Kernel 2.4.20-8 为例):

```
# depmod -a
```

```
# cd /lib/modules/2.4.20-8
```

```
# ls
```

← 分析所有可用的模块

← 模块默认所在的目录

```
build modules.dep modules.ieee1394map modules.parportmap
```

这是depmod分析模块后所产生的文件

```
modules.pnpbiosmap kernel modules.generic_string
modules.isapnpmap modules.pcimap modules.usbmap
# more modules.dep ← 显示modules.dep的内容
```

...

```
/lib/modules/2.4.20-8/kernel/fs/minix/minix.o:
/lib/modules/2.4.20-8/kernel/fs/msdos/msdos.o:
    /lib/modules/2.4.20-8/kernel/fs/fat/fat.o
```

模块相依性，载入fat.o模块之前，先载入msdos.o模块

```
/lib/modules/2.4.20-8/kernel/fs/ncpfs/ncpfs.o:
```

...

② 列出nfs.o无法参照的符号：

```
# depmod -e /lib/modules/2.4.20-8/kernel/fs/nfs/nfs.o
```

df (disk free)

常用指数：

功能说明：显示磁盘的文件系统与使用情形。

语 法：df [-ahHiklmPT] [-t <文件系统类型>] [-x <文件系统类型>] [--block-size= <块大小>] [--help] [--no-sync] [--sync] [--version] [文件或设备]

包名称：coreutils。



相关命令： du, sync。

参 数：

-a或--all	包含全部的文件系统，即使空间为0
--block-size =<块大小>	以指定的块大小来显示块数目
-h或--human-readable	以可读性较高的方式来显示信息
-H或--si	与-h参数相同，但在计算时，是以1 000字节为换算单位，而非1 024字节
-i或--inodes	显示inode的信息
-k或--kilobytes	指定块大小为1 024字节
-l或--local	仅显示本地端的文件系统
-m或--megabytes	指定块大小为1 048 576字节
--no-sync	在取得磁盘使用信息前，不要执行sync命令，这是默认值
-P或--portability	使用POSIX的输出格式
--sync	在取得磁盘使用信息前，先执行sync命令，以将内存的数据写入磁盘中
-t <文件系统类型>或 --type=<文件系统类型>	仅显示指定文件系统类型的磁盘信息。Linux默认的文件系统类型为ext2，DOS默认的文件系统类型为msdos
-T或--print-type	显示文件系统的类型
-x <文件系统类型>或 --exclude-type=<文件系统类型>	不要显示指定文件系统类型的磁盘信息
--help	显示帮助
--version	显示版本信息
[文件或设备]	指定磁盘设备。若是参数中为文件名称，则表示指定该文件所在的磁盘设备

范 例:

① 以默认的方式显示文件系统当前的使用情形:

```
# df
```

文件系统	1k-区段	已用	可用	已用%	挂载点
/dev/hda5	2063504	1130548	828136	58%	/
/dev/hda1	4088532	2423600	1664932	59%	/mnt/c
none	58976	0	58976	0%	/dev/shm

↑ ↑ ↑ ↑ ↑

装置名称	此文件系统所占有的总Kbytes数	目前使用的Kbytes数	剩余可用空间的Kbytes数	空间使用比率	挂载点

② 以可读性较高的方式显示文件系统当前的使用情形:

```
# df -h
```

文件系统	容量	已用	可用	已用%	挂载点
/dev/hda5	2.0G	1.1G	809M	58%	/
/dev/hda1	3.9G	2.3G	1.6G	59%	/mnt/c
none	58M	0	58M	0%	/dev/shm

会以较适当的单位和大小显示

③ 以可读性较高的方式显示文件系统当前的使用情形,但换算KB、MB及GB时不是以1 024字节,而是以1 000字节为单位:

```
# df -H
```

文件系统	容量	已用	可用	已用%	挂载点
/dev/hda5	2.1G	1.2G	848M	58%	/
/dev/hda1	4.2G	2.5G	1.7G	59%	/mnt/c
none	61M	0	61M	0%	/dev/shm

④ 显示文件系统inode的使用情形:

```
# df -i
```

文件系统	Inode	(I)已用	(I)可用	(I)已用%	挂载点
/dev/hda5	262144	75565	186579	29%	/
/dev/hda1	0	0	0	0%	/mnt/c
none	14744	1	14743	1%	/dev/shm

↑ ↑ ↑ ↑
 inode总数 已使用的 剩余可用 使用比率
 的数目 的数目

⑤ 显示全部文件系统的使用情形:

```
# df -a
```

文件系统	1k-区段	已用	可用	已用%	挂载点
/dev/hda5	2063504	1130816	827868	58%	/
none	0	0	0	-	/proc
/dev/hda1	4088532	2423608	1664924	59%	/mnt/c
none	0	0	0	-	/dev/pts
none	58976	0	58976	0%	/dev/shm

也列出了空间为0的文件系统

⑥ 以MBytes为单位来显示文件系统当前的使用情形:

```
# df -m
```

文件系统	1k-区段	已用	可用	已用%	挂载点
/dev/hda5	2015	1104	809	58%	/
/dev/hda1	3993	2367	1626	59%	/mnt/c
none	58	0	58	0%	/dev/pts

|
以MBytes为单位

⑦ 以2048字节为单位来显示文件系统当前的使用情形:

```
# df --block-size=2048
```

⑧ 显示文件系统使用情形时，同时显示文件系统的类型：

```
# df -T
```

文件系统	类型	1k-区段	已用	可用	已用%	挂载点
/dev/hda5	ext2	2063504	1130728	827956	58%	/
/dev/hda1	vfat	4088532	2423600	1664932	59%	/mnt/c
none	tmpfs	58976	0	58976	0%	/dev/shm

↑
多出了此列：文件系统的类型

⑨ 仅显示ext3的文件系统当前的使用情形：

```
# df -t ext3 -T
```

指定ext3文件系统

文件系统	类型	1k-区段	已用	可用	已用%	挂载点
/dev/hda5	ext3	2063504	1130728	827956	58%	/

↑
仅显示ext3文件系统

⑩ 显示文件系统当前使用情形时，不要显示 ext3 类型的文件系统：

```
# df -x ext3 -T
```

排除ext3文件系统

文件系统	类型	1k-区段	已用	可用	已用%	挂载点
/dev/hda1	vfat	4088532	2423600	1664932	59%	/mnt/c

↑
仅显示非ext3的文件系统

⑪ 如果有加载远程的文件系统，显示时只列出本地端的文件系统：

```
# df -l
```


d

⑫ 在显示文件系统当前使用情形前，执行sync命令，以先将内存的数据写入磁盘中：

```
# df --sync
```

⑬ 显示command.com文件所在的文件系统使用情形：

```
# df /mnt/c/command.com
```



指定文件名称

文件系统	1k-区段	已用	可用	已用%	加载点
/dev/hda1	4088532	2423600	1664932	59%	/mnt/c

⑭ 显示/dev/hda5设备的文件系统使用情形：

```
# df /dev/hda5
```

文件系统	1k-区段	已用	可用	已用%	加载点
/dev/hda5	2063504	1130816	827868	58%	/

diff (differential)

常用指数：



功能说明：比较文件的差异。

语 法：diff [-abBcdefHilnNpPqrstTuvxy] [-<行数>] [-C <行数>] [-D <宏名称>] [-I <字符或字符串>] [-S <文件>] [-W <宽度>] [-x <文件名或目录>] [-X <文件>] [--help] [--left-column] [--suppress-common-line][文件或目录1][文件或目录2]

包名称：diffutils。

相关命令：cmp, diffstat。



补充说明：diff默认会逐行比较文本文件的异同。此外，也可以指定要比较的目录，diff会比较目录中相同文件名的文件，但不会比较其中的子目录（若要连子目录一起比较，必须使用-r参数）。

比较后若发现不同处，则会显示文件1与文件2中该行的内容。若以“-”取代文件名，则以标准输入来替代文件。

参 数：

-<行数>	指定要显示多少行的文本。此参数必须与-c或-u参数一并使用
-a或--text	diff默认只会逐行比较文本文件。此参数可强制diff逐行比较二进制编码的文件
-b或--ignore-space-change	不比较空白字符数目的不同
-B或--ignore-blank-lines	不比较空白行数目的不同
-c	显示相异处前后部分内文，并标出不同之处
-C<行数>或--context<行数>	与执行“-c -<行数>”命令相同
-d或--minimal	使用不同的算法，以较小的单位来作比较
-D<宏名称>或ifdef<宏名称>	此参数的输出格式，可用于前置处理器宏
-e或--ed	此参数的输出格式，可用于ed的script文件
-f或--forward-ed	输出的格式类似ed的script文件，但按照原来文件中的顺序来显示不同的地方
-H或--speed-large-files	比较大文件时，可加快速度
-I<字符或字符串>或 --ignore-matching-lines <字符或字符串>	若两个文件在某几行有所不同，而这几行同时都包含了选项中指定的字符或字符串，则不显示这两个文件的差异
-i或--ignore-case	不检查大小写的不同
-l或--paginate	将结果交由pr程序来分页
-n或--rcs	将比较结果以RCS的格式来显示

-N或--new-file	在比较目录时，若文件A仅出现在某个目录中，默认会显示： Only in目录：文件A 若使用-N参数，则diff会将文件A与一个空白的文件比较
-P	若比较的文件为C语言的程序文件时，列出差异所在的函数
-P或 --unidirectional-new-file	与-N类似，但只有当第二个目录包含了在第一个目录所没有的文件时，才会将这个文件与空白的文件作比较
-q或--brief	仅显示有无差异，不显示详细的信息
-r或--recursive	比较子目录中的文件
-s或--report-identical-files	若没有发现任何差异，仍然显示信息
-S <文件>或 --starting-file=<文件>	在比较目录时，从指定的文件开始比较。默认则是按文件名的字母顺序来比较
-t或--expand-tabs	在输出时，将tab字符展开（默认的显示与空白字符相同）
-T或--initial-tab	在每行前面加上tab字符以便对齐
-u、-U<列数>或 --unified=<列数>	以合并的方式来显示文件内容的不同
-v或--version	显示版本信息
-w或--ignore-all-space	忽略全部的空白字符
-W <宽度>或 --width=<宽度>	在使用-y参数时，指定栏宽
-x <文件名或目录>或 --exclude=<文件名或目录>	不比较选项中所指定的文件或目录。<文件或目录>中可以包含通配字符
-X <文件> 或 --exclude-from=<文件>	可以将文件或目录类型存成文本文件，然后在<文件>中指定此文本文件。diff在比较时会排除这些文件
-y或--side-by-side	将两个文件以并列的方式，显示文件的异同之处

<code>--help</code>	显示帮助
<code>--left-column</code>	在使用-y参数时，若两个文件某一行内容相同，则仅在左侧的列（也就是文件1）显示该行内容
<code>--suppress-common-lines</code>	在使用-y参数时，仅显示不同之处

范 例：

- ① 列出file1及file2文件的内容（稍后将陆续用这两个文件作比较），
并比较file1与file2两个文件的异同：

```
# cat file1                                ← 列出file1的内容
line 1
line 2 :differences
# cat file2                                ← 列出file2的内容
line 1
line 2
# diff file1 file2                         ← 比较文件的异同
2c2                                         ← 这两个文件在第2行有所不同
< line 2 :differences                       ← file1第2行的内容
---
> line 2                                    ← file2第2行的内容
```

- ② 比较file1与file2两个文件，只列出有无差异，并不显示详细信息：

```
# diff -q file1 file2
文件file1与file2不同                       ← 只指出两个文件有所不同
```

- ③ 比较file1与file2两个文件，将两个文件的全部内容分别显示在左右两侧：

```
# diff -y file1 file2
line 1
line 2 :differences
┌──────────────────┴──────────────────┐
file1文件的内容                       file2文件的内容
```

④ 比较file1与file2两个文件，若两个文件某一行内容相同，则仅在左侧显示该行内容；若内容不同，则分别显示在左右两侧：

```
# diff -y --left-column file1 file2
```

由于file2的第1行与file1相同，所以并不列出该行的内容

```
line 1
line 2 :differences
| line 2
```



⑤ 比较file1与file2两个文件，只将两个文件不同的行分别显示在左右两侧：

```
# diff -y --suppress-common-line file1 file2
```

```
line 2 :differences
| line 2
```

⑥ 列出file3及file4文件的内容（稍后将陆续利用这两个文件作比较），比较file3与file4两个文件，显示不同之处前后部分的内文，并标出两个文件的不同之处：

```
# cat file3
```

← 列出file3的内容

```
line 1
line 2
line 3
line 4 : differences
line 5
line 6
```

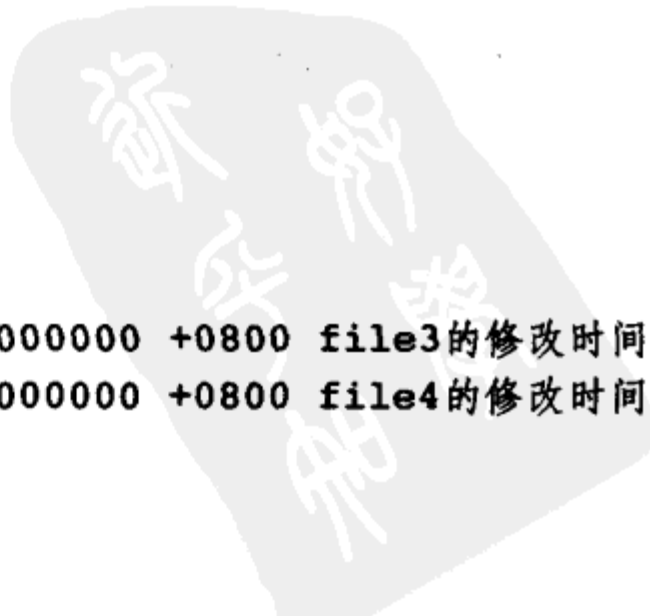
```
# cat file4
```

← 列出file4的内容

```
line 1
line 2
line 3
line 4
line 5
line 6
```

```
# diff -c file3 file4
```

```
*** file3 2003-06-05 14:07:45.000000000 +0800 file3的修改时间
--- file4 2003-06-05 14:08:07.000000000 +0800 file4的修改时间
*****
*** 1,6 ****
```



```

line 1
line 2
line 3
! line 4 : differences

```

```

line 5
line 6
--- 1,6 ---
line 1
line 2
line 3
! line 4
line 5
line 6

```

“!” 表示此行不同

D

⑦ 比较file3与file4两个文件，显示不同之处前后各2行的内文，并标出两个文件的不同处：

```
# diff -c -2 file3 file4
```

↑
指定列出前后各2行

```

*** file3      2003-06-05 14:07:45.000000000 +0800
--- file4      2003-06-05 14:08:07.000000000 +0800

```

```
*****
```

```
*** 2,6 ****
```

```

line 2
line 3
! line 4 : differences
line 5
line 6

```

列出前后各2行的内容

```
--- 2,6 ---
```

```

line 2
line 3
! line 4
line 5
line 6

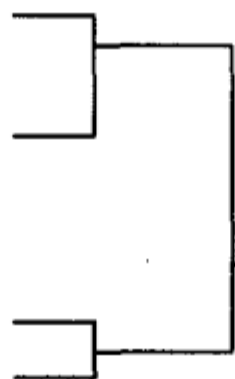
```

列出前后各2行的内容

⑧ 比较file3与file4两个文件，显示不同之处前后部分的内文，并将结果以合并的方式列出：

```
# diff -u file3 file4
--- file3      2003-06-05 14:07:45.000000000 +0800
+++ file4      2003-06-05 14:08:07.000000000 +0800
@@ -1,6 +1,6 @@
```

```
line 1
line 2
line 3
-line 4 : differences
+line 4
line 5
line 6
```



并不重复显示相同的行

⑨ 比较file1与file2两个文件的异同，比较后显示不同之处前后各2行的内文，并将结果以合并的方式列出：

```
# diff -U 2 file3 file4
```

指定以合并方式，并列出行后 2 行

```
--- file3      2003-06-05 14:07:45.000000000 +0800
+++ file4      2003-06-05 14:08:07.000000000 +0800
@@ -2,5 +2,5 @@
```

```
line 2
line 3
-line 4 : differences
+line 4
line 5
line 6
```



列出前后各2行的内容



⑩ 比较dir1与dir2两个目录中文件的异同:

ls dir1 dir2 ← 显示dir1与dir2目录下的文件

dir1:
file1 file2 file3

dir2:
file1 file2 file3

diff dir1 dir2 ← 比较dir1与dir2两个目录中的文件

diff dir1/file3 dir2/file3

2c2

< That's a question.

> That's A question.

← 比较结果显示dir1/file3与dir2/file3在这一行有所差异(小写a与大写A的不同)



⑪ 接上例, 忽略内文中大小写的差异, 若没有任何差异, 仍显示信息:

diff -is dir1 dir2

文件 dir1/file1 和 dir2/file1 相同
文件 dir1/file2 和 dir2/file2 相同
文件 dir1/file3 和 dir2/file3 相同

忽略大小写的差异后, 则两个目录下的文件完全相同

⑫ 接上例, 在比较dir1与dir2目录中的文件时, 若文本中包含“question”字符串, 则忽略不比较该行:

diff -I question -s dir1 dir2

文件 dir1/file1 和 dir2/file1 相同
文件 dir1/file2 和 dir2/file2 相同
文件 dir1/file3 和 dir2/file3 相同

忽略包含“question”字符串的行后, 则两个目录下的文件完全相同



d

- ⑩ 接上例，比较dir1与dir2目录中的文件，并以并列方式显示文件的相异之处：

```
# diff -y --suppress-common-lines dir1 dir2
diff -y --suppress-common-lines dir1/file3 dir2/file3
That's a question. | That's A question.
```

以并列方式显示文件中相异的行列，左边是dir1/file3中的内容，右边是dir2/file3中的内容

D

- ⑪ 比较dirA与dirB两个目录下文件的异同，比较时连同子目录下的文件一并比较：

```
# diff -r dirA dirB
```

- ⑫ 比较dirA与dirB两个目录下文件的异同，比较时忽略名称为file1的文件：

```
# diff -x file1 dirA dirB
```

- ⑬ 比较fileA与fileB两个文件的异同，但比较时不比较空白字符数目的不同：

```
# diff -b fileA fileB
```

- ⑭ 比较fileA与fileB两个文件的异同，但比较时不比较空白行数目的不同：

```
# diff -B fileA fileB
```

- ⑮ test1与test2两个文件是二进制文件，强制diff逐行比较两个文件的异同：

```
# diff -a test1 test2
```

- ⑯ func1.c与func2.c两个文件是C语言的程序文件，比较两者的异同，若有不一样的地方，则列出差异所在的函数：

```
# diff -p func1.c func2.c
```

diffstat (differential status)

常用指数:



功能说明: 根据diff的比较结果, 显示统计数字。

语 法: diffstat [-V] [-n <文件名长度>] [-p <文件名长度>] [-w <列位宽度>]

包 名 称: diffstat。

相关命令: diff。

补充说明: diffstat读取diff的输出结果, 然后统计各文件的插入、删除、修改等差异数量。

参 数:

-n <文件名长度>	指定文件名长度, 指定的长度必须大于或等于所有文件中最长的文件名。若不指定, diffstat将使用所有文件中最长文件名的长度
-p <文件名长度>	与-n参数相同, 但此处的<文件名长度>包括了文件的路径
-w <列宽度>	指定输出时列的宽度
-V	显示版本信息

范 例:

① 比较dir1与dir2两个目录中的文件, 并将结果交由diffstat统计整理:

```
# diff dir1 dir2 | diffstat
```

```
fileA | 2 --
```

1 files changed, 2 deletions(-) ←表示dir1/fileA中有2行的内容是dir2/fileA中所没有的

② 比较dir3与dir4两个目录中的文件, 并将结果交由diffstat统计整理:

```
# diff dir3 dir4 | diffstat
```

```
fileB | 4 ++++
```

1 files changed, 4 insertions(+) ←表示dir4/fileB中有4行的内容是dir3/fileB中所没有的

③ 比较dir5与dir6两个目录中的文件，并将结果交由diffstat统计整理：

```
# diff dir5 dir6 | diffstat
```

```
fileC |      4 ++--
```

```
1 files changed, 2 insertions(+), 2 deletions(-)
```

表示dir5/fileC中有2行的内容是dir6/fileC中所没有的，且dir6/fileC中有2行的内容是dir5/fileC中所没有的

D

dip

常用指数：

功能说明：IP拨号连接。

语 法：dip [-aikt] [-l <连接位置>] [-m <MTU数目>] [-p <协议>]
[拨号 script 文件]

包名称：dip (Mandrake Linux)。

相关命令：minicom, setserial。

补充说明：这是Mandrake Linux才有的命令。dip可控制调制解调器，以拨号的方式建立对外的双向连接。

参 数：

-a	询问用户名与密码
-i	启动拨号服务器功能
-k	删除执行中的dip程序。除了root以外，一般用户只能删除由自己所启动的dip程序
-l <连接位置>	指定要删除的连接，必须配合-k参数一起使用
-m <MTU数目>	设置最大传输单位 (MTU)，默认值为296字节
-p <协议>	设置通信协议，可用的选项有SLIP、CSLIP、PPP或TERM
-t	进入dip的命令模式，通常都以此模式来拨号
-v	执行时显示详细的信息

范 例:

① 以下实例示范如何以dip拨号连接:

```
# dip -tv
```

```
DIP: Dialup IP Protocol Driver version 3.3.7o-uri (8 Feb 96)
Written by Fred N. van Kempen, MicroWalt Corporation.
```

```
DIP: name=root home=/tmp
      host=free.flag.com.tw IP=192.168.0.135
      prot=SLIP MTU=296
```

```
Modem set to "HAYES".
```

```
DIP [0 ]> port /dev/ttySI ← 设置调制解调器的设备名称
```

```
>> port /dev/
```

```
PORT: terminal port set to "/dev/modem".
```

```
DIP: tty_open: /dev/modem (3) DIP: tty_open: IBUF=1024 OBUF=1020
```

```
DIP: tty: set_speed: 38400
```

```
DIP: tty: set_databits: 8
```

```
DIP: tty: set_stopbits: 1
```

```
DIP: tty: set_parity: N
```

```
DIP: tty: set_speed: 38400
```

```
DIP [0 ]> speed 57600 ← 设置调制解调器速度
```

```
>> speed 57600
```

```
DIP: tty: set_speed: 38400
```

```
DIP [0 ]> reset ← 初始化调制解调器 (可不作)
```

```
...
```

```
DIP [0 ]> term ← 进入终端模式
```

```
[ Entering TERMINAL mode. Use CTRL-] to get back ]
```

```
atdt????????? ← ??????????为ISP的拨接号码
```

```
CONNECT 19200/arg/v34/lapm/v42bis
```

```
...
```

```
← 按下 [ ] + [ ] 键
```

```
[ Back to LOCAL mode. ]
```

```
DIP [0 ]> default ← 使用默认的网关
```

```
Destination net/address set to 'default'
```

```
DIP [0 ]> mode ppp ← 使用ppp模式
```

② 指定以PPP模式，最大传输单位为512字节来进入dip命令模式：

```
# dip -tv -p PPP -m 512
```

```
DIP: Dialup IP Protocol Driver version 3.3.7o-uri (8 Feb 96)
Written by Fred N. van Kempen, MicroWalt Corporation.
```

```
DIP: name=root home=/tmp
      host=free.flag.com.tw IP=192.168.0.135
      prot=PPP MTU=512
```

↑ ↑
通信协议 最大传输单位

...

③ 启动拨号服务器，启动前请先确定有/etc/diphosts设置文件：

```
# dip -i
```

④ 删除/dev/modem设备执行中的连接：

```
# dip -k -l modem
```

```
DIP: Dialup IP Protocol Driver version 3.3.7o-uri (8 Feb 96)
Written by Fred N. van Kempen, MicroWalt Corporation.
```

```
DIP: process 926 killed.
```

已经成功地
删除连接

操作命令：以下为dip命令模式中的一些常用命令：

default	将网关设为远端电脑的网关
dial <电话号码>	拨号
mode	设置通信协议，默认为SLIP
port <调制解调器设备>	设置调制解调器通信端口
reset	将调制解调器初始化

speed <传输速率>	设置传输速率，单位为bit/s
term	进入终端模式
quit	中断连接并结束程序

dircolors

常用指数:



D

功能说明: 设置ls命令在显示目录或文件时所用的颜色。

语 法: dircolors [颜色设置文件] 或
dircolors[-bcp] [--help] [--version]

包 名 称: coreutils。

相关命令: ls。

补充说明: dircolors可根据 [颜色设置文件] 来设置LS_COLORS环境变量（第一种语法），或是显示设置LS_COLORS环境变量的shell命令（第二种语法）。ls命令会根据LS_COLORS的变量值，在显示文件与目录时，以颜色来区别不同类型的文件与目录。一般的使用方法是在.bash_profile中加入dircolors命令，以便开机时即设好LS_COLORS变量。

参 数:

-b、--sh或--bourne-shell	显示在Bourne shell中，设置LS_COLORS环境变量的shell命令
-c、--csh或--c-shell	显示在C shell中，设置LS_COLORS环境变量的shell命令
-p或--print-database	显示默认值。此参数的显示结果可直接作为[颜色设置文件]
--help	显示帮助
--version	显示版本信息

范 例:

① 将默认值存为.colorrc颜色设置文件, 修改后再从.bash_profile执行:

```
# dircolors -p > .colorrc
# cat .colorrc
...
# Attribute codes:
# 00=none 01=bold 04=underscore 05=blink 07=reverse 08=concealed
# Text color codes:
# 30=black 31=red 32=green 33=yellow 34=blue 35=magenta ... 37=white
# Background color codes:
# 40=black 41=red 42=green 43=yellow 44=blue 45=magenta ... 47=white
```

这里的注释文字说明如何设置颜色

```
NORMAL 00 # global default, although everything should be
something.
```

```
FILE 00 # normal file
```

```
DIR 01;34 # directory
```

```
LINK 01;36 # symbolic link
```

```
FIFO 40;33 # pipe
```

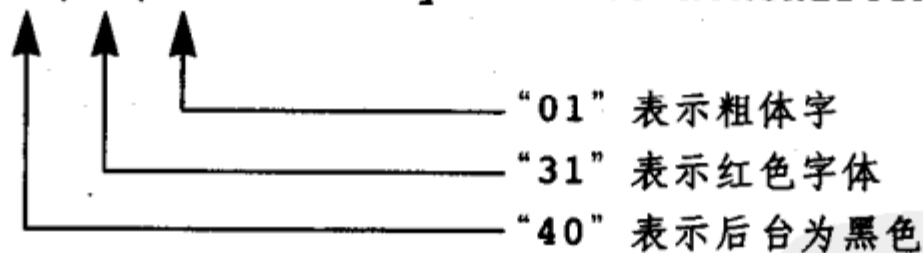
```
SOCK 01;35 # socket
```

```
DOOR 01;35 # door
```

```
BLK 40;33;01 # block device driver
```

```
CHR 40;33;01 # character device driver
```

```
ORPHAN 40;31;01 # symlink to nonexistent file
```



...

请根据自己喜欢的方式来更改.colorrc文件。改好后, 请在.bash_profile加入下列语句:

```
eval `dircolors ~/.colorrc`
```

注意这个符号是键盘左上角, 数字1左边的那个按键

重新登录后, 颜色设置便会生效。

② 在 Bourne shell 中，变更 LS_COLORS 环境变量值：

```
# dircolors -b
LS_COLORS='no=00:fi=00:di=01;34:ln=01;36:pi=40;33:so=01;35:do=01;35:
bd=40;33;01:cd=40;33;01:or=40;31;01:ex=01;32:*.tar=01;31:*.tgz=01;31:
*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:
*.gz=01;31:*.bz2=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.jpg=01;35:
*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:
*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:
*.mpg=01;35:*.mpeg=01;35:*.avi=01;35:*.fli=01;35:*.gl=01;35:*.dl=01;35:
*.xcf=01;35:*.xwd=01;35:*.ogg=01;35:*.mp3=01;35:*.wav=01;35: ';
export LS_COLORS
```

上面的文字其实只执行了两个语句：“LS_COLORS=[变量值]”与“export LS_COLORS”。第一个语句设置 LS_COLORS 的变量值，第二个语句则将 LS_COLORS 输出为环境变量。dircolors -b 的输出结果可作为 script 文件，执行后即可变更 LS_COLORS 环境变量。

```
# dircolors -b > setcolor
# sed -e s/"di=01;34"/"di=05;34"/1 setcolor > newcolor
# source newcolor
# ls
```

也可以使用任
何一种文本编辑器来修改
setcolor 中的设置


将目录的字型属性改为“05”（闪烁）

必须以 source 来执行 script，才
能将环境变量输出到 shell 中

执行 ls 时，即可看见目录会一闪一闪



dirs (directory stack)

常用指数: 

功能说明: 显示目录记录。

语 法: dirs [-clnpv] [+n]

包 名 称: bash。

相关命令: pushd, popd。

补充说明: 显示目录堆栈中的记录。显示时，最左边的目录是最后加入的记录，最右边的目录是最早加入的记录。可利用pushd命令将目录加至堆栈中，利用popd命令将目录从堆栈中删除。

若不加任何参数，即显示当前目录堆栈中的记录。

参 数:

-c	删除目录堆栈中的所有记录
-l	以完整格式显示目录中的记录，当显示用户根目录时，列出完整路径，而不以“~”表示。请注意此参数为英文小写字母的“l”，不是阿拉伯数字的“1”
-n	显示从右边算起第n条的目录（n从0开始）
-p	以每行一个记录的方式列出目录堆栈中的所有记录
-v	以每行一个记录的方式列出目录堆栈中的所有记录，并在每行前面加上流水号
+n	显示从左边算起第n条的目录（n从0开始）

范 例:

① 显示并加入目录堆栈中的记录:

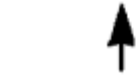
```
# dirs ← 显示当前堆栈中的记录
```

```
~
```

```
↑
```

以“~”表示用户根目录

```
# pushd /etc ← 将/etc目录加入堆栈中
/etc ~
```



每条记录之间以空白字符隔开

```
# pushd /mnt ← 将/mnt目录加入堆栈中
/mnt /etc ~
```

```
# dirs -l ← 以完整路径来显示堆栈中的记录
/mnt /etc /root
```

② 接上例，以每行一个记录的方式列出目录堆栈中的记录：

```
# dirs -p
/mnt
/etc
~
```

③ 接上例，以每行一个记录的方式列出目录堆栈中的记录，并在每行前面加上流水号：

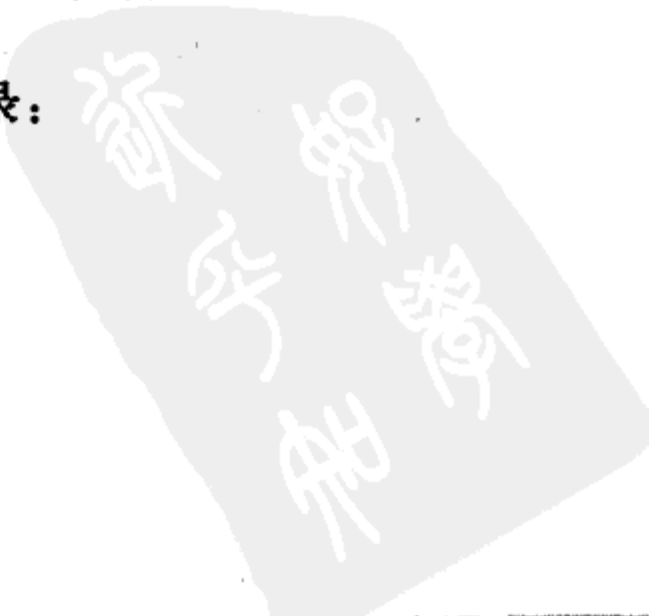
```
# dirs -v
0 /mnt
1 /etc
2 ~
```

④ 接上例，列出目录堆栈中从左边算起第1条目录：

```
# dirs +1
/etc
```

⑤ 接上例，列出目录堆栈中从右边算起第1条目录：


```
# dirs -1
/etc
```



⑥ 接上例，删除目录堆栈中的所有记录：

```
# dirs -c
```

dmesg

常用指数：

功能说明：显示开机信息。

语 法：dmesg [-cn] [-s <缓冲区大小>]

包 名 称：util-linux。

补充说明：kernel会将开机信息存储在ring buffer中。如果开机时来不及查看信息，可利用dmesg来查看。开机信息也存储在/var/log目录中名为dmesg的文件里。

参 数：

-c	显示信息后，清除ring buffer中的内容
-s <缓冲区大小>	默认值为8 196，刚好等于ring buffer的大小。若自定义较大的ring buffer，则必须在此设置相对的缓冲区大小，才能查看整个ring buffer中的内容
-n	设置记录信息的层级

范 例：

① 显示ring buffer中的信息：

```
# dmesg
```

```
Linux version 2.4.20-8 (bhcompile@stripples.devel.redhat.com)
```

```
(gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))
```

```
...
```

② 将开机信息存储在bootlog文件中:

```
# dmesg > bootlog
```

③ 清除ring buffer中的内容:


```
# dmesg -c
```

④ 将记录层级设为1, 往后开机时只有严重的错误信息才会出现在屏幕上:

```
# dmesg -n 1
```

D

du (disk usage)

常用指数: 

功能说明: 显示目录或文件的大小。

语法: du [-abcDhHklmsSx] [-L <符号链接>] [-X <文件>] [--block-size=<块大小>] [--exclude=<目录或文件>] [--help] [--max-depth=<目录层数>] [--version] [目录或文件]

包名称: coreutils。

相关命令: df, mdu。

补充说明: du会显示指定的目录或文件所占用的磁盘空间, 若不加任何参数, 则默认的单位为1 KB (1 024字节)。可以一次指定多个目录或文件。



参 数:

-a或-all	显示目录中个别文件的大小
-b或-bytes	显示目录或文件大小时,以字节为单位
-c或--total	除了显示个别目录或文件的大小外,同时也显示所有目录或文件的总和
-D或--dereference-args	显示指定符号链接的来源文件大小
-h或--human-readable	以k(KB)、M(MB)、G(GB)为单位,提高信息的可读性。KB、MB、GB是以1 024为换算单位
-H或--si	与-h参数相同,但是KB、MB、GB是以1 000为换算单位
-k或--kilobytes	以1 024字节为块单位。这是程序的默认单位
-l或--count-links	重复计算硬链接文件所占用的磁盘空间
-L <符号链接>或 --dereference <符号链接>	显示选项中所指定符号链接的来源文件大小
-m 或 --megabytes	以1 MB (1 048 576字节) 为单位
-s 或 --summarize	仅显示总计
-S 或 --separate-dirs	显示个别目录的大小时,不包含其子目录的大小
-x 或--one-file-system	以最先处理目录的文件系统为准,而不显示其他不同文件系统的目录
-X <文件> 或 --exclude-from=<文件>	du会略过<文件>中所指定的目录或文件
--block-size=<块大小>	指定显示时以多少字节当成一个块
--exclude=<目录或文件>	略过指定的目录或文件
--max-depth=<目录层数>	超过指定层数的目录后,予以忽略
--help	显示帮助
--version	显示版本信息

范 例:

① 显示当前目录下，所有子目录所占用的磁盘空间：

```
# du
4      ./dirA/dirAA/dirAAA/dirAAAA
8      ./dirA/dirAA/dirAAA
4512   ./dirA/dirAA
```

↑
每个目录所占用的块数，默认块单位为 1024 字节。

...

② 显示当前目录下，有子目录所占用的磁盘空间，并指定以512字节当成一个块显示：

```
# du --block-size=512
8      ./dirA/dirAA/dirAAA/dirAAAA
16     ./dirA/dirAA/dirAAA
9024   ./dirA/dirAA
```

↑
所占用的块数比上例增加了一倍

...

③ 显示当前目录下，所有子目录所占用的磁盘空间，显示时以 1 MB 为单位：

```
# du -m
0      ./dirA/dirAA/dirAAA/dirAAAA
0      ./dirA/dirAA/dirAAA
4      ./dirA/dirAA
```

↑
所占用的 MB 数

...

- ④ 以可读性较高的方式显示当前目录下，所有子目录所占用的磁盘空间，其中KB、MB、GB以1 024为换算单位：

```
# du -h
4.0k    ./dirA/dirAA/dirAAA/dirAAAA
8.0k    ./dirA/dirAA/dirAAA
4.4M    ./dirA/dirAA
```



按照所占用磁盘空间的不同，以适当的单位显示

...

- ⑤ 以可读性较高的方式显示当前目录下，所有子目录所占用的磁盘空间，其中KB、MB、GB以1 000为换算单位：

```
# du -H
4.1k    ./dirA/dirAA/dirAAA/dirAAAA
8.2k    ./dirA/dirAA/dirAAA
4.6M    ./dirA/dirAA
```



由于是以 1 000 为换算单位，跟上例比较，可发现结果变多了

...

- ⑥ 显示当前目录下，所有子目录所占用的磁盘空间。显示时只列出个别目录的大小，并不包含其子目录大小：

```
# du -s
4        ./dirA/dirAA/dirAAA/dirAAAA
4        ./dirA/dirAA/dirAAA
4504    ./dirA/dirAA
```



该目录所占用的磁盘空间，并不包含子目录

- ⑦ 显示当前目录下，所有子目录所占用的磁盘空间。最多只显示2层子目录，超过的部分则不予显示：

```
# du --max-depth=2
4512    ./dirA/dirAA
4516    ./dirA
...
```

最多显示2层的子目录

- ⑧ 显示当前目录下，所有子目录所占用的磁盘空间，但不显示名称为dirAAAA的目录：

```
# du --exclude=dirAAAA
          ↑
          排除dirAAAA目录
4        ./dirA/dirAA/dirAAA
4508    ./dirA/dirAA
...
```

- ⑨ 显示当前目录所占用的磁盘空间，只显示整个目录的大小，并不列出个别目录占用的空间：

```
# du -s
9040    .
↑
整个目录所占用的磁盘空间
```

- ⑩ 显示当前目录下，所有子目录及个别文件所占用的磁盘空间：

```
# du -a
4        ./file2
12       ./file1
4        ./dirA/dirAA/dirAAA/dirAAAA
...
```

除了目录之外，也会显示个别文件所占用的磁盘空间

⑩ 除了filelist文本文件所指定的文件之外，显示当前目录下，所有子目录及文件所占用的磁盘空间：

```
# ls
file1 file2 file3 filelist
# cat filelist
file1
file2
# du -a -x filelist
```

↑
排除 filelist 指定的文件

```
4500 ./file3
4 ./filelist
4508 .
...
```

} 不会列出filelist指定文件file1及file2所占用的磁盘空间

⑪ 显示file文件所占用的磁盘空间，显示时以字节为单位：

```
# ls -l file
-rw-r--r-- 1 root root 17 6月 5 20:42 file
```

↑
file文件大小为17字节

```
# du -b file
4096 file
```

虽然file文件大小只有17字节，但却占用了4 096字节的磁盘空间

⑫ 显示file1、file2及file3文件各占用的磁盘空间，并显示占用的总磁盘空间：

```
# du -c file1 file2 file3
12 file1
4 file2
4500 file3
4516 总计
```

← 此3个文件所占用的总磁盘空间

- ⑭ 显示当前目录所有子目录及文件所占用的磁盘空间，若有硬链接的文件，重复计算所占用的磁盘空间：

```
# ls -l
总计 24
-rw-r--r--  2 root  root  9482  6月 5   22:20  fileA
-rw-r--r--  2 root  root  9482  6月 5   22:20  fileB
```

fileA与fileB互为硬链接文件

```
# du -a
12      ./fileB
16      .
```

← 以默认的方式显示
← 只显示其中一个文件所占用的磁盘空间

```
# du -a -l
12      ./fileB
12      ./fileA
28      .
```

↑ 重复计算硬链接文件所占用的空间
← 也会显示硬链接文件所占用的空间

- ⑮ file2是一个符号链接文件，显示此符号链接来源文件所占用的磁盘空间：

```
# du file2
0      file2
```

↑ 此符号链接所占用的磁盘空间为 0

```
# du -L file2
12     file2
```

↑ 显示符号链接来源文件所占用的磁盘空间
↑ 来源文件所占用的磁盘空间

- ⑥ 显示/mnt目录下所有子目录所占用的磁盘空间，显示时只列出最先处理目录的文件系统，而不显示其他不同文件系统的目录：

ls /mnt ← 列出/mnt目录的内容

```
cdrom  ext3  floppy  win98
```

此目录属vfat文件系统
 此目录属msdos文件系统
 此目录属ext2文件系统
 此目录属iso9660文件系统

du -x /mnt

```
4      /mnt/ext3
8      /mnt
```

由于/mnt目录属ext3文件系统，所以只会列出同属于ext3文件系统的目录，而不列出其他文件系统的目录

dump

常用指数：



功能说明：备份文件系统。

语法：dump [-cnu] [-0123456789] [-b <块大小>] [-B <块数目>] [-d <密度>] [-f <设备名称>] [-h <层级>] [-s <磁带长度>] [-T <日期>] [目录或文件系统] 或dump [-wW]

包名称：dump。

相关命令：restore。

补充说明：dump为备份工具程序，可将目录或整个文件系统备份至指定的设备（磁带机、硬盘等等），或备份成一个大文件。当备份内容大于指定的存储设备时，dump会自动将输出分割成数个备份卷（Volume），以便分开存储。dump也提供多种备份的层级，可让管理员选择完整备份或是只备份新的文件。若要回复备份数据，请使用restore命令。

参 数:

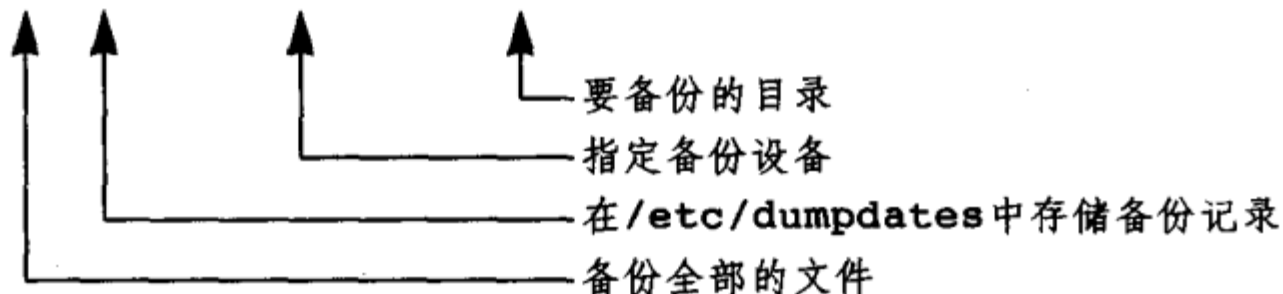
-0123456789	备份的层级。0为最完整的备份，会备份所有的文件。若指定0以上的层级，则备份自上一次备份以来（备份层级相同或较低）修改或新增的文件
-b <块大小>	指定块的大小，单位为KB
-B <块数目>	指定备份卷（Volume）的块数目
-c	修改备份磁带默认的密度与容量
-d <密度>	设置磁带的密度，单位为BPI。默认的磁带密度为1 600BPI
-f <设备名称>	指定备份设备。备份设备可以是实际的设备或文件
-h <层级>	当备份层级等于或大于指定的层级时，将不备份用户标示为“nodump”的文件
-n	当备份工作需要管理员介入时，向所有“operator”组中的用户发出通知
-s <磁带长度>	备份磁带的长度，单位为英尺。超过指定的长度时，即通知管理员更换磁带。默认值为2 300英尺
-T <日期>	指定开始备份的时间与日期。此参数通常用在script文件中，执行自动备份的工作
-u	备份完毕后，在/etc/dumpdates中记录备份的文件系统、层级、日期与时间等等
-w	与-W类似，但仅显示需要备份的文件
-W	显示需要备份的文件及其最后一次备份的层级、时间与日期。dump会根据/etc/dumpdates与/etc/fstab中的记录，来判断需要备份的文件

范 例:

- ① 将/root（需为独立的分区）下的所有文件完整备份至/dev/hdd1，并将备份记录存到/etc/dumpdates，此备份操作将会删除/dev/hdd1上的所有数据：

d

```
# dump -0 -u -f /dev/hdd1 /root
```



```
DUMP: Date of this level 0 dump: Mon Jun 9 19:52:42 2003
```

```
DUMP: Dumping /dev/hda3 (/) to /dev/hdd1
```

```
DUMP: Excluding inode 8 (journal inode) from dump
```

```
DUMP: Excluding inode 7 (resize inode) from dump
```

```
DUMP: Label: none
```

```
DUMP: mapping (Pass I) [regular files]
```

```
DUMP: mapping (Pass II) [directories]
```

```
DUMP: estimated 1573784 tape blocks.
```

```
DUMP: Volume 1 started with block 1 at: Mon Jun 9
```

```
19:52:46 2003
```

```
DUMP: dumping (Pass III) [directories]
```

```
DUMP: dumping (Pass IV) [regular files]
```

```
DUMP: 63.42% done at 3326 kB/s, finished in 0:02
```

```
DUMP: Closing /dev/hdd1
```

```
DUMP: Volume 1 completed at: Mon Jun 9 20:01:32 2003
```

```
DUMP: Volume 1 1749450 tape blocks (1708.45MB)
```

```
DUMP: Volume 1 took 0:08:46
```

```
DUMP: Volume 1 transfer rate: 3325 kB/s
```

```
DUMP: 1749450 tape blocks (1708.45MB) on 1 volume(s)
```

```
DUMP: finished in 521 seconds, throughput 3357 kBytes/sec
```

```
DUMP: Date of this level 0 dump: Mon Jun 9 19:52:42 2003
```

```
DUMP: Date this dump completed: Mon Jun 9 20:01:32 2003
```

```
DUMP: Average transfer rate: 3325 kB/s
```

```
DUMP: DUMP IS DONE
```

```
# cat /etc/dumpdates ← 列出dumpdates中的记录
```

```
/dev/hda3 0 Mon Jun 9 19:52:42 2003
```

D

② 根据/etc/fstab及/etc/dumpdates记录文件列出需要备份的文件:

```
# dump -W
Last dump(s) done (Dump '>' file systems):
  /dev/hda3      (      /) Last dump: Level 0, Date Mon Jun9
19:52:42 2003
```

上次备份文件在 /dev/hda3, 备份层级为0, 备份时间6月9日19时52分

③ 将/root目录备份到/mnt/hdb/dump_file文件:

```
# dump -f /mnt/hdb/dump_file /root
```

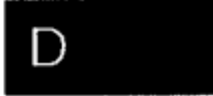
④ 将/tmp目录下的所有文件完整备份到软驱中的软盘:

```
# dump -f /dev/fd0 /tmp
DUMP: Date of this level 0 dump: Mon Jun 9 22:03:22 2003
...
```

DUMP: Change Volumes: Mount volume #2 ← 当使用完第1张软盘, 会要求放入第2张软盘

DUMP: Is the new volume mounted and ready to go?: ("yes" or "no") yes ← 放入第2张软盘后, 输入 "yes" 继续备份

...



e2fsck (ext2 file system check)

常用指数:



功能说明: 检查ext2文件系统的正确性。

语法: e2fsck [-acCdFfnprsStvVy] [-b <superblock>] [-B <块大小>] [-j <日志位置>] [-l <文件>] [-L <文件>] [设备名称]

包名称: e2fsprogs。

相关命令: badblocks, mke2fs, fsck, fsck.ext2, fsck.ext3。

补充说明: e2fsck 执行后的返回值及表示意义如下:

0	没有任何错误发生
1	文件系统发生错误, 并且已经修正
2	文件系统发生错误, 并且已经修正。如果文件系统已经卸载, 则须重新开机
4	文件系统发生错误, 但没有修正
8	运作时发生错误
16	使用的语法发生错误
128	共享的函数库发生错误

参数:

-a	不询问用户意见, 便自动修复文件系统。与-p参数有相同的功能, 是为了兼容性而存在, 建议仍使用-p参数
-b <superblock>	指定superblock, 而不使用默认的superblock
-B <块大小>	指定块的大小, 单位为字节。e2fsck默认会查找适当的块大小, 使用此参数可强制指定块的大小
-c	一并执行badblocks, 以标示损毁的块

-C	将检查过程的信息完整记录在file descriptor中，使得整个检查过程都能完整监控
-d	显示排错信息。除非要对程序进行排错，否则不用此参数
-f	即使文件系统没有错误的迹象，仍强制检查正确性
-F	执行前先清除设备的缓冲区
-j<日志位置>	指定日志文件系统存在的分区。通常检查ext3日志文件系统时，需用此参数
-l <文件>	将文件中指定的块加到损毁块列表。可利用badblocks命令找出损毁的块，并存储在文件中
-L <文件>	先清除损毁块列表，再将文件中指定的块加到损毁块列表，因此损毁块列表的块跟文件中指定的块是一样的。可利用badblocks命令找出损毁的块，并存储在文件中
-n	以只读模式打开文件系统，并采取非互动方式执行，所有的问题对话均设置以“no”回答。如与-C、-l或-L参数一并使用，则只允许更新文件系统的损毁块列表，而不能写入其他部分
-p	不询问用户意见，便自动修复文件系统
-r	此参数只为了兼容性而存在，并无实际作用
-s	如果文件系统的字节顺序 (byte-order) 不适当，就交换字节顺序，否则不做任何操作。如80×86系列的CPU的字节顺序是从最低有效字节开始的little endian
-S	不管文件系统的字节顺序，一律交换字节顺序
-t	显示时间信息。若输入这个参数2次，则显示更详细的时间信息
-v	执行时显示详细的信息
-V	显示版本信息
-y	采取非互动方式执行，所有的问题均设置以“yes”回答

范 例:**① 检查/dev/hdd1文件系统的正确性:**

```
# e2fsck /dev/hdd1
e2fsck 1.32 (09-Nov-2002)
/dev/hdd1 is mounted.
```

WARNING!!! Running e2fsck on a mounted filesystem may cause SEVERE filesystem damage. ← 因为检查的文件系统已经载入，所以会出现警告信息：此动作可能会导致数据损毁，建议执行前先卸载文件系统

Do you really want to continue (y/n)? yes ← 按 **■** 键继续

```
/dev/hdd1 was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/hdd1: 11/251392 files (0.0% non-contiguous),
7905/502023 blocks ← 检查结果
```

]—— 检查流程

② 检查/dev/hdd1文件系统的正确性（执行前已先将文件系统卸载）:

```
# e2fsck /dev/hdd1
e2fsck 1.32 (09-Nov-2002) ←
/dev/hdd1: clean, 11/251392 files, 7905/502023 blocks
```

由于已经卸载了文件系统，所以没有出现警告信息

③ 检查/dev/hdd2文件系统（ext3）的正确性，若发现错误，不询问用户意见，便自动修复文件系统:

```
# e2fsck -j ext3 -p /dev/hdd2
```

- ④ 检查/dev/hdd2文件系统 (ext3) 的正确性, 同时执行badblocks以检查是否有损毁的块:

```
# e2fsck -j ext3 -c /dev/hdd2
e2fsck 1.32 (09-Nov-2002)
Checking for bad blocks (read-only test): 172336/552234
```

当检查空间较大时, 检查损毁块的过程需要较长的时间

- ⑤ 即使文件系统没有错误, 仍强制检查/dev/hdd2文件系统的正确性:

```
# e2fsck -j ext3 -f /dev/hdd2
e2fsck 1.32 (09-Nov-2002)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/hdd2: 11/276352 files (0.0% non-contiguous), 16889/
552234 blocks
```

不询问就强制检查正确性

- ⑥ 即使文件系统没有错误, 仍强制检查/dev/hdd2文件系统的正确性, 同时显示检查时间的信息:

```
# e2fsck -j ext3 -f -t /dev/hdd2
e2fsck 1.32 (09-Nov-2002)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Peak memory: Memory used: 450k/0k (406k/44k), time:
4.86/ 0.22/ 3.16
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/hdd2: 11/276352 files (0.0% non-contiguous),
16889/552234 blocks
Memory used: 50k/0k (5k/46k), time: 5.30/ 0.32/ 3.21
```

与上例比较, 增加了这2行的时间信息

- ⑦ 即使文件系统没有错误，仍强制检查/dev/hdd2文件系统的正确性，并显示更详细的时间信息：

```
# e2fsck -j ext3 -f -tt /dev/hdd2
```

↑ 输入2次t参数

- ⑧ 检查/dev/hdd2文件系统的正确性，并将badblockslist文件中指定的块加到损毁块列表中：

```
# e2fsck -j ext3 -l badblockslist /dev/hdd2
```

↑ 此文件是由badblocks命令产生

- ⑨ 检查/dev/hdd2文件系统的正确性，并将损毁块列表设成跟badblockslist文件指定的块一致：

```
# e2fsck -L badblockslist /dev/hdd2
```

- ⑩ 检查/dev/hdd1文件系统的正确性，执行前先清除缓冲区的数据：

```
# e2fsck -F /dev/hdd1
```


- ⑪ 以非互动的方式，所有的问题均设置以“yes”回答，检查/dev/hdd1文件系统的正确性：

```
# e2fsck -y /dev/hdd1
```

- ⑫ 以非互动的方式，所有的问题均设置以“no”回答，并以只读模式检查/dev/hdd1文件系统的正确性：

```
# e2fsck -n /dev/hdd1
```

echo

常用指数: 

功能说明: 显示文本。

语法: echo [-ne] [字符串...] 或
echo [--help] [--version]

包名称: coreutils。

补充说明: echo会将输入的字符串送往标准输出。输出的字符串间以空白字节隔开，并在最后加上换行符号。

参数:

-n	不要在最后自动换行
-e	若字符串中出现以下字节，则特别加以处理，而不会将它当成一般文字输出： \a 发出警告声 \b 删除前一个字符 \c 最后不加上换行符号 \f 换行但光标仍旧停留在原来的位置 \n 换行且光标移到行首 \r 光标移到行首，但不换行 \t 插入tab \v 与\f相同 \\ 插入\字节 \nnn 插入nnn（八进制）ASCII码所表示的字符
--help	显示帮助
--version	显示版本信息

范例:

① 用echo显示字符串，利用-n参数使得结果不要在最后自动换行：

```
# echo -n "Hello Alkane"
```

```
Hello Alkane # ← 在此处出现提示符号，并不会自动换行
```

E

② 用echo显示字符串，利用-e参数使得不要在输出结果的最后加上换行符号：

```
# echo -e "Hello Alkane\c"
Hello Alkane # ←—— 跟上例有相同的结果
```

③ 用echo显示字符串，并对某些特殊字符作处理：

```
# echo -e "Don't forget \164he date of our memories, \t12
June 2003."
Don't forget the date of our memories, 12 June 2003.
      ↑                ↑
      插入的小写t      这里加上了一个tab
```

④ 显示变量内容：

```
# echo $PATH
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr
/local/bin:/bin:/sbin:
/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin
```

⑤ 用echo显示字符串，并在最后发出一声警告声：

```
# echo -e "Warning message\a"
Warning message
```

⑥ 用echo显示字符串，并删除此字符串中的一个字符“s”：

```
# echo -e "Merrys\b Christmas"
Merry Christmas
```

ed (editor)

常用指数:




功能说明: 文本编辑器。

语 法: ed [-] [-Gs] [-p <字符串>] [--help] [--version] [文件]

包 名 称: ed。

相关命令: jed, joe, sed, vi, vim。

补充说明: ed是Linux中功能最简单的文本编辑程序，一次仅能编辑一行而非全屏幕方式的操作。

ed有command mode（命令模式）与input mode（输入模式）两种模式。刚开始执行ed时为command mode，值得注意的是此时没有任何的说明文本。在command mode中使用a、c或i命令可进入input mode，若要回到command mode，请在新的一行输入“.”，按  键后即回到command mode。

参 数:

-G或--traditional	这是为了兼容于以前版本所提供的功能
-p <字符串>	指定ed在command mode的提示符号（默认是没有提示符号）
-s、-、--quiet或--silent	不执行打开文件时的检查功能
--help	显示帮助
--version	显示版本信息

范 例:

① 指定command mode的提示符号为“command:”:

```
# ed -p command:
command;
```

② 利用ed编辑器编辑textfile文件:

ed textfile

533 ← 会先出现此文件的字节数

...

操作命令: command mode命令说明如下:

a	进入input mode, 输入的内容会接在最后一行后面
c	进入input mode, 输入的内容会取代最后一行
i	进入input mode, 输入的内容会加到最后一行的前一行
d	删除最后一行
n	显示最后一行的行号与内容
w<文件名>	存储文件
q	结束程序

操作范例:

③ 以下实际示范如何编辑文字:

ed

a ← 进入文本模式

This is the 1st line.

This is the wnd line. ← 这一行不小心输错了

. ← 回command mode

c ← 取代最后一行

* This is the 2nd line.

.
i ← 插入最后一行的前一行

This line is between 1st and 2nd line.

.

w mydoc ← 将文件存为mydoc

83 ← ed会显示文件的字符数 (包括空白字符与换行符号)

q

← 结束程序

cat mydoc

This is the 1st line.

This line is between 1st and 2nd line.

This is the 2nd line.

mydoc的内容

edquota (edit quota)

常用指数: **功能说明:** 编辑用户或组的quota。**语法:** edquota [-p <来源用户名称>] [-ug] [用户或组名称...]或
edquota [-ug] -t**包名称:** quota。**相关命令:** quota, quotacheck, quotaoff, quotaon, requota。**补充说明:** edquota默认会使用vi来编辑用户或组的quota设置。
Mandrake Linux的用户也可利用linuxconf, 以图形界面的方式来设置quota。**参数:**

-u	设置用户的quota, 这是默认的参数
-g	设置组的quota
-p <来源用户名称>	将来源用户的quota设置应用至其他用户或组
-t	设置过渡期 (Grace period)



范 例:

① 将用户user1的quota设置应用到用户user2:

```
# edquota -p user1 user2
```

② 编辑用户user1的quota设置。进入vi后直接修改要设置的数字即可:

```
# edquota user1
```

Disk quotas for user user1 (uid 503): ← 设置user1用户

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/hdd2	0	15360	20480	0	500	550

磁盘空间的限制

文件数目的限制

③ 编辑sphenoid组的quota设置:

```
# edquota sphenoid
```

Disk quotas for user sphenoid (gid 503): ← 设置sphenoid用户

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/hdd2	0	15360	20480	0	500	550

磁盘空间的限制

文件数目的限制

④ 设置组的过渡期, 进入vi后直接修改所要更改的过渡期即可:

```
# edquota -g -t
```

Grace period before enforcing soft limits for groups:

Time units may be: days, hours, minutes, or seconds

Filesystem	Block grace period	Inode grace period
/dev/hdd2	7days	7days

修改此处

efax

常用指数:



功能说明: 收发传真。

语 法: efax [-sw] [-a <AT 命令>] [-c <调制解调器属性>] [-d <驱动程序>] [-f <字体文件>] [-g <命令>] [-h <传真标题字符串>] [-i <AT 命令>] [-j <AT 命令>] [-k <AT 命令>] [-l <识别码>] [-o <选项>] [-q <错误次数>] [-r <文件名>] [-v <信息类型>] [-x <UUCP 锁定文件>] [-t <电话号码> <传真文件>]

包 名 称: efax。

补充说明: 支持Class 1与Class 2的调制解调器来收发传真。

参 数:

-a <AT 命令>	以指定的AT命令来接电话。默认为A命令。AT指的是调制解调器通用的AT命令集
-c <调制解调器属性>	<p>设置本机调制解调器的属性。总共有8个属性，分别以vr、br、wd、ln、df、ec、bf、st来表示。属性之间以逗号来区隔。每个属性的表示意义如下：</p> <p>vr 垂直分辨率。0表示98 lpi（每英寸线条数），1表示196 lpi。</p> <p>br 传输速率，单位为bit/s（位/秒）： 0表示2400 bit/s 1表示4800 bit/s 2表示7200 bit/s 3表示9600 bit/s 4表示12000 bit/s 5表示14400 bit/s</p> <p>wd 面宽度： 0表示8.5英寸（等于Letter的宽度） 1表示10英寸 2表示12英寸</p> <p>ln 页面长度： 0表示11英寸（等于Letter的长度） 1表示14英寸 2表示没有长度限制</p>

	<p>df 数据编码格式： 0表示1-D编码 1表示2-D编码</p> <p>ec 错误修正机制： 0表示没有错误修正 1表示启动错误修正的功能</p> <p>bf 二进制文件传输： 0表示不传输二进制文件 1表示可以传输二进制文件</p> <p>st 最少扫描时间： 0表示每条线条需要的扫描时间为0秒 1表示每条线条需要的扫描时间为5毫秒 3表示每条线条需要的扫描时间为10毫秒 5表示每条线条需要的扫描时间为20毫秒 7表示每条线条需要的扫描时间为40毫秒</p>
-d <驱动程序>	指定调制解调器驱动程序，默认值为/dev/fax
-f <字体文件>	使用指定的字体文件来建立传真标题
-g <命令>	若接到的电话为数据（不是传真），则执行指定的命令。例如，“exec /bin/sh/getty -h”，可启动登录的程序
-h <传真标题字符串>	指定字符串为每页最前端的标题。字符串中第一个%d变量会替换为页码，第二个%d变量则替换为总页数
-i <AT 命令>	在调制解调器进入传真模式前，传送AT命令至调制解调器
-j <AT 命令>	在调制解调器进入传真模式后，传送AT命令至调制解调器
-k <AT 命令>	在调制解调器离开传真模式前，传送AT命令至调制解调器
-l <识别码>	设置本机调制解调器的识别码，通常使用本机的电话号码
-o <选项>	<p>使用非标准调制解调器时，须在此设置相关选项：</p> <p>1 使用Class 1传真调制解调器命令</p> <p>0 使用Class 2传真调制解调器命令，这是默认值</p> <p>a 先以数据模式接听电话，若无法连接，再为传真模式接听</p> <p>e 忽略调制解调器初始化时产生的错误</p> <p>n 忽略对方调制解调器要求重送页面的要求</p> <p>x 以XON (DC1) 模式来接收数据，而非DC2模式。此选项只适用于Class 2调制解调器</p> <p>z 在每个AT命令之间插入100毫秒，以避免信号传送过快，调制解调器无法辨识</p>

-q <错误次数>	接收传真时，当每页发生错误的次数超过指定的数目时，即要求对方重送。默认值为10
-r <文件名>	在接收传真时，将每页分别存储成文件。文件的命名方式为<文件名>.001、<文件名>.002 ... 依此类推
-s	调制解调器初始化之后，解除调制解调器的锁定。如此在等待接听时，也能对外拨号。当efax检测到来电时，即重新锁定调制解调器
-v <信息类型>	<p>选择要印出的信息类型。-v参数最多可使用两次，第一次出现的-v会将信息送至标准错误输出设备；第二次出现的-v则送至标准输出设备。默认仅使用一次-v参数送至标准错误输出设备，默认选项为ewin。各信息类型如下：</p> <ul style="list-style-type: none"> e 错误信息 w 警告信息 i 执行进度 n 与其他调制解调器沟通时的信息 a 程序的参数 c 调制解调器的命令与响应信息 f 有关影像文件的详细信息 m 调制解调器的输出信息 r 接收数据不正常时所产生信息 t 有关传输过程的详细信息W x 锁定调制解调器的过程
-w	不要接听电话，等待OK或CONNECT的信号。此参数可将调制解调器设置为自动接听的状态。若要在接听电话前，先等待数声响铃，请加入<iS0=n>选项，n为等待的铃声数
-x <UUCP 锁定文件>	使用UUCP (Unix-to-Unix copy) 格式的锁定文件来锁定调制解调器
-t <电话号码> <传真文件>	以<电话号码>中的号码来拨号，并将<传真文件>传真出去。若是采用按键式电话的音频来拨号，则可在<电话号码>前加“T”

范 例:

① 将faxdoc1与faxdoc2传真至2345-6789:

```
# efax -d /dev/fax -o1 -t T23456789 faxdoc1 faxdoc2
```

要传真的图文件

T表示以按键式电话的音频来拨号

使用Class 1的调制解调器

② 自动接听来电:

```
# efax -d /dev/fax -w -iS0=3 -r received
```

收到的传真将存为received.
001、received.002 ...等文件

等铃声响三声后再接电话

egrep (extended regexp)

常用指数:



功能说明: 查找文件里符合条件的字符串。

语 法: egrep [范本样式] [文件或目录...]

包 名 称: grep。

相关命令: fgrep, grep, rgrep。

补充说明: 在文件内查找指定的字符串。egrep执行效果如同grep -E, 使用的语法及参数可参照grep命令, 与grep不同点在于解读字符串的方法, egrep是用extended regular expression语法来解读, 而grep则用basic regular expression语法, extended regular expression比basic regular expression有更完整的表达规范。

eject

常用指数:



功能说明: 退出抽取式设备。

语 法: eject [-dfhnqrstv] [-a <开关>] [-c <光驱编号>] [设备]

包 名 称: eject。

相关命令: mount, umount。

补充说明: 若设备已加载, 则eject会先将该设备卸载再退出。请注意, 有些硬件设备可能不支持eject命令。

参 数:

[设备]	设备可以是驱动程序名称 (例如/dev/hdd), 也可以是加载点 (例如/mnt/cdrom)
-a <开关> 或 --auto <开关>	控制设备的自动退出功能。在<开关>选项中输入on或1, 表示启动自动退出功能; 输入off或0, 表示关闭自动退出功能。启动设备的自动退出功能后, 一旦卸载, 该设备即自动退出
-c <光驱编号>或 --changerslut<光驱编号>	选择光驱柜中的光驱。第一个光驱的编号是0
-d或--default	显示默认的设备, 而不实际执行操作
-f或--floppy	退出抽取式磁盘 (例如zip磁盘)
-h或--hlep	显示帮助
-n或--noop	显示指定的设备
-q或--tape	退出磁带
-r或--cdrom	退出光盘
-s或--scsi	以SCSI命令来退出设备
-t或--trayclose	关闭光驱的托盘。请注意, 一些旧式的光驱可能不支持此命令
-v或--verbose	执行时, 显示详细的说明

E

范 例:

① 显示当前所选择的设备:

```
# eject -n
eject 设备是 '/dev/cdrom'
```

② 退出光盘, 然后关闭光驱托盘:

```
# eject /mnt/cdrom      ←退出光盘
# eject -t /mnt/cdrom  ←关闭光驱托盘
```

③ 启动 hdd 设备的自动退盘功能, 并在执行时显示详细的信息:

```
# eject -v -a on /dev/hdd
```

启动自动退盘的功能

④ 显示默认的设备:

```
# eject -d
eject: 内定设备为: `cdrom'      ←默认设备为cdrom
```

enable

常用指数:



功能说明: 启动或关闭 shell 内建命令。

语 法: enable [-a] 或
enable [-n] [内建命令...]

包名称: cups (Red Hat Linux/Fedora Core), cups-common (Mandrake Linux)。

补充说明：若要执行的文件名称与shell内建命令相同，可用enable -n来关闭shell内建命令。若不加-n参数，enable可重新启动关闭的命令。

参 数：

a	显示shell所有关闭与启动的命令
n	关闭指定的shell内建命令，若使用此参数而不指定内建命令，则可列出shell当前关闭的命令

范 例：

① 显示当前 shell 所有内建命令的状态：

```
# enable -a
enable .
enable :
enable [
enable alias
enable bg
enable bind
enable break
enable builtin
...
```

② 以下将cd命令关闭后，再重新启动：

```
# enable -n cd          ← 关闭cd命令
# cd /root             ← 试试看执行cd命令的结果
bash: cd: command not found ← 无法执行cd命令
# enable -n           ← 显示当前关闭的命令
enable -n cd
# enable cd           ← 重新启动cd命令
```


enscript

常用指数:



功能说明: 将文本文件转为 PostScript。

语 法: enscript [-BcgGhjkKlmOqrRvVzZ] [-# <打印份数>] [-1] [-2] [-a <打印范围>] [-A <奇偶数>] [-b <页首>] [-C<行号>] [-d <打印机名称>] [-D <Page device 选项>] [-e <字符>] [-E <程序语言>] [-f <字体类型名称>] [-F <字体类型名称>] [-H<行数>] [-i<字节数>] [-I <筛选命令>] [-L <打印行数>] [-M <纸张类型>] [-n <份数>] [-N <换行符号>] [-p <文件>] [-P <打印机名称>] [-s <行距>] [-S <打印机命令:值>] [-t <标题>] [-T <tab 大小>] [-u<背景文字>] [-U <页数>] [-W <输出语言>] [-X <字符集名称>] [--color=<色彩模式>] [--columns=<每页栏数>] [--download-font=<字体文件>] [--help] [--help-pretty-print] [--highlight-bar-gray=<灰度>] [--list-media] [--list-options] [--margins=<左:右:上:下>] [--mark-wrapped-lines=<换行标示>] [--non-printable-format=<表示方法>] [--page-label-format=<页面标签>] [--ps-level=<层级>] [--printer-options=<打印机命令参数>] [--rotate-even-pages] [--toc] [--ul-angle=<背景文字角度>] [--ul-font=<字体类型名称>] [--ul-gray=<灰度>] [--ul-position=<起始位置>] [--ul-style=<背景文字样式>] [--word-wrap] [要打印的文件...]

包 名 称: enscript。

相关命令: mpage。

补充说明: enscript可将文本文件转为 PostScript 文件,再送往指定的打印机。enscript提供许多选项供用户自定义格式,同时也支持许多类型的打印机。

参 数:

-# <打印份数>	指定每页打印的份数												
-l、2或--columns =<每页栏数>	指定每页的栏数为1栏、2栏或参数中的栏数												
-a <打印范围>或 --pages=<打印范围>	指定要打印的范围。<打印范围>的格式如下： <table border="0"> <tr> <td>起始页码 - 结束页码</td> <td>打印起始页码与结束页码之间的页面</td> </tr> <tr> <td>- 结束页码</td> <td>从第一页打印至指定的页面</td> </tr> <tr> <td>起始页码 - 页码</td> <td>从指定页面打印至最后一页</td> </tr> <tr> <td>odd</td> <td>仅打印指定的页面</td> </tr> <tr> <td>even</td> <td>打印奇数页</td> </tr> <tr> <td></td> <td>打印偶数页</td> </tr> </table>	起始页码 - 结束页码	打印起始页码与结束页码之间的页面	- 结束页码	从第一页打印至指定的页面	起始页码 - 页码	从指定页面打印至最后一页	odd	仅打印指定的页面	even	打印奇数页		打印偶数页
起始页码 - 结束页码	打印起始页码与结束页码之间的页面												
- 结束页码	从第一页打印至指定的页面												
起始页码 - 页码	从指定页面打印至最后一页												
odd	仅打印指定的页面												
even	打印奇数页												
	打印偶数页												
-A <奇偶数>或 --file-align=<奇偶数>	在打印多份文件时，指定文件对齐的方式。在双面打印时使用偶数选项，可使每个文件的第一页都朝上												
-b <页首> 或 --header=<页首>	将<页首>中的字符串设置为页首。默认值为文件名称、文件修改日期与页码												
-B 或 --no-header	不打印页首												
-c 或 --truncate-lines	若某行的长度超出打印页面，则加以截断。默认为自动换行												
-C<行号> 或--line -numbers=<行号>	在每行前面加上行号。<行号> 选项表示开始打印的行号，默认值为1												
-d <打印机名称>	指定打印机												
-D<Page device选项> 或--setpagedevice= <Page device选项>	Page device为PostScript的规格。选项必须加上布林值来控制，例如，Collate:true表示启动Collate选项。以下为常用的一些选项： <table border="0"> <tr> <td>Collate 多重打印时</td> <td>页面输出的顺序</td> </tr> <tr> <td>Duplex</td> <td>双面打印</td> </tr> <tr> <td>ManualFeed</td> <td>手动送纸</td> </tr> <tr> <td>OutputFaceUp</td> <td>打印面朝上或朝下</td> </tr> <tr> <td>Tumble</td> <td>双面打印时的方向</td> </tr> </table>	Collate 多重打印时	页面输出的顺序	Duplex	双面打印	ManualFeed	手动送纸	OutputFaceUp	打印面朝上或朝下	Tumble	双面打印时的方向		
Collate 多重打印时	页面输出的顺序												
Duplex	双面打印												
ManualFeed	手动送纸												
OutputFaceUp	打印面朝上或朝下												
Tumble	双面打印时的方向												
-e <字符> 或--escapes=<字符>	指定脱离 (Escape) 字符。在<字符>选项中指定的字符，将取代默认的脱离字符												

-E <程序语言>或--pretty-print=<程序语言>	若要打印的是程序语言的源代码，利用此参数可自动加上格式。若不加上<程序语言>选项，则enscript会自行判断
-f <字体类型名称>或--font=<字体类型名称>	指定内文的字体类型，默认为Courier字体，文字大小为10
-F <字体类型名称>或--header-font=<字体类型名称>	指定页首的字体类型
-g或--print-anyway	不管文件内容是什么，即使是二进制文件，仍然照常打印
-G 或--fancy-header	用花哨的方式打印页首
-h 或 --no-job-header	省略打印作业的工作说明页
-H<行数> 或--highlight-bars=<行数>	在后台打印等宽的横条。<行数>选项可指定横条的高度，默认值为2行。若不指定此参数，则默认不打印横条
-i<字节数>或--indent=<字节数>	设置缩排。<字节数>选项可指定缩排的宽度，单位为字节
-I <筛选命令> 或--filter=<筛选命令>	以<筛选命令>处理文件后再打印。筛选命令可以是单一命令，或是一连串的命令。在<筛选命令>选项中，以%s来表示要打印的文件
-j或--borders	打印时加上边框。若页面分栏，则在栏边加框
-k或--page-prefeed	设置预先送纸
-K或--no-page-prefeed	设置不预先送纸，这是默认值
-l或--lineprinter	模拟行列式打印机 (Line printer)
-L<打印行数>或--lines-per-page=<打印行数>	指定每页打印行数
-m 或 --mail	完成打印后，以E-mail通知用户
-M <纸张类型>或--media=<纸张类型>	设置打印纸张的类型，默认为Letter
-n <份数>或--copies=<份数>	指定每页要打印的份数
-N <换行符号>或--newline=<换行符号>	指定换行符号。常见的换行符号为'\n'或'\r'

-O或--missing-characters	显示无法打印的字符码
-p <文件> 或 --output=<文件>	输出至文件，所产生的文件为PostScript的格式，可利用Ghostview等工具程序来查看
-P <打印机名称> 或--printer=<打印机名称>	指定打印机。此参数与-d相同
-q、-quiet或--silent	除非发生错误，否则不显示任何信息
-r或--landscape	指定横式打印
-R或--portrait	指定竖式打印
-s <行距>或--base-lineskip=<行距>	指定行距，默认值为1
-S <打印机命令：值> 或--statusdict=<打印机命令：值>	加上打印机命令，例如，指定纸匣等。打印机命令因机型而异，请参考打印机使用手册
-t <标题> 或--title=<标题>	设置打印作业首页的标题
-T <tab 大小>或 --tabsize=<tab大小>	定义tab的大小，默认值为8
-u<背景文字>或--underlay=<背景文字>	页面上加入背景文字，会产生像浮水印的效果
-U <页数>或--nup=<页数>	将许多页的内容缩小打印在一页上。指定的页数必须为2的幂方（2、4、8等）
-v或--verbose	打印时，在屏幕上显示详细的信息
-V或--version	在屏幕上显示enscript的版本信息
-W <输出语言>或 --language=<输出语言>	设置输出的语言。默认值为PostScript，此外还有html、overstrike与rtf等选项
-X <字符集名称> 或--encoding=<字符集名称>	设置输入的字符集，默认值为88591
-z或--no-formfeed	不处理送纸字节
-Z或--pass-through	直接打印PostScript与PCL文件，不做额外的处理
--color=<色彩模式>	设置-E或--pretty-print参数打印时所用的色彩模式。默认值为emacs，此外也可选择使用blackwhite模式

<code>--download-font= <字体文件></code>	传送指定的字体文件给打印机
<code>--help</code>	在屏幕上显示帮助
<code>--help-pretty-print</code>	在屏幕上显示-E参数所支持的程序语言
<code>--highlight-bar- gray=<灰度></code>	指定以-H参数打印时，横条的灰度设置。灰度值为小数点下一位，最大值为1（颜色最浅），最小值为0（颜色最深）
<code>--list-media</code>	在屏幕上显示所有支持的纸张类型
<code>--list-options</code>	在屏幕上显示所有的选项与设置值
<code>--margins=<左:右: 上:下></code>	设置纸张的打印边界，单位为PostScript点数
<code>--mark-wrapped- lines=<换行标示></code>	以指定的符号来标示自动换行的地方，选项如下： none 不标示（默认值） plus 加号 box 黑色方块 arrow 箭头
<code>--non-printable- format=<表示方法></code>	设置当有无法打印的字节时，所采取的处理方式。选项如下： caret 以“^”来表示 octal 以该字符的八进制来表示，这是默认值 questionmark 以问号来表示 space 以空白字符来表示
<code>--page-label-format =<页面标签></code>	设置页面标签。选项如下： short 标签仅包含页码（默认值） Long 标签包含文件名与页码
<code>--ps-level=<层级></code>	设置PostScript语言的层级，共有1与2两种选项
<code>--printer-options= <打印机命令参数></code>	加上打印机命令的参数
<code>--rotate-even-pages</code>	打印偶数页时，倒转180度
<code>--toc</code>	在文件最后加上目录页
<code>--ul-angle=<背景 文字角度></code>	设置背景文字的角度
<code>--ul-font=<字体类 型名称></code>	设置背景文字的字体类型。默认字体类型为Times-Roman 200

<code>--ul-gray=<灰度></code>	设置背景文字的灰度值。灰度值可达小数点下一位。最大值为1（颜色最浅），最小值为0（颜色最深）
<code>--ul-position=<起始位置></code>	设置背景文字的起始位置。位置的格式如：(+/-) <X轴坐标>(+/-)<Y轴坐标>。例如，+5-5中间不可加上空白字节
<code>--ul-style =<背景文字样式></code>	设置背景文字的样式。样式选项如下： outline ← 描边字体（默认值） filled ← 实心字体
<code>--word-wrap</code>	换行时，避免截断文字

范 例：

① 打印3份textfile文件，并设置每页的标签包含文件名和页码：

```
# enscript -#3 --page-label-format=long textfile
```

↑
打印3份 每页标签包含文件名和页码

② 以每页2栏的方式打印textfile文件，如遇到换行，则尽量避免截断文字：

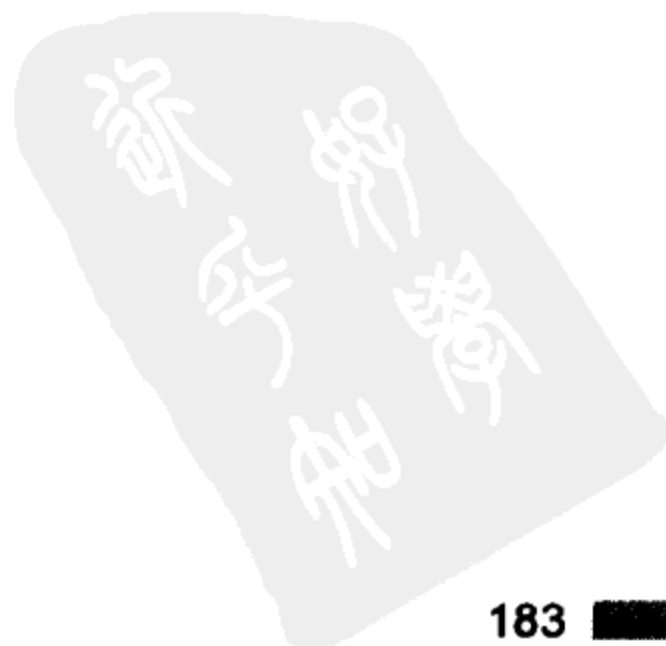
```
# enscript -2 --word-wrap textfile
```

↑
每页2栏 换行时避免截断文字

③ 以每页3栏的方式打印textfile文件，并为每栏加上边框：

```
# enscript --columns=3 -j textfile
```

↑
每页3栏 加边框



④ 打印textfile文件的第3页至第7页，并设置行距为3：

```
# enscript -a 3-7 -s 3 textfile
```

打印第3-7页 行距为3

⑤ 打印textfile文件的奇数页，并指定以横向的方式打印：

```
# enscript -a odd -r textfile
```

打印奇数页 横式打印

⑥ 打印textfile文件时，不打印页首，并以双面的方式打印：

```
# enscript -B -D Duplex:true textfile
```

不打印页首 以双面的方式打印

⑦ 打印textfile文件时，以指定的格式打印页首：

```
# enscript --header='#n %W Page %# of #' textfile
```

文件名称 打印日期 此页的页码 此文件的总页数

⑧ 打印textfile文件时，在每行前面加上行号，并在最后加上目录页：

```
# enscript -C --toc textfile
```

每行加上行号 加上目录页

- ⑨ 指定以Times-Roman字体打印textfile文件，并以宽度10点、高度12点的文字大小打印：

```
# enscript -f Times-Roman@10/12 textfile
```

|
↑
↑

内文的字体
字体高度
字体宽度

- ⑩ 以花哨的方式，并以Courier字体、大小为7.5点来打印textfile文件：

```
# enscript -F Courier7.5 -G textfile
```

|
↑

页首的字体种类及大小
以较花哨的方式打印页首

- ⑪ 打印textfile时，在后台打印等宽的横条，并指定横条的高度为3行，灰度值为0.9：

```
# enscript -H3 --highlight-bar-gray=0.9 textfile
```

↑
|

横条高度
横条的灰度值

- ⑫ 打印textfile时，指定每页以3个字节的宽度缩排：

```
# enscript -i3 textfile
```

↑

缩排3个字节

- ⑬ 打印textfile时，将每4页缩小为一页打印：


```
# enscript -U 4 textfile
```


⑨ 打印textfile时，加上Linux的背景文字，指定背景文字的旋转角度为10度，并从左上角(0.0)的位置开始打印背景文字：

```
# enscript -uLinux --ul-angle=10 --ul-position=+0-0 textfile
```

↑
↑
 背景文字的角度 背景文字的起始位置

eval

常用指数：

功能说明：重新计算求出参数的内容。

语法：eval [参数...]

包名称：bash。

补充说明：eval可读取一连串的参数，然后再根据参数本身的特性来执行。通常是用来处理shell script中的变量。

参数：参数不限数目，彼此之间以分号(;)隔开。参数可以是命令或变量。


范例：

以下的范例包含var1与var2两个变量，分别利用eval及不用eval命令显示var2的内容：

```
# var2=\$var1                    ← 将var2变量设成var1
# var1=tyrell                   ← 将var1设成tyrell
# eval echo $var2               ← 利用eval显示var2变量的内容
tyrell
# echo $var2                   ← 直接显示var2变量的内容
$var1
```

E


ex

常用指数: 

功能说明: 在Ex模式下启动vim文本编辑器。ex执行效果如同vi-E, 使用语法及参数可参照vi命令, 如要从Ex模式回到普通模式, 则在vim中输入:vi或:visual即可。

包名称: Vim-minimal。

exec

常用指数: 

功能说明: shell执行指定的命令后即交出控制权。

语法: exec [命令 <命令参数>]

包名称: bash。

补充说明: 在shell中执行其他命令或程序后, 执行控制权不会交回给shell。因此, 除非在参数中指定了其他shell, 否则执行后会回到登录的画面。

参数: 指定要执行的命令及参数。

范例:

① 执行完ls命令后, 即退出当前的shell:

```
# exec ls
```



② 切换到C shell:

```
# exec csh      ← 指定其他的shell
~              ← 已经切换到C shell了
```

exit

常用指数: **功能说明:** 退出当前的 shell。**语 法:** exit [状态值]**包 名 称:** bash。


补充说明: 执行exit可使shell以指定的状态值退出。若不设置状态值参数，则shell以默认值（最后执行命令的状态值）退出。状态值0表示执行成功，其他值表示执行失败。

exit也可用在script，离开正在执行的script，回到shell。

参 数: 无。**范 例:**

退出 shell:

```
# exit
Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8.0 on an i686
flag login:
```


 重新登录画面

export

常用指数:



功能说明: 设置或显示环境变量。

语法: export [-fn] [变量名称[=变量设置值]] 或
export -p

包名称: bash。

补充说明: 在shell中执行程序时, shell会提供一组环境变量。export可新增、修改或删除环境变量, 供后续执行的程序使用。export的效力仅及于该次登录作业。

参数:

-f	表示 [变量名称] 中为函数名称
-n	删除指定的变量。变量实际上并未删除, 只是不会输出到后续命令的执行环境中
-p	列出所有shell赋予程序的环境变量

范例:

① 设置HISTFILESIZE变量:

```
# export HISTFILESIZE=750
# echo $HISTFILESIZE
750
```



② 删除HOME变量:


```

# export -p
declare -x BASH_ENV="/root/.bashrc"
declare -x G_BROKEN_FILENAMES="1"
declare -x HISTFILESIZE="750"
declare -x HISTSIZE="1000"
declare -x HOME="/root"    ←存在declare -x HOME="/root"
declare -x HOSTNAME="free.flag.com.tw"
...
# export -n HOME
# echo $HOME
/root
# export -p
declare -x BASH_ENV="/root/.bashrc"
declare -x G_BROKEN_FILENAMES="1"
declare -x HISTFILESIZE="750"
declare -x HISTSIZE="1000"
declare -x HOSTNAME="free.flag.com.tw"
...

```

找不到declare -x HOME="/root"了

ext2ed (ext2 file system editor)

常用指数: **功能说明:** ext2文件系统编辑程序。**语 法:** ext2ed**包 名 称:** ext2ed (Mandrake Linux)。**相关命令:** mke2fs, fsck, fsck.ext2。**补充说明:** 这是Mandrake Linux才有的命令。ext2ed可直接处理硬盘分区上的数据,该命令只有某些Linux发行版才提供。不过,除非相当熟悉Linux ext2文件系统的架构,否则建议不要轻易尝试以ext2ed来修改分区中的数据。

ext2ed当前只支持GB以下的分区。请注意不要将此程序用在已加载的文件系统上，且使用时务必格外小心。与ext2ed相关的文件如下：

/usr/bin/ext2ed	ext2ed程序
/var/lib/ext2ed/ext2ed.conf	ext2ed的设置文件
/var/lib/ext2ed/ext2.descriptors	ext2文件系统中各项对象的定义
/var/lib/ext2ed/ext2ed.log	ext2ed执行的记录文件

E

参 数：无。

范 例：

要将superblock还原成最近一次的备份，可利用稍后所介绍的操作命令操作：

```
# ext2ed
...
ext2ed > setdevice /dev/hda2  ← 设置分区
ext2ed > enablewrite         ← 设成可写入状态
ext2ed > super               ← 进入superblock模式
ext2ed > gocopy 1            ← 移到superblock最近的备份
ext2ed > setactivecopy       ← 将当前所在的superblock复制
                             至main superblock
ext2ed > writedata          ← 实际执行修改的操作
ext2ed > quit
```

操作命令：进入ext2ed之后，请在屏幕下方的ext2ed >提示符下输入命令：

一般操作命令	
setdevice [设备名称]	指定要处理的设备，例如，setdevice /dev/hdb1。建议不要使用当前加载的设备。如果指定的分区当前已加载，则将无法执行enablewrite（写入）的命令。即使只是要用ext2ed来查看分区的状态，仍然建议先将该设备卸载
disablewrite	将ext2ed设为只读的状态，这也是程序执行时的默认值
enablewrite	将ext2ed设为可读写的状态
help [命令]	显示个别命令的帮助
next	移到下一个单位，单位会根据当前所在的模式而异
prev	移到前一个单位，单位会根据当前所在的模式而异
pgup	移到下一页
pgdn	移到前一页
set	修改当前的数据，参数会根据当前所在的模式而异
writedata	在执行此命令之后，才会实际修改分区中的数据。也就是说，如果在修改过程中反悔了，只要尚未执行writedata命令，结束ext2ed程序即可还原

ext2ed进入3种模式的操作命令

super	进入main superblock，即Superblock模式。Super block会有很多备份，main superblock是指分区当前的superblock
group <编号>	进入指定的group（组），即Group模式
cd <目录或文件>	在Inode模式下，进入指定的目录或文件。即在Inode模式下，用此命令可轻易地找到所要修改的目录或文件

Superblock 模式

gocopy<备份编号>	进入指定的superblock备份。gocopy 0即回到main superblock
setactivecopy	将当前所在的superblock复制到main superblock。若是分区因superblock损毁而无法读取文件数据，可利用此命令将数据挽回

Group 模式	
blockbitmap	显示当前group的位图
inode	进入当前group的第一个inode
inodebitmap	显示当前group的inode二进制码
Inode 模式	
dir	进入目录模式。在目录模式中可用set <变量名称=设置值>命令来修改数据
file	进入文件模式。在文件模式中可用set <变量名称=设置值>命令来修改文件中的数据

E



fbset (frame buffer setup)

常用指数: 

功能说明: 设定帧缓冲 (Frame Buffer)。

语 法: fbset [-ahinsvVx] [-db <信息文件>] [-fb <外围设备代号>]
[--test][显示模式]

包 名 称: fbset。

补充说明: 在执行此命令前, 需先启用帧缓冲 (Frame Buffer)。fbset 命令可用于设定帧缓冲的大小, 还能调整画面的分辨率、位置、高低宽窄、颜色深度 (即颜色多少), 并可决定是否要启动显卡的各项硬件特性。每一个帧相当于一个画面由左至右、从上到下完整显示一次, 可说成是一页或一幅画面。下面是有关显示模式设定的列表, 首先是与几何特性相关的设定:

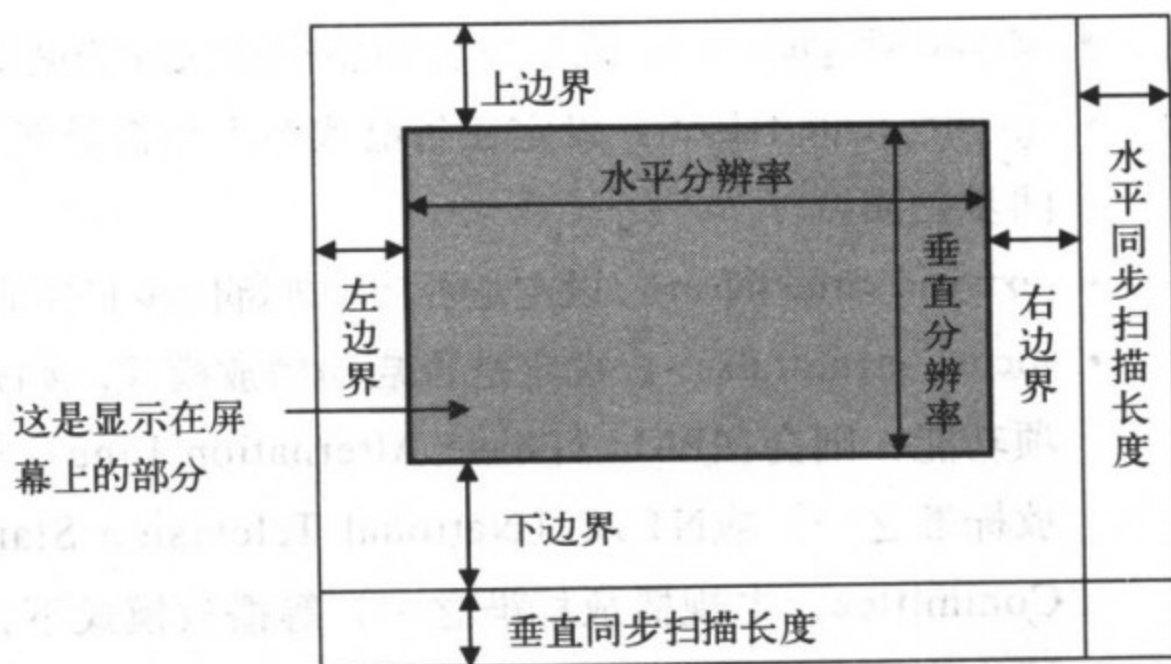
- -xres <设定数值>: 设定画面的水平分辨率, 单位为像素 (Pixel)。例如分辨率设成640, 则显示器的水平方向就会以640像素来显示画面。
- -yres <设定数值>: 设定画面的垂直分辨率, 单位为像素。例如分辨率设成480, 则显示器的垂直方向就会以480像素来显示画面。
- -vxres <设定数值>: 设定虚拟水平分辨率, 单位为像素。
- -vyres <设定数值>: 设定虚拟垂直分辨率, 单位为像素。
- -depth <设定数值>: 设定颜色深度, 单位为位。例如深度设成16位, 则每个像素能使用16个位来记录颜色的变化, 即可显示2的16次方颜色, 也就是65 536种颜色。

- **-g <设定数值>或--geometry <设定数值>**: 同时指定全部有关几何特性的显示数值, 即上述5项显示模式的设定。例如设定的数值为“640 480 800 600 16”, 就表示水平分辨率为640像素、垂直分辨率为480像素、虚拟水平分辨率为800像素、虚拟垂直分辨率为600像素, 而颜色深度则是16位。

其次是与时序特性相关的设定:

- **-pixclock <设定数值>**: 设定每个像素的时序长短, 单位为一兆分之一秒, 也就是10的12次方分之一秒。
- **-left <设定数值>**: 设定画面的左边界, 单位为像素。
- **-right <设定数值>**: 设定画面的右边界, 单位为像素。
- **-upper <设定数值>**: 设定画面的上边界, 单位为像素行。
- **-lower <设定数值>**: 设定画面的下边界, 单位为像素行。
- **-hslen <设定数值>**: 设定水平同步扫描长度, 单位为像素。
- **-vslen <设定数值>**: 设定垂直同步扫描长度, 单位为像素行。
- **-t <设定数值>或--timings <设定数值>**: 同时指定全部有关时序特性的显示数值, 即上述7项显示模式的设定。例如设定的数值为“32768 64 96 32 16 108 2”, 就表示像素时序为32 768、左边界为64像素、右边界为96像素、上边界为32像素行、下边界为16像素行, 而水平同步扫描长度为108像素, 垂直同步扫描长度为2像素行。

以图解方式说明上述各边界及长度间的相对关系, 更易于了解:



因此，水平和垂直方向的总长度分别为：

水平方向总长度 = 左边界 + 水平分辨率 + 右边界 + 水平同步扫描长度

垂直方向总长度 = 上边界 + 垂直分辨率 + 下边界 + 垂直同步扫描长度

水平和垂直扫描频率可依此计算得知：

每列长度 = 每一像素的时序 × 水平方向总长度

水平扫描频率 = 10^{12} 次方 / 每行长度

若为交错扫描，则垂直方向总长度减半；若为双重扫描，则垂直方向总长度加倍。

每个帧 = 垂直方向总长度 × 每行长度

垂直扫描频率 = 10^{12} 次方 / 每个帧

最后是与硬件特性相关的设定：

- `-accel <true/false>`：设定是否启动纯文字模式的硬件加速，“true”为开启，“false”为关闭。
- `-hsync <high/low>`：设定水平同步扫描极性的高低。
- `-vsync <high/low>`：设定垂直同步扫描极性的高低。

- `-csync <high/low>`: 设定复合同步扫描极性的高低。
- `-gsync <true/false>`: 设定在省电状态下是否仍然要启动同步扫描。
- `-extsync <true/false>`: 设定是否启动外部同步扫描的时序。
- `-bcast <true/false>`: 设定是否启动播放模式。如开启这项功能, 则会在PAL (Phase Alternation Line, 电视播放标准之一) 或NTSC (National Television Standards Committee, 电视播放标准之一) 等播放模式下, 使用精确的时序。
- `-laced <true/false>`: 设定是否启动交错扫描模式。如开启这项功能, 则每个显示画面将一分为二, 单数列与双数列交错显示。
- `-double <true/false>`: 设定是否启动双重扫描模式。如开启这项功能, 则每列将显示两次。
- `-move <up/down/left/right>`: 移动画面的可视范围。
- `-step <设定数值>`: 设定移动画面时每次的位移量, 单位为像素 (水平方向) 或像素行 (垂直方向), 默认值是水平方向每次8个像素, 垂直方向每次2个像素行。

参 数: 除了前述的参数之外, 其余参数列表如下:

<code>-a</code> 或 <code>--all</code>	改变所有使用该设备的虚拟终端的显示模式
<code>-db <信息文件></code>	指定显示模式的信息文件, 默认的文件名称为 <code>fb.modes</code> , 存放在 <code>/etc</code> 目录下
<code>-fb <外围设备代号></code>	指定用来作为输出帧缓冲的外围设备, 默认值为 <code>"/dev/fb0"</code>
<code>-h</code> 或 <code>--help</code>	显示帮助
<code>-i</code> 或 <code>--info</code>	列出所有帧缓冲的相关信息
<code>-n</code> 或 <code>--now</code>	马上改变显示模式。若无使用 <code>"-ofb"</code> 参数, 则此项参数为默认值

s 或 --show	列出当前显示模式的设定。若无使用任何参数，或仅指定“-ifb”和“-ofb”参数，则此项参数为默认值
-v 或 --verbose	显示命令执行过程
-V 或 --version	显示版本信息
-x 或 --xfree86	使用XFree86兼容模式
--test	仅做测试，并不改变当前显示模式

范 例：

① 显示帧缓冲当前的状态：

fbset

mode "1024x768-76"

D: 78.653 MHz, H: 59.949 kHz, V: 75.694 Hz ← 扫描频率

geometry 1024 768 1024 768 16 ← 显示模式

timings 12714 128 32 16 4 128 4 ← 时序特性

rgba 5/11,6/5,5/0,0/0 ← RGB颜色的对应模式

endmode

② 设定画面的像素时序为32 768、左边界为64像素、右边界为96像素、上边界为32像素、下边界为16像素，而水平同步扫描长度为108像素，垂直同步扫描长度为2像素行：

fbset -t 32768 64 96 32 16 108 2 name

③ 启动纯文字模式硬件加速的功能：

fbset -accel true name

④ 将水平、垂直及复合同步扫描属性设成较高的状态：

fbset -hsync high -vsync high -csync high name




⑤ 关闭省电模式下同步扫描的功能：

```
# fbset -gsync false name
```

⑥ 仅测试帧缓冲的状态，并不实际改变当前的显示模式：

```
# fbset -test
```

fc (fix command)

常用指数：

功能说明：修改命令，且执行该命令。

语法：fc [-e <文本编辑程序>] [-lnr] [第一个命令] [最后一个命令] 或
fc [-s <命令日志内的命令>]

F

包名称：bash。

相关命令：history。

补充说明：利用fc命令，能够让用户选择记录于命令日志中的命令，该命令可以是单独一个或是某特定范围内的所有命令，用文本编辑程序修改过后，以批处理方式一起依次执行，大幅节省输入命令的时间。假设没有指定最后一个命令，则fc仅会编辑所给予的单一命令，若搭配“-l”参数，将会列出从该命令起之后的所有命令。如果没有设定任何命令，则fc会编辑最后执行过的命令，若搭配“-l”参数使用，将会列出最后执行过的16个命令。



参 数:

-e <文本编辑程序>	指定欲使用的文本编辑程序，默认的编辑程序为vi
-l	列出第一个与最后一个命令范围内的所有命令
-n	列出命令列表时，不显示编号
-r	列出命令列表时，采用逆向排序
-s <命令日志内的命令>	使用这项参数，fc命令会到命令日志内由后往前查找合乎条件的最后一个命令，然后执行它

范 例:

① 编辑最后执行的命令:

```
# fc
```

② 依次执行过数个命令之后，利用 fc命令回头执行已执行过的命令:

```
# rm *.txt
```

← 当稍后执行“fc -s rm”时，因后面还有符合条件的命令，故不会执行此命令

```
# cd /var/tmp
```

```
# rm *.jpg
```

```
# cd /root
```

```
# fc -s rm
```

← 回头执行符合“rm”条件的最后一个rm命令

```
# fc -s
```

← 假如还要执行“rm *.jpg”，无需再设定命令，直接执行即可

③ 假设最后一个命令的日志编号是1 000，欲逆序执行最后5 个命令:

```
# fc -e joe 1000 996
```

和使用“-r 996 1000”的效果相同

指定使用文本编辑程序joe，编辑后保存文件退出，就会一口气执行这5项命令

④ 列出之前所执行的 5 个命令：

```
# fc -l -5
584      fc -e pico 582 580
585      fc -l 10
586      fc -l
587      ls
588      fc -l -10
```

⑤ 列出编号586到588号的命令列表，列出时并不显示编号：

```
# fc -n -l 586 588
        fc -l
        ls
        fc -l -10
```

F



与上例比较，不会列出命令列表的编号

fdisk

常用指数：



功能说明：磁盘分区。

语 法：fdisk [-b <扇区大小>] [-uv] [外围设备代号] 或
fdisk [-l] [-b <扇区大小>] [-uv] [外围设备代号...] 或
fdisk [-s <分区编号>]

包 名 称：util-linux。

相关命令：cfdisk, sfdisk。

补充说明：fdisk是用来磁盘分区的程序，它采用传统的问答式界面，而非类似DOS fdisk的cfdisk交互式操作界面，因此在使用上较为不便，但功能却丝毫不打折扣。

参 数:

-b <扇区大小>	指定每个扇区的大小。通常硬盘的每个扇区是512字节，其他存储媒体则不一定，例如磁光盘（Magnetic Optical Disc）的扇区一般为2 048字节
-l	列出指定的外围设备的分区表状况。硬盘的外围设备代号随着各种接口不同而不同，IDE 硬盘为/dev/had ~ /dev/hdh，SCSI硬盘为/dev/sda ~ /dev/sdp，ESDI硬盘为/dev/eda ~ /dev/edd，XT硬盘则只有/dev/xda与/dev/xdb可用
-s <分区编号>	将指定的分区大小输出至标准输出上，单位为块（Block）。每台硬盘最多可划分4个主要（Primary）分区，其分区编号由/dev/hda1 ~ /dev/hda4。如更多的分区，则要再建立逻辑（Logical）分区，它存在于扩展（Extended）分区之内，分区编号由/dev/hda5开始。扩展分区必须占据1个主要分区的空间
-u	搭配“-l”参数列表，会用扇区数目取代柱面数目，来标示每个分区的起始位置
-v	显示版本信息

范 例:

① 为系统上第3台 IDE接口硬盘分区，可利用稍后介绍的操作命令进行操作：

```
# fdisk /dev/hdc
```

↑
第3台IDE硬盘的外围设备代号

Command (m for help): ← 提示符号

② 以柱面为单位列出第1台IDE硬盘分区的状态:

```
# fdisk -l /dev/hda
```

```
Disk /dev/hda: 20.5 GB, 20547841536bytes  
255 heads, 63 sectors/track, 2498 cylinders
```

此硬盘有255个磁头、63个扇区及2498个柱面

```
Units = cylinders of 16065 * 512 = 8225280bytes
```

以柱面为显示单位, 每个柱面大小为
16065 * 512bytes, 即8 225 280bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	730	5863693+	83	Linux

此分区的名称

"*" 表示此分区为引导分区

此分区的起始柱面

此分区的结束柱面

此分区的容量, 以块为单位

此分区文件系统的编号

此分区文件系统的类型



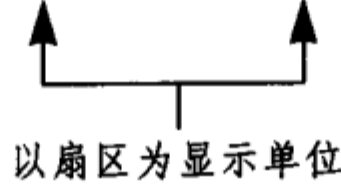
③ 以扇区为单位列出第1个IDE硬盘分区状态:

```
# fdisk -lu /dev/hda
```

```
Disk /dev/hda: 20.5 GB, 20547841536bytes
255 heads, 63 sectors/track, 2498 cylinders, total
40132503 sectors
```

```
Units = sectors of 1 * 512 = 512bytes ← 以扇区为显示单位, 每个扇区大小为512bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	63	11727449	5863693+	83	Linux
/dev/hda2		11727450	23454899	5863725	83	Linux



...

④ 显示/dev/hda1分区的大小:

```
# fdisk -s /dev/hda1
```

```
5863693 ← 此分区有5863693个块
```

操作命令:

a	将分区设成引导分区
b	编辑BSD分区的磁盘标签
C	设成与DOS兼容的分区
d	删除分区
l	列出所有支持分区文件系统的类型
m	显示帮助
n	建立新的分区
o	建立新的DOS分区
p	列出当前分区的状态

q	不存储分区更改后的状态，直接结束程序
s	建立新的SUN分区
t	改变分区文件系统的类型
u	切换以扇区或柱面为单位来显示分区容量
v	检查分区的正确性
w	存储分区更改后的状态，并结束程序
x	切换到高级模式

F

fetchmail

常用指数：



功能说明：接收电子邮件。

语 法：fetchmail [-?aceEFkKmnNqsUvVw] [-A <认证类型>] [-b <邮件数目>] [-B <邮件数目>] [-d <间隔秒数>] [-D <域名>] [-f <设定文件>] [-i <识别文件>] [-I <传输接口>] [-l <邮件大小>] [-L <记录文件>] [-M <传输接口>] [-p <通信协议>] [-P <通信端口>] [-Q <删除字符串>] [-r <邮件目录>,...] [-S <寄信主机>] [-t <超时秒数>] [-u <账号名称>] [-Z <错误号码>, ...] [--bsmtp] [--configdump] [--invisible] [--lmtpt] [--nobounce] [--nosyslog] [--plugin <外挂程序>] [--plugout <外挂程序>] [--postmaster] [--syslog] [邮件服务器...]

包名称：fetchmail。

相关命令：mail, mutt, pine。

补充说明：fetchmail命令用于电子邮件的收发，它从指定的邮件服务器上接收电子邮件，然后分发给信件的收件者。fetchmail可当作常驻服务执行，重复地定时收发邮件。

参 数:

-? 或 --help	显示帮助
-a 或 --all	接收所有邮件，不论新旧。所谓新邮件是指尚未标示成已阅读的邮件，默认fetchmail 仅会下载新邮件。当邮件通信协议为 ETRN 模式时，本参数无效
-A <认证类型>或 --auth <认证类型>	设定连接邮件服务器时的认证类型。可使用的类型有：“kerberos”、“kerberos_v5”和“password”。当通信协议采用“KPOP”时，fetchmail命令会自动使用“kerberos”认证类型，其余则默认为“password”。当邮件通信协议为ETRN模式时，本参数无效
-b <邮件数目>或 --batchlimit <邮件数目>	设定在与邮件服务器连接之前，要传送给SMTP的邮件数目。默认值为0，表示没有限制。当邮件通信协议为 ETRN模式时，本参数无效
-B <邮件数目>或 --fetchlimit<邮件数目>	设定每次与邮件服务器连接之后，要下载的邮件数目。默认值为0，表示没有限制。当邮件传输协议为ETRN模式时，本参数无效
-c 或--check	只检查邮件状况，而不真正去接收或删除它。当邮件通信协议为ETRN模式时，本参数无效
-d <间隔秒数>或 --daemon <间隔秒数>	将fetchmail命令以常驻服务的类型执行。可以指定fetchmail收发电子邮件的间隔时间，单位以秒计算
-D <域名>或--smtpaddress <域名>	指定要放在电子邮件中，“RCPT TO”列的域名。如果没有特别设定，默认会采用SMTP服务器的域名
-e或--expunge	以IMAP通信协议连接邮件服务器接收信件之后，立即删除已下载的电子邮件。本参数会在每个删除命令后面加上 EXPUNGE 命令，强制立刻删除邮件。当邮件通信协议为 ETRN、POP2或 POP3模式时，本参数无效
-E或--envelope	让fetchmail命令复制邮件文件头的信封位置

-f <设定文件>或 --fetchmailrc <设定文件>	指定fetchmail的设定文件名称。默认的设定文件为用户专属目录下的.fetchmailrc，可用这项参数改变它。该文件的权限数值至少为0600
-F或--flush	在接收新邮件之前，先删除旧邮件。当邮件传输协议为POP3或IMAP模式时，才能使用本参数
-i <识别文件>或 --idfile <识别文件>	指定fetchmail用来存储POP3用户识别码的文件名称。默认的文件名为.fetchids，可用这项参数改变它
-I <传输界面>或 --interface <传输界面>	指定传输界面供fetchmail命令使用。例如，传输界面的设定格式为“ppp0/205.47.202.83/255.255.255.0”，最前面是传输界面的代号，其次是可接受的IP地址，最后面则是网络掩码（Netmask），彼此之间以“/”分割。如不设定网络掩码，默认值为255.255.255.255
-k或--keep	保留已接收的邮件。通常电子邮件在下载之后，就会将其从邮件服务器上删除，使用这项参数可保留这些电子邮件。当邮件通信协议为ETRN模式时，本参数无效
-K或--nokeep	效果和“-k”参数相反。当邮件通信协议ETRN模式时，本参数无效
-l <邮件大小>或 --limit <邮件大小>	设定每封电子邮件的大小限制。超过这项限制的邮件将不会下载，而留在邮件服务器上。指定邮件大小为0时，表示没有任何限制。当邮件通信协议为ETRN模式时，本参数无效
-L <记录文件>或 --logfile <记录文件>	设定记录文件的文件名称。可以在fetchmail命令当成常驻服务执行时，记录它返回的相关信息。本参数的主要目的在于方便进行程序debug
-m或--mda	强制采用MDA模式传送邮件。一般外寄邮件时，SMTP所使用的通信端口为25，可用此参数直接启动MDA模式，而不经25通信端口
-M <传输接口>或 --monitor <传输接口>	指定监控网络用的传输接口，供fetchmail命令使用

-n或--norewrite	不更改RFC-822文件头。fetchmail命令在接收信件时，会改写每封邮件的文件头（如From、To、Cc、Bcc、Reply-To等），以便该邮件得以正确地回复。可通过这个参数，让电子邮件的文件头不被更改。当邮件通信协议为ETRN模式，本参数无效
-N或--nodetach	在启用常驻服务模式执行fetchmail命令时，不使用后台执行。本参数的主要目的在于方便进行程序debug
-p <通信协议> 或 --protocol <通信协议>	指定邮件的通信协议。可以通过这个参数，设定和邮件服务器沟通的通信协议。默认值为“AUTO”，它会自行尝试有支持的全部协议，也可以直接指定。当前支持的通信协议为：“APOP”、“ETRN”、“IMAP”、“IMAP-GSS”、“IMAP-k4”、“KPOP”、“POP2”、“POP3”、“RPOP”和“SDPS”
-P <通信端口> 或 --port <通信端口>	设定连接邮件服务器的TCP/IP通信端口
-q或--quit	关闭fetchmail命令的常驻服务
-Q <删除字符串> 或--qvirtual <删除字符串>	设定欲从邮件文件头删除的字符串。例如字符串设为“Delivered”，则fetchmail命令会从信件文件头寻找“Delivered”字符串，并将其删除
-r <邮件目录>或 --folder <邮件目录>	指定接收该目录的电子邮件。邮件服务器上都有默认的邮件目录，可利用此参数去设定。可同时指定多个目录，两两之间用“,”号加以区隔
-s或--silent	不显示错误信息。如同时搭配“-v”参数使用，“-s”参数将失去其效果
-S <寄信主机>或 --smtp host <寄信主机>	指定寄发电子邮件的主机
-t <超时秒数> 或 --timeout <超时秒数>	设定等待服务器返回的时间，单位以秒计算，超过此时间则为超时，fetchmail命令会切断连接。默认值在使用参数“-V”时可得知

-u <账号名称>或 --username<账号 名称>	指定登录邮件服务器的账号名称
-U或--uidl	强制使用UIDL (Unique ID Listing)。当邮件通信协议为POP3模式时，才能使用本参数
-v或--verbose	显示命令执行过程。如重复指定这项参数两次，则会显示更为详细的相关信息
-V或--version	显示fetchmail当前相关的设定值及版本信息
-w或--warnings	当邮件大小超过限制时，便提出警告信息。本参数通常搭配参数“-l”使用
-Z <错误号码> 或 --antispam <错误 号码>	设定SMTP错误号码的列表。同时指定多个错误号码，两两之间请用“,”号加以区隔
--bsmtp	把邮件附上BSMTP文件。该文件含有信件通过SMTP常驻服务时所产生的一些命令
--configdump	把读取.fetchmailrc设定文件时的返回信息呈现至标准输出设备
--invisible	试着隐藏fetchmail命令。通常fetch-mail命令在接收信件时，会在每封邮件的文件头写下收件信息。假设使用这项参数，fetchmail将不会在文件头写入这些信息，并尝试让信件看起来像是直接从邮件服务器所发出
--lmtpl	让邮件传输通过LMTP (Local Mail Transfer Protocol) 进行。使用此参数时，默认的25通信端口将不接受
--nobounce	把RFC-1894的错误信息返回给邮件管理者，而非该信件的发信者
--nosyslog	关闭记录fetchmail命令执行的错误信息
-plugin <外挂程序>	设定外挂程序，用来建立TCP (Transmission Control Protocol) 通信协议
-plugout <外挂程序>	此参数的效果和指定“-plugin”参数类似，但用于建立SMTP通信连接时
--postmaster	把没有适当收件者的邮件转寄给电子邮件管理者
--syslog	启动记录fetchmail命令执行的错误信息。此参数必须在系统有执行syslog常驻服务时才能使用

范 例:

① 接收指定的邮件服务器电子邮件:

```
# fetchmail -u testlinux -p POP3 msg.pchome.com.cn
```

↑
用户账号为testlinux

↑
邮件服务器地址

↑
指定使用POP3通信协议

```
Enter password for testlinux@msg.pchome.com.cn:
```

←
输入该账号的密码

```
1 message for testlinux at msg.pchome.com.tw (1287 octets).
```

↑
该账号有一封新的信

```
reading message testlinux@msg.pchome.com.tw 1 of 1
```

```
(1287 octets) . flushed
```

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

←
通知有新的信

② 如果不想每次执行fetchmail都要输入邮件服务器、用户账号、密码及通信协议等参数，也可以在用户根目录下自行建立.fetchmailrc设定文件，其文件的内容如下（以下以root账号为例）:

```
set postmaster "root"
```

← 如果执行fetchmail时，发生任何错误，通知用户root

```
poll msg.pchome.com.tw with proto POP3
```

↑
邮件服务器地址

↑
指定使用的通信协议

```
user "testlinux" there with password "secrets" is root here
```

↑
账号名称

↑
账号密码

↑
此账号的邮件要分发给用户root

```
options fetchall
```

↑
下载该账号的所有邮件

F

设定好.fetchmailrc文件后，可将该文件的权限设成0600，使得只有该账号的用户才能读取这个设定文件。接下来便可直接执行fetchmail来收取邮件：

```
# fetchmail
1 message for testlinux at msg.pchome.com.tw (916
octets).
reading testlinux@msg.pchome.com.tw message
1 of 1 (916 octets) flushed
You have new mail in /var/spool/mail/root
```

不用输入任何参数，即可取得邮件

③ 接收电子邮件时，每次只收取一封信：

```
# fetchmail -B 1
2 messages for testlinux at msg.pchome.com.tw (1480 octets).
```

一共有2封信

```
reading message 1 of 2 (740 octets) flushed
fetchmail: fetchlimit 1 reached; 1 messages left on
server msg.pchome.com.tw account testlinux
```

只下载一封信，留一封信在服务器上

```
fetchmail: Query status=13
You have new mail in /var/spool/mail/root
```

④ 只检查邮件服务器的状态，并不下载邮件：

```
# fetchmail -c
1 message for testlinux at msg.pchome.com.tw (740 octets).
```

有一封电子邮件

⑤ 以常驻服务的类型执行fetchmail，并设定每隔300秒检查邮件服务器上是否有新的邮件，并将相关信息记录在fetchmail.log文件中：

```
# fetchmail -d 300 -L fetchmail.log
```

间隔时间

指定记录文件

⑥ 关闭执行中的fetchmail常驻服务:

```
# fetchmail -q
fetchmail: background fetchmail at 2200 killed.
```

已经成功地关闭 fetchmail 常驻服务

⑦ 接收电子邮件时保留信件在服务器上:

```
# fetchmail -k
1 message for testlinux at msg.pchome.com.tw (740 octets).
reading message testlinux@msg.pchome.com.tw 1 of 1
(740 octets) not flushed
```

接收了一封信, 但保留该信件在服务器上

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

F

⑧ 接收电子邮件时, 如果有超过500k的电子邮件, 则不下载该邮件:

```
# fetchmail -l 500000
      ↑
    500 000字节
```

⑨ 接收新的电子邮件之前, 先删除旧的信件:

```
# fetchmail -F
```

⑩ 显示fetchmail当前的设定值:

```
# fetchmail -v
```



⑩ 接收电子邮件时，若邮件服务器超过180秒仍没响应，则断开连接：

```
# fetchmail -t 180
```

```
fetchmail: timeout after 180 seconds waiting to connect  
to      server msg.pchome.com.tw.  
fetchmail: client/server synchronization error while  
fetching  
from msg.pchome.com.tw  
fetchmail: Query status=7
```

在此例，该主机超过180秒仍没有响应

⑪ 执行fetchmail时，读取指定的设定文件，而不使用默认的文件：

```
# fetchmail -f myfetchmailrc
```

F

fg (foreground)

常用指数：



功能说明：将程序或命令切换至前台执行。

语 法：fg [命令名称]

包 名 称：bash。

相关命令：bg, jobs, ps。

补充说明：使用bg命令可把执行的程序或命令放到后台去工作。而利用fg命令则能将这些在后台执行的程序或命令调回到前台来操作。可先用ps命令查看程序或命令的命令名称，再用fg命令去调用它们。

参 数：无。



范 例:

把setiathome的计算程序放置到后台去执行,然后再切换回前台来操作:

```
# ./setiathome -nice 19 > /dev/null & ← 把程序放到后台去执行
[1]5122                               ← 该程序执行的程序识别码 (Process ID)
# fg ./setiathome                       ← 把程序调回前台来操作
```

fgrep (fixed regexp)

常用指数: 

功能说明: 查找文件里符合条件的字符串。



语 法: fgrep [范本样式] [文件或目录...]

包 名 称: grep。

相关命令: egrep, grep, rgrep。

补充说明: 本命令相当于执行grep命令加上参数“-F”,详见grep命令的说明。

file

常用指数:  

功能说明: 识别文件类型。

语 法: file [-bcLsvz] [-f <名称文件>] [-m <魔法数字文件>,...]
[文件或目录...]

包 名 称: file。

相关命令: od。



补充说明：通过file命令，我们可以识别该文件的类型。file命令会尝试去辨别所指定的文件类型是什么。如果为文本文件，它会告诉用户是哪一类文本文件，如ASCII、English、International等；如果为执行文件，它会告诉用户是哪一种执行文件，如ELF、Bourne-Again Shell Script、Bourne Shell Script等；若为符号链接，它会显示该连接所指向的文件。其他诸如目录、特殊文件（Block、Character、Socket、Pipe...）、图形文件（JPEG、GIF、BMP、PCX...）、备份文件或压缩文件（TAR、GZip、Zip...）等等，只要该文件类型在魔法数字（Magic Number）文件内有记录数据，file命令就会返回其类型的相关信息。

参 数：

-b	列出识别结果时，不显示文件名称
-c	详细显示命令执行过程，以便于debug或分析程序执行的情形
-f <名称文件>	指定名称文件，其内含有一个或多个文件名称，让file依次识别这些文件，名称文件的格式为每列一个文件名称
-L	直接显示符号链接所指向的文件类别。例如文件gamma为ASCII文本文件sigma的符号链接，识别gamma文件时加上这项参数，就会显示文件类型为ASCII文本文件，而非符号链接
-m <魔法数字文件>	指定魔法数字文件。默认是名称为magic的文件，存放在/usr/share目录下，可运用此参数另行设定。同时指定多个魔法数字文件时，文件之间以“，”号分开，例如“-mlanda, nova”就是设定文件landa和文件nova为魔法数字文件
-s	当所要识别的是较特殊的装置，例如分区时，可使用此参数
-v	显示版本信息
-z	尝试去解读压缩文件的内容

范 例:

① 识别textfile文件的类型:

file textfile
textfile: ASCII text ← 是一个ASCII的文本文件

② 识别textfile文件的类型, 列出结果时并不显示文件名:

file -b textfile
ASCII text ← 与上例比较, 可看出此例并没有列出文件名称

③ 识别filelist内指定文件的文件类型:

cat filelist
test1
test2
test3
filelist 指定了此3个文件

file -f filelist
test1: ASCII text ← 识别filelist指定的文件
test2: directory ← test1是个ASCII文本文件
test3: news text ← test2是个目录
← test3是个新闻组的文本文件

④ testsymbol是个符号链接, 识别其来源文件的类型:

file testsymbol
testsymbol: symbolic link to testfile ← testsymbol是一个符号链接

file -L testsymbol
testsymbol: HTML document text ← 此符号链接的来源文件是HTML文本文件



⑤ textfile.gz是个压缩文件，利用file命令尝试识别源文件的类型：

```
# file textfile.gz
```

```
textfile.gz: gzip compressed data, was "textfile", from Unix
```

↑
此文件为gzip所压缩的压缩文件

```
# file -z textfile.gz
```

```
textfile.gz: ASCII text (gzip compressed data, was "textfile",  
from Unix)
```

原始文件是ASCII文本文件

⑥ 指定利用mymagic魔法数字文件来识别SETI目录下的文件类型：

```
# file -m mymagic SETI/*
```

```
file: Using regular magic file 'mymagic'
```

SETI/README:	English text
SETI/lock.txt:	empty
SETI/outfile.txt:	ASCII text
SETI/result_header.txt:	ASCII text
SETI/setiathome:	ELF 32-bit LSB executable, Intel 80386
SETI/state.txt:	ASCII text
SETI/temp.txt:	ASCII text
SETI/user_info.txt:	ASCII text
SETI/version.txt:	ASCII text
SETI/work_unit.txt:	ASCII text

识别结果

⑦ 识别硬盘分区文件系统的类型：

```
# file -s /dev/hda{,1,5,6}
```

↑
识别/dev/hda、/dev/hda1、/dev/hda5及/dev/hda6分区

```
/dev/hda: x86 boot sector
/dev/hda1: Linux rev 1.0 ext3 filesystem data
/dev/hda5: Linux rev 1.0 ext3 filesystem data (needs
journal recovery)
/dev/hda6: Linux rev 1.0 ext3 filesystem data (needs
journal recovery)
```

find

常用指数: 

功能说明: 查找文件或目录。

语 法: find [目录...] [-amin <分钟>] [-anewer <参考文件或目录>] [-atime <24 小时数>] [-cmin <分钟>] [-cnewer <参考文件或目录>] [-ctime <24 小时数>] [-daystart] [-depth] [-empty] [-exec <执行命令>] [-false] [-fls <列表文件>] [-follow] [-fprint <列表文件>] [-fprint0 <列表文件>] [-fprintf <列表文件> <输出格式>] [-fstype <文件系统类型>] [-gid <组织识别码>] [-group <组名称>] [-help] [-ilname <范本样式>] [-iname <范本样式>] [-inum <inode 编号>] [-ipath <范本样式>] [-iregex <范本样式>] [-links <连接数目>] [-lname <范本样式>] [-ls] [-maxdepth <目录层级>] [-mindepth <目录层级>] [-mmin <分钟>] [-mount] [-mtime <24 小时数>] [-name <范本样式>] [-newer <参考文件或目录>] [-nogroup] [-noleaf] [-nouser] [-ok <执行命令>] [-path <范本样式>] [-perm <权限数值>] [-print] [-print0] [-printf <输出格式>] [-prune] [-regex <范本样式>] [-size <文件大小>] [-true] [-type <文件类型>] [-uid <用户识别码>] [-used <日数>] [-user <拥有者名称>] [-version] [-xdev] [-xtype <文件类型>]

包 名 称: findutils。

相关命令: locate, slocate, whereis, which。

补充说明: find命令用于查找符合条件的文件，任何在参数之前的字符串都将视为欲查找的目录。假如没有指定目录，则会查找当前的目录，假如没有设定参数，则会以“-print”参数作为默认值。

当指定参数时，可在参数之前加上“1”，表示查找不符合此参数的文件或目录。也可将两个参数用“-o”连接，表示只要符合其中一个参数的条件即可。

参 数：

-amin <分钟>	查找在指定时间曾存取过的文件或目录，单位以分钟计算。例如设定成10，find命令会寻找刚好在10分钟之前存取过的文件或目录；设定为+10，表示存过10分钟以前所有存取过的文件或目录；设定为-10，则表示10分钟之内所有存取过的文件或目录
-anewer <参考文件或目录>	查找其存取时间较指定文件或目录的存取时间，更接近现在的文件或目录。如搭配参数“-follow”使用，则必须放在“-follow”参数的后面才有效果
-atime <24 小时数>	查找在指定时间曾存取过的文件或目录，单位以24小时计算。例如设定成2，find命令会寻找刚好在48 小时之前存取过的文件或目录；设定为+2，表示超过48小时以前所有存取过的文件或目录；设定为-2，则表示48小时之内所有存取过的文件或目录
-cmin <分钟>	查找在指定时间更改的文件或目录，单位以分钟计算。例如设定成10，find命令会寻找刚好在10分钟之前更改的文件或目录；设定为+10，表示超过10分钟以前所有更改的文件或目录；设定为-10，则表示10分钟之内所有更改的文件或目录
-cnewer <参考文件或目录>	查找其更改时间比指定文件或目录的更改时间更接近现在的文件或目录。如搭配参数“-follow”使用，它必须放在“-follow”参数的后面才有效果

-ctime <24 小时数>	查找在指定时间更改的文件或目录，单位以24小时计算。例如设定成2，find命令会寻找刚好在48小时之前更改的文件或目录；设定为+2，表示超过48小时以前所有更改的文件或目录；设定为-2，则表示48小时之内所有更改的文件或目录
-daystart	从本日开始计算时间，而非从24小时之前开始计算。此参数通常用来配合“-amin”、“-atime”、“-cmin”、“-time”、“-mmin”、“-mtime”等参数使用
-depth	从指定目录下最深层的子目录开始查找
-empty	寻找文件大小为0字节的文件，或目录下没有任何子目录或文件的空目录
-exec <执行命令>	假设find命令的返回值为True，执行该命令，命令的结尾必须是“\;”号。可利用{}表示当前处理的文件
-false	将find命令的返回值都设为False
-fls <列表文件>	此参数的效果和指定“-ls”参数类似，但会把结果存储成指定的列表文件
-follow	排除符号链接。本参数同时具有“-noleaf”参数的效果
-fprint <列表文件>	此参数的效果和指定“-print”参数类似，但会把结果存储成指定的列表文件
-fprint0 <列表文件>	此参数的效果和指定“-print0”参数类似，但会把结果存储成指定的列表文件
-fprintf <列表文件> <输出格式>	此参数的效果和指定“-printf”参数类似，但会把结果存储成指定的列表文件

fstype <文件系统类型>	只寻找该文件系统类型下的文件或目录。文件系统类型可以是：“adfs” (Advanced Disk File System)、“affs” (Amiga Fast File System)、“hfs” (Macintosh Filesystem)、“fat” (DOS File Allocation Table)、“msdos”、“umsdos” (UNIX-Like MS-DOS)、“vfat” (Virtual FAT)、“isofs” (ISO 9660)、“minix”、“ntfs” (NT File System)、“hpfs” (High Performance File System)、“qnx4” (QNX Filesystem)、“ext2” (Second Extended Filesystem)、“sysv” (System V Filesystem)、“ufs” (UNIX File System) 以及 “efs” (IRIX File System)
-gid <组识别码>	查找符合指定的组识别码的文件或目录。例如设定成500，find命令会寻找组识别码为500的文件或目录；设定为+500，表示寻找所有识别码大于500的文件或目录；设定为-500，则表示寻找所有识别码小于500的文件或目录
-group <组名称>	查找符合指定组名称的文件或目录，也能以组识别码指定
-help或--help	显示帮助
-iname <范本样式>	此参数的效果和指定“-lname”参数类似，但忽略字符大小写差别
-iname <范本样式>	此参数的效果和指定“-name”参数类似，但忽略字符大小写差别
-inum <inode 编号>	查找符合指定的inode编号的文件或目录。例如设定成6150，find命令会寻找inode编号为6150的文件或目录；设定为+6150，表示寻找所有编号大于6150的文件或目录；设定为-6150，则表示寻找所有编号小于6150的文件或目录
-ipath <范本样式>	此参数的效果和指定“-ipath”参数类似，但忽略字符大小写的差别
-iregex <范本样式>	此参数的效果和指定“-regex”参数类似，但忽略字符大小写的差别

<code>-links <连接数目></code>	查找符合指定的硬链接数目的文件或目录。例如设定成10, find命令会寻找硬链接数目为10的文件或目录; 设定为+10, 表示寻找所有硬链接数目大于10的文件或目录; 设定为-10, 则表示寻找所有硬链接数目小于10的文件或目录
<code>-lname <范本样式></code>	指定字符串作为寻找符号链接的范本样式。所谓范本样式, 就是作为查找符号链接的名称依据。可以使用特殊字符来指定范本样式, 例如指定的范本为“eos*dawn”, find命令会找出所有起始字符串为“eos”, 且结尾字符串是“dawn”的符号链接。注意特殊字符并不特别看待符号“/”与“.”
<code>-ls</code>	假设find命令的返回值为True, 就将文件或目录名称列出到标准输出, 格式类似ls命令加上“-ils”参数, 但每个名称之前都有“./”字符串。默认的单位是块, 这里每个块是1 KB而非512字节, 除非设定环境变量POSIXLY_CORRECT
<code>-maxdepth <目录层级></code>	设定最大目录层级。用此参数可决定执行find命令的范围, 例如目录层级设为5, 就会从指定目录算起, 再往下寻找4层, 目录层级设成1, 则不会寻找任何子目录的内容。 <code>-mindepth <目录层级></code> 设定最小目录层级。用此参数可决定执行find命令的范围, 例如目录层级设为2, 就会从指定目录下1层的子目录开始寻找, 但不会寻找指定的目录, 依此类推
<code>-mmin <分钟></code>	查找在指定时间曾变动过的文件或目录, 单位以分钟计算。例如设定成10, find命令会寻找刚好在10分钟之前变动过的文件或目录; 设定为+10, 表示超过10分钟以前所有变动过的文件或目录; 设定为-10, 则表示10分钟之内所有变动过的文件或目录
<code>-mount</code>	此参数的效果和指定“-xdev”参数相同

mtime <24 小时数>	查找在指定时间曾变动过的文件或目录，单位以24小时计算。例如设定成2，find命令会寻找刚好在48小时之前变动过的文件或目录；设定为+2，表示超过48小时以前所有变动过的文件或目录；设定为-2，则表示48小时之内所有变动过的文件或目录
-name <范本样式>	指定字符串作为寻找文件或目录的范本样式。所谓范本样式，就是作为查找文件或目录的名称依据。可以使用特殊字符（如“*”或“?”等等）来指定范本样式，例如指定的范本为“eos*dawn”，find命令会找出所有起始字符串为“eos”，且结尾字符串是“dawn”的文件或目录，如名称为eosbringdawn的文件或目录就是。注意特殊字符并不处理文件或目录名称前面的“.”号，举例来说，名称为.eosbringdawn的文件或目录就无法用“*eos*dawn”字符串来找到
-newer<参考文件或目录>	查找其变动时间比指定文件或目录的变动时间更接近现在的文件或目录。如搭配参数“-follow”使用，它必须放在“-follow”参数的后面才有效果
-nogroup	找出不属于本地主机组织别码的文件或目录
-noleaf	不为“目录至少拥有两个硬链接（Hard Link）”的情况做最优化。一般在UNIX的文件系统里，每个目录至少都会拥有两个硬链接，其中一个为目录名称，另一个则是当前目录“.”。但其他文件系统则不一定具备这项特性，比如CD-ROM或DOS的文件系统，以及AFS（Andrew Filesystem）等。find命令默认会针对这种特性去把查找方式最优化，在没有类似连接性质的文件系统中取消最优化的步骤，可有效减少耗费的时间
-nouser	找出不属于本地主机用户识别码的文件或目录

-ok <执行命令>	此参数的效果和指定“-exec”参数类似，但在执行命令之前会先询问用户，若回答非“y”或“Y”，则放弃执行命令
-path <范本样式>	指定字符串作为寻找目录的范本样式。所谓范本样式，就是作为查找目录的名称依据。可以使用特殊字符来指定范本样式，例如指定的范本为“eos*dawn”，find命令会找出所有起始字符串为“eos”，且结尾字符串是“dawn”的目录。注意特殊字符并不特别看待符号“/”与“.”，举例而言，可以用“eos*dawn”字符串来找到名称为eos/bring.dawn或eos.bring/dawn等目录
-perm <权限数值>	查找符合指定的权限数值的文件或目录。例如设定成0700，find命令会寻找权限数值正好是0700的文件或目录；设定为+0700，表示寻找所有权限部分大于0700的文件或目录；设定为-0700，则表示寻找所有权限部分少于符合0700的文件或目录
-print	假设find命令的返回值为True，就将文件或目录名称列出到标准输出，格式为每行一个名称，每个名称之前皆有“/”字符串
-print0	假设find命令的返回值为True，就将文件或目录名称列出到标准输出，格式为全部的名称皆在同一行
-printf <输出格式>	假设find命令的返回值为True，就将文件或目录名称列出到标准输出，格式可以自行指定
-prune	不寻找当前的目录。如同时搭配“-depth”参数使用，“-prune”参数将失去其效果
-regex <范本样式>	指定字符串作为寻找文件或目录的范本样式。所谓范本样式，就是作为查找文件或目录的名称依据。可以使用特殊字符来指定范本样式，例如指定的范本为“eos*dawn”，find命令会找出所有起始字符串为“eos”，且结尾字符串是“dawn”的文件或目录

-size <文件大小>	查找符合指定的文件大小的文件。例如设定成100, find命令会寻找占用块数目为100的文件; 设定为+100, 表示寻找所有占用块数目大于100的文件; 设定为-10, 则表示寻找所有占用块数目小于100的文件。可以自行指定文件大小的计算单位, 在文件大小字尾加上“b”, 表示用块为单位, 这是默认值, 每个块 512 字节。加上“c”表示用Byte, 加上“k”表示用KB, 加上“w”表示用个字节为单位。例如文件大小为“1048000c”表示1 048 000字节, 依此类推
-true	将find命令的返回值皆设为True
-type <文件类型>	只寻找符合指定文件类型的文件。文件类型共有7种, 分别是: “b” (Block Special)、 “c” (Character Special)、 “d” (Directory)、 “p” (Named Pipe)、 “f” (Regular File)、 “l” (Symbolic Link) 和 “s” (Socket)。请注意, 目录在Linuxext2文件系统上可视为一种特别文件
-uid <用户识别码>	查找符合指定用户识别码的文件或目录。例如设定成500, find命令会寻找用户识别码为500的文件或目录; 设定为+500, 表示寻找所有识别码大于500的文件或目录; 设定为-500, 则表示寻找所有识别码小于500的文件或目录
-used <日数>	查找文件或目录变动之后, 在指定时间曾存取过的文件或目录, 单位以日计算。例如设定成3, find命令会寻找刚好在变动后3日存取过的文件或目录; 设定为+3, 表示变动后超过3日才存取过的所有文件或目录; 设定为-3, 则表示变动后3日内曾存取过的文件或目录
-user <拥有者名称>	查找符合指定拥有者名称的文件或目录, 也能以用户识别码指定
-version或--version	显示版本信息

-xdev	将范围局限在当前的文件系统中，若指定目录下的某些子目录，其存放于另一个文件系统中，则将该子目录予以排除在寻找范围外
-xtype <文件类型>	此参数的效果和指定“-type”参数类似，差别在于它针对符号链接检查。如果该符号链接所指向的原始文件符合所设定的文件类型，且没有搭配“-follow”参数使用，则find命令的返回值将为True。假如配合参数“-follow”使用，必须指定文件类型为“l”，其返回值才是True

范 例：

① 列出当前目录下所有子目录及文件的名称：

```
# find
```

② 寻找当前目录下，文件名称以 april 起始的文件：

```
# find -name april\*
```

③ 寻找当前目录下文件名称以 april 起始的文件，并将结果输出到 filelist 文件中：

```
# find -name april\* -fprint filelist
```

↑
指定输出到filelist文件

④ 寻找当前目录下文件名称以april或may起始的文件：

```
# find -name april\* -o -name may\*
```

↑
只要符合前后两个参数之中的一个条件即可

⑤ 寻找/mnt目录下文件系统类型为vfat，且名称为searchfile.txt的文件：

```
# find /mnt -name searchfile.txt -fstype vfat
```

⑥ 寻找/mnt目录下名称为searchfile.txt文件，但其所在的文件系统不能是vfat：

```
# find /mnt -name searchfile.txt ! -fstype vfat
```

↑
不能符合后面参数所指定的条件

⑦ 寻找/tmp目录下文件类型为符号链接，且名称为wra起始的文件：

```
# find /tmp -name wra\* -type l
```

⑧ 列出/home目录下最近两天之内有变动的文件：

```
# find /home -mtime -2
```

↑
“-”表示指定的时间之后

⑨ 列出/home目录下最近60分钟之前有变动的文件：

```
# find /home -mmin +60
```

↑
“+”表示指定的时间之前

⑩ 列出/home目录下变动时间比modifyfile文件更近的文件或目录：

```
# find /home -newer modifyfile
```



⑪ 列出/home目录下最近一天之内有存取过的文件:

```
# find /home -atime -1
```

⑫ 列出/home目录下最近30分钟之前有存取过的文件:

```
# find /home -amin +30
```

⑬ 列出/home目录下存取时间比accessfile文件更近的文件或目录:

```
# find /home -anewer accessfile
```

⑭ 列出/home目录下文件或目录变动之后, 在两日内曾存取过的文件或目录:

```
# find /home -used -2
```

⑮ 列出/tmp目录下属于用户alkane的文件或目录:

```
# find /tmp -user alkane
```

⑯ 列出/tmp目录下文件或目录所属的用户, 其用户识别码大于501的文件或目录:

```
# find /tmp -uid +501
```

⑰ 列出/tmp目录下属于sphenoid组的文件或目录:

```
# find /tmp -group sphenoid
```



⑧ 列出/tmp目录下属于501组识别码的文件或目录:

```
# find /tmp -gid 501
```

⑨ 列出/tmp目录下不属于本地用户的文件或目录:

```
# find /tmp -nouser
```

⑩ 列出/tmp目录下不属于本地组的文件或目录:

```
# find /tmp -nogroup
```

⑪ 寻找/tmp目录下文件名称为alpha.txt的文件, 寻找时最多只往下找3层子目录:

```
# find /tmp -name alpha.txt -maxdepth 4
```

F

⑫ 寻找/tmp目录下文件名称为beta.txt的文件, 寻找时从/tmp目录下2层子目录开始找起:

```
# find /tmp -name beta.txt -mindepth 3
```

⑬ 列出/tmp目录下文件大小为0, 或目录下没有任何子目录或文件的空目录:

```
# find /tmp -empty
```

⑭ 列出/mnt目录下大于5 120 KB的文件:

```
# find /mnt -size +5120k
```



⑤ 列出/tmp目录下硬链接数目大于2的文件或目录:

```
# find /tmp -links +2
```

⑥ 列出/tmp目录下权限刚好为0700的文件或目录:

```
# find /tmp -perm 0700
```

⑦ 寻找/tmp目录下名称为text的文件, 如果有该文件, 则接着列出这个文件的内容:

```
# find /tmp -name text -exec cat {} \;
```

所执行的命令, “{ }”表示此文件

⑧ 寻找/tmp目录下名称为text的文件, 如果有该文件, 则询问用户是否要删除此文件:

```
# find /tmp -name text -ok rm {} \;
```

执行此命令前, 先询问用户意见

```
< rm ... /tmp/text > ? y
```

按“y”确定, 若按“n”则放弃执行此命令

finger

常用指数:



功能说明: 查找并显示用户信息。

语法: finger [-lmps] [账号名称...]

包名称: finger。

相关命令: chfn, w, who。

补充说明：finger命令会去寻找并显示指定账号的用户相关信息，包括本地与远程主机的用户皆可，账号名称没有大小写的差别。单独执行finger命令，它会显示本地主机现在所有用户的登录信息，包括账号名称、真实姓名、登录终端、空闲时间、登录时间，以及地址和电话。

参 数：

-l	列出该用户的账号名称、真实姓名、用户根目录、登录所用的shell、登录时间、邮件地址、电子邮件状态，还有项目文件(.project)和计划文件(.plan)内容。这项参数为默认值，若仅给予finger命令账号名称，有无加此参数效果相同
-m	排除查找用户的真实姓名。用户通常具备两个名称，一个是登录系统的用户账号，另一个则是真实姓名。假如没有加上这个参数，finger命令会去查找所有符合指定账号名称的用户账号与真实姓名，搭配此项参数之后，真实姓名将不在查找范围内
-p	列出该用户的账号名称、真实姓名、用户根目录、登录所用的shell、登录时间、邮件地址、电子邮件状态，但不显示该用户的项目和计划文件内容
-s	列出该用户的账号名称、真实姓名、登录终端、空闲时间、登录时间，以及地址和电话

范 例：

① 以默认值的方式查询本地主机上名称为alkane的用户信息：

```
# finger alkane
```

```
Login: alkane
```

↑
登录账号名称

```
Name: shinewin
```

↑
真实姓名

```

Directory: /home/alkane
           ↑
         用户根目录
Office: beijing, 010-909-909
        ↑      ↑
    办公室地址  办公室电话
Shell: /bin/bash
        ↑
    登录所用的shell
Home Phone: 123-456-789
           ↑
        家庭电话
Last login Mon Jun 2 23:58 (CST) on pts/1 from 192.168.0.35
           ↑
        上次登录的信息
No mail. ← 电子邮件状态
No Plan. ← 计划文件内容 (~/.plan)

```

如果该用户当前已登录到本地主机上，则显示以下信息：

```

# finger alkane
Login: alkane           Name: shinewin
Directory: /home/alkane Shell: /bin/bash
Office: taipei, 909-909-909 Home Phone: 123-456-789
On since Tue Jun 10 17:30 (CST) on pts/3 from 192.168.0.35
  1 minute 49 seconds idle ← 用户当前空闲的时间
No mail.
No Plan.

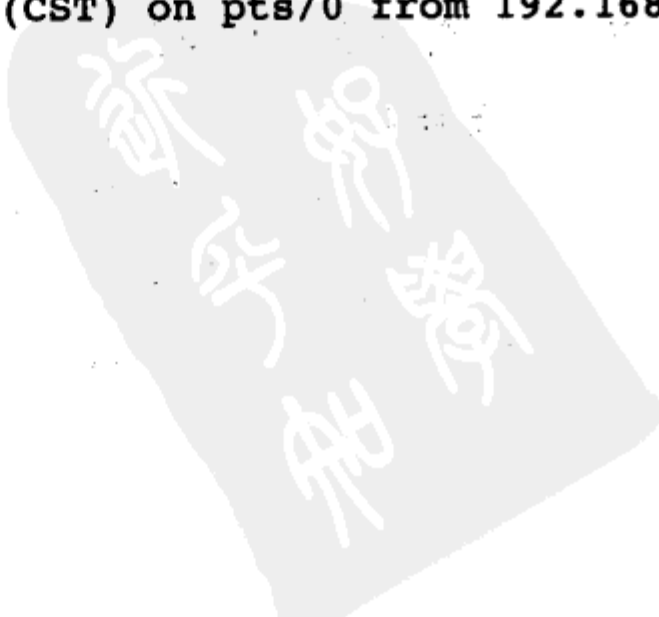
```

② 查询本地主机上名称为alkane的用户信息，但不显示该用户项目及计划文件内容：

```

# finger -p alkane
Login: alkane           Name: shinewin
Directory: /home/alkane Shell: /bin/bash
Office: taipei, 909-909-909 Home Phone: 123-456-789
Last login Tue Jun 10 21:06 (CST) on pts/0 from 192.168.0.35
No mail.

```



- ③ 查询本地主机上名称为alkane的用户信息，显示时以较简要的格式列出：

```
# finger -s alkane
```

```
Login Name      Tty Idle Login Time Office Office Phone
alkane shinewin  1   * Jun  10   12:12 (203.74.205.39)
```

- ④ 查询本地主机上账号名称为shinewin，但不包括真实姓名在内：

```
# finger -m shinewin
```

```
finger: shinewin: no such user.
```

← 虽然alkane的真实姓名为shinewin，但因为加上了-m参数，所以并不会显示出alkane的信息

F

- ⑤ 查询远程主机flash.com.tw上，账号名称为dark的用户信息：

```
# finger dark@flash.com.tw
```



远程主机的名称之前要加上“@”号

fmt (format)

常用指数：



功能说明：编排文本文件。

语 法：fmt [-cstu] [-p <行起始字符串>] [-w <每行字符数>] [--help] [--version] [文件...]

包 名 称：coreutils。

相关命令：fold。

补充说明：fmt命令会从指定的文件里读取内容，将其依照指定格式重新编排后，输出至标准输出设备。除非指定参数“-u”，否则文件内容中的空白列都会保留下来。若不指定任何文件名称，或是所给予的文件名为“-”，则fmt命令会从标准输入设备读取数据。

参 数:

-c或--crown-margin	每段前两列缩排
-p <行起始字符串>或 --prefix=<行起始字符串>	仅合并含有指定字符串的行，通常运用在程序语言的注解
-s 或 --split-only	只拆开字数超出每行字符数的行，但不合并字数不足每行字符数的行
-t或--tagged-paragraph	每段前两列缩排，但第1列和第2列的缩排格式不同
-u或--uniform-spacing	每个英文字之间都以一个空格字符间隔，每个句子之间则用两个空格字符分隔
-w <每行字符数>、 --width=<每行字符数> 或-<每行字符数>	设定每行的最大字符数。默认值为每行75个字符，可用这项参数去改变它。fmt命令会预留约7%的字符空间，以便尽量调整每行的宽度达到一致。举例来说，假如每行字符数设为100，则fmt命令会用到约93个字符，剩下7个字符的空间则用来平衡每行的宽度
--help	显示帮助
--version	显示版本信息

范 例:

① 将textfile文件重新编排成每行78列，并将结果输出到屏幕上：

```
# fmt -w 78 textfile
```

② 将textfile以默认的每行75列的方式重新编排，但编排时只拆开超过75列的行，并不合并字数不足的行，最后则将结果存成outfile：


```
# fmt -s textfile > outfile
```

↑ 用特殊字符“>”导向新文件

③ 将textfile以默认每行75列的方式重新编排，但每个英文单词以一个空格字符分开，而每个句子则以两个空格字符分隔，最后将结果存成outfile：

```
# fmt -u textfile > outfile
```

fold

常用指数: 

功能说明: 限制文件行宽。

语法: fold [-bs] [-w <每行列数>] [--help] [--version] [文件...]

包名称: coreutils。

相关命令: fmt。

补充说明: fold命令会从指定的文件里读取内容，将超过限定行宽的行加入增行字符后，输出至标准输出设备。由于fold命令所根据的是画面显示结果，因此跳格字符将视为多个字符、倒退字符会让行数退回一行，而CR字符则会重置列数编号使编号变成0，也就是回到该行的起始点。若不指定任何文件名称，或是所给予的文件名为“-”，则fold命令会从标准输入设备读取数据。

参数:

-b或--bytes	以Byte为单位计算行宽，而非采用列数编号为单位。使用这项参数之后，无论是跳格字符、倒退字符或是CR字符，都视为1个字节
-s或--spaces	以空格字符作为换行点。此参数可避免文字遭到截断，导致纯文本文件的文章无法阅读的状况发生。fold命令将在最大行宽的限制之内，于最后一个空格字符后加上增行字符，假使该列没有空格字符存在，则fold命令仍旧会在达到每行列数限制时，强制截断该行，并把超出部分新增成另一行
-w <每行列数>或 --width <每行列数>	设定每行的最大列数。默认值为每行80列，可利用此参数改变它
--help	显示帮助
--version	显示版本信息

范 例:

① textfile文件重新编排成每行78列，并将结果输出到屏幕上：

```
# fold -w 78 textfile
```

② textfile文件重新编排成每行78列，转换时以空格字符当做换行点，以避免切断文字，最后则将结果存成outfile文件：

```
# fold -s -w 78 textfile > outfile
```

③ 以Byte为单位，将文件textfile重新编排成每行100列，并将结果输出到屏幕上：

```
# fold -b -w 100 textfile
```

free

常用指数：



F

功能说明：查看内存状态。

语 法：free [-bkmotV] [-s <间隔秒数>]

包 名 称：procps。

相关命令：ps, procinfo, tload, top, w。

补充说明：free命令会显示内存的使用情形，包括物理（Physical）内存、虚拟的交换文件内存、共享内存区段，以及系统核心使用的缓冲区等。



参 数:

-b	以Byte为单位显示内存使用情形
-k	以KB为单位显示内存使用情形。这项参数为默认值，其效果相当于仅执行free命令而不加任何参数
-m	以MB为单位显示内存使用情形
-o	不显示缓冲区调节行
-s <间隔秒数>	持续观察内存使用状况。间隔秒数设成3，则每隔3秒就会查看内存状态一次；设为5，则每5秒查看一次，依此类推
-t	显示内存总和行
-v	显示版本信息

范 例:

① 查询当前内存的使用情况:

free

```

                total    used    free    shared    buffers    cached
Mem:           127872  124840   3032   66564      2636     37188
-/+ buffers/cache:    85016  42856
Swap:          104380    13572  90808
    
```

② 查询当前内存的使用情况，但不显示缓冲区调节行:

free -o

```

                total    used    free    shared    buffers    cached
Mem:           127872  123492   4380   66608      2704     36976
Swap:          104380    13560  90820
    
```



③ 查询当前内存的使用情况，显示时并列所有内存总和行：

```
# free -t
      total      used      free  shared  buffers  cached
Mem:   127872     96380     31492    56900     3816    44516
-/+buffers/cache: 48048    79824
Swap:   104380     13036     91344
Total:  232252    109416    122836
```

← 多增加了这一行

④ 以Byte为单位来显示当前内存的使用情况：

```
# free -b
      total      used      free  shared  buffers  cached
Mem: 130940928 127836160 3104768 68165632 2699264 38080512
-/+ buffers/cache: 87056384 43884544
Swap:  106885120 13897728 92987392
```

F

全部结果均以Byte为单位显示

⑤ 以MB为单位来显示现在内存的使用情况：

```
# free -m
      total  used  free  shared  buffers  ached
Mem:      124   121    2      65        2    36
-/+ buffers/cache: 83    41
Swap:     101   13    88
```

全部结果均以MB为单位显示

⑥ 每10秒钟检查一次内存的使用情形：

```
# free -s 10
```



fsck (file system check)

常用指数: **功能说明:** 检查文件系统并尝试修复错误。**语法:** fsck [-aANPrRsTV] [-t <文件系统类型>] [文件系统...]**包名称:** e2fsprogs。**相关命令:** e2fsck, fsck.ext2, fsck.minix, mkfs。**补充说明:** 当文件系统发生错误时, 可用fsck命令尝试加以修复。直接采用分区编号 (如/dev/hda5), 或使用加载点 (Mount Point, 如/、/usr等) 指定文件系统皆可。假设一次指定多个文件系统, 而这些系统分别位于不同的物理磁盘上, 则fsck将会尝试同步的方式去检查它们, 以节省操作时间。**参 数:**

-a	自动修复文件系统, 不询问任何问题
-A	根据/etc/fstab 设定文件的内容, 检查该文件内所列的全部文件系统。若没有附加参数“-P”, 则会先检查根目录/的文件系统, 而不会以同步的方式检查所有文件系统
-N	不执行命令, 仅列出实际执行会进行的操作
-P	当搭配“-A”参数使用时, 则会以同步的方式检查所有的文件系统
-r	采用互动模式, 在执行修复时询问问题, 让用户得以确认并决定处理方式
-R	当搭配“-A”参数使用时, 则会略过根目录/的文件系统不予检查
-s	依次执行检查操作, 而非同时执行。当一次指定多个文件系统, 且采用互动模式进行检查时, 请使用此参数依次执行, 否则fsck可能同时询问数个问题, 让用户不知所措

<code>-t <文件系统类型></code>	指定要检查的文件系统类型。当配合参数“-A”使用时，只有符合指定类型的文件系统会被检查，不在指定类型内的文件系统则会略过。若在文件系统类型前面加上“no”字符串，则仅有不符合指定类型的文件系统会被检查。例如类型指定为“ext2”，只会检查所有ext2的文件系统；指定成“noext2”，则检查所有非ext2的文件系统
<code>-T</code>	执行fsck命令时，不显示标题信息
<code>-V</code>	显示命令执行过程

范 例：

- ① 检查/dev/hdd2文件的正确性：

```
# fsck /dev/hdd2
fsck 1.32 (09-Nov-2002)
e2fsck 1.32 (09-Nov-2002)
/dev/hdd2: clean, 15/276352 files, 16893/552234 blocks
                  此文件系统没有任何问题
```

- ② 检查所有文件的正确性，若发现任何问题，不询问用户意见，便自动修复文件系统：

```
# fsck -A -a
              ↑      ↑
              |      |
              |      |————— 采用非互动模式
              |      |————— 检查所有的文件系统
```

- ③ 除了根目录的文件系统之外，检查所有文件的正确性：

```
# fsck -AR
```

- ④ 采用同步的方式检查所有文件的正确性：

```
# fsck -AP
```

⑤ 采用循环的方式检查所有文件系统的正确性:

```
# fsck -As
```

⑥ 检查所有ext3文件系统的正确性:

```
# fsck -A -t ext3
```

⑦ 检查除了ext3文件系统之外, 所有文件系统的正确性:

```
# fsck -A -t noext3
```

⑧ 采用互动模式, 循环检查所有文件系统的正确性:

```
# fsck -Asr
```

F fsck.ext3

常用指数:



(file system check - second extended filesystem)

功能说明: 检查文件系统并尝试修复错误。

语法: `fsck.ext3 [-acdfFnprsStvVy] [-b <分区第1个扇区位置>] [-B <块大小>] [-C <反叙述器>] [-I <inode 缓冲块数>] [-l <损毁块文件>] [-L <损毁块文件>] [-P <处理inode大小>] [外围设备代号]`

包名称: e2fsprogs。

相关命令: e2fsck, fsck, mkfs。



补充说明：当ext3文件系统发生错误时，可用fsck.ext3命令尝试加以修复。[外围设备代号]对硬盘之类的存储媒体而言，就是分区编号（如/dev/hda1），对软驱来说，则是外围设备代号（如/dev/fd0）。

参 数：

-a	自动修复文件系统，不询问任何问题
-b <分区第1个扇区位置>	指定分区的第1个扇区的起始位置，也就是Super Block。当欲修复的文件系统所在分区，其第1个扇区已经毁损时，可借由此参数强制指定其位置所在，例如“8193”、“16385”等等
-B <块大小>	设定该分区每个块的大小。一般而言，fsck.ext3命令会自动确认块大小，可使用这项参数，强制指定块大小
-c	检查指定的文件系统内，是否存在毁损的块。如发现毁损块，便会将其标示出来，避免应用程序使用该块
-C <反叙述器>	指定反叙述器（Descriptor），fsck.ext3命令会把全部的执行过程都交由其逆向描述，便于debug或监控程序执行的情形
-d	详细显示命令执行过程，以便于debug或分析程序执行的情形
-f	强制对该文件系统进行全面检查，即使该文件系统在概略检查下没有问题
-F	检查文件系统之前，先清理该存储设备缓冲区的数据
-l <inode缓冲块数>	设定欲检查的文件系统，其inode缓冲区的块数目
-l <损毁块文件>	把文件中所列出的块视为损毁块并将其标示出来，避免应用程序使用该块。至于损毁块文件，可执行 badblocks 命令来产生

-L <毁损块文件>	此参数的效果和指定“-l”参数类似，但在参考毁损块文件标示毁损块之前，会先将原来标示成毁损块者统统清除，即全部重新设定，而非仅是加入新的毁损块标示
-n	把欲检查的文件系统设成只读，并关闭互动模式，否定所有询问的问题。此参数对于“-c”、“-l”和“-L”参数没有作用，毁损块依旧会标示出来
-p	此参数的效果和指定“-a”参数相同
-P <处理inode大小>	设定fsck.ext3命令所能处理的inode大小为多少
-r	此参数将忽略不予处理，仅负责解决兼容性的问题
-s	检查文件系统时，交换每对字节的内容。1对字节含有2字节，即这2字节的内容互换，成为标准字节顺序。若该文件系统已经是标准字节顺序，则此参数将没有任何效果
-S	此参数的效果和指定“-s”参数类似，但不论该文件系统是否已是标准字节顺序，一律交换每对字节的内容
-t	显示fsck.ext3命令的时序信息。如重复指定这项参数两次，则会详细显示时序信息，每个执行过程之后都会跟着显示相关信息
-v	详细显示命令执行过程
-V	显示版本信息
-y	关闭互动模式，且同意所有询问的问题

范 例：

① 检查指定的ext3文件系统正确性：

```
# fsck.ext3/dev/hdc1
e2fsck 1.32, (09-Nov-2002) for fsck.ext3
/dev/hdc1: clean, 12/105952 files, 3339/211672 blocks
```

没有发现任何错误

若检查时发现有任何错误，fsck.ext3会自动尝试修复错误，如下例所示：

```
# fsck.ext3 /dev/hdc1
e2fsck 1.32, (09-Nov-2002) for fsck.ext3
/dev/hdc1 contains a file system with errors, check
forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Setting filetype for entry 'lost+found' in / (2) to 2.
Setting filetype for entry '#11' in /lost+found (12) to 2.
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
```

发现有错误

```
/dev/hdc1: ***** FILE SYSTEM WAS MODIFIED *****
```

此文件系统的错误已经修正

```
/dev/hdc1: 12/105952 files (0.0% non-contiguous),
3339/211672 blocks
```

② 强制检查指定的ext3文件系统正确性：

```
# fsck.ext3 -f /dev/hdd2
e2fsck 1.32 (09-Nov-2002)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/hdd2: 15/276352 files (0.0% non-contiguous),
16893/552234 blocks
```

③ 强制检查指定的ext3文件系统正确性，并且显示时序信息：

```
# fsck.ext3 -ft /dev/hdd2
e2fsck 1.32 (09-Nov-2002)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Peak memory: Memory used: 450k/0k (406k/44k),
time:3.48/0.15/0.31
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/hdd2: 15/276352 files (0.0% non-contiguous),
16893/552234 blocks
Memory used: 50k/0k (5k/46k), time: 3.83/ 0.25/ 0.33
```

增加了这2行与时序有关的信息

④ 强制检查指定的ext3文件系统正确性，并显示更详细的时序信息：

```
# fsck.ext3 -f -t -t /dev/hdd2
```



```
e2fsck 1.32 (09-Nov-2002)
Pass 1: Checking inodes, blocks, and sizes
Pass 1: Memory used: 374k/0k (296k/79k),time: 3.42/ 0.20/0.36
Pass 2: Checking directory structure
Pass 2: Memory used: 450k/0k (372k/78k),time: 0.00/ 0.00/0.00
Pass 3: Checking directory connectivity
Peak memory: Memory used: 450k/0k (406k/44k),time:
3.45/0.20/ 0.36
Pass 3A: Memory used: 450k/0k (406k/44k),time: 0.00/ 0.00/0.00
Pass 3: Memory used: 450k/0k (372k/78k),time: 0.00/ 0.00/ 0.00
Pass 4: Checking reference counts
Pass 4: Memory used: 450k/0k (184k/266k),time: 0.03/ 0.03/ 0.00
Pass 5: Checking group summary information
Pass 5: Memory used: 450k/0k (150k/300k),time: 0.26/ 0.06/ 0.00
/dev/hdd2: 15/276352 files (0.0% non-contiguous),16893/552234 blocks
Memory used: 50k/0k (5k/46k), time: 3.77/ 0.29/ 0.38
```

与上例比较，每个检查过程都会显示时序信息

- ⑤ 检查指定的ext3文件系统正确性，若发现任何错误，不询问用户意见，便自动修复文件系统：

```
# fsck.ext3 -a /dev/hdc1
```

- ⑥ 检查指定的ext3文件系统是否存在损毁的块，若发现则将其标示起来：

```
# fsck.ext3 -c /dev/hdd2
e2fsck 1.32 (09-Nov-2002)
Checking for bad blocks (read-only test): 303856/552234
```

检查的过程

...

- ⑦ 检查指定的ext3文件系统的正确性，并将badblocklist文件中指定的块加到损毁块列表中：

```
# fsck.ext3 -l badblocklist /dev/hdc1
```

↑
此文件是由badblocks命令产生

- ⑧ 检查指定的ext3文件系统的正确性，并将损毁块列表设成跟badblocklist文件指定的块一致：

```
# fsck.ext3 -L badblocklist /dev/hdc1
```

- ⑨ 检查指定的ext3文件系统的正确性，执行前先清除缓冲区的数据：

```
# fsck.ext3 -F /dev/hdc1
```



⑩ 以非交互的方式，所有的问题均设定以“yes”回答，检查指定的ext3文件系统的正确性：

```
# fsck.ext3 -y /dev/hdc1
```

⑪ 以非交互的方式，所有的问题均设定以“no”回答，并以只读模式检查指定的ext3文件系统的正确性：

```
# fsck.ext3 -n /dev/hdc1
```

fsck.minix

常用指数：

(file system check-minix filesystem)

F

功能说明：检查文件系统并尝试修复错误。

语 法：fsck.minix [-aflmrsv] [外围设备代号]

包 名 称：util-linux。

相关命令：fsck, mkfs, mkfs.minix。

补充说明：当minix文件系统发生错误时，可用fsck.minix命令尝试加以修复。外围设备代号对硬盘之类的存储媒体而言，就是分区编号，对软驱来说，则是外围设备代号。



参 数:

-a	自动修复文件系统，不询问任何问题
-f	强制对该文件系统进行全面检查，即使该文件系统在概略检查下没有问题
-l	列出所有文件名称
-m	使用类似MINIX操作系统的警告信息
-r	采用互动模式，在执行修复时询问问题，让用户得以确认并决定处理方式
-s	显示该分区第1个扇区的相关信息
-v	显示命令执行过程

范 例:

① 检查指定的MINIX文件系统正确性:

```
# fsck.minix /dev/hdc1
```

② 采用非互动的方式检查指定的MINIX文件系统正确性:

```
# fsck.minix -a /dev/hdc1
```

③ 采用互动的方式检查指定的MINIX文件系统正确性:

```
# fsck.minix -r /dev/hdc1
```

④ 强制检查指定的MINIX文件系统正确性:

```
# fsck.minix -f /dev/hdc
```

```
Forcing filesystem check on /dev/hdc1.
```

ftp (file transfer protocol)常用指数: **功能说明:** 传输文件。**语法:** ftp [-dginv] [主机名称或 IP 地址]**包名称:** ftp。**相关命令:** ncftp, ncftpget, ncftpput。

补充说明: FTP (File Transfer Protocol) 是ARPANet (Advanced Research Project Agency Network) 的标准文件通信协议, 该网络就是Internet的前身。FTP让用户得以下载存放于远程主机的文件, 也能将文件上传至远程主机放置。FTP所使用的命令和Linux几乎相同, 进入FTP界面之后, 执行?或help命令就可查询所有可用命令, 执行?[命令名称]或help[命令名称]命令则能显示该命令说明。

参 数:

-d	详细显示命令执行过程, 便于debug或分析程序执行的情形
-g	关闭本地主机文件名称支持特殊字符的扩充特性
-i	关闭交谈模式, 不询问任何问题
-n	不使用自动登录。FTP在启动时会尝试自动登录远程系统, 它会从用户专属目录中读取.netrc文件的内容, 以便自行登录系统。若该文件不存在, 则FTP会放弃自动登录, 并询问用户账号名称
-v	显示命令执行过程

范 例:

① 用FTP登录ftp.ncku.edu.tw主机，可利用稍后介绍的操作命令来下载及上传文件：

```
# ftp ftp.ncku.edu.tw
```

② 以非互动模式登录ftp.ncku.edu.tw主机：

```
# ftp -i ftp.ncku.edu.tw
```

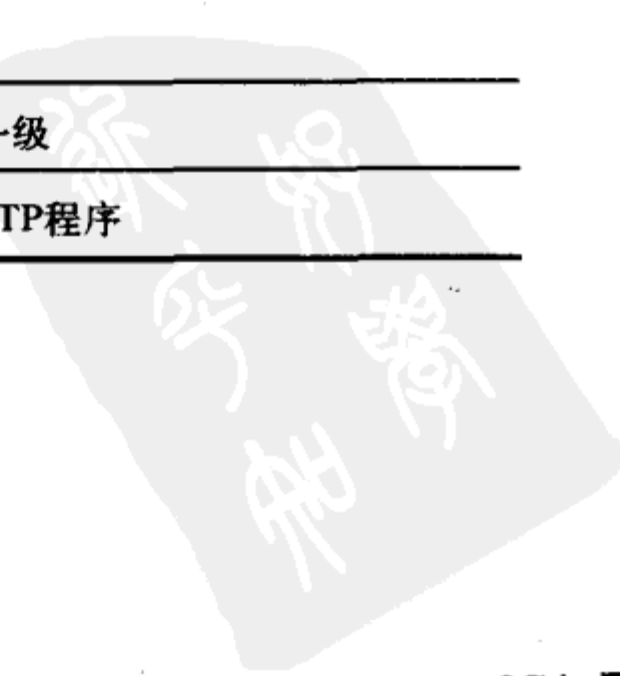
③ 用FTP登录ftp.ncku.edu.tw主机，并显示详细的命令执行过程：

```
# ftp -d ftp.ncku.edu.tw
```

操作命令:

! <命令>	在本地端执行指定的命令
? <操作命令>	列出操作命令的简要说明。若不指定 <操作命令> 参数，则列出所有可用的操作命令
append <本地文件名称><远程文件名称>	将本地文件附加到指定的远程文件之后
ascii或asc	将传输数据的模式设成ASCII模式
bell	切换是否每传完一个文件，就发出一声警告声
binary或bin	将传输数据的模式设成二进制模式
bye	结束此次连接并结束FTP程序
case	当下载文件时，是否将原本为大写的文件名称改为小写
cd <目录名称>	切换远程所在的目录
cdup	切换到远程所在目录的上一级
close	结束此次连接但并不结束FTP程序

F



delete <文件名称>	删除远程的文件
dir <目录名称>	列出远程目录的内容
disconnect	结束此次连接但并不结束FTP程序
get <文件名称>	下载远程的文件
glob	确定当使用mdelete、mget及mput操作命令时，是否可以使用通配字符
hash	确定是否每传输一个块的数据后就列出一个“#”号。当连接速度很慢时，开启此功能就可知道是不是还在传送文件
help <操作命令>	列出操作命令的简要说明。若不指定<操作命令>参数，则列出所有可用的操作命令
lcd <目录名称>	切换本地端所在的目录
ls <目录名称>	列出远程目录的内容
mdelete<文件名称...>	删除多个远程的文件
mget <文件名称...>	下载多个远程的文件
mkdir <目录名称>	在远程建立目录
modtime <文件名称>	显示远程文件上次修改的时间
mput <文件名称...>	上传多个本地端的文件
newer <文件名称>	只有当远程文件的更新时间比本地文件更新时间更近，才下载此文件
open <主机位置>	与指定FTP服务器建立FTP连接
prompt	切换互动模式和非互动模式
put <文件名称>	上传本地端的文件
pwd	列出远程当前所在的目录位置
quit	结束此次连接并结束FTP程序
recv <文件名称>	下载远程的文件
rename <来源文件名称><目的文件名称>	将远程的来源文件名称改为目的文件名称
rmdir <目录名称>	删除远程的目录

runique	当传输文件时，如果已经存在同样名称的文件，确定是否自动更改文件名。若启动此功能，当下载或上传同样文件名的文件时，系统会自动在文件名称末端加上“.1”，第2次则加上“.2”，依此类推
send <文件名称>	上传本地端的文件
status	显示当前连接的状态
system	显示远程操作系统的类型
user <用户账号>	以指定的用户账号及密码登录FTP主机<密码>
verbose	切换执行时是否显示详细的执行过程

ftpcount

常用指数：



功能说明：显示当前以FTP登录的用户人数。

语 法：ftpcount [-n] [-f <设定文件>]

包名称：proftpd (Mandrake Linux)。

相关命令：ftpsht, ftpwho。

补充说明：这是Mandrake Linux才有的命令。执行这项命令可得知当前用FTP登录系统的人数，以及FTP登录人数的上限。

参 数：

-f <设定文件>或--file<设定文件>	指定设定文件所在的路径
-h或--help	显示帮助


范 例:

查询当前FTP用户的人数:

```
# ftpcount
Master proftpd process 2085:
Service class          -    3 user
```

↑
当前共有3个用户登录FTP

ftpshut (ftp shutdown)

常用指数: 

功能说明: 在指定的时间关闭 FTP 服务器。

语 法: ftpshut [-d <分钟>] [-l <分钟>] [关闭时间] [“警告信息”]

包 名 称: proftpd (Mandrake Linux)。

相关命令: ftpcount, ftpwho。

补充说明: 这是Mandrake Linux才有的命令。本命令提供系统管理者在设定的时间关闭FTP服务器，且能在关闭之前发出警告信息通知用户。关闭时间若设定为“now”，则会马上关闭服务器。如果采用+30的方式来设定，表示服务器将在30分钟之后关闭，依此类推。假设使用1130的格式，则表示服务器会在每日的11时30分关闭，时间的格式为24小时制。可以在关闭服务器之前，使用警告信息通知所有的FTP用户。FTP服务器关闭之后，在/etc目录下会产生一个名称为 shutmsg的文件，删除该文件后即可再次启动FTP服务器的功能。

参 数:

-d <分钟>	切断所有FTP连接的时间。默认在服务器关闭前5分钟之内，将会切断现在所有的FTP连接，可以使用这个参数来改变它
-l <分钟>	停止接受FTP登录的时间。默认在服务器关闭前10分钟之内，将不再接受新的FTP登录，可以使用这个参数来改变它

范 例:

① 在午夜0时0分关闭FTP服务器，并在关闭前5分钟拒绝新的FTP登录，前3分钟切断所有FTP的连接，且送出警告信息“Server will be shutdown at 00:00:00”给登录用户：

```
# ftpshut -d 3 -l 5 0000 "Server will be shutdown at 00:00:00"
```

② 现在立即关闭FTP服务器，并送出警告信息“Server will be shutdown now”给登录的用户：

```
# ftpshut now "Server will be shutdown now"
```

ftpwho

常用指数:



功能说明: 显示当前所有以FTP登录的用户信息。

语 法: ftpwho [-v]

包 名 称: proftpd (Mandrake Linux)。

相关命令: ftpcount, ftpshut。



补充说明：这是Mandrake Linux才有的命令。执行这项命令可得知当前用FTP登录系统的用户有哪些人，以及他们正在进行的操作。

参 数：

-v	显示版本信息
----	--------

范 例：


查询当前有哪些用户正以FTP登录系统：

ftpwho

```
standalone FTP daemon [2085]:  
 3547 wyw      [ 2m20s]  2m13s (idle)  
Service class          -    1 user
```



gemtopbm

常用指数: 

(gem to portable bitmap)

功能说明: 转换图像文件。

语法: gemtopbm [-d] [GEM 图像文件]

包名称: netpbm-progs (Red Hat Linux/Fedora Core), netpbm (Mandrake Linux)。

补充说明: 执行gemtopbm命令, 读取指定的GEM格式图像文件(默认最后的扩展名为.img), 并把该文件转成Portable Bitmap(默认最后的扩展名为.pbm)格式后, 输出到标准输出设备。若不指定任何文件名称, 则gemtopbm命令会从标准输入设备读取数据。

参数:

-d或-debug	详细显示命令执行过程, 便于debug或了解程序执行的情况
-----------	-------------------------------

范例:

把mypicture.img图像文件转成Portable Bitmap格式:

```
# gemtopbm mypicture > mypicture.pbm
```

getlist

常用指数: 

功能说明: 下载新闻组列表。

语法: getlist [-h <新闻服务器>] [-p <通信端口>] [组类型 <范本样式><组名称>]

包名称: inn。

相关命令: nntpget, slrn。

补充说明: getlist命令会从新闻服务器下载新闻组列表, 并把列表输出到标准输出设备。其中组类型可分为“active”、“active.times”、“distributions”和“newsgroups”。所谓范本样式, 例如指定的范本为“alt.*”, getlist命令会列出所有起始字符串为“alt.”的新闻组, 如alt.3d、alt.albus.freehand、alt.amiga.demos等。

参 数:

-h<新闻服务器>	指定欲连接的新闻服务器
-p<通信端口>	设定连接新闻组服务器的通信端口。默认值为119, 可以用这项参数改变它

范 例:


① 从news.sinica.edu.tw下载所有的新闻组列表:

```
# /usr/lib/news/bin/getlist -h news.sinica.edu.tw
comp 0000001300 0000001300 y
comp.sys 0000000144 0000000144 y
comp.sys.laptops 0000206050 0000200426 y
comp.sys.atari 0000000389 0000000389 y
comp.sys.atari.st 0000060223 0000059916 y
...
```

② 从news.sinica.edu.tw下载与tw.bbs.comp有关的新闻组列表:

```
# /usr/lib/news/bin/getlist -h news.sinica.edu.tw active
tw.bbs.comp.*
tw.bbs.comp.linux 0000313975 0000311302 y
tw.bbs.comp.hacker 0000415498 0000412622 y
tw.bbs.comp.book 0000042110 0000041874 y
tw.bbs.comp.www 0000113143 0000112482 y
tw.bbs.comp.software 0000207829 0000205700 y
...
```

giftopnm

常用指数: 

(graphics interchange format to portable anymap)

功能说明: 转换图像文件。**语 法:** giftopnm [-comments] [-image <页数编号>] [-verbose]
[GIF 图像文件]**包 名 称:** netpbm-progs (Red Hat Linux/Fedora Core), netpbm
(Mandrake Linux)。**补充说明:** 执行giftopnm命令, 读取指定的GIF格式图像文件(默认最后的扩展名为.gif), 并把该文件转成Portable Anymap(默认最后的扩展名为.pnm)格式后, 输出到标准输出设备。Anymap的格式可以是Bitmap、Graymap或Pixmap, 根据输入的文件格式而定。**参 数:**

-comments	仅输出GIF89格式里的注释部分
-image <页数编号>	某些GIF图像文件内, 包含不只一张图像(如GIF89a格式, 即GIF动画文件), 这些图像每张一页, 依次轮流显示, 展现动画效果。可用此参数指定图像的页数编号, 将其一一解出
-verbose	显示命令执行过程

范 例:

① 文件gifanime.gif是一个GIF动画文件, 内含5张图像, 我们用giftopnm命令将这5张图像分别存成5个Portable Anymap图像文件:

```
# giftopnm -image 1 gifanime.gif > ga_5-1.pnm
# giftopnm -image 2 gifanime.gif > ga_5-2.pnm
# giftopnm -image 3 gifanime.gif > ga_5-3.pnm
# giftopnm -image 4 gifanime.gif > ga_5-4.pnm
```

gouldtoppm

常用指数: 

(gould to portable pixmap)

功能说明: 转换扫描文件。**语法:** `gouldtoppm [Gould 扫描文件]`**包名称:** `netpbm-progs` (Red Hat Linux/Fedora Core), `netpbm` (Mandrake Linux)。**补充说明:** 执行 `gouldtoppm` 命令, 读取指定的 Gould 扫描器产生的扫描文件, 并把该文件转成 Portable Pixmap (默认的扩展名为 `.ppm`) 格式后, 输出到标准输出设备。**参数:** 无。**范例:**把扫描文件 `fromgouldscanner` 转成 Portable Pixmap 格式:

```
# gouldtoppm fromgouldscanner > nowitsapixmap.ppm
```

grep

常用指数: **功能说明:** 查找文件里符合条件的字符串。**语法:** `grep [-abcCEFGhHilLnqrsVwxy] [-A <显示行数>] [-B <显示行数>] [-d <进行操作>] [-e <范本样式>] [-f <范本文件>] [--help] [范本样式] [文件或目录...]`

包名称: grep。

相关命令: egrep, fgrep, rgrep。

补充说明: grep命令用于查找内容包含指定范本样式的文件，如果发现某文件的内容符合所指定的范本样式，默认grep命令会把含有范本样式的那一行显示出来。若不指定任何文件名称，或是所给予的文件名为“-”，则grep命令会从标准输入设备读取数据。

参 数:

-a或--text	不要忽略二进制的文件。若grep命令找到符合范本样式的内容，但该文的数据是二进制格式而非文本文件时，通常grep命令仅会返回是否符合条件，而不显示其内容。使用本参数可强制grep命令显示内容，无论是否为二进制文件
-A<显示行数>或--after -context=<显示行数>	除了显示符合范本样式的那一行之外，并显示该行之后的内容。指定显示行数为3，就会显示从该行算起，到后3行之间的全部内容。指定显示行数为5，就会显示从该行算起，到后5行之间的所有内容，依此类推
-b或--byte-offset	在显示符合范本样式的那一行之前，标示出该行第1个字符的位编号。编号数从文件的第1个字符算起，即每个文件的第1个字符，其位编号为1，该文件第100个字符的位编号就是100，依此类推
-B<显示行数>或--before -context=<显示行数>	除了显示符合范本样式的那一行之外，并显示该行之前的内容。指定显示行数为3，就会显示从该行算起，到前3行之间的全部内容。指定显示行数为5，就会显示从该行算起，到前5行之间的所有内容，依此类推
-c或--count	计算符合范本样式的行数。使用本参数将不会显示合乎条件之行的内容，而会显示每个文件里共有几行符合范本样式。搭配参数“-v”使用，则会显示每个文件里共有几行不符合范本样式

-C、--context=<显示行数>或-<显示行数>	除了显示符合范本样式的那一行之外，并显示该行前后的内容。指定显示行数为3，就会显示从该行算起，往前后各3行之间的全部内容。指定显示行数为5，就会显示从该行算起，往前后各5行之间的所有内容，依此类推。默认的显示行数为2
-d <进行操作>或--directories=<进行操作>	当指定要查找的是目录而非文件时，必须使用这项参数，否则grep命令将返回信息并停止操作。有3个进行操作得以设定：“read”、“recurse”和“skip”。“read”为默认值，表示grep命令会把目录当作一般文件进行查找字符串。“recurse”是递归处理，表示将该目录下的所有文件及子目录一并处理。“skip”则是略过该目录不予处理
-e <范本样式> 或--regexp=<范本样式>	指定字符串作为寻找文件内容的范本样式。这和直接指定范本样式并无不同，但对于以“-”号为首的字符串来说，可有效避免与参数混淆，导致grep命令发生错误
-E或--extended-regexp	将范本样式作为延伸的普通表示法来使用。此参数的效果和egrep命令类似
-f <范本文件>或--file=<范本文件>	指定范本文件，其中含有一个或多个范本样式，让grep查找符合范本条件的文件内容，格式为每行一个范本样式。一个空的范本文件等于没有指定范本样式，因此也就不会有任何文件符合条件
-F或--fixed-regexp	将范本样式视为固定字符串的列表，两两之间以新增行控制字符分隔，任一个字符串符合条件即可。此参数的效果和fgrep命令相同
-G或--basic-regexp	将范本样式作为普通的表示法来使用。此为默认值
-h或--no-filename	在显示符合范本样式的那一行之前，不标示该行所属的文件名称
-H或--with-filename	在显示符合范本样式的那一行之前，标示该行所属的文件名称。此为默认值
-i或--ignore-case	忽略字符大小写区别。包括范本样式和指定要查找的文件名称
-l或--files-with-matches	列出文件内容符合指定范本样式的文件名称

-L或--files-without-match	列出文件内容不符合指定范本样式的文件名称
-n或--line-number	在显示符合范本样式的那一行之前，标示出该行的行数编号
-q、--quiet或--silent	不显示任何信息
-r或--recursive	此参数的效果和指定“-d recurse”参数相同
-s或--no-messages	不显示错误信息
-v或--invert-match	反向查找。寻找并显示所有不包含指定范本样式的行的内容。举例而言，假设范本样式为1 481 010，使用此参数之后，在查找范围内的所有文件内容，每个不包含1 481 010字符串的行，都会显示出来
-V或--version	显示版本信息
-w或--word-regexp	只显示全字符合的行。假设范本样式为“email”，原本仅需符合该字符串即可，例如“gotemail”或“emailnow”都算。使用这项参数之后，必须全字符合才算数，也就是“email”需为一个完整的单字
-x或--line-regexp	只显示全行符合的行。假设范本样式为“email”，原本仅需符合该字符串即可，例如“gotemail”或“emailnow”都算。使用这项参数之后，必须全行符合才算数，也就是“email”需为单独的一行
-y	此参数的效果和指定“-i”参数相同
--help	显示帮助

范 例：

① 对当前目录中，所有扩展名为.txt的文件内容寻找包含“seti”字符串的文件：

```
# grep seti *.txt
```

② 对目录/home/sniper中，所有最后扩展名为.doc的文件内容寻找包含“shut”字符串的文件，不论大小写皆可：

```
# grep -i shut /home/sniper/*.doc
```

- ③ 对当前目录中的所有文件，包括子目录在内，寻找包含“deepwater”字符串的文件：

```
# grep -r deepwater *
```

- ④ 对当前目录中，寻找包含“alpha”字符串的文件，在寻找的过程中，若遇到子目录，并不寻找此目录中的文件：

```
# grep -d skip alpha *
```

不寻找目录中的文件

- ⑤ 对当前目录中，名称以text起始的文件，寻找包含“alpha”字符串的文件，如果找到，标示出该字符串是在第几行：

```
# grep -n alpha text*
```

- ⑥ 对当前目录中，名称以text起始的文件，寻找完全符合“alpha”字符串的文件：

```
# grep -w alpha text*
```

- ⑦ pattern文本文件中包含了两个范本样式。对当前目录中，名称以text起始的文件，寻找至少包含这两个范本样式之一的文件：

```
# cat pattern
```

```
alpha
```

```
beta
```

pattern文件内包含了2个字符串“alpha”及“beta”

```
# grep -f pattern text* ← 寻找包含“alpha”或“beta”字符串的文件
```

- ⑧ 对当前目录中，名称以text起始的所有文件，寻找包含“alpha”字符串的文件，如果找到，列出包含此字符串之后的2行：

```
# grep -A 2 alpha text*
```

- ⑨ 对当前目录中，名称以text起始的所有文件，寻找包含“alpha”字符串的文件，如果找到，列出包含此字符串之前的3行：

```
# grep -B 3 alpha text*
```

- ⑩ 对当前目录中，名称以text起始的所有文件，寻找包含“alpha”字符串的文件，如果找到，列出包含此字符串之前后各3行：

```
# grep --context=3 alpha text*
```

- ⑪ 对当前目录中，名称以text起始的所有文件，寻找包含“alpha”字符串的文件，如果找到，只列出包含此字符串的该行，并不列出文件名：

```
# grep -h alpha text*
```

- ⑫ 对当前目录中，名称以text起始的所有文件，寻找包含“alpha”字符串的文件，如果找到，只列出文件名称，并不显示包含此字符串的该行：

```
# grep -l alpha text*
```

- ⑬ 对当前目录中，名称以text起始的所有文件，寻找不包含“alpha”字符串的文件，如果找到，列出文件名称：

```
# grep -L alpha text*
```

- ⑭ 对当前目录中，名称以text起始的文件，寻找不包含“alpha”字符串的文件，如果找到，列出其文件名称，以及不包含此字符串的该行：

```
# grep -v alpha text*
```

↑
利用反向查找

- ⑮ 对当前目录中，名称以text起始的文件，寻找某行为“For Love of the Game”的文件，如果找到，列出其文件名称及该行：

```
# grep -x 'For Love of the Game' text*
```

- ⑯ 对当前目录中，名称以text起始的所有文件，计算“alpha”字符串出现的次数：

```
# grep -c alpha text*
```

- ⑰ 对当前目录中，名称以text起始的所有文件，寻找包含“alpha”字符串的文件，若寻找过程有遇到任何错误，不显示错误信息：

```
# grep -s alpha text*
```

groupadd (group add)

常用指数：



功能说明： 建立组。

语法： groupadd [-fr] [-g <组识别码> <-o>] [组名称]

包名称： shadow-utils。

相关命令： groupdel, groupmod, id, newgrp, useradd, usermod。

补充说明：需要在系统上建立组时，可用groupadd命令来完成这项工作。

参 数：

-f	强制建立已存在的组。每个组名称在该系统中都应该是唯一且不重复的，假设所给予的组名称已经存在，则groupadd命令会返回错误信息。使用本参数可强制系统接受已存在的组名称
-g <组识别码>	设定新建立组的识别码。若不附加这项参数，则系统会自行从编号500开始，依次分派给新建立的组使用，编号500之前则保留给系统各项服务的账号使用
-o	重复使用组识别码。每个组识别码在该系统中都应该是唯一且不重复的，假设使用“-g”参数指派组识别码，该编号却已经存在，则groupadd命令会返回错误信息。配合参数“-o”使用，可强制系统使用已存在的组识别码。本参数仅和“-g”参数搭配使用才有效果
-r	建立系统组。系统提供各项服务时，必须使用某些组名，这些组的识别码都在第0~499号，一般组的编号从500开始。使用此参数所建立的组识别码，其值将小于500，可利用它来建立系统服务所需的组

范 例：

① 新增一个名称为sphenoid的组：

```
# groupadd sphenoid
```

② 新增一个名称为cirware的组，并指定组识别码为700：

```
# groupadd -g 700 cirware
```

③ 新增一个名称为sysd的系统组：

```
# groupadd -r sysd
```

groupdel (group delete)

常用指数:



功能说明: 删除组。

语法: groupdel [组名称]

包名称: shadow-utils。

相关命令: groupadd, groupmod, id, newgrp, useradd, usermod。

补充说明: 需要从系统上删除组时, 可用groupdel命令来完成这项工作。假如该组中仍包括某些用户, 则必须先删除这些用户后, 才能删除组。

参数: 无。

范例:

删除sphenoid组:

```
# groupdel sphenoid
```

groupmod

常用指数:



(group modify)

功能说明: 更改组识别码或名称。

语法: groupmod [-g <组识别码> <-o>] [-n <新组名称>] [组名称]

包名称: shadow-utils。

相关命令: groupadd, groupdel, id, newgrp, useradd, usermod。

补充说明：需要更改组的识别码或名称时，可用groupmod命令来完成这项工作。

参 数：

-g <组识别码>	设定欲使用的组识别码
-n <新组名称>	设定欲使用的组名称。每个组名称在该系统中都应该是唯一且不重复的，假设所给予的组名称已经存在，则groupmod命令会返回错误信息
-o	重复使用组识别码。每个组识别码在该系统中都应该是唯一且不重复的，假设使用“-g”参数分配组识别码，该编号却已经存在，则groupmod命令会返回错误信息。配合参数“-o”使用，可强制系统使用已存在的组识别码。本参数仅和“-g”参数搭配使用才有效果

范 例：

① 将oldname组名称更名为newname：

```
# groupmod -n newname oldname
```

② 将newname组的组识别码更改为510：

```
# groupmod -g 510 newname
```

grpconv

常用指数：



(group convert to shadow password)

功能说明：打开组的影子密码 (Shadow Password)。

语 法： grpconv

包名称: shadow-utils。

相关命令: grpunconv, pwconv, pwunconv。

补充说明: Linux系统里的用户和组密码, 分别存放在/etc目录下的passwd和group文件中。因系统运作所需, 任何程序都能读取它们, 这造成安全上的漏洞。影子密码将文件内的密码改存在/etc目录下的shadow和gshadow文件内, 只允许系统管理者读取, 同时把原密码替换为“x”字符, 有效地强化了系统的安全性。影子密码的功能可随时打开或关闭, 只需执行grpconv命令就能打开组影子密码。

参 数: 无。

范 例:

打开组影子密码, 加强安全防护:

```
# grpconv
```

grpunconv

常用指数:



(group unconvert from shadow password)

功能说明: 关闭组的影子密码。

语 法: grpunconv

包名称: shadow-utilss。

相关命令: grpconv, pwconv, pwunconv。



补充说明：执行grpunconv命令可关闭组影子密码，它会把密码从gshadow文件内存回到group文件里。


参 数：无。

范 例：

关闭组影子密码，方便直接编辑group文件建立、删除组名称：

```
# grpunconv
```

grub-install

常用指数：

功能说明：安装GRUB启动管理程序。

语 法：grub-install [-hv] [设备名称]

包 名 称：grub。

相关命令：lilo。

补充说明：将GRUB启动管理程序安装到指定的磁盘或磁盘分区。

参 数：

-h	显示帮助
-v	安装时显示详细信息

范 例：

① 将GRUB安装在hda的MBR：

```
# grub-install /dev/hda
```



② 将GRUB安装在hda6的Linux分区，并显示详细信息：

```
# grub-install -v /dev/hda6
grub-install (GNU GRUB 0.93)
```

③ 将GRUB安装在软盘上：

```
# grub-install /dev/fd0
```

gunzip (gnu unzip)

常用指数：



功能说明：解压文件。

语法： gunzip [-acfhLnNqrtvV] [-S <压缩字尾字符串>] [文件...]
或gunzip [-acfhLnNqrtvV] [-S <压缩字尾字符串>] [目录]

包名称：gzip。

相关命令：gzip, zcat。

补充说明：gunzip是个使用广泛的解压程序，它用于解开被gzip压缩过的文件，这些压缩文件默认最后的扩展名为.gz。事实上，gunzip就是gzip的硬链接，因此不论是压缩或解压，都可借由gzip命令单独完成。

参数：

-a或-ascii

使用ASCII文字模式。由于非UNIX系统家族的操作系统（例如DOS），其文本文件每行以LF+CR字符当作结尾，但UNIX却仅以LF字符做结尾，造成在非UNIX的操作系统下观看UNIX格式的文本文件，会出现阶梯状的排列问题。使用本参数之后，gunzip命令在解压时，会把文本文件内的LF控制字符置换成LF+CR字符，以便供非UNIX之类的操作系统使用。此参数仅供非UNIX家族的操作系统使用

-c、--stdout或--to-stdout	把解压后的文件输出到标准输出设备，不去改变原始文件。默认gunzip命令会将压缩文件直接解压还原，而不保留原来的文件。可用“>”等特殊字符将解压后的结果导向一个新的文件，存储解压缩后的文件却不会变动原来的压缩文件。假设不把输出导向到某个存储设备，将会在显示器上看见解压缩的内容，除非原本是纯文本文件，否则只是一堆乱码
-f或-force	强行解开压缩文件，不理睬文件名称或硬链接是否存在，以及该文件是否为符号链接。如不使用这个参数，则解压缩时若遇到已存在的文件名称，gunzip命令会一一询问用户，确认是否要覆盖已有文件。至于硬链接或符号链接，gunzip默认是忽略不予处理，加上本参数后，gunzip命令就会解开该连接所指向、连接的压缩文件。注意这些连接在解压之后，无论是硬链接或符号链接都不复存在，而成为另一个文件。另外需留意权限问题，当强制解开这些硬链接或符号链接压缩文件时，其指向、连接的原始文件的权限将会全部开放，请注意系统的安全问题
-h或-help	显示帮助
-l或-list	列出压缩文件的相关信息。这些信息包括：压缩后的文件大小、未压缩的原始大小、压缩比，以及压缩前的文件名称。本参数如搭配“-v”参数使用，则会增列压缩法、CRC校验码，还有压缩文件的日期和时间
-L或-license	显示版本与版权信息
-n或--no-name	解压时，若压缩文件内含有原来的文件名称及时间戳，则将其忽略不予处理。即压缩文件在解压之后，其文件名就是原名去掉压缩字尾字符串。这个参数对于gunzip命令而言为默认值
-N或-name	解压时，若压缩文件内含有原来的文件名称及时间戳，则将其回存至解开的文件上
-q或-quiet	不显示警告信息
-r或-recursive	递归处理，将指定目录下的所有文件及子目录一并处理

-S <压缩字尾字符串> 或--suffix <压缩字尾 字符串>	更改压缩字尾字符串。默认的字尾字符串为“.gz”， 可以利用这项参数改变它。gunzip命令能接受的压缩 字尾字符串，包括：“-gz”、“.z”、“-z”、“_z”和“.Z”。 建议没有特殊需求，请勿随意改变字尾字符串，因为 此举将造成文件类型的混淆
-t或--test	测试压缩文件是否正确无误
-v或--verbose	显示命令执行过程。配合参数“-l”使用，可列出更 详细的相关信息
-V或--version	显示版本信息

范 例：

① 解开file1.gz压缩文件：

```
# gunzip file1.gz
```

② 解开file2.gz压缩文件，解开后保留原来文件：

```
# gunzip -c file2.gz > file2
```

↑
将输出转到另一个文件

```
# ls
```

```
file2  file2.gz
```

↑
原来文件被保留下来

③ 把/home/darkseed目录下的所有压缩文件一并解压，包括子目录：

```
# gunzip -r /home/darkseed
```



④ 列出当前目录中所有gzip压缩文件的信息：

```
# gunzip -l *.gz
```

```
compressed  uncompressed.  ratio  uncompressed_name
    15430      53170      71.0%    file1
    35583      202868     82.4%    file2
     6361      22579     71.9%    file3
    57374      278617     79.4%    (totals)
```

↑
压缩后文件大小

↑
压缩前文件大小

↑
压缩比例

↑
压缩前的文件名称

⑤ 检查当前目录中所有gzip压缩文件是否正确无误：

```
# gunzip -t *.gz
```

gzexe (gzip executable)

常用指数：



功能说明： 压缩执行文件。

语 法： gzexe [-d] [执行文件...]

包 名 称： gzip。

补充说明： gzexe是用来压缩执行文件的程序。当去执行被压缩过的执行文件时，该文件会先自动解压然后继续执行，和执行一般的执行文件相同（自动解压的过程会花上一些时间）。假如磁盘空间十分有限，可考虑利用它来减少可执行文件所占用的空间。

参 数：

-d	解开压缩文件。使用本参数后，gzexe命令会把该执行文件解压而非压缩
----	------------------------------------

范 例:

① `execfile`文件是一个可执行文件，我们把它压缩起来以节省磁盘空间：

```
# gzexe execfile
```

```
test:                55.9%    55.9%的压缩比
```

```
# ls -l
```

```
总计 44
```

```
-rwxr-xr-x  1 root  root  13284  6月 2  12:20  execfile
```

↑
压缩后文件的大小

↑
压缩后的执行文件

```
-rwxr-xr-x  1 root  root  28628  6月 2  12:20  execfile -
```

↑
压缩前文件的大小

↑
原始文件改名成`execfile-`备份起来。如果确定压缩后没问题，即可删除此文件

② 解开压缩过的`execfile`执行文件：

```
# gzexe -d execfile
```

gzip (gnu zip)

常用指数：



功能说明：压缩文件。

语 法：`gzip [-acdfhlLnNqrtvV] [-S <压缩字尾字符串>] [-<压缩比>] [--best] [--fast] [文件...] 或 gzip [-acdfhlLnNqrtvV] [-S <压缩字尾字符串>] [-<压缩比>] [--best] [--fast] [目录]`

包 名 称：`gzip`。

相关命令：`gunzip`, `zcat`。

补充说明：gzip是个使用广泛的压缩程序，文件经它压缩过后，其名称后面会多出“.gz”的扩展名。当要解压时，可附加参数或执行gunzip命令。事实上，gunzip就是gzip的硬链接，因此不论是压缩或解压缩，都可使用gzip命令单独完成。若不指定任何文件名称，或是所给予的文件名为“-”，则gzip命令会从标准输入设备读取数据，压缩后再把结果输入到输出设备。

参 数：

-a或-ascii	使用ASCII文字模式。由于非UNIX系统家族的操作系统(例如DOS)，其文本文件每行以LF+CR字符当作结尾，但UNIX却仅以LF字符作结尾，造成在非UNIX的操作系统下观看UNIX格式的文本文件，会出现阶梯状的排列问题。使用本参数之后，gzip命令在压缩时，会把文本文件内的LF+CR控制字符置换成LF字符，以便供UNIX之类的操作系统使用。此参数仅供非UNIX系统家族的操作系统使用
-c、--stdout或 --to-stdout	把压缩后的文件输出到标准输出设备，不去改变原始文件。默认gzip命令会将文件直接压缩后加上“.gz”扩展名，而不保留原来的文件。可用“>”等特殊字符将压缩后的结果导向一个新的文件，存储压缩后的文件却不会改变源文件。假设不把输出导向到某个存储设备，将会在显示器上看见压缩文件的内容，也就是一堆乱码
-d、--decompress或 --uncompress	解开压缩文件。gzip命令加上这个参数就等于gunzip命令
-f或--force	强行压缩文件，不管文件名称或硬链接是否存在，以及该文件是否为符号链接。如不使用这项参数，则压缩后加上“.gz”扩展名的文件名称若已存在，gzip命令便会一一询问用户，确认是否要覆盖已有的文件。至于硬链接或符号链接，gzip默认是忽略不予处理，加上本参数后，gzip命令就会读取该连接所指向、连接的原始文件的内容，并将它压缩后保存文件。注意这些压缩文件在解压之后，无论是硬链接或符号链接都不复存在，而成为与原始文件内容完全相同的另一个文件。另外需留意权限问题，当强制解开这些硬链接或符号链接压缩文件时，其指向、连接的原始文件的权限将会全部开放，请注意系统的安全问题

-h或-help	显示帮助
-l或-list	列出压缩文件的相关信息。这些信息包括：压缩后的文件大小、未压缩的原始大小、压缩比，以及压缩前的文件名称。本参数如搭配“-v”参数使用，则会增加压缩法、CRC校验码，还有压缩文件的日期和时间
-L或-license	显示版本与版权信息
-n或--no-name	压缩文件时，不存储原来的文件名称及时间戳
-N或-name	压缩文件时，存储原来的文件名称及时间戳。这个参数对于 gzip 命令而言为默认值
-q或-quiet	不显示警告信息
-r或-recursive	递归处理，将指定目录下的所有文件及子目录一并处理
-S<压缩字尾字符串> 或--suffix<压缩字尾字符串>	更改压缩字尾字符串。默认的字尾字符串为“.gz”，可以利用这项参数改变它。gzip命令能接受的压缩字尾字符串，包括：“-gz”、“.z”、“-z”、“_z”和“.Z”。建议没有特殊需求，请勿随意改变字尾字符串，因为此举将造成文件类型的混淆
-t或-test	测试压缩文件是否正确无误
-v或-verbose	显示命令执行过程。配合参数“-l”使用，可列出更详细的相关信息
-V或-version	显示版本信息
-<压缩比>	压缩比是一个介于1~9的数值，默认值为6，指定越大的数值，压缩比就会越高。压缩比更好，产生的压缩文件就更小，但耗费在压缩过程的时间也就越多
--best	此参数的效果和指定“-9”参数相同
--fast	此参数的效果和指定“-1”参数相同

范 例：

① 压缩当前目录中所有文件：

```
# gzip *
```

② 列出当前目录中所有gzip压缩文件的信息:

```
# gzip -l *.gz
```

compressed	uncompressed.	ratio	uncompressed_name
15432	53170	71.0%	AAA.TXT
18834	77079	75.6%	ch0D.txt
35585	202868	82.4%	chod.ps
278	2979	91.5%	extract
2943	11810	75.3%	install.log
47581	173852	72.6%	mymagic
6364	22579	71.9%	textfile
127017	544337	76.6%	(totals)

↑
压缩后文件大小

↑
压缩前文件大小

↑
压缩比例

↑
压缩前的文件名

③ 压缩file1文件, 压缩后保留原来文件:

```
# gzip -c file1 > file.gz
```

↑
将输出转到另一个文件

```
# ls
```

```
file1  file1.gz
```

↑
原来文件被保留下来

④ filelink是一个硬链接文件, 强制压缩此文件:

```
# gzip filelink ← 以默认的方式压缩此硬链接文件
```

```
gzip: filelink has 1 other link -- unchanged ← 压缩失败
```

```
# gzip -f filelink ← 强制压缩此文件
```

⑤ 压缩/home/aurora目录下所有文件及子目录下的文件：

```
# gzip -r /home/aurora/*
```

⑥ 以压缩比最高的方式压缩当前目录中所有文件，压缩时显示命令执行的过程：

```
# gzip --best -v *
```

```
file1:          71.1% -- replaced with file1.gz
file2:          83.1% -- replaced with file2.gz
file3:          72.0% -- replaced with file3.gz
```

↑
与后面2个例子比较，此方式的压缩比最高

⑦ 指定以压缩比5的方式压缩当前目录中所有文件，压缩时显示命令执行的过程：

```
# gzip -5 -v *
```

```
file1:          70.8% -- replaced with file1.gz
file2:          81.1% -- replaced with file2.gz
file3:          71.7% -- replaced with file3.gz
```

↑
与前后例子比较，此方式的压缩比居中

⑧ 以压缩时间最快的方式压缩当前目录中所有文件，压缩时显示命令执行的过程：

```
# gzip --fast -v *
```

```
file1:          67.8% -- replaced with file1.gz
file2:          76.0% -- replaced with file2.gz
file3:          67.8% -- replaced with file3.gz
```

↑
与前面2个例子比较，此方式的压缩比最低

⑨ 压缩file1文件，并指定压缩后字尾字符串为“.secret”：

```
# gzip -S .secret file1
```

```
# ls
```

```
file1.secret ← 压缩后的文件名
```

⑩ 检查当前目录中所有gzip压缩文件是否正确无误：

```
# gzip -t *.gz
```

```
gzip: errorfile.gz: unknown method 149 -- get newer  
version of gzip ← errorfile.gz 压缩文件有错误
```

⑪ 解开当前目录中所有gzip压缩文件：

```
# gzip -d *.gz
```



halt

常用指数:



功能说明: 关闭系统。

语法: halt [-dfinpw]

包名称: SysVinit。

相关命令: reboot, shutdown, sync, runlevel。

补充说明: halt会先检测系统的runlevel (执行等级)。若执行等级不为0或6, 则关闭系统, 否则调用shutdown来关闭系统。

参数:

-d	不要在zwtmp (登录日志文件) 中日志
-f	不论当前的runlevel是什么, 不调用shutdown就强制关闭系统
-i	在halt之前关闭全部的网络接口
-n	halt前不用先执行sync (将内存中的数据写入硬盘)
-p	halt之后执行poweroff (关闭电源)
-w	仅在wtmp中日志, 而不实际结束系统

范例:

① 关闭系统, 并且不要记录在登录记录文件中:

```
# halt -d
```

② 关闭系统, 并关闭电源:

```
# halt -p
```



③ 先关闭全部的网络接口设备之后，再关闭系统：

```
# halt -i
```

④ 不先将内存数据写入硬盘就将系统关闭：

```
# halt -n
```

hdparm (hard disk parameters)

常用指数：

功能说明：显示与设置硬盘的参数。

语法：hdparm [-CfghiIqtTvyYZ] [-a <缓存扇区>] [-A <0 或 1>]
[-c <I/O 模式>] [-d <0 或 1>] [-k <0 或 1>] [-K <0 或 1>]
[-m <扇区数>] [-n <0 或 1>] [-p <PIO 模式>] [-P <扇区数>]
[-r <0 或 1>] [-S <时间>] [-u <0 或 1>] [-W <0 或 1>] [-X <
传输模式>] [设备]

包名称：hdparm。

补充说明：hdparm可检测、显示与设置IDE或SCSI硬盘的参数。其中有些参数若设置错误，可能导致硬盘读取发生问题。因此，使用时务必格外小心。请尽可能使用最新版的kernel，才能正确地配合较新的硬盘设备。

若不指定参数，则IDE硬盘默认会使用-acdgkmnru参数，SCSI硬盘则使用“-gr”参数。

参数：

-a <缓存扇区>	设置读取文件时，预先存入缓存的扇区数（默认值为8）若不加上<缓存扇区>选项，则显示当前的设置。若预估硬盘的循环存取机会较高，可考虑增加缓存扇区的数量。若预估硬盘的随机存取机会较高，可考虑减少缓存扇区的数量
-----------	--

-a <缓存扇区>	设置读取文件时，预先存入缓存的扇区数（默认值为8）
-A <0或1>	启动或关闭读取文件时的缓存功能。0表示关闭，1表示启动（默认值为1）。此参数仅适用于IDE硬盘
-c <I/O模式>	设置IDE 32位I/O模式。<I/O模式>的选项如下： 0 关闭32位模式（也就是使用16位模式），此为默认值 1 启动32位I/O模式 3 启动特殊的32位I/O模式。有些硬盘必须使用此参数才能启动32位I/O模式 若不加上选项，则显示当前设置
-C	检测IDE硬盘的电源管理模式。有下列几种模式： unknown 无法检测 active/idle 正常作业模式 standby 省电模式 sleeping 睡眠模式
-d <0或1>	设置硬盘的DMA模式。0表示关闭，1表示启动（默认值为0）。若不加上<0或1>选项，则显示当前的设置。本参数仅适用于IDE硬盘
-f	将内存缓冲区的数据写入硬盘，并清除缓冲区
-g	显示硬盘的磁道、磁头、扇区等参数
-h	显示帮助
-i	显示硬盘的硬件规格信息，这些信息是在开机时由硬盘本身所提供。此参数仅适用于IDE硬盘
-l	直接读取硬盘所提供的硬件规格信息。此参数仅适用于IDE硬盘
-k <0或1>	重设硬盘时，保留“-dmu”参数的设置。0表示关闭，1表示启动（默认值为0）。当“-dmu”参数设置有误时，会使硬盘重设，而恢复“-dmu”的默认值（-k的选项设为0时）。因此，将“-k”参数的选项设为1之前，务必先确认“-dmu”参数可正常运作，否则会使硬盘不停地执行重设的动作。此参数仅适用于IDE硬盘
-K <0或1>	重设硬盘时，保留“-APSWXZ”参数的设置。0表示关闭，1表示启动。此参数仅适用于IDE硬盘

-m <扇区数>	设置硬盘多重扇区存取的扇区数。若不加上<扇区数>选项，则显示当前的设置。此参数仅适用于IDE硬盘。所谓多重扇区存取是指每次可传输许多扇区的功能，可显著改善硬盘存取的效率。至于每次可存取的扇区数，依每个硬盘的设计而有所不同。可以使用“-i”参数，查看硬盘MaxMultSect（多重扇区存取时，扇区数的上限值）的值。默认值为0，也就是关闭多重扇区存取的功能。请注意，若设置的<扇区数>超出硬盘本身的许可，可能会导致硬盘数据损毁
-n <0或1>	忽略硬盘写入时所发生的错误（选项设为1）。若不加上<0或1>选项，则显示当前的设置。此参数仅适用于IDE硬盘
-p <PIO模式>	设置硬盘的PIO模式。PIO模式总共有6种（从0~5）。若不加上<PIO模式>选项，则显示当前的设置。此参数仅适用IDE硬盘。可以使用“-i”参数，查看硬盘所支援的PIO模式。若设置有误，可能导致硬盘数据损毁
-P <扇区数>	设置硬盘内部缓存的扇区数。此参数仅适用于IDE硬盘
-q	在执行后续的参数时，不在屏幕上显示任何信息
-r <0或1>	设置硬盘读写模式。0表示可写入，1表示只读，默认值为1
-S <时间>	设置硬盘进入省电模式前的等待时间。<时间>选项的单位为5秒钟，最小值为0（表示关闭省电模式的功能），最大值为240（表示1200秒，也就是20分钟）
-t	评估硬盘的读取效率
-T	评估硬盘缓存的读取效率
-u <0或1>	在硬盘存取时，允许其他中断要求同时执行（选项设为1）。默认值为0，表示关闭此功能。此参数仅适用于IDE硬盘
-v	显示硬盘的相关设置。效果与不加任何参数相同
-W <0或1>	设置硬盘的写入缓存。0表示关闭，1表示启动，默认值为0。虽然启动硬盘的写入缓存可提高写入效率，但相对的也提高数据遗失的几率。此参数仅适用于IDE硬盘
-X <传输模式>	设置硬盘的传输模式。例如-X 34表示DMA2模式，-X 09表示PIO模式1，-X 10则采用默认的PIO模式。通常不设置此参数。此参数仅适用于IDE硬盘
-y	使IDE硬盘进入省电模式
-Y	使IDE硬盘进入睡眠模式
-Z	关闭某些Seagate硬盘（如ST3系列的硬盘）的自动省电功能

范 例:

① 显示硬盘的相关设置:

```
# hdparm /dev/hdc
```

```
/dev/hdc:
```

```
multcount      = 0 (off)
I/O support    = 0 (default 16-bit)
unmaskirq      = 0 (off)
using_dma      = 0 (off)
keepsettings   = 0 (off)
nowerr         = 0 (off)
readonly       = 0 (off)
readahead      = 8 (on)
geometry       = 2484/16/63, sectors = 2503872, start = 0
```

② 显示硬盘的柱面、磁头、扇区等参数:

```
# hdparm -g /dev/hdc
```

```
/dev/hdc:
```

```
geometry       = 2484/16/63, sectors = 2503872, start = 0
                ↑   ↑   ↑           ↑           ↑
                柱面 磁头 扇区       总扇区数   起始扇区数
```

③ 评估硬盘的读取效率:

```
# hdparm -t /dev/hdc
```

```
/dev/hdc:
```

```
Timing buffered disk reads: 64 MB in 19.22 seconds =
3.33 MB/sec
```

④ 评估硬盘缓存的读取效率:

```
# hdparm -T /dev/hdc
```

```
/dev/hdc:
```

```
Timing buffer-cache reads: 128 MB in 2.20 seconds  
=58.18 MB/sec
```

⑤ 检测IDE硬盘的电源管理模式。

```
# hdparm -C /dev/hdc
```

```
/dev/hdc:
```

```
drive state is: standby ← 省电模式
```

⑥ 显示IDE硬盘本身内定的硬件规格:

```
# hdparm -i /dev/hdc
```

```
/dev/hdc:
```

```
Model=QUANTUM FIREBALL1280A, FwRev=A63.0E00, SerialNo=822611390974  
Config={ HardSect NotMFM HdSw>15uSec Fixed DTR>10Mbs }  
RawCHS=2484/16/63, TrkSize=32256, SectSize=512, ECCbytes=4  
BuffType=3(DualPortCache), BuffSize=83kB, MaxMultSect=8,  
MultSect=off  
DblWordIO=no, maxPIO=2(fast), DMA=yes, maxDMA=2(fast)  
CurCHS=2484/16/63, CurSects=884998182, LBA=yes  
LBA CHS=621/64/63 Remapping, LBA=yes, LBAsects=2503872  
tDMA={min:120,rec:120}, DMA modes: sword0 sword1 sword2  
mword0 mword1 *mword2  
IORDY=on/off, tPIO={min:333,w/IORDY:120}, PIO modes:  
mode3 mode4
```

⑦ 直接检测硬盘的硬件规格:

```
# hdparm -I /dev/hdc
```

```
/dev/hdc:
```

```
Model=, FwRev=, SerialNo=
```

```
Config={ }
```

```
RawCHS=0/0/0, TrkSize=0, SectSize=0, ECCbytes=0
```

```
BuffType=0(?), BuffSize=0kB, MaxMultSect=0
```

```
DblWordIO=no, maxPIO=0(slow), DMA=no
```

```
(maybe): CurCHS=0/0/0, CurSects=0, LBA=no
```

⑧ 查询并启动IDE硬盘的32位I/O模式:

```
# hdparm -c /dev/hdc
```

← 查询当前32位I/O模式:

```
/dev/hdc:
```

```
I/O support = 0 (default 16-bit) ← 32位I/O模式是关闭的
```

```
# hdparm -c 1 /dev/hdc
```

← 启动32位I/O模式

```
/dev/hdc:
```

```
setting 32-bit I/O support flag to 1
```

```
I/O support = 1 (32-bit) ← 32位I/O模式已经启动
```

⑨ 查询并启动IDE硬盘的DMA模式:

```
# hdparm -d /dev/hdc
```

← 查询当前的DMA模式

```
/dev/hdc:
```

```
using_dma = 0 (off) ← DMA模式是关闭的
```

```
# hdparm -d 1 /dev/hdc
```

← 启动DMA模式

```
/dev/hdc:
```

```
setting using_dma to 1 (on)
```

```
using_dma = 1 (on) ← DMA模式已经启动
```

⑩ 查询并设置读取文件时，预先存入缓存的扇区数为16个区块：

```
# hdparm -a /dev/hdc
```

← 查询当前设置值

```
/dev/hdc:
```

```
readahead      = 8 (on)
```

← 预先存入扇区数为8

```
# hdparm -a 16 /dev/hdc
```

← 将预先存入扇区数设成16

```
/dev/hdc:
```

```
setting fs readahead to 16
```

```
readahead      = 16 (on)
```

← 设置成功

⑪ 查询并设置硬盘多重扇区存取的扇区数为8，以提高硬盘的存取效率：

```
# hdparm -m /dev/hdc
```

← 查询当前设置值

```
/dev/hdc:
```

```
multcount      = 0 (off)
```

← 多重扇区存取的功能是关闭的

```
# hdparm -m 8 /dev/hdc
```

← 将多重扇区存取的扇区数设成8

```
/dev/hdc:
```

```
setting multcount to 8
```

```
multcount      = 8 (on)
```

← 设置成功

⑫ 查询并指定当硬盘存取时，是否允许其他中断要求同步执行：

```
# hdparm -u /dev/hdc
```

← 查询当前设置

```
/dev/hdc:
```

```
unmaskirq      = 0 (off)
```

← 此功能当前关闭

```
# hdparm -u 1 /dev/hdc
```

← 启动允许其他中断要求同步执行的功能

```
/dev/hdc:
```

```
setting unmaskirq to 1 (on)
```

```
unmaskirq      = 1 (on)
```

← 此功能已经启动

⑬ 启动硬盘写入缓存的功能:

```
# hdparm -W 1 /dev/hdc
```

```
/dev/hdc:
```

```
setting drive write-caching to 1 (on) ← 已经启动硬盘缓存
```

⑭ 设置硬盘用DMA2来当作传输模式:

```
# hdparm -X 34 /dev/hdc
```

```
/dev/hdc:
```

```
setting xfermode to 34 (multiword DMA mode2) ←
```

已经将硬盘设成DMA2传输

⑮ 设置硬盘经过5分钟的等待时间后, 则进入省电模式:

```
# hdparm -S 60 /dev/hdc
```

```
/dev/hdc:
```

```
setting standby to 60 (5 minutes)
```

⑯ 使IDE硬盘立即进入省电模式:

```
# hdparm -y /dev/hdc
```

```
/dev/hdc:
```

```
issuing standby command ← 进入省电模式
```

⑰ 使IDE硬盘进入睡眠模式:

```
# hdparm -Y /dev/hdc
```


```
/dev/hdc:
```

```
issuing sleep command ← 进入睡眠模式
```


⑧ 将内存缓冲区的数据写入硬盘，并清除缓冲区的数据：

```
# hdparm -f /dev/hdc
```

head

常用指数：

功能说明：输出文件内容的最前面部分。

语法：head [-qv] [-c <显示数目>] [-n <显示行数>] [--help] [--version] [文件...]

包名称：coreutils。

相关命令：cat, cut, tac, tail。

补充说明：head命令会读取所给予的文件的内容，并将其内容最前的部分输出到标准输出设备。若不指定任何文件名称，或是所给予的文件名为“-”，则head命令会从标准输入设备读取数据。默认读取的行数是最前10行，假设所给予的文件超过一个以上，head会在列出的文件内容前面加上符号“==>”和“<==”，两个符号之间则标出文件名称。

参数：

<p>-c <显示数目>或 --bytes=<显示数目></p>	<p>设置要显示多少数据，单位以Byte计算。可在指定的数字后面加上b、k和m，各表示512字节、1 024字节和1 048 576字节。例如，位设成20，表示列出20个字节的数据，若是2m，则是列出2 MB大小的数据</p>
<p>-n <显示行数>或 --lines=<显示行数></p>	<p>设置要显示多少行的内容。如显示行数设为5，head命令会显示文件最前5行的内容。我们可在显示行数前面加上“+”号，表示要从该指定行数开始列出文件内容，例如显示行数设成“+10”，表示从该文件内容的第10行开始列出，直到文件结束为止</p>

-q、--quiet或--silent	不显示文件名称
-v或--verbose	显示文件名称
--help	显示帮助
--version	显示版本信息

范 例:

① 列出textfile文字文件最前面的内容:

```
# head textfile
```

← 默认会列出最前面10行的内容

② 列出textfile文字文件最前面15行的内容:

```
# head -n 15 textfile
```

③ 列出textfile文字文件最前面500字节的内容:

```
# head -c 500 textfile
```

④ 列出textfile文字文件最前面512字节的内容:

```
# head -c 1b textfile
```

↑
“b”表示512字节


⑤ 列出当前目录下所有副文件名为“.txt”文件最前面的内容:

```
# head *.txt
```

⑥ 列出当前目录下所有文件名称以file起始的文件最前面内容, 列出时不显示每个文件的名称:

```
# head -q file*
```

help

常用指数: 

功能说明: 显示shell内建命令的说明。

语 法: help [shell内建命令]

包 名 称: bash。

相关命令: info, man。

补充说明: 显示shell内建命令的说明, 或列出全部的shell内建命令。

参 数: 若help后面不加上命令参数, 则列出全部的shell内建命令。

范 例:

① 显示shell所有的内建命令:

```
# help
```


② 显示fg命令的说明:

```
# help fg
```

```
fg: fg [job_spec]
```

```
Place JOB_SPEC in the foreground, and make it the current job.  
If JOB_SPEC is not present, the shell's notion of the  
current job is used.
```

history

常用指数: 

功能说明: 列出之前用过的命令。

h

语 法: history [N] [-arw] [文件名]

包 名 称: bash。

补充说明: history会列出所有使用过的命令并加以编号。这些信息默认存储在每个使用者根目录的.bash_history文件中，默认可存储1 000条命令记录。

若要执行history中记录的命令，请输入“!”加上命令编号即可。

参 数:

N	列出最近N次所输入的命令
-a	在history文件中加上记录
-r	仅读取history文件，不会加上记录
-w	覆盖原有的history文件


范 例:

从最近10次所输入的命令中，选出要执行的命令：

```
# history 10          ← 列出最近10次的命令
995 ls -la | more
996 mount -t msdos /dev/hda4 /mnt/dosdir
997 cd /mnt/dos622/doszip
998 cp doc /root/temp
999 umount /mnt/dosdir
1000 logout
1001 smbclient //deckard/temp -N
1002 cls
1003 ls
1004 history 10

# !996                ← 执行在history中编号996的命令
mount -t msdos /dev/hda4 /mnt/dosdir
```

host

常用指数: 

功能说明: DNS查询工具。

语法: host [-aCdrTv] [-c <class>] [-N <点数>] [-R <次数>] [-t <类型>] [-W <秒数>] [查询的名称] [DNS服务器]

包名称: bind-utils。

相关命令: hostname。

补充说明: 客户端DNS正查与反查工具，使用者可以由域名查出所对应的IP，或是由IP反查出所对应的域名。

参数:

-a	功能与使用“-v”参数相同
-c <class>	指定non-IN的数据
-C	直接跟权威 (authoritative) DNS服务器查询SOA记录
-d	与参数“-v”相同
-N <点数>	自动于网址列后补“.”的数目，默认为1
-r	取消递归查询
-R <次数>	重试次数
-t <类型>	指定查询类型。有CNAME、NS、SOA、SIG、KEY等
-T	开启TCP/IP模式
-v	详细输出模式
-W <秒数>	指定要等待的秒数

范 例:

① 直接跟openfind.com.tw的权威DNS服务器查询SOA记录:

```
# host -C openfind.com.tw
Nameserver dns.openfind.com.tw:
    openfind.com.tw      SOA      openfind.com.tw.
    openadm.openfind.com.tw.
    20030517 3600 900 3600000 3600
Nameserver openfind.com.tw:
    openfind.com.tw      SOA      openfind.com.tw.
    openadm.openfind.com.tw.
    20030517 3600 900 3600000 3600
```

② 查询openfind.com.tw的mx记录:

```
# host -t mx openfind.com.tw
openfind.com.tw mail is handled by 100
dns.openfind.com.tw.
openfind.com.tw mail is handled by 0
mail.office.openfind.com.tw.
```

由这两台邮件主机负责该域名的信件收发

③ 指定向168.95.1.1查询www.yahoo.com.tw的IP:

```
# host www.yahoo.com.tw 168.95.1.1
Using domain server:
Name: 168.95.1.1
Address: 168.95.1.1#53
Aliases:

www.yahoo.com.tw is an alias for rc.yahoo.com.
rc.yahoo.com is an alias for w1.rc.vip.scd.yahoo.com.
w1.rc.vip.scd.yahoo.com has address 66.218.71.198
```

查询到的结果

hostname

常用指数:



功能说明: 查询及设置主机名称。

语法: hostname [-adhinsvVy] [-F <文件名称>]

包名称: net-tools。

相关命令: netconfig, host。

补充说明: 当开机时, 系统会读取/etc/hosts设置文件, 由此设置主机的IP及对应的全名 (FQDN, Fully Qualified Domain Name) 与别名。而hostname命令可用来查询当前执行主机的主机名称、全名、前置名称、域名、别名、FQDN、IP等。

现将相关的名称解释如下:

linux.flag.com.tw ← 全名 (FQDN)

↑ ↑

前置名称 域名

参数:

-a或--alias	查询主机名称的别名
-d或--domain	查询主机的域名
-F <文件名称>或--file <文件名称>	将主机名称设成文件中指定的名称
-f、--fqdn或--long	查询主机的全名
-h或--help	显示帮助
-i或--ip-address	查询主机的IP地址
-n或--node	查询DECnet网络的节点名称

h

-s或--short	查询主机的前置名称
-v或--verbose	显示命令执行过程
-V或--version	显示版本信息
-y、--yp或--nis	查询NIS域名

范 例:

① 查询主机别名:

```
# hostname -a  
linux
```

② 查询主机的IP地址:

```
# hostname -i  
203.74.205.36
```

③ 查询主机的域名:

```
# hostname -d  
flag.com.tw
```

④ 查询主机的前置名称:

```
# hostname -s  
irchen
```

⑤ 查询主机的全名:

```
# hostname -f  
irchen.flag.com.tw ← 全名等于前置名称加域名
```



⑥ 查询主机的别名，并显示执行的过程：

```
# hostname -av
gethostname()='linux'
Resolving 'linux' ...
Result: h_name='irchen.flag.com.tw'
Result: h_aliases='linux'
Result: h_aliases='mylinux'
Result: h_addr_list='203.74.205.36'
linux mylinux
```

命令执行的过程

← 有2个别名

⑦ 将主机名称设成assignhost文件中指定的名称：

```
# cat assignhost
mylinux
# hostname -F assignhost
# hostname
mylinux
```

← 列出assignhost文件的内容

← 指定主机名称为mylinux

← 查询主机名称

← 主机名称已经改成mylinux

httpd (http daemon)

常用指数：



功能说明： Apache HTTP服务器程序。

语法： httpd [-X] [-c <httpd 命令>] [-C <httpd 命令>] [-d <服务器根目录>] [-D <设置文件参数>] [-f <设置文件>] 或 httpd [-hLLStTvV]

包名称： httpd (Red Hat Linux/Fedora Core), apache2 (Mandrake Linux)。

相关命令： apachectl, lynx。

补充说明： httpd为Apache HTTP服务器程序（在Mandrake Linux中，此程序称为httpd2）。直接执行程序即可启动服务器的服务。若要停止服务器的服务，则必须以kill命令将httpd的主程序删除。

若不加任何参数，则httpd会去读取默认的设置文件/etc/httpd/conf/httpd.conf（Mandrake Linux则为httpd2.conf），根据文件中的设置来启动HTTP服务器。

参 数：

-c <httpd 命令>	在读取设置文件前，先执行选项中的命令
-C <httpd 命令>	在读取设置文件后，再执行选项中的命令
-d <服务器根目录>	指定服务器的根目录。默认目录为/etc/httpd（Mandrake Linux则为/etc/httpd/2.0）
-D <设置文件参数>	指定要传入设置文件（用于<IfDefine>段）的参数
-f <设置文件>	指定设置文件，而不使用默认的设置文件
-h	显示帮助
-l	显示服务器编译时所包含的模组
-L	显示httpd命令的说明
-S	显示设置文件中的设置
-t	测试设置文件的语法是否正确
-T	除了根目录设置文件不检查之外，测试设置文件的语法是否正确
-v	显示版本信息
-V	显示版本信息以及其建立时的环境
-X	以单一程序的方式来启动服务器。通常仅在排错时使用

范 例：

① 启动HTTP服务器：

```
# ps -aux | grep httpd
```

← 查看是否有httpd的执行程序

```
# httpd
```

← 启动HTTP服务器

```
# ps -aux | grep httpd
```

← 再查看一次httpd的执行程序

```
root 13724 0.0 0.0 21108 76 ? S Jun1 0:00 /usr/sbin/httpd
apache 13727 0.0 1.3 21436 1552 ? S Jun17 0:00 [httpd]
apache 13728 0.0 1.7 21396 2044 ? S Jun17 0:00 [httpd]
apache 13729 0.0 0.8 21440 960 ? S Jun17 0:00 [httpd]
apache 13730 0.0 1.2 21388 1508 ? S Jun17 0:00 [httpd]
apache 13731 0.0 2.0 21432 2436 ? S Jun17 0:00 [httpd]
apache 13732 0.0 1.7 21840 2044 ? S Jun17 0:00 [httpd]
apache 13733 0.0 1.1 21360 1396 ? S Jun17 0:00 [httpd]
apache 13734 0.0 1.7 21412 2080 ? S Jun17 0:00 [httpd]
```

② 启动httpd时使用指定的设置文件:

```
# httpd -f httpd.conf
```

③ 测试设置文件的语法是否正确:

```
# httpd -t
Syntax OK
```

④ 显示httpd编译时所包含的模组:

```
# httpd -l
Compiled in modules:
  core.c
  prefork.c
  http_core.c
  mod_so.c
```

H



hwclock (hardware clock)

常用指数:



功能说明: 显示与设置硬件时钟。

语法: hwclock [--adjust] [--debug] [--directisa] [--hctosys] [--set --date=<日期与时间>] [--show] [--systohc] [--test] [--utc] [--version]

包名称: util-linux。

相关命令: clock, date, rdate, timeconfig。

补充说明: 在Linux中有硬件时钟 (RTC, Real Time Clock) 与系统时钟 (System Clock) 两种时钟。硬件时钟是指主机板上的时钟设备, 也就是通常可在BIOS画面设置的时钟。系统时钟则是指kernel中的时钟。当Linux启动时, 系统时钟会去读取硬件时钟的设置, 之后系统时钟即独立运作。所有Linux相关命令与函数都是读取系统时钟的设置。

参数:

--adjust	hwclock每次更改硬件时钟时, 都会记录在/etc/adjtime文件中。使用“--adjust”参数, 可使hwclock根据先前的记录来估算硬件时钟的偏差, 并用来校正当前的硬件时钟
--debug	显示hwclock执行时详细的信息
--directisa	hwclock默认从/dev/rtc设备来存取硬件时钟。若无法存取时, 可用此参数直接以I/O命令来存取硬件时钟
--hctosys	将系统时钟调整为与当前的硬件时钟一致
--set -date =<日期与时间>	设置硬件时钟。选项的格式为“月/日/公元年 时:分:秒”
--show	显示硬件时钟的时间与日期
--systohc	将硬件时钟调整为与当前的系统时钟一致
--test	仅测试程序, 而不会实际更改硬件时钟
--utc	以格林威治时间显示
--version	显示版本信息

范 例:

① 显示硬件时钟的时间与日期:

```
# hwclock
公元2003年06月19日 (周四) 15时07分18秒 -1.022002 seconds
```

② 以格林威治时间显示硬件时钟的时间:

```
# hwclock ← 先以默认的方式显示硬件时钟的时间
公元2003年06月19日 (周四) 15时07分30秒 -1.022002 seconds
# hwclock --utc ← 以格林威治时间显示
公元2003年06月19日 (周四) 23时07分50秒 -0.263501 seconds
```

③ 设置硬件时钟为2003年6月19号23点59分0秒:

```
# hwclock --set --date="6/19/2003 23:59:00" ← 设置硬件时钟
```

④ 将系统时钟调整与当前的硬件时钟一致:

```
# hwclock --hctosys
```

⑤ 将硬件时钟调整与当前的系统时钟一致:

```
# hwclock --systohc
```

⑥ 校正当前的硬件时钟, 并显示详细的信息:

```
# hwclock --adjust --debug
hwclock from util-linux-2.11y
Using /dev/rtc interface to clock.
Last drift adjustment done at 1056007535 seconds after 1969
...
Need to insert 0 seconds and refer time back 0.000676
seconds ago
Needed adjustment is less than one second, so not setting
clock. ← 不需任何校正
```

id

常用指数:



功能说明: 显示用户ID, 以及所属组ID。

语法: id [-gGnru] [--help] [--version] [用户名称]

包名称: coreutils。

相关命令: groupmod, usermod, whoami。

补充说明: id会显示用户以及所属组的实际ID与有效ID。若两个ID相同, 则仅显示实际ID。

若仅指定用户名称, 而不加任何参数, 则显示用户ID, 以及所属组的ID。若不指定用户名称, 则显示当前用户的ID。

参 数:

-g 或 --group	显示用户所属组的ID
-G 或 --groups	显示用户所属附加组 (supplementary group) 的ID。一个用户除了原先建立账号的组之外, 也可指定多个附加组给同一个用户。使得该用户具有这些附加组的执行权限
-n 或 --name	显示用户、所属组或附加组的名称。必须与-g、-G或-u一起使用
-r 或 --real	显示实际ID。必须与-g、-G或-u一起使用
-u 或 --user	显示用户ID
--help	显示帮助
--version	显示版本信息

范 例:

① 显示当前用户及其所属组的ID:

```
# id
```

```
uid=503(tony) gid=503(tony) groups=503(tony)
```

↑
所属组ID及名称

↑
所属附加组ID及名称

② 显示用户alkane的ID:

```
# id -u alkane  
500
```

③ 显示用户alkane的名称:

```
# id -un alkane  
alkane
```

④ 显示用户alkane所属组的 ID:

```
# id -g alkane  
501
```

⑤ 显示用户alkane所属组的名称:

```
# id -gn alkane  
alkane
```

⑥ 显示用户alkane所属附加组的 ID:

```
# id -G alkane  
501 502
```

↑ 所属组ID

← 也会列出其所属组

← 所属附加组ID

⑦ 显示用户alkane所属附加组的名称:

```
# id -Gn alkane  
alkane sphenoid
```



ifconfig (interface configure) 常用指数:

功能说明: 显示或设置网络设备。

语 法: ifconfig [网络设备] 或

```
ifconfig [网络设备] [add <地址>] [del <地址>] [down]
[dstaddr <地址>] [hw <网络设备类型> <硬件地址>]
[io_addr <I/O 地址>] [irq <IRQ地址>] [media <网络媒介
类型>] [mem_start <内存地址>] [metric <数目>] [mtu <字
节>] [netmask <子网络掩码>] [tunnel <地址>] [up] [-
allmulti] [-arp] [-broadcast <地址>] [-pointopoint <地址>]
[-promisc] [IP 地址]
```

包 名 称: net-tools。

相关命令: netconfig, netstat, ping。

补充说明: ifconfig可设置网络设备（通常是网卡）的状态，或是显示当前的设置。常设置的项目有IP地址、子网络掩码以及硬件本身的IRQ与I/O等，其他项目通常保留默认值即可。若不加任何参数，则显示当前使用中网络设备的设置。若指定[网络设备] [IP地址]，则是设置该网络设备的IP地址。

参 数:

add <地址>	设置网络设备IPv6的IP地址
del <地址>	删除网络设备IPv6的IP地址
down	关闭指定的网络设备
dstaddr <地址>	设置点对点连接时，对应端的IP地址

hw<网络设备类型><硬件地址>	设置网络设备的类型与硬件地址。当前支援的类型有etherax25、RCnet与netrom。以网卡为例，通常制造商在出厂前就设好硬件地址，实际上很少有更改硬件地址的需要
io_addr <I/O地址>	设置网络设备的I/O地址。此项设置与IRQ情况类似，请参阅网络设备的说明手册
irq <IRQ地址>	设置网络设备的IRQ。以网卡为例，并不是所有网卡都能通过此参数来设置IRQ，有些网卡IRQ必须通过网卡上的跳线来设置，有些则需要用制造商所提供的程序来修改。相关信息请参阅网络设备的说明手册
Media <网络媒介类型>	设置网络设备的媒介类型。并不是所有网络设备都可以用此参数来设置，常见的选项有10base2、10baseT、AUI或auto
mem_start <内存地址>	设置网络设备在主内存所占用的起始地址，只有少部分网络设备需要设置此参数
metric <数目>	指定在计算包的转送次数（routing）时，所要加上的数目。负责路由的协议（例如，RIP）在计算最短路径时，会用到此项信息。通常是设为1
mtu <字节>	设置网络设备的MTU（最大传输单位）。单位为字节
netmask <子网络掩码>	设置网络设备的子网络掩码
tunnel <地址>	建立IPv4与IPv6之间的通道通信地址
up	启动指定的网络设备
-allmulti	决定网络设备是否要接收multicast的包。负号表示要接收，未加负号表示不接收
-arp	关闭或启动指定网络设备的ARP（用来取得网络设备硬件地址的通信协议）。负号表示关闭，未加负号表示启动
-broadcast <地址>	将要送往指定地址的包当成广播包来处理。负号表示关闭此设置，不加负号表示使用此设置
-pointopoint <地址>	与指定地址的网络设备建立直接连接，此模式具有保密功能。负号表示关闭此次连接，未加负号表示启动
-promisc	关闭或启动指定网络设备的promiscuous模式（就是不管包的目的地地址为何，接收所有的网络包，通常用于网络监控的程序）。负号表示关闭，未加负号表示启动
[IP 地址]	指定网络设备的IP地址
[网络设备]	指定网络设备的名称。例如，eth0是指第一块网卡

范 例:

① 显示第一块网卡当前的设置值:

```
# ifconfig eth0
```

```
eth0  Link encap:Ethernet  HWaddr 00:00:B4:55:DF:45
      inet  addr:203.74.205.36  Bcast:203.74.205.255
      Mask:255.255.255.0
      UP BROADCAST RUNNING  MTU:1500  Metric:1
      RX packets:6315  errors:0  dropped:0  overruns:0
      frame:0
      TX packets:2166  errors:0  dropped:0  overruns:0  car
      rier:0
      collisions:0 txqueuelen:100
      RX bytes:8127488 (7.7 Mb)TX bytes:1896998(1.8 Mb)
      Interrupt:19 Base address:0xa420
```

② 将第一块网卡的IP地址设为203.74.205.123:

```
# ifconfig eth0 203.74.205.123
```

③ 将第一块网卡的子网络掩码设为255.255.0.0:

```
# ifconfig eth0 netmask 255.255.0.0
```

④ 本机IP地址为203.74.205.38, 关闭第一块网卡后, 再重新启动:

```
# ifconfig eth0 down
```

← 关闭第一块网卡

```
# ping 203.74.205.38
```

```
connect: Network is unreachable
```

← 网络当前是不通的情况

```
# ifconfig eth0 up
```

← 重新启动第一块网卡

```
# ping 203.74.205.38
```

```
PING 203.74.205.38 (203.74.205.38) 56(84) bytes of data.
```

```
64 bytes from 203.74.205.38: icmp_seq=0 ttl=255 time=1.7 ms
```

网络已经恢复连接

⑤ 将第一块网卡的`最大传输单位`设为1 200个字节:

```
# ifconfig eth0 mtu 1200
```

⑥ 将第一块网卡设成允许接收multicast的包:

```
# ifconfig eth0 allmulti
```

⑦ 启动第一块网卡的`网络监控模式`:

```
# ifconfig eth0 promisc
```

indent

常用指数:



功能说明: 调整 C 源代码文件的格式。

语法: indent [参数] [来源文件]或indent [参数] [来源文件] [-o目的文件]

包名称: indent。

补充说明: indent可辨识C的源代码文件,并加以格式化,以方便程序员阅读。indent同时也具有语法检查的功能。

使用时若不指定目的文件(第一种语法),indent会先备份来源文件,然后再直接修改来源文件。备份文件的文件名为“来源文件~”。例如,来源文件为replicant.h,则备份文件的文件名为replicant.h~。

参 数:

-bad或--blank-lines-after-declarations	在声明后加上空白行
-bap或--blank-lines-after-procedures	在程序后加上空白行
-bbb或--blank-lines-after-block-comments	在注释后加上空白行
-bc或--blank-lines-after-commas	在声明中, 若出现逗号即换行
-bl或--braces-after-if-line	if (或是else、for等) 与后面执行部分的“{”不同行, 且“}”自成一。例如: <pre>if (x > 0) ← 不同行 { x++; } ← 自成一</pre>
-bli<缩进格数>或 --brace-indent<缩进格数>	设置{ }缩进的格数
-br或--braces-on-if-line	if (或是else、for等) 与后面执行部分的“{”同行。且“}”自成一。例如: <pre>if (x > 0) { ← 同行 x++; } ← 自成一</pre>
-bs或--blank-before-sizeof	在sizeof之后空一格
-c<行数>或--comment-indentation<行数>	将注释置于程序代码右侧指定的行
-cd<行数>或--declaration-comment-column<行数>	将注释置于声明右侧指定的行
-cdb或--comment-delimiters-on-blank-lines	注释符号自成一。例如: <pre>/* Comments here. */</pre>
-ce或--cuddle-else	将else置于“}” (if执行部分的结尾) 之后

-ci<缩进格数>或--continuation-indentation<缩进格数>	语句过长而换行时，指定换行后缩进的格数
-cli<缩进格数>或--case-indentation<缩进格数>	使用case时，switch缩进的格数
-cp<行数> 或 --else-endif-column<行数>	将注释置于else与elseif语句右侧指定的行
-cs或--space-after-cast	在cast之后空一格
-d<缩进格数>或--line-comments-indentation<缩进格数>	针对不是放在程序代码右侧的注释设置其缩进格数
-di<行数>或--declaration-indentation<行数>	将声明部分的变量置于指定的行
-fcl或--format-first-column-comments	针对放在每行最前端的注释设置其格式。请注意，-fcl的“1”是阿拉伯数字的“1”，不是小写的英文字母“l”
-fca或--format-all-comments	设置所有注释的格式
-gnu或--gnu-style	指定使用GNU的格式，此为默认值。GNU的格式设置如下： -nbad -bap -nbbb -nbc -bl -bli2 -c33 -cd33 -ncdb -nce -cli0 -cp1 -di0 -nfcl -nfca -i2 -ip5 -lp -pcs -psl -nsc -nsob -nss -ts8
-i<格数>或--indent-level <格数>	设置缩进的格数
-ip<格数>或--parameter-indentation<格数>	设置参数的缩进格数
-kr或--k-and-r-style	指定使用Kernighan & Ritchie的格式。Kernighan & Ritchie的格式设置如下： -nbad -bap -nbbb -nbc -br -c33 -cd33 -ncdb -ce -ci4 -cli0 -cp33 -d0 -di1 -nfcl -nfca -i4 -ip0 -l75 -lp -npcs -npsl -nsc -nsob -nss -ts8
-lp或--continue-at-parentheses	语句过长而换行，且语句中包含了括号（大、中、小括号）时，将括号中的每行起始栏位内容垂直对齐排列
-nbad或--no-blank-lines-after-declarations	在声明部分后不要加上空白行

-nbap或--no-blank-lines-after-procedures	在程序后不要加上空白行
-nbbb或--no-blank-lines-after-block-comments	在注释部分后不要加上空白行
-nbc或--no-blank-lines-after-commas	在声明部分中，即使出现逗号，仍旧不要换行
-ncdb或--no-comment-delimiters-on-blank-lines	注释符号不要自成一。例如： /* Comments here */
-nce或--dont-cuddle-else	不要将else置于“}”（if执行部分的结尾）之后
-ncs或--no-space-after-casts	不要在cast之后空一格
-nfc1或--dont-format-first-column-comments	不要格式化放在每行最前端的注释
-nfca或--dont-format-comments	不要格式化任何的注释
-nip或--no-parameter-indentation	参数不要缩进
-nlp或--dont-line-up-parentheses	语句过长而换行，且语句中包含了括号（大、中、小括号）时，不用将括号中的每行起始列垂直对齐排列
-npcs或--no-space-after-function-call-names	在调用的函数名称之后，不要加上空格
-npro或--ignore-profile	不要读取indent的设置文件“.indent.pro”
-npsl或--dont-break-procedure-type	程序类型与程序名称放在同一行
-nsc或--dont-star-comments	注释左侧不要加上星号（*）
-nsob或--leave-optional-blank-lines	不用处理多余的空白行
-nss或--dont-space-special-semicolon	若for或while部分（含条件与执行语句）仅有一行时，在分号前不用加上空格
-nv或--no-verbosity	不显示详细的信息
-orig或--original	使用Berkeley的格式。

	Berkeley的格式设置如下： -nbap -nbad -nbbb -bc -br -c33 -cd33 -cdb -ce -ci4 -cli0 -cp33 -d4 -di16 -fc1 -fca -i4 -ip4 -l75 -lp -npcs -psl -sc -nsob -nss -ts8
-pcs或--space-after-procedure-calls	在调用的函数名称与“(”之间加上空格
-psl或--procnames-start-lines	程序类型置于程序名称的前一行
-sc或--start-left-side-of-comments	在每行注释左侧加上星号(*)
-sob或--swallow-optional-blank-lines	删除多余的空白行
-ss或--space-special-semicolon	若for或while部分(含条件与执行语句)仅有一行时,在分号前加上空格
-st或--standard-output	将结果显示在标准输出设备
-T	资料类型名称缩进
-ts<格数>或--tab-size<格数>	设置tab的长度
-v或--verbose	执行时显示详细的信息
--version	显示版本信息

范 例:

① 以默认的GNU格式来处理source.c,并将结果存储至modified.c:

```
# indent source.c -o modified.c
```

② 以Berkeley的格式来处理source.c文件,最后将结果输出到标准输出设备:

```
# indent -st -orig source.c
```



- ③ 以Kernighan & Ritchie的格式来处理source.c文件，由于不指定输出文件文件名，indent会将来源文件备份成source.c~，转换后的文件则存成source.c：

```
# indent -kr source.c
```

- ④ 以默认的GNU格式来处理source.c文件，将结果存储至modified.c，格式转换时，删除多余的空白行：

```
# indent -sob source.c -o modified.c
```

- ⑤ 以默认的GNU格式来处理source.c文件，将结果存储至modified.c，格式转换时，注释符号自成一行：

```
# indent -cdb source.c -o modified.c
```

- ⑥ 以默认的GNU格式来处理source.c文件，将结果存储至modified.c，格式转换时，将注释置于程序代码右侧第20列位处：

```
# indent -c 20 source.c -o modified.c
```

- ⑦ 以默认的GNU格式来处理source.c文件，将结果存储至modified.c，格式转换时，将else置于“}”之后：

```
# indent -ce source.c -o modified.c
```

- ⑧ 以默认的GNU格式来处理source.c文件，将结果存储至modified.c，格式转换时，将程序类型与程序名称放在同一行：

```
# indent -npsl source.c -o modified.c
```

- ⑨ 以默认的GNU格式来处理source.c文件，将结果存储至modified.c，格式转换时，将{}缩进3格：

```
# indent -bli 3 source.c -o modified.c
```


- ⑩ 以默认的GNU格式来处理source.c文件，将结果存储至modified.c，格式转换时，若在声明部分中出现逗号即换行：

```
# indent -bc source.c -o modified.c
```

- ⑪ 以默认的GNU格式来处理source.c文件，将结果存储至modified.c，格式转换时，在声明部分后加上空白行：

```
# indent -bad source.c -o modified.c
```

info (infomation)

常用指数：

功能说明：显示说明。

语 法： info [--directory=<目录>] [--dribble=<文件>] [--file=<文件>] [--help] [--index-search=<字符串>] [--node=<主题>] [--output=<文件>] [--restore=<文件>] [--show-options] [--subnodes] [--usage] [--version]

包 名 称： info。

相关命令： help, man。

补充说明： info为读取info说明文件的程序。在说明文件中，可再分为许多主题。用户可以循环阅读主题，或通过主题选单，选择指定的主题。

参 数：

--directory=<目录>	将目录加入info搜寻的路径中
--dribble=<文件>	将执行info过程所输入的按键，记载到指定的文件中
--file=<文件>	指定要读取的info文件

--index-search=<字符串>	根据选项中的字符串，在索引中寻找参照的主题，并显示该主题
--node=<主题>	指定info文件中的主题。若主题包含了空白字符，则必须在主题的前后加上双引号（"）
--output=<文件>	将所选主题输出指定的文件
--restore=<文件>	开启说明文件之后，先执行文件所记载的按键
--show-options或--usage	显示所有info说明文件
--subnodes	输出所有主题
--help	显示帮助
--version	显示版本信息

范 例：

① 执行info，可利用稍后所介绍的操作命令来查看说明文件：

info

此主题的文件名称 当前主题的名称

```
File: dir      Node: Top      This is the top of the INFO tree

This (the Directory node) gives a menu of major topics.
Typing "q" exits, "?" lists all Info commands, "d" returns here,
"h" gives a primer for first-timers,
"mEmacs<Return>" visits the Emacs topic, etc.

In Emacs, you can click mouse button 2 on a menu item or cross reference
to select it.

* Menu:
Texinfo documentation system
* Texinfo: (texinfo).      The GNU documentation format.
* info standalone: (info-std).      Read Info documents without Emacs.
* infokey: (info-std)Invoking infokey.      Compile Info customizations.
* install-info: (texinfo)Invoking install-info. Update info/dir entries.
* makeinfo: (texinfo)Invoking makeinfo.      Translate Texinfo source.
* texi2dvi: (texinfo)Format with texi2dvi.      Print Texinfo documents.
* texindex: (texinfo)Format with tex/texindex.      Sort Texinfo index files.

Miscellaneous
* Am-utils: (am-utils).      The Amd automounter suite of utilities
* As: (as).      The GNU assembler.
* Bfd: (bfd).      The Binary File Descriptor library.
* Binutils: (binutils).      The GNU binary utilities.

-----Info: (dir)Top, 21/37 行 -----
欢迎使用 Info 4.7 版。输入 ? 以获得帮助, m 将得到菜单。
```

可供选择的主题

此主题的总行数

目前画面在此主题的显示比率
(在此例中，是在此主题的最顶端)

- ② 指定要读取的info文件为info.info，主题为“Creating an Info File”，并将主题内容输出到info.txt文件中：

```
# info --file=info.info --node="Creating an Info File"
                                --output=info.txt
```

```
info: 写入节点 (info.info.gz)Creating an Info File 中...
info: 完成.
```

- ③ 指定要读取的info文件为info.info，并将执行过程所输入的按键记录到keystrokes.txt文件中：

```
# info --file=info.info --dribble=keystrokes.txt
```

操作命令：以info开启说明文件后，常用的操作命令如下：

b	移到此主题开始处
d	回到主题目录画面，可供进一步选择主题
e	移到此主题结束处
f	移到相关的主题
g	移到指定的主题
h	教您如何使用info
i	寻找当前文件的特定主题
l	移到前一个主题
m	输入所要查看的主题
n	下一个主题
p	上一个主题
q	结束程序
s	往前寻找此主题的特定字符串

1	回到第一个（最上面）的主题
?	显示所有的命令
0	回到上一个主题选单
1 ~ 9	选择当前选单中第1到9号的主题
↑	将游标上移一行
↓	将游标下移一行
空白键或 Page Up	切换至下一页
Del 或 Page Down	切换至上一页
Ctrl + L	重新显示画面
Tab	切换到当前选单的下一个主题索引上
Esc - s	往回找此主题的特定字符串
Enter	进入游标当前所在的主题

inncheck (inn check)

常用指数：



功能说明：检查inn相关设置文件的语法是否正确。

语法：inncheck [-afv] [-noperm] [-pedantic] [-perm] [文件]

包名称：inn。

相关命令：archive, ctlinnd。

补充说明：inncheck会检查inn设置文件的语法是否正确。默认会检查的文件为active、controlctl、expirectl、incoming.conf、inn.conf、moderators、newsfeeds、overview.fmt、nnrp.access、nntpsendctl和passwd.nntp。

inncheck只负责检查语法，并不会实际修改这些设置文件。执行此命令前，请先切换到/usr/lib/news/bin目录下。

参 数:

-a	不管是否出现[文件]参数，仍然检查全部的设置文件。若[文件]参数中指定要检查有别于默认的文件，则检查参数中指定的文件，其他未指定部分仍检查默认文件
-f	若发现设置文件有误，则列出可修正错误的命令。此参数必须配合“-perm”一起使用
-v	执行时显示详细的信息
--noperm	不检查文件的权限设置
-pedantic	以较为严格的方式来检查。以此参数检查时，列出的设置不见得一定有错误，只是警告该设置可能不正确
-perm	检查文件权限设置是否有误
[文件]	若不指定此参数，则inncheck会检查所有默认的文件。若设置了此参数，则inncheck仅检查指定的文件。若要检查非默认的文件，请用[文件=实际要检查的文件]方式来指定。例如，inn.conf=/testconf/inn.conf

范 例:

① 以默认方式检查inn设置文件是否正确:

```
# ./inncheck
```

```
/var/lib/news/active:0: mode 644, should be 664
/etc/news/inn.conf:0: owned by root, should be news
/etc/news/inn.conf:0: in group root, should be news
/etc/news/inn.conf:97: fromhost isn' t a valid FQDN
/etc/news/inn.conf:99: modmailer has bad address
```

发现
有这
些问
题

② 以较为严格的方式检查全部的inn设置文件，执行时显示详细的信息:

```
# ./inncheck -v -pedantic
```

```
Looking at /var/lib/news/active...
/var/lib/news/active:0: mode 644, should be 664
Looking at /etc/news/controlctl...
```



```

Looking at /etc/news/expirectl...
Looking at /etc/news/incoming.conf...
Looking at /etc/news/inn.conf...
/etc/news/inn.conf:0: owned by root, should be news
/etc/news/inn.conf:0: in group root, should be news
/etc/news/inn.conf:97: fromhost isn't a valid FQDN
/etc/news/inn.conf:99: modmailer has bad address
Looking at /etc/news/moderators...
Looking at /etc/news/newsfeeds...
ME, controlchan!, done.
Looking at /etc/news/nntpsendctl...
Looking at /etc/news/overview.fmt...
Looking at /etc/news/passwd.nntp...
Looking at /etc/news/readers.conf...

```

列出了检查过程

③ 检查所有的inn设置文件，但不检查默认的incoming.conf，而是检查/tmp目录下的incoming.conf：

```
# ./inncheck -a incoming.conf=/tmp/incoming.conf
```

④ 检查相关的文件权限设置是否有误，若发现错误，则列出可修正的命令：

```
# ./inncheck -perm -f
```

```
# /usr/lib/news/bin:0: owned by root, should be news
```

发现的错误

```
chown news /usr/lib/news/bin
```

建议修正的命令

...



insmod (install module)

常用指数:



功能说明: 载入模块。

语法: insmod [-fkmpsVX] [-o <模块名称>] [模块文件] [符号名称=符号值]

包名称: modutils。

相关命令: depmod, lsmod, modinfo, modprobe, rmmod。

补充说明: Linux有许多功能是通过模块的方式, 在需要时才载入kernel。如此可使kernel较为精简, 进而提高效率, 以及保有较大的弹性。这类可载入的模块, 通常是设备驱动程序, 例如光驱、声卡或各种文件系统的驱动程序等等。insmod可将模块载入执行中的kernel。载入时, insmod会建立kernel符号表 (symbol table) 与模块的对应关系。

参数:

-f	不检查当前kernel版本与模块编译时的kernel版本是否一致, 强制将模块载入。除非确定不同版本之间不会有相容问题, 否则建议不使用此参数
-k	将模块设置为自动卸除。当模块闲置时, kerneld (Kernel Daemon, 可视需求自动载入或卸除模块的服务程序) 会自动卸除模块
-m	当载入模块时, 一并输出模块的载入信息
-o <模块名称>	指定模块的名称, 可使用模块文件 (.o) 的文件名。编译好的模块文件默认存放在/lib/modules下, 以kernel版本编号为名称的目录中
-p	测试模块是否能正确地载入kernel
-s	将所有信息记录在系统记录文件中, 而不要输出到标准输出设备
-v	执行时显示详细的信息
-x	不要输出模块的外部符号
-X	输出模块所有的外部符号, 此为默认值

范 例:

① 载入OS/2的HPFS文件系统的驱动程序:

```
# insmod hpfs
```

② 载入Windows NT的NTFS文件系统的驱动程序, 载入时并将此模块设为可自动卸除的模块:

```
# insmod -k ntfs
```

③ 测试能否正确地载入nfs.o模块:

```
# insmod -p nfs
```

```
Using /lib/modules/2.4.20-8/kernel/fs/nfs/nfs.o
```

←
可以正确载入

iptables

常用指数:



功能说明: 包过滤功能和 NAT 的管理工具。

语 法: iptables [-h] [-A <链名>] [-D <链名>] [-E <链名>] [-F <链名>] [-I <链名>] [-L <链名>] [-N <链名>] [-P <链名>] [-t <表格名称>] [-X <链名>] [过滤规则] [选项]

包 名 称: iptables。

补充说明: iptables命令主要是用来设置、维护及检查Linux核心中IP包过滤原则的表格。



参 数:

-A <链名>	将规则附加到链的最后
-D <链名>	删除链中规则
-E <链名>	更改链名
-F <链名>	清除表格中的规则
-h	列出简短的使用说明
-I <链名>	将规则加到链的最前面
-L <链名>	列出表格所有的规则
-N <链名>	建立新的链
-P <链名>	设置链中默认的原则
-t <表格名称>	指定表格名称
-X <链名>	此参数可以删除自定义的链，但该链必须为空的

范 例:

① 列出filter表格的所有规则:

```
# iptables -t filter -L
Chain INPUT (policy ACCEPT)
target      prot opt source
destination

Chain FORWARD (policy ACCEPT)
target      prot opt source
destination

Chain OUTPUT (policy ACCEPT)
target      prot opt source
destination
```

← 列出filter表格的所有规则

默认的3个
链都是空的

② 建立一条名称为bad-domain的新链:

```
# iptables -N bad-domain
# iptables -L
Chain INPUT (policy ACCEPT)
target      prot opt source      destination

Chain FORWARD (policy ACCEPT)
target      prot opt source      destination

Chain OUTPUT (policy ACCEPT)
target      prot opt source      destination

Chain bad-domain (0 references)
target      prot opt source      destination
```

← 建立新链

← 列出所有的链

} 刚才新增的链, 当前并没有任何规则

③ 新增过滤规则:

```
# iptables -A bad-domain -j DROP
# iptables -L
...
Chain bad-domain (0 references)
target      prot opt source      destination
DROP        all  --  anywhere    anywhere
```

← 于bad-domain链中新增一条过滤规则

← 刚才新增的规则

④ 清空及删除自定义的bad-domain链:

```
# iptables -X bad-domain
iptables: Chain is not empty
# iptables -F bad-domain
# iptables -X bad-domain
```


← 删除自定义的bad-domain链

← 若链中含有规则, 就会有错误信息

← 先清空bad-domain链中的规则

← 这时删除就不会有错误信息

ispell (interactive spelling check)

常用指数: 

功能说明: 拼写检查程序。



语 法: ispell [-aAbBCImMnPSstx] [-d <字典文件>] [-p <字典文件>]
[-w <非字母字符>] [-W <字符串长度>] [要检查的文件]

包 名 称: aspell。

相关命令: look, spell。

补充说明: ispell默认会使用/usr/lib/ispell/english.hash字典文件来检查文本文件。若在检查的文件中找到字典文件没有的词汇, ispell会建议可使用的词汇, 或是将新的词汇加入个人字典文件。

参 数:

-a	当其他程序将输出送至ispell时 (如通过管线), 必须使用此参数
-A	读取到“&Include File&”字符串时, 就去检查字符串后所指定文件的内容。检查完后, 再回来继续检查原来的文件
-b	产生备份文件, 文件名为.bak, 或在原始文件名后加上“~”
-B	检查连字错误。连字错误就是两个英文字之间没有加上空白字符, 例如, “are the”打成“arethe”。此为默认值
-C	不检查连字错误
-d <字典文件>	指定字典文件
-l	从标准输入设备读取字符串, 结束后 (按  +  键) 显示拼错的词汇
-m	自动考虑字尾的变化 (例如, 动词的过去式或复数名词等等)
-M	进入ispell后, 在画面下方显示操作命令的按键。此为默认值
-n	检查的文件为nroff或troff的格式
-p <字典文件>	指定个人字典文件

-P	不考虑字尾变化的情形
-S	不排序建议取代的词汇
-t	检查的文件为TeX或LaTeX的格式
-w <非字母字符>	检查时，特别挑出含有指定的字符（例如，数字、百分比符号等等），且此字符不能为字母。选项前后必须加上双引号（" "），选项中可以输入一个以上的字符
-W <字符串长度>	不检查指定长度的词汇。例如，“-W 1”表示不检查单一字母的词汇。此为默认值
-x	不要产生备份文件，此为默认值

范 例：

- ① 检查text文件的拼写，若发现有疑虑的词汇，则会出现以下画面，可用稍后介绍操作命令决定是否要更改此词汇：

```
# ispell text
```

```

                                                                    发现疑虑的词汇
Frequently Asked Questions (FAQ) for PostgreSQL
Last updated: Tue Dec 17 23:56:27 EST 2002
Current maintainer: Bruce Momjian (pgman@candle.pha.pa.us)
The most recent version of this document can be viewed at
http://www.PostgreSQL.org/docs/faq-english.html.

1) Picturesquely          3) Paradoxically
2) Paradoxical

i) Ignore                I) Ignore all
r) Replace               R) Replace all
a) Add                  x) Exit

?
    可用的组合键                建议可取代的词汇

```

② 检查textfile文件拼写的正确性，若有修改则产生备份文件：

```
# ispell -b textfile
```

③ 检查textfile文件拼写的正确性，检查时需考虑字尾的变化情形：

```
# ispell -m textfile
```

④ 检查textfile文件的正确性，检查时，特别挑出含有“百分比符号”的词汇：

```
# ispell -w "%" textfile
```

⑤ 检查textfile文件拼写的正确性，但不检查该文件的连字错误：

```
# ispell -C textfile
```

⑥ 检查textfile文件拼写的正确性，但不检查长度为2个字母的词汇：

```
# ispell -W 2 textfile
```

⑦ 使用美式英文的字典检查textfile文件拼写的正确性：

```
# ispell -d american textfile
```

⑧ 检查TeX格式的mydoc.lyx文件拼写的正确性：

```
# ispell -t mydoc.lyx
```

⑨ 从键盘输入一段文字，利用ispell命令检查此文字是否有拼写错误：

```
# ispell -l
```

```
How important is information to your organization? ←
```

```
information
```

← 此字拼错了

输入完后，按 **Ctrl** + **Q** 结束



操作命令:

■	不改变当前的词汇，后文中若出现相同的词汇，也加以忽略
■	不改变当前的词汇，且将此词汇改为大写后，加入个人字典
■	查字典
■	不存储变更即结束程序
■	存储变更后结束程序
■	不改变当前的词汇，且将此词汇改为小写后，加入个人字典
■	以输入的词汇来取代
■	执行其他的命令
■	显示命令说明
■ + ■	重新显示画面
■ + ■	结束程序的执行
数字键	指定建议的词汇来取代
空白键	不改变当前的词汇，后文中若出现相同的词汇，仍然加以处理



jed

常用指数: **功能说明:** 编辑文本文件。**语 法:** jed [-2n] [-f <函数>] [-g <行数>] [-i <文件>] [-l <文件>]
[-s <字符串>] [-batch] [文件]**包 名 称:** jed。**相关命令:** ed, joe, sed, vi, vim。**补充说明:** Jed是以Slang (一个和C很像的语言) 所写成的程序, 适合用来编辑程序原始码。Jed能识别原始码并加以格式化, 以方便程序设计师编写程序。

Jed可模拟Emacs、EDT与Wordstar等编辑器。若要改变模拟的设定, 请在jed.rc设定文件 (默认目录为/usr/share/jed/lib) 中加入以下的语句:

- 模拟Emacs (默认): ()=evalfile ("emacs");
- 模拟EDT: ()=evalfile ("edt");
- 模拟Wordstar: ()=evalfile ("wordstar");

Jed可同时处理多个编辑区, 每个编辑区有各自或共同的缓冲区。直接打开文件时, 即建立以该文件为名的缓冲区。可以在缓冲区之间交互切换, 并执行剪贴的操作。



参 数:

-2	显示上下两个编辑区
-batch	以批处理模式来执行
-f <函数>	执行Slang函数
-g <行数>	移至缓冲区中指定的行数
-i <文件>	将指定的文件载入缓冲区
-l <文件>	载入Slang原始代码文件
-n	不要载入jed.rc设定文件
-s <字符串>	查找并移至指定的字符串

范 例:

- ① 编辑mysource.c程序代码文件，可利用稍后所介绍的操作命令来编辑此文件：

```
# jed mysource.c
```

功能表

The screenshot shows the jed editor interface. At the top is a menu bar with the following items: F10 key ==> File Edit Mode Search Buffers Windows System Help. Below the menu bar is a dark window containing the following code:

```
#include <linux/config.h>
#include <linux/module.h>

#include <stdarg.h>

#include <asm/bitops.h>
#include <asm/uaccess.h>
#include <asm/system.h>

#include <linux/errno.h>
#include <linux/fs.h>
#include <linux/ufs fs.h>
#include <linux/slab.h>
#include <linux/sched.h>
#include <linux/stat.h>
#include <linux/string.h>
#include <linux/locks.h>
#include <linux/blkdev.h>
#include <linux/init.h>

#include "swab.h"
```

On the left side of the screenshot, there is a bracket labeled "编辑区" (Editing Area) that encompasses the list of include statements.

jed的版本编号

打开的文件名称

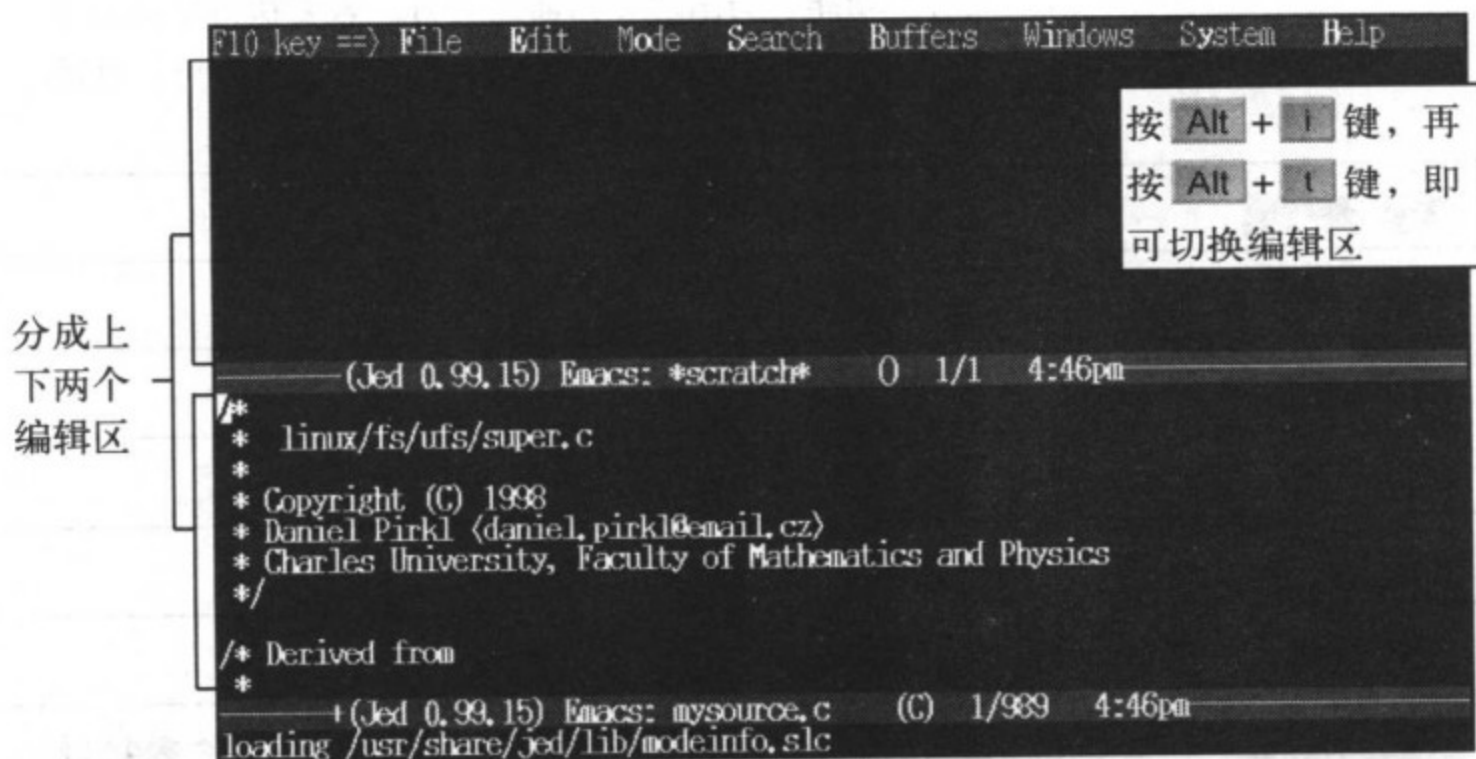
现在时间

当前是模拟Emacs编辑器

光标所在的行数及列数

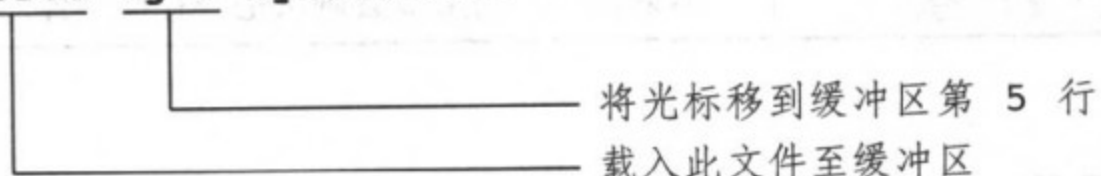
② 以上下两个编辑区的方式，打开mysouce.c程序代码文件：

jed -2 mysource.c



③ 编辑mysource.c程序代码文件，载入程序时并将ref.c文件载入到缓冲区，同时将光标移到缓冲区第5行。若要切换到此缓冲区，请按 Alt + B 键，再按 Alt + C 键，即可切换：

jed -i ref.c -g 5 mysource.c



操作命令：由于Jed可模拟多种编辑器，其各自按键命令也有所不同。

下面以模拟Emacs为例，说明在编辑器中的操作方法：

方向键	在编辑器中移动光标
Page Up 或 Esc V	将光标往上移一页
Page Down 或 Ctrl + V	将光标往下移一页

Esc ? ? 或 F10	切换至功能表。按键的方式是一次按一个键，依次按 Esc 键后，再按 ? 键，再按 ? 键（或直接按 F10 键）。功能表会出现在画面最上方。若要执行功能表上的选项，请直接按该选项的英文字母代号即可。功能表中提供了文件、编辑、查找等许多功能
Esc ? h	显示按键的说明
Ctrl + @	标示选取区域
Ctrl + a	将光标移至此行的起始处
Ctrl + d	删除字符
Ctrl + e	将光标移至此行的结束处
Ctrl + l	重新显示画面
Ctrl + X u	复原上一个操作。按键的方式是先同时按 Ctrl 键与 X 键，然后再单独按 u 键
Ctrl + X + f	打开旧文件
Ctrl + X + W	存储文件
Ctrl + X + C	结束程序，结束前会确认是否存储文件

jobs

常用指数：



功能说明：显示正在后台执行的工作。

语 法：jobs [-lnp] [工作编号]

包 名 称：bash。

相关命令：bg, fg, kill, ps。

补充说明：jobs可显示当前正在后台中执行的工作。可以根据工作编号，将工作移至前台（fg命令），或删除工作（kill命令）。

参 数:

-l	除了一般信息外，并列出工作的执行程序ID
-n	仅列出状态更改过的工作
-p	仅列出工作的执行程序ID

范 例:

① 显示在后台中执行的工作与其执行程序 ID:

```
# jobs -l
[1]+  9442停止 (tty输出)      man jobs      ← 当前有一个在后台执行的程序
```

② 仅显示在后台中执行程序的 ID:

```
# jobs -p
9442      ← 只列出程序ID
```

③ 列出状态变更过的工作:

```
# jobs -n
[1]+  Done      find / -name swatchrc
```



joe (Joe's Own Editor)

常用指数: 

功能说明: 编辑文本文件。

语 法: joe [-asis] [-autoindent] [-backpath <目录>] [-beep]
 [-columns <列>] [-crlf] [-csmode] [-dopadding] [-exask]
 [-force] [-help] [-indentc <缩进字符>] [-istep <缩进字符数>]
 [-keepup] [-keymap <按键设定文件>] [-lightoff] [-lines<行
 数>] [-linums] [-lmargin <栏数>] [-marking] [-mid]
 [-nobackups] [-nonotice] [-nosta] [-noxon] [-orphan] [-
 overwrite] [-pg <行数>] [-rdonly] [-rmargin <栏数>]
 [-skiptop <行数>] [-tab <栏数>] [-wordwrap] [+<行数>]
 [要编辑的文件...]

包 名 称: joe。

相关命令: ed, jed, sed, vi, vim。

补充说明: Joe是一个功能强大的全屏幕文本编辑程序，一次可打开多个文件，每个文件各放在一个编辑区内，并可在文件之间执行剪贴的操作。

Joe可以模拟许多种编辑程序。使用时，直接执行其符号链接，即可进入模拟的环境：

- jstar模拟WordStar的环境。
- jmacs模拟Emacs的环境。
- jpico模拟Pico的环境。
- rjoe则是精简版的Joe。

启动程序时，可以加上两个种类的参数。第一类参数是程序参数，负责控制程序。第二类参数为文件参数，控制所打开的文件。

参 数:

以下为程序参数:	
-asis	字符码超过127的字符不做任何处理
-backpath <目录>	指定备份文件的目录。默认存储在原始文件所在的目录
-beep	编辑时,若有错误(例如,不正确的命令)即发出哔声
-columns <列>	设定列数
-csmode	可执行连续查找模式,此为默认值
-dopadding	使程序跟tty间存在缓冲区
-exask	在程序中,按 Ctrl + K + X 键(存储后结束程序)时,会先确认是否要存储文件
-force	强制在最后一行的结尾处加上换行符
-help	执行程序时一并显示帮助
-keepup	在进入程序后,画面上方为状态栏(包含当前行数、栏数与时间等等)。使用此参数可即时更新状态栏,但会增加系统的负荷。默认值为非即时更新
-lightoff	选取的块在执行完块命令后,就会回复成原来的状态(不反白显示)。块命令是指移动、复制、将块存文件等命令
-lines <行数>	设定行数
-marking	在选取块时(先按 Ctrl + K + b 键,标示块的起点),反白块会随着光标移动
-mid	当光标移出画面时,即自动卷页,使光标回到中央。此功能通常用于一些卷动画面较慢的老式终端机
-nobackups	不建立备份文件。默认值会自动建立名称为原始文件名加上“~”的备份文件
-nonotice	程序执行时,不显示版权信息
-nosta	程序执行时,不显示状态栏(画面最上方的反白行)
-noxon	尝试取消 Ctrl + s 与 Ctrl + q 键的功能
-orphan	若同时打开一个以上的文件,则其他文件会置于独立的缓冲区,而不会另外打开编辑区

pg <行数>	按 Page Up 或 Page Down 键 (或是 Ctrl + u 与 Ctrl + V) 换页时, 所要保留前一页的行数。默认为1行, 也就是每次换页时, 仍然可以看见原来页的第一行或最后一行
-skiptop <行数>	不使用屏幕上方指定的行数
以下为文件参数:	
+<行数>	指定打开文件时, 光标所在的行数
-autoindent	自动缩进。当当前正在编辑的这一行为缩进时, 按 Enter 键后, 下一行也会缩进。
-crlf	在换行时, 使用CR+LF字符 (换行并将光标移至该行最前端)。这是为了要与MS-DOS的文件系统兼容
-indentc <缩进字符>	执行缩进时, 实际插入的字符。如为空白字符则指32, tab则为9。默认为空白字符
-istep <缩进字符数>	每次执行缩进时, 所移动的缩进字符数
-keymap<按键设定文件>	使用不同的按键设定文件
-linums	在每行前面加上行号
-lmargin <栏数>	设定左侧边界
-overwrite	设定覆写模式 (默认为插入模式)
-rmargin <栏数>	设定右侧边界
-tab <栏数>	设定tab的宽度。默认宽度为8个字符
-rdonly	以只读的方式打开文件
-wordwrap	编辑时若超过右侧边界, 则自动换行

范 例:

① 以joe文本编辑器打开text文本文件, 可用稍后介绍的操作命令编辑文件:

```
# joe text
```

打开的文件名称 光标所在的行数及列数 现在时间 显示操作说明的按键

编辑区

```

IW (text) (Row 441 Col 11) (5:25) (Ctrl-K H for help)
Command arguments which have embedded spaces may be quoted with quote ""
marks.

To abort a file transfer, use the terminal interrupt key (usually Ctrl-
C). Sending transfers will be immediately halted. Receiving transfers
will be halted by sending a ftp protocol ABOR command to the remote serv
er, and discarding any further data received. The speed at which this is
accomplished depends upon the remote server's support for ABOR process
ing. If the remote server does not support the ABOR command, an 'ftp'
prompt will not appear until the remote server has completed sending the
requested file.

The terminal interrupt key sequence will be ignored when ftp has complet
ed any local processing and is awaiting a reply from the remote server.
A long delay in this mode may result from the ABOR processing described
above, or from unexpected behavior by the remote server, including viola
tions of the ftp protocol. If the delay results from unexpected remote
server behavior, the local ftp program must be killed by hand.

FILE NAMING CONVENTIONS
Files specified as arguments to ftp commands are processed according to
the following rules.

```

② 利用上下两个编辑区打开text1及text2文本文件：

```
# joe text1 text2
```

③ 不使用两个编辑区，而是利用不同的缓冲区来打开text1及text2文本文件：

```
# joe -orphan text1 text2
```

④ 打开text1文本文件，并指定存储时不建立备份文件：

```
# joe -nobackups text1
```

⑤ 打开text1文本文件，并指定备份文件存储在/root/doc目录下：

```
# joe -backpath /root/doc text1
```

J

⑫ 打开text1文本文件，若执行换页操作时，保留前一页6行的内容在画面上：

```
# joe -pg 6 text1
```

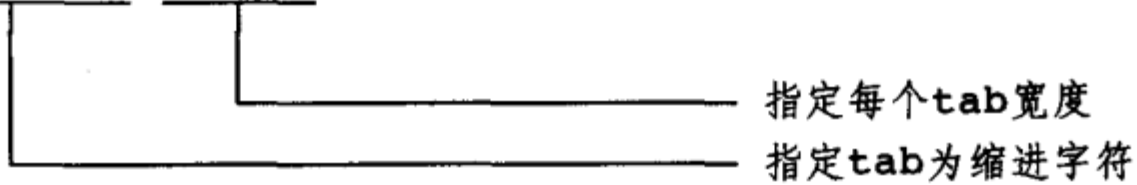
⑬ 打开text1文本文件，打开之后直接将光标移至第50行，并设定为覆写模式：

```
# joe -overwrite +50 text1
```



⑭ 打开text1文本文件，当执行缩进时，指定要插入的字符为tab，并且每个tab的宽度为3个字符：

```
# joe -indentc 9 -tab 3 text1
```



⑮ 打开在MS-DOS下建立的msdos.txt文本文件：

```
# joe -crlf msdos.txt
```

↑
若不使用此参数，则在MS-DOS下阅读该文件将会发生问题

⑯ 以只读的方式打开text1文本文件：

```
# joe +ronly text1
```




操作命令:

← 或 Ctrl + b	将光标向左移一个字符
→ 或 Ctrl + f	将光标向右移一个字符
↑ 或 Ctrl + p	将光标向上移一个字符
↓ 或 Ctrl + n	将光标向下移一个字符
Ctrl + X	将光标移到下一个英文字符
Ctrl + Z	将光标移到上一个英文字符
Page Up 或 Ctrl + u	将光标往上移一页
Page Down 或 Ctrl + v	将光标往下移一页
Ctrl + a	移到该行起始处
Ctrl + e	移到该行结尾处
Ctrl + k + u	移到文件的起始处
Ctrl + k + v	移到文件的结尾处
Ctrl + k + h	在画面上方显示操作命令的说明。按 Esc + . 键可显示帮助的下一页, 按 Esc + , 键可显示帮助的上 一页, 若要关闭说明, 请再按一次 Ctrl + k + h 键
Ctrl + k + f	查找字符串, 并可执行取代的功能
Ctrl + l	查找下一个字符串
Del 或 Ctrl + d	删除光标所在的字符
Ctrl + w	删除光标所在的英文字符
Ctrl + y	删除光标所在的行
Ctrl + k + b	标示选取区域开始的地方
Ctrl + k + k	标示选取区域结束的地方
Ctrl + k + c	将选取区域复制到光标当前所在的位置
Ctrl + k + m	将选取区域移至光标当前所在的位置
Ctrl + k + y	删除选取区域
Ctrl +] +	将缩进左移
Ctrl + .	将缩进右移
Ctrl + k + o	将当前编辑区一分为二
Ctrl + k + n	切换到下一个编辑区
Ctrl + k + p	切换到上一个编辑区
Ctrl + k + g	放大当前编辑区的显示区域
Ctrl + k + t	缩小当前编辑区的显示区域
Ctrl + k + i	只用一个编辑区显示

Ctrl + k + e	打开其他文件（或新文件）来编辑
Ctrl + k + e	在光标当前所在的编辑区载入文件
Ctrl + c	关闭编辑区，若内文有更动，则会确认是否不存储，并立即关闭文件
Ctrl + k + r	插入文件
Ctrl + k + d	存储文件
Ctrl + k + x	存储文件后关闭编辑区，若仅剩一个编辑区，则结束程序
Ctrl + k + z	暂时切换到shell下执行程序，可在shell执行fg命令，重新回到joe程序

join

常用指数：

功能说明：将两个文件中，指定列内容相同的行连接起来。

语 法：join [-i] [-a <1 或 2>] [-e <字符串>] [-j <列>] [-o <格式>] [-t <字符>] [-v <1 或 2>] [-1 <列 A> -2 <列 B>] [--help] [--version] [文件 1] [文件 2]

包名称：coreutils。

补充说明：找出两个文件，指定列（默认为第1个列）内容相同的行，并加以合并，再输出到标准输出装置。

输出内容的第1个列为两个文件共同的列，接着显示[文件1]该行的其他内容，然后再显示[文件2]该行的其他内容。默认是以空白字符来区隔列。

参 数:

-a <1 或 2>	除了显示原来的输出内容之外，还显示命令文件中没有相同列的行。1 表示 [文件1]，2 表示 [文件2]
-e <字符串>	若[文件1]与[文件2]中找不到指定的列（例如，空白行），则在输出中填入选项中的字符串
-i 或--ignore-case	比较列内容时，忽略大小写的差异
-j <列>	比较指定的列，而不比较默认的第1个列
-o <格式>	按照指定的格式来显示结果。<格式>中必须先指定文件（1或2），加上句号，再加上列编号。例如，“1.3”表示[文件1]的第3个列
-t <字符>	指定列的分隔字符
-v <1或2>	跟“-a”相同；但是只显示指定文件中没有相同列的行
-1 <列A> -2 <列B>	比较[文件1]指定的列A，及[文件2]指定的列B
--help	显示帮助
--version	显示版本信息

范 例:

① 列出file1及file2文件的内容，然后寻找到两个文件第1个列相同的行，若找到则列出该行：

```
# cat file1          列出file1的内容
apple      2kg      NT440      90%
banana     5kg      NT170      50%
Cherry     6kg      NT300      45%
marmalade  3kg      NT100      20%
```

└───┘
└───┘
└───┘

第1列
第2列
第3列
第4列

```
# cat file2 ← 列出file2的内容
apple      Taipei      NT440      2kg
amarelle   Taichun     NT120      4kg
banana     Kaohsiung   NT170      5kg
cherry     Pingtung    NT300      7kg
```

第1列 第2列 第3列 第4列

```
# join file1 file2 ← 以默认的方法比较 2 个文件
apple 2kg NT440 90% Taipei NT440 2kg
banana 5kg NT170 50% Kaohsiung NT170 5kg
```

文件2的第4列
文件2的第3列
文件2的第2列
文件1的第4列
文件1的第3列
文件1的第2列
2个文件有相同的第1列

② 同第1个范例的两个文件，若找到两个文件第1个列相同的行，除了显示该行内容之外，还显示文件1没有相同列的行：

```
# join -a 1 file1 file2
apple 2kg NT440 90% Taipei NT440 2kg
banana 5kg NT170 50% Kaohsiung NT170 5kg
Cherry 6kg NT300 45%
marmalade 3kg NT100 20%
```

列出文件1没有相同列的其他2行

③ 同第1个范例的两个文件，若找到两个文件第1个列相同的行，只显示文件1没有相同列的行，并不输出有相同列的行：

```
# join -v 1 file1 file2
Cherry 6kg NT300 45%
marmalade 3kg NT100 20%
```

仅列出文件1没有相同列的行

J

- ④ 同第1个范例的两个文件，若找到两个文件第1个列相同的行，则列出该行，但比较时忽略大小写的差异：

```
# join -i file1 file2
apple 2kg NT440 90% Taipei NT440 2kg
banana 5kg NT170 50% Kaohsiung NT170 5kg
Cherry 6kg NT300 45% Pingtung NT300 7kg
```

与第1例比较，
← 多出了此行

- ⑤ 同第1个范例的两个文件，若找到两个文件第3列相同的行，则列出该行：

```
# join -j 3 file1 file2
NT440 apple 2kg 90% apple Taipei 2kg
NT170 banana 5kg 50% banana Kaohsiung 5kg
NT300 Cherry 6kg 45% cherry Pingtung 7kg
```

以两个文件第3列为比较条件

- ⑥ 同第1个范例的两个文件，比较文件1第2列，及文件2第4列，若找到相同的行，则列出该行：

```
# join -1 2 -2 4 file1 file2
```

文件2的第4列
文件1的第2列

```
2kg apple NT440 90% apple Taipei NT440
5kg banana NT170 50% banana Kaohsiung NT170
```

改以此列比较

- ⑦ 同第1个范例的两个文件，然后若找到两个文件第1个列相同的行，则依次列出文件1的第1、2、3、4列，文件2的第2列：

```
# join -o 1.1 1.2 1.3 1.4 2.2 file1 file2
apple 2kg NT440 90% Taipei
banana 5kg NT170 50% Kaohsiung
```

kill

常用指数:



功能说明: 删除执行中的程序或工作。

语法: kill [-s <信息名称或编号>] [程序...]或kill [-l <信息编号>]

包名称: util-linux。

相关命令: ps, jobs, top。

补充说明: kill可将指定的信息送至执行中的程序。默认的信息为SIGTERM(15), 可将指定程序终止。若仍无法终止该程序, 可使用SIGKILL(9)信息尝试强制删除程序。

程序或工作的编号可利用ps命令或jobs命令查看。

参数:

-l <信息编号>	若不加<信息编号>选项, 则“-l”参数会列出全部的信息名称。若加上<信息编号>选项, 则列出该信息编号的信息名称
-s <信息名称或编号>	指定要送出的信息。若是指定信息名称, 可以用全名, 或是仅使用“SIG”后的字符串。例如, 若要指定“SIGTERM”, 只要用“TERM”即可
[程序]	[程序]可以是程序的PID或是PGID, 也可以是工作编号。若是工作编号, 编号前要加上百分比符号(%)

范 例:

① 列出所有信息名称:

```
# kill -l
```

```
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL
5) SIGTRAP     6) SIGIOT     7) SIGBUS      8) SIGFPE
9) SIGKILL     10) SIGUSR1   11) SIGSEGV    12) SIGUSR2
13) SIGPIPE    14) SIGALRM   15) SIGTERM    17) SIGCHLD
18) SIGCONT    19) SIGSTOP   20) SIGTSTP    21) SIGTTIN
22) SIGTTOU    23) SIGURG    24) SIGXCPU    25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF   28) SIGWINCH   29) SIGIO
...
```

② 列出信息编号9的信息名称:

```
# kill -l 9
```

```
KILL
```

③ 列出当前系统执行的程序，并强制删除编号 1143 的程序:

```
# ps
```

```
  PID TTY          TIME CMD
 1127 tty1      00:00:00 bash
 1142 tty1      00:00:00 swatch
 1143 tty1      00:00:00 swatch
 1144 tty1      00:00:00 perl
 1146 tty1      00:00:00 tail
 1150 tty1      00:00:00 ps
```

```
# kill-s 9 1143
```

K

last

常用指数:



功能说明: 列出当前与过去登录系统的用户相关信息。

语法: last [-adRx] [-f <记录文件>] [-n <显示列数>] [账号名称...]
[终端编号...]

包名称: SysVinit。

相关命令: login。

补充说明: 单独执行last命令，它会读取位于/var/log目录下，名称为wtmp的文件，并把该文件内容记录的登录系统的用户名单全部显示出来。可以指定账号名称或是终端编号，让last命令仅列出指定范围的清单。

参数:

-a	把从何处登录系统的主机名称或IP地址显示在最后一行
-d	将IP地址转换成主机名称。当用户从远程而非本地主机登录系统时，所记录的将不仅是主机名称，还包括该远程主机的IP地址。默认last命令会用IP地址来显示远程的主机，使用这项参数可将其换成主机名称
-f <记录文件>	指定记录文件。默认last命令会去读取/var/log目录里的wtmp文件，如果您有另外的记录文件，便可使用本参数指定它
-n <显示行数>或-<显示行数>	设置显示名单的显示行数。如只想查询最后登录系统的10位用户名称，可将显示行数设成10，想查询最后的30位用户，则设为30，依此类推
-R	不显示登录系统的主机名称或IP地址
-x	显示系统关机、重新启动，以及执行等级的改变等信息

范 例:

① 查询最后登录系统的20位用户:

last -n 20

```

benny pts/2 205.47.203.13 Thu Jun 19 10:38 still logged in
irchen tty4 103.4.50.136 Thu Jun 19 10:00 still logged in
twomix pts/4 163.88.37.250 Thu Jun 19 09:37 -10:00 (00:22)

```

...

```

votoms pts/0 :0 Wed Jun 18 18:04 -19:08 (01:03)
chaos pts/1 198.162.3.27 Wed Jun 18 11:48 -18:03 (06:14)
nike pts/0 :1 Wed Jun 18 11:17 -18:03 (06:46)

```

用户 名称	终端 编号	显示用户从 何处登录系统	登录日期	登录系统 起始时间	登录系统 合计时间

wtmp begins Mon Jun 2 11:17:07 2003 ← 显示记录文件开始
记录信息的时间

② 查询用户claire登录系统的情况, 还有第3号终端的使用情形:

last claire 3

```

claire ftp 163.82.21.198 Wed Jun 18 16:54 - 16:55 (00:00)
apple2 tty3 Wed Jun 18 16:50 - 16:53 (00:02)
claire ftp 163.82.21.198 Tue Jun 18 17:23 - 17:23 (00:00)

```

...

```

rochen tty3 228.16.37.201 Tue Jun 17 11:44 - 14:18 (02:33)
claire tty4 Tue Jun 17 11:25 - 15:18 (03:53)
orange tty3 Tue Jun 17 11:21 - 11:25 (00:03)

```

wtmp begins Mon Jun 2 11:17:07 2003



- ③ 列出当前与过去登录系统的用户信息，并将登录的主机名称或IP地址显示在最后一行：

```
# last -a
irchen pts/1 Thu Jun 19 16:47 still logged in 103.4.50.136
```

↑
在最后一行列出登录的IP

...

- ④ 列出当前与过去登录系统的用户信息，但不列出登录系统的主机或IP地址：

```
# last -R
irchen pts/1 Thu Jun 19 16:47 still logged in
```

↑
与前面例子比较，可发觉此例已没有显示登录的主机或IP地址

...

- ⑤ 除了列出当前与过去登录系统的用户信息外，还列出系统关机、重新启动及改变执行等级等信息：

```
# last -x
...
runlevel (to lvl 3) 2.4.20-8 Sat May 6 15:00 - 16:11(2+01:11)
```

执行等级改变

```
reboot system boot 2.4.20-8 Sat May 6 15:00 (2+02:12)
```

重新启动

```
shutdown system down 2.4.20-8 Sat May 6 15:02 - 17:13(2+02:10)
```

系统关机

```
runlevel (to lvl 6) 2.4.20-8 Sat May 6 15:02 - 15:02 (00:00)
```

执行等级改变

...

less


常用指数: **功能说明:** 显示文件内容。**语法:** less [-?aBcCdeEfgGiImMnNqQrsSuUVwX] [-b<缓冲区大小>] [-h<回卷行数>] [-j<行数编号>] [-k<按键定义文件>] [-o<输出文件>] [-O<输出文件>] [-p<范本样式>] [-P<s/m/M/h/=>] [-t<标签>] [-T<标签文件>] [-x<跳格字数>] [-y<前卷行数>] [-z<显示行数>] [--] [文件...]**包名称:** less。**相关命令:** more。**补充说明:** less命令的效用类似more, 但它具有互动式操作界面, 可以轻易地利用各种组合键来浏览文件。遇到大型文件时, less命令不会一次读取整个文件, 因此在执行速度上较文本编辑程序快。

如果文件名称含有“-”或“+”号, 可用符号“--”来区分参数与文件名的差别。在“--”号之后的字符串都视为文件名称。

参数:

-?或--help	显示帮助
-a	查找字符串时, 略过画面上显示的部分
-b<缓冲区大小>	设置缓冲区的大小, 单位以KB计算
-B	关闭自动配置管道 (Pipe) 缓冲区。从管道读入数据时, less命令会自动分配所需的缓冲区内存。当大量数据涌入时, 为避免耗费过多的内存可附加这项参数关闭自动分配, 仅使用“-b”参数指定的缓冲区大小
-c	重新刷新整个画面
-C	此参数的效果和指定“-c”参数类似, 但在重新刷新之前会先清除画面

-d	忽略错误信息
-e	自动离开less浏览模式。当第2次浏览到EOF (End of File) 行时, less命令会自动离开其浏览模式。默认必须按下 q 或 Q 键才能离开
-E	此参数的效果和指定“-e”参数类似,但在第1次浏览到EOF 列就会离开
-f	强制开启特殊文件。所谓特殊文件是指外围设备代号、目录或二进制文件,一般而言,less命令会拒绝开启此类文件,或先询问用户。使用本参数后,可强制less命令直接开启特殊文件(不包括目录),且忽略警告信息
-g	只反白显示当前符合查找条件的字符串。通常less会反白显示所有符合查找条件的字符串,可用此参数来改变默认值
-G	不反白显示任何符合查找条件的字符串
-h<回卷行数>	设置最大回卷行数。超过此设置值,就重新刷新画面
-i	忽略字符大小写差别。整个查找的范本样式皆为大写或小写字符串者,其大小写一视同仁没有差别。若指定的范本样式里混杂着大小写字符,则不会忽略该范本样式的字符大小写差别
-I	此参数的效果和指定“-i”参数类似,但就算范本样式有大小写字符混杂,也一概忽略
-j<行数编号>	设置目标行的行数编号。举例来说,当目标行的编号设为2时,则要查找的字符串会从画面上的第3行开始查找,依此类推
-k<按键定义文件>	另行指定less命令的按键定义文件
-m	在画面底部的提示符号上,使用类似more命令的百分比模式。less命令默认是使用“:”号
-M	此参数的效果和指定“-m”参数类似,除了显示百分比之外,还显示当前所在行数以及总行数
-n	忽略行数编号。当读入的文件内容其数据量过于庞大时,关闭行数的编号可以提升less命令执行的速度
-N	显示行数编号。在每行开头的部分,标示出该行的行数编号
-o<输出文件>	将less命令读入的数据输出成文件存储起来。必须指定输出文件的名称,当该文件名称已经存在时,less命令会询问用户是否要将已存在的文件覆盖。本参数仅适用于从管道导入的数据

-O<输出文件>	此参数的效果和指定“-m”参数类似，但遇到已存在的文件名称时，不会询问用户
-p<范本样式>	从指定的范本样式处开始显示。less命令会找寻所指定的范本样式，并从第1个符合条件的地方开始显示
-P<s/m/M/h/=>	设置不同的提示符号。本参数通常在LESS环境变量中指定，避免一再反复执行。其中“s”、“m”和“M”各自表示短、中、长格式的提示符号，“h”会改变帮助的提示符号，而“=”会改变按下  键时所显示的信息
-q	不使用终端钟鸣声响。假如该终端可用提醒符号代替声音，则会用它取代真正的声响
-Q	此参数的效果和指定“-q”参数类似，但不发出任何声响，也不会显示提醒符号
-r	显示控制字符。注意本参数可能造成显示上的错误
-s	当内容某部分的空白行超过一行时，则该处仅以一行空白列表示
-S	当某列的内容超过显示画面的宽度时，直接将超出的部分截断舍弃。默认 less 命令会把超出的部分新增成另一行
-t<标签>	编辑内含指定标签的文件。欲使用这项参数，则在执行less命令的当前目录下，必须含有一个名称为tags的标签文件，该文件由ctags命令所产生
-T<标签文件>	另行指定标签文件，而不使用默认的tags文件
-u	将倒退和CR字符视为可打印的字符
-U	将倒退、跳格以及CR字符视为控制字符
-V或--version	显示版本信息
-w	使用空白行显示EOF列之后的列
-x<跳格字数>	设置跳格字符的位移字数，默认值为8个字符
-X	不传送termcap初始化、反初始化字符串到终端
-y<前卷行数>	设置最大前卷行数。超过此设置值，就重新刷新画面
-z<显示行数>或 -<显示行数>	改变画面的显示行数，默认值视终端的类型而定。如果所指定的显示行数为负数，例如“-5”，而该终端原本可显示25行，则less命令将会使用20行来显示文件的内容

范 例:

- ① 显示textfile文本文件的内容, 可利用稍后所介绍的操作命令来浏览文件内容:

```
# less textfile
```

- ② 显示text文本文件的内容, 并从第一次出现“display”字符串的地方开始显示:

```
# less -p display textfile
```

- ③ 显示textfile文本文件的内容, 当执行查找命令时, 只反白显示当前这个符合查找条件的字符串:

```
# less -g textfile
```

- ④ 显示textfile文本文件的内容, 当执行查找命令时, 不反白显示所有符合查找条件的字符串:

```
# less -G textfile
```

- ⑤ 显示textfile文本文件的内容, 当执行查找命令时, 忽略字符大小写的差别:

```
# less -I textfile
```

- ⑥ 显示textfile文本文件的内容, 并标示每行的编号:

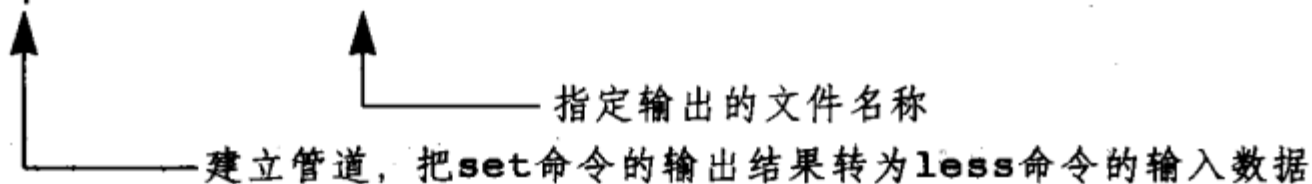
```
# less -N textfile
```

- ⑦ 显示textfile文本文件的内容, 当连续的空白行超过一行时, 仅以一行显示:

```
# less -s textfile
```

- ⑧ 将数据经由管道导入less命令，以便阅读其内容，最后并存成outfile文件：

```
# set | less -o outfile
```



- ⑨ 显示textfile文本文件的内容，并使用百分比来作为提示符号：

```
# less -m textfile
```

- ⑩ 显示textfile文本文件的内容，提示符号除了列出百分比之外，并显示当前所在的行数及总行数：

```
# less -M textfile
```

- ⑪ 显示textfile文本文件的内容，并将提示符号改为“input”字符串：

```
# less -Pinput textfile
```

- ⑫ 显示textfile文本文件的内容，若文本的长度超过画面可显示范围时，则直接截断此行：

```
# less -S textfile
```

- ⑬ 文件dexxa是一个二进制文件，使用less命令查看其内容，设置缓冲区大小为10KB，且不发出任何声响，并在碰到EOF行时自动离开less浏览模式：

```
# less -f -b10 -q -E dexxa
```

- ⑭ 同时载入text1及text2两个文件的内容，可用稍后所介绍操作命令，切换这两个文件：

```
# less text1 text2
```

操作命令：

E 、 Ctrl + E 、 Ctrl + N 或 Enter	往下移一行
Y 、 Ctrl + Y 、 K 、 Ctrl + K 或 Ctrl + P	往上移一行
F 、 Ctrl + F 、 Ctrl + V 或 Space	往下移一页
B 、 Ctrl + B 或 Esc V	往上移一页
D 或 Ctrl + D	往上移半页
U 或 Ctrl + U	往下移半页
g 、 < 或 Esc <	移至文件起始处
G 、 > 或 Esc >	移至文件结束处
Esc) 或 →	向右移8列
Esc (或 ←	向左移8列
/ <字符串>	往后查找指定的字符串
? <字符串>	往前查找指定的字符串
N	查找下一个字符串
N	反方向查找字符串
Esc U	取消查找的反白标示
m <字母>	可指定一个字母，以记忆当前的位置
\ <字母>	回到此字母指定的位置。其中“^”表示文件起始处，“\$”表示文件结束处
: E <文件名称> 或 Ctrl + K Ctrl + V <文件名称>	载入指定的文件
: N	切换到当前已载入文件的下一个文件

: P	切换到当前已载入文件的上一个文件
: X	切换到当前已载入文件的第一个文件
: D	不再载入当前文件
= 、 Ctrl + G 或 : F	显示文件名称
q 、 : q 、 Q 、 : Q 或 z Z	结束程序
R 、 Ctrl + R 或 Ctrl + L	重新显示画面
h + H 或	显示操作说明

lha

常用指数:



功能说明: 压缩或解压文件。

语 法: lha [-acdefglmnpqtuvx] [-a<0/1/2>/u<0/1/2>] [-<a/c/u>d] [-<e/x>i][-<a/u>o] [-<e/x>w=<目的目录>] [-<a/u>z] [压缩文件] [文件...]或lha [-acdefglmnpqtuvx] [-a<0/1/2>/u<0/1/2>] [-<a/c/u>d] [-<e/x>i] [-<a/u>o] [-<e/x>w=<目的目录>] [-<a/u>z] [压缩文件][目录]

包名称: lha。

补充说明: lha是从lharc演变而来的压缩程序，文件经它压缩过后，会另外产生具有.lzh扩展名的压缩文件。因为它是由日本用户所发展的程序，所以在日本FTP站点经常可见到最后扩展名为.lzh的文件。不论压缩或解压缩，lha命令都要附加参数执行。

假设欲压缩、解压的文件是硬链接或符号链接，lha命令会读取该连接所指向、连接的原始文件的内容，将它压缩、解压后存文件。注意连接的文件经过压缩再解压之后，其连接将不复存在，而成为与原始文件内容完全相同的另一个文件。

参 数:

-a或a	压缩文件，并加入至压缩文件内。若该压缩文件中已有相同名称的文件存在，则压缩文件内的文件将被置换为加入的文件
-a<0/1/2>或a<0/1/2>	压缩文件时，采用不同的文件头。默认值为1，指定为2较节省空间，设置成0则不会记录每个文件的路径地址
-c或c	压缩文件，重新建构新的压缩文件后，再将其加入
-d或d	从压缩文件内删除指定的文件
-<a/c/u>d或<a/c/u>d	压缩文件，然后将其加入、重新构建、更新压缩文件后，删除原始文件，即把文件移到压缩文件中
-e或e	解开压缩文件
-f或f	强制执行lha命令，在解压缩时会直接覆盖已有文件，而不加以询问
-g或g	使用通用的压缩格式，便于解决兼容性的问题
-<e/x>i或<e/x>i	解开压缩文件时，忽略存储在压缩文件内的文件路径，直接将其解压后存放在当前目录下，或是指定的目录中
-l或l	列出压缩文件的相关信息
-m或m	此参数的效果和同时指定“-ad”参数相同
-n或n	不执行命令，仅列出实际执行会进行的动作
-<a/u>o或<a/u>o	采用lharc兼容格式，将压缩后的文件加入、更新压缩文件
-p或p	从压缩文件内输出至标准输出设备
-q或q	不显示命令执行过程
-t或t	检查备份文件内的每个文件是否正确无误
-u或u	更换较新的文件至压缩文件内。利用此参数可把较新的文件压缩后，加入压缩文件中，置换原本压缩文件里较旧的文件
-u<0/1/2>或u<0/1/2>	在文件压缩时采用不同的文件头，然后更新至压缩文件内。默认值为1，指定为2较节省空间，设置成0则不会记录每个文件的路径地址
-v或v	详细列出压缩文件的相关信息。比参数“-l”多了压缩后的文件大小、CRC校验码及时间戳

-<e/x>w=<目的目录> 或<e/x>w=<目的目录>	指定解压缩的目录。可使用这项参数，设置存放解压后的文件的目录
-x或x	解开压缩文件
-<a/u>z或<a/u>z	不压缩文件，直接把它加入、更新压缩文件

范 例:

① 将source文件压缩成destine.lzh压缩文件:

```
# lha -a destine.lzh source
source - Frozen(-35%) o
```

② 把/home/olympus目录下的所有文件一起压缩，包括子目录，压缩文件名为pegasus.lzh:

```
# lha -a pegasus.lzh /home/olympus
```

③ 压缩source1、source2及source3等3个文件，压缩文件名为multifiles.lzh，压缩时采用较节省空间的方式来压缩文件头:

```
# lha -a2 multifiles.lzh source1 source2 source3
```

④ 列出multifiles.lzh压缩文件的相关信息:

```
# lha -l multifiles.lzh
```

PERMSSN	UID GID	SIZE	RATIO	STAMP	NAME
-----	-----	-----	-----	-----	-----
-rw-r--r--	0/0	15371	30.6%	Jun 9 14:02	source1
-rw-r--r--	0/0	17974	22.7%	Jun 9 14:03	source2
-rwxr-xr-x	0/0	26700	44.4%	Jun 9 14:04	source3
-----	-----	-----	-----	-----	-----
Total	3 files	60045	34.4%	Jun 9 14:07	

⑤ 以更详细的方式列出multifiles.lzh压缩文件的相关信息:

```
# lha -v multifiles.lzh
PERMSSN  UID GID  PACKED  SIZE RATIO  CRC  STAMP NAME
-----
rw-r--r--0/0 4715 15371 30.6% -lh5- 296b Jun 9 14:02 source1
rw-r--r--0/0 4095 17974 22.7% -lh5- 4665 Jun 9 14:03 source2
rwxr-xr-x0/0 11861 26700 44.4% -lh5- 2808 Jun 9 14:04 source3
-----
Total      3files 20671 60045 34.4% Jun 9 14:07
```

⑥ 测试multifiles.lzh压缩文件的正确性:

```
# lha -t multifiles.lzh
source1 - Tested : o
source2 - Tested : o
source3 - Tested : o
```

每个文件都正确无误

⑦ 将source4文件压缩后, 加入multifiles.lzh压缩文件内:

```
# lha -a multifiles.lzh source4
source4 - Frozen
# lha -l multifiles.lzh ← 查看multifiles.lzh压缩文件
PERMSSN  UID GID  SIZE  RATIO  STAMP  NAME
-----
rw-r--r-- 0/0 15371 30.6% Jun 9 14:02 source1
rw-r--r-- 0/0 17974 22.7% Jun 9 14:03 source2
rwxr-xr-x 0/0 26700 44.4% Jun 9 14:04 source3
rwxr-xr-x 0/0 28796 50.4% Jun 9 14:31 source4 ←
-----
Total      4 files 88841 39.6% Jun 9 14:32
```

新加入的文件



⑧ 删除multifiles.lzh压缩文件内的source3文件：

```
# lha -d multifiles.lzh source3
# lha -l multifiles.lzh
```

PERMSSN	UID	GID	SIZE	RATIO	STAMP	NAME
rw-r--r--	0/0	15371	30.6%	Jun 9 14:02	source1	← source3 已经删除
rw-r--r--	0/0	17974	22.7%	Jun 9 14:03	source2	
rw-r-xr-x	0/0	28796	50.4%	Jun 9 14:31	source4	

Total	3 files	62141	37.5%	Jun 9 14:36		

⑨ 解压缩multifiles.lzh压缩文件：

```
# lha -x multifiles.lzh
```


⑩ 把压缩文件artemis.lzh解压到当前目录下的 agis 子目录里，并忽略压缩文件内所记录的路径地址：

```
# lha -xiw=agis artemis.lzh
```

↑ 这边也能指定绝对路径，例如/home/Olympus/agis
 如果压缩文件内含有子目录twilight，使用“-i”参数
 让lha命令把所有文件解
 压后放在agis目录里，若不附加这个参数，则会放到
 agis/twilight目录下



lilo (linux loader)

常用指数: 

功能说明: 安装内核载入、启动管理程序。


语 法: lilo [-clqtV] [-b <外围设备代号>] [-C <设置文件>] [-d <延迟时间>] [-D <识别标签>] [-f <几何参数文件>] [-i <启动扇区文件>] [-I <识别标签>] [-m <映射文件>] [-P <fix/ignore>] [-r <根目录>] [-R <执行命令>...] [-s <备份文件>] [-S <备份文件>] [-uU <外围设备代号>] [-v...]

包名称: lilo。

相关命令: grub-install。

补充说明: lilo是一个Linux系统内核载入程序，同时具备管理启动的功能。单独执行lilo命令，它会读取/etc目录下的lilo.conf设置文件，然后根据其内容安装lilo。

参 数:

-b <外围设备代号>	指定安装lilo之处的外围设备代号。例如要把lilo装在第1台IDE硬盘的主引导记录 (Main Boot Record)，其外围设备代号就是/dev/hda，若是装在第1个IDE硬盘的第2个主要分区，其外围设备代号就是/dev/hda2，依此类推 (逻辑分区编号从5开始)
-c	使用紧致映射模式。本参数会把系统启动时所需的文件放置在连续的扇区里，以便增进读取效率 (对于软盘效果显著)
-C <设置文件>	指定lilo的设置文件。默认的文件名为lilo.conf，放置在/etc目录下。使用此参数可改变默认值,另行指定文件
-d <延迟时间>	设置启动延迟的时间。在启动之后，lilo默认会等待5秒，以便指定欲启动的操作系统，或是欲使用的系统内核。等待时间是从看到“LILO boot:”字符串后开始算起，若5秒之内按下任何按键，则重新计时5秒，直到不再按任何按键后5秒，或按下  键为止，lilo才会继续启动操作。延迟时间的单位以十分之一秒计算

-D <识别标签>	指定启动后默认启动的操作系统，或系统内核识别标签。如果没有设置默认启动的识别标签，则lilo会自行启动设置文件内排在第一个的操作系统或系统内核
-f <几何参数文件>	指定磁盘的几何参数设置文件。默认是/etc目录里的disktab文件，可借由这个参数去改变它
-i <启动扇区文件>	指定欲使用的启动扇区文件，默认是/boot目录里的boot.b文件，可借由这个参数去改变它
-I <识别标签>	显示系统内核存放之处。指定启动系统时的识别标签，lilo命令会把该识别标签所使用的内核映像文件，其座落位置及文件名称输出到标准输出设备
-l	产生线性扇区地址，而不采用柱面/磁头/扇区(Cylinder/Head/Sector)的方式定义硬盘的位
-m <映射文件>	指定映射文件。默认的文件名为map，放置在/boot目录下。使用此参数可改变默认值，另行指定文件
-P <fix/ignore>	决定要修复或忽略分割表的错误。当硬盘之类的存储媒体的分割表发生错误时，可通过“fix”选项尝试去修复它，或是使用“ignore”选项忽略错误不予处理
-q	列出映射的系统内核文件。lilo会把系统内核文件的名称及存放地址，记录在映射文件中，默认的映射文件名称为map，放置在/boot目录里
-r <根目录>	设置系统启动时欲加载成为根目录的目录。这项参数通常是从软盘启动，准备安装操作系统时才会用到
-R <执行命令>	设置下次启动系统时，首先执行的命令。该命令仅会执行一次，然后lilo便会将其删除
-s <备份文件>	指定备份文件。当执行lilo命令，在数据写入启动扇区之前，lilo会先读取该扇区的内容，并将其存放在/boot目录下，文件名称为boot.xxxx，其中“xxxx”字符串为一组数字代码，视启动设备而定。本参数让用户可以另行指定备份文件的名称，而不采用默认值。与参数“-u”一起使用，则会从指定的备份文件读取数据，然后写回启动扇区
-S <备份文件>	强制指定备份文件。用参数“-s”指定备份文件，若该文件名称已经存在，则lilo将不会覆盖已有文件。使用本参数可强制lilo命令覆盖已存在的文件
-t	不执行命令，仅列出实际执行会进行的动作。搭配参数“-v”使用，可得知进一步的相关信息

-u <外围设备代号>	删除lilo。使用这个参数，可删除指定的外围设备上的lilo程序，同时lilo会检查时间戳，并把当初备份下来的扇区数据，从备份文件读取数据后写回启动扇区
-U <外围设备代号>	此参数的效果和指定“-u”参数类似，但不检查时间戳
-v	显示命令执行过程。如重复指定这项参数多次，则会显示更为详细的相关信息
-V	显示版本信息

范 例：

① 根据/etc/lilo.conf设置文件，执行lilo命令：

```
# lilo
Added linux
Added dos *
```

② 安装lilo到第1台SCSI硬盘的第4个主要分区，并采用第3级的详细模式，显示命令执行过程：

```
# lilo -b /dev/sda4 -v -v -v
```

指定1个参数“-v”是第1级模式，指定2个是第2级，依此类推，重复指定多次，显示的相关信息就更加详细

③ 安装lilo时，指定使用/etc/mylilo.conf设置文件：

```
# lilo -C /etc/mylilo.conf
```

④ 列出所有的系统内核映射文件：

```
# lilo -q
dos          *
linux
```

- ⑤ 把lilo安装在第1台SCSI硬盘的主引导记录，并指定设置文件、识别标签和备份文件：

```
# lilo -D linux -s /boot/boot.Backup
```

指定默认启动的
系统识别标签

将原来关机扇区备份到
/boot目录下的
boot.Backup文件

- ⑥ 删除/dev/hda的lilo程序：

```
# lilo -u /dev/had
```

lispmtopgm

常用指数：



(lisp to portable graymap)

功能说明：转换图像文件。

语 法：lispmtopgm [Lisp 图像文件]

包名称：netpbm-progs (Red Hat Linux/Fedora Core), netpbm (Mandrake Linux)。

补充说明：执行lispmtopbm命令，读取指定的Lisp主机使用的图像文件，并把该文件转成Portable Graymap（默认最后的扩展名为.pgm）格式后，输出到标准输出设备。虽然在Lisp主机上可看见彩色的图像文件，但由于该图像文件并不包含色彩配置表，因此转文件之后色彩仅剩灰度，而非彩色。

参 数：无。

范 例:

把Lisp主机的图像文件fromlispmachine转成Portable Graymap格式:

```
# lispmtoppm fromlispmachine > nowitsagraymap.pgm
```

ln (link)

常用指数: 

功能说明: 连接文件或目录。

语 法: ln [-bdfinsv] [-S <字尾备份字符串>] [-V <备份方式>] [--help] [--version] [来源文件或目录] [目的文件或目录] 或
ln [-bdfinsv] [-S <字尾备份字符串>] [-V <备份方式>] [--help] [--version] [来源文件或目录...] [目的目录]

包 名 称: coreutils。

相关命令: cp, lndir, symlinks。

补充说明: ln命令用在连接文件或目录, 如同时指定两个以上的文件或目录, 且最后的目的地是一个已经存在的目录, 则它会把前面指定的所有文件或目录复制到该目录中。若同时指定多个文件或目录, 而最后的目的地并非是一个已存在的目录, 则会出现错误信息。默认ln命令会产生硬链接, 由于硬链接之间拥有相同的inode, 故其来源的文件或目录必须已经存在。



参 数:

-b或--backup	删除、覆盖目的文件之前先备份，备份的文件则会在字尾加上一个备份字符串
-d、-F或--directory	建立目录的硬链接。默认ln命令无法建立目录的硬链接，也无法建立指向目录的符号链接的硬链接。本参数可建立指向目录的符号链接的硬链接，但仍旧不能直接建立目录的硬链接
-f或--force	强行建立文件或目录的连接，不论目的文件或目录是否存在
-i或--interactive	覆盖已有文件之前先询问用户
-n或--no-dereference	把符号链接的目的目录视为一般文件。当目的目录是某个目录的符号链接时，ln命令默认会把该符号链接视为目录看待，而在其内建立来源文件或目录的连接。如果加上这个参数，则ln命令会把该符号链接视为一般文件，因此将返回文件已存在，或是直接删除、覆盖它
-s或--symbolic	对来源文件建立符号链接，而非硬链接
-S <字尾备份字符串> 或--suffix=<字尾备份字符串>	用“-b”参数备份目的文件后，备份文件的字尾会被加上一个备份字符串，默认的字尾备份字符串是符号“~”，可通过“-S”参数来改变备份它
-v或--verbose	显示命令执行过程
-V<备份方式>或--version control=<备份方式>	用“-b”参数备份目的文件后，备份文件的字尾会被上一个备份字符串，加个字符串不仅可用“-S”参数变更，当使用“-V”参数指定不同的备份方式时，也会产生不同字尾的备份字符串。备份方式指定为“numbered”（或“t”）将使用备份编号，在字尾会加上“~1~”字符串，且数字编号依次递增。备份方式指定为“simple”（或“never”）将使用简单备份，默认的字尾备份字符串是“~”，也能通过“-S”参数来设置。备份方式指定为“existing”（或“nil”）将使用当前方式，程序会先检查是否存在着备份编号，若有则采用备份编号，若无则采用简单备份
--help	显示帮助
--version	显示版本信息

范 例:

① 建立source文件的硬链接,并命名为target:

```
# ln source target
```

② 建立source文件的符号链接,并命名为target:

```
# ln -s source target
```

③ 建立source文件的硬链接,并命名为target,如果要覆盖已有的文件,则先询问用户:

```
# ls ← 列出当前目录的文件
source target ← target文件已经存在
# ln source target
ln: 'target' 文件已存在 ← target文件已经存在,所以执行不成功
# ln -i source target ← 以交谈的方式建立符号链接
ln: 是否替换 `target'? y ← 先询问用户的意见,如果要覆盖文件,按 Y 确定
```

④ 建立source文件的硬链接,并命名为target,如果要覆盖已有的文件,不询问用户意见便直接覆盖:

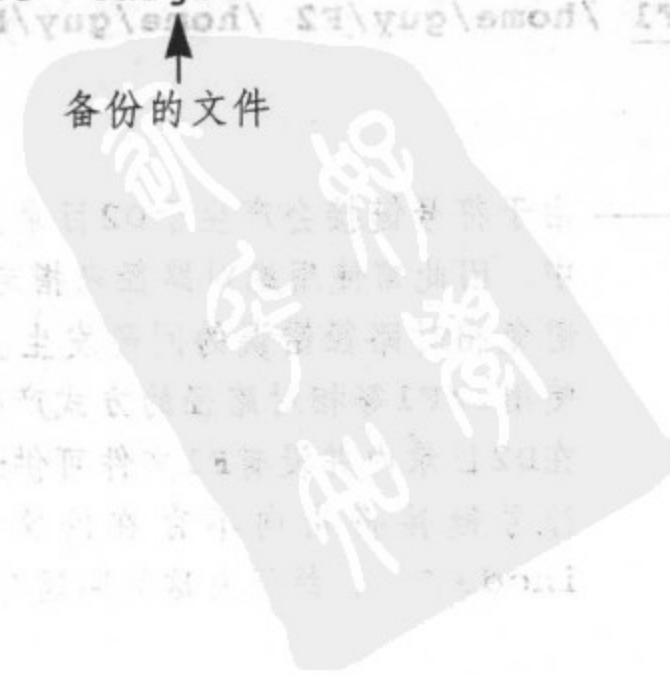
```
# ln -f source target
```

⑤ 建立source文件的硬链接,并命名为target,如果要覆盖已有的文件,先备份目的文件后再覆盖:

```
# ln -b source target
```

```
# ls
source target target~
```

↑
备份的文件



- ⑥ 建立source文件的硬链接，并命名为target，如果要覆盖已有的文件，先备份目的文件再覆盖文件，并指名备份文件名称的字尾为“_backup”：

```
# ln -b -s _backup source target
```

```
# ls
```

```
source target target_backup
```

|
备份的文件

- ⑦ 建立source文件的硬链接，并命名为target，如果要覆盖已有的文件，先备份目的文件再覆盖文件，并指定备份文件命名采用字尾编号的方式：

```
# ln -b -v t source target ← 建立第1个硬链接
```

```
# ls
```

```
source target target.~1~
```

|
第1个备份文件

```
# ln -b -v t source target ← 建立第2个硬链接
```

```
# ls
```

```
source target target.~1~ target.~2~
```

|
第2个备份文件

- ⑧ 现有文件F1、F2、F3与目录D1，欲在D2下建立其符号链接：

```
# ln -s /home/guy/F1 /home/guy/F2 /home/guy/F3
```

```
/home/guy/D1 D2
```

由于符号链接会产生于D2目录里，而非当前目录中，因此需使用绝对路径来指定来源文件或目录，避免相对路径错误的问题发生。举例来说，假如使用./F1等相对路径的方式产生符号链接，由于在D2目录内并没有F1文件可供连接，所以产生的符号链接将指向不存在的文件。硬链接则因inode相同，故没有这类问题存在

⑨ 假设D3是一个目录的符号链接，现在欲建立其硬链接D4：

```
# ln -d D3 D4
```



假如不加上这项参数，则ln命令会报告错误信息

ln (link directory)

常用指数：



功能说明：连接目录内容。

语法：ln [-ignorelinks] [-silent] [来源目录] [目的目录]

包名称：XFree86。

相关命令：cp, ln, symlinks。

补充说明：执行ln命令，可一口气把来源目录下的文件和子目录统统建立起相互对应的符号链接。若没指定目的目录，则会在当前目录中建立来源目录下所有内容的符号链接，来源目录必须用绝对路径标示。

参数：

-ignorelinks	直接建立符号链接的符号链接。当ln命令遇到符号链接时，默认会维持该连接指向原始文件的路径，例如符号链接lucifer指向文件satan，执行ln命令建立的符号链接lucifer依旧会指向文件satan。假设使用这项参数，则新建立的符号链接lucifer会指向符号链接lucifer，而非原始文件satan
-silent	不显示命令执行过程。若来源目录下包含子目录，则ln命令每深入一个子目录里面，就会返回该目录的路径名称。此参数可关闭这些返回的信息

范 例:

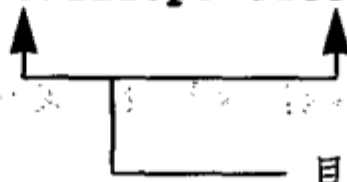
① 把目录melpomene下的所有文件，包括子目录在内，统统放在当前目录里，个别建立符号链接：

```
# ln -s melpomene .
```

② 把目录calliope下的所有文件与子目录在内，统统放在目录euterpe里，个别建立符号链接：

```
# ln -s /home/olympus/calliope euterpe
```

请用绝对路径
指定来源目录



目录calliope和euterpe的结构将完全相同，就像分身一样。即euterpe目录为calliope目录的投影

③ 把目录melpomene下的所有文件，包括子目录在内，统统放在当前目录里，个别建立符号链接。建立时如遇到符号链接的文件，则建立指向该符号链接的符号链接：

```
# ln -s -ignorelinks melpomene .
```

locate

常用指数:



功能说明: 查找文件。

语 法: locate [-d <数据库文件>] [--help] [--version] [范本样式...]

包 名 称: slocate。

相关命令: find, slocate, updatedb, whereis, which。

补充说明： locate命令用于查找符合条件的文件，它会去存储文件与目录名称的数据库内，查找合乎范本样式条件的文件或目录。可以使用特殊字符（如“*”或“?”等等）来指定范本样式，例如指定的范本为“twi*light”，locate命令会找出所有起始字符串为twi，且结尾字符串是“light”的文件或目录，如名称为twi-light的文件。假如有目录的名称为twi_light，则它会列出该目录下包括子目录在内的所有文件。

注意特殊字符并不特别看待符号“/”与“.”，也就是说，将会一起找到类似twi.light的文件，或者是twi/light的路径名称（如果有这种文件或路径名称存在）。由于locate命令是从数据库里面查找文件，因此速度较find命令快上许多，但是对于刚新增或已删除的文件、目录，locate命令便无法自行察觉，所以必须经常以updatedb命令维持该数据库的内容正确。

参 数：

-d<数据库文件>或 --database=<数据库文件>	设置locate命令使用的数据库。locate命令默认的数据库位于/var/lib/slocate目录里，文件名为slocate.db，可使用这个参数另行指定
--help	显示帮助
--version	显示版本信息

范 例：

① 利用默认的数据库，查找 favorite_game 文件：

```
# locate favorite_game
/home/olympus/favorite_game
```

└── 该文件存放的目录名称

② 使用数据库文件phoebe.db，查找artemis.mov文件：

```
# locate -d /root/phoebe.db artemis.mov
```

login

常用指数: 

功能说明: 登录系统。

语 法: login

包 名 称: util-linux。

相关命令: last, logout, mingetty。

补充说明: login命令让用户登录系统,也可借由它的功能随时更换登录身份。在Slackware发行版中,可在命令后面附加欲登录的用户名称,它会直接询问密码,等待用户输入。当/etc目录里含有名称为nologin的文件时,系统只允许root账号登录系统,其他用户一律不准登录。


参 数: 无。

范 例:

离开现在使用的身份,再重新登录系统:

```
# login
```

logname

常用指数: 

功能说明: 显示用户名称。

语 法: logname [--help] [--version]

包 名 称: coreutils。

相关命令： usermod。

补充说明： 执行logname命令，它会显示当前用户的名称。

参 数：

--help	显示帮助
--verison	显示版本信息

范 例：

查询自己登录系统的账号名称：

`$ logname` ← 登录系统时使用的账号名称，就算经过su命令改变身份，还是会显示最初登录系统时的用户名称

slayer

logout

常用指数：



功能说明： 退出系统。

语 法： logout

包 名 称： bash。

相关命令： login。

补充说明： logout命令让用户退出系统，其功能和login命令相互对应。

参 数： 无。



范 例:

直接执行 `logout` 命令即可退出系统:

```
# logout
```

logrotate (log rotate)

常用指数:



功能说明: 管理记录文件。

语 法: `logrotate [-?dfv] [-s <状态文件>] [--usage] [设置文件]`

包 名 称: `logrotate`。

补充说明: 使用 `logrotate` 命令可轻松管理系统所产生的记录文件。它提供自动替换、压缩、删除和邮寄记录文件，每个记录文件都可被设置成每日、每周或每月处理，也能在文件太大时立即处理。必须自行编辑、指定设置文件，默认的设置文件存放在 `/etc` 目录下，文件名称为 `logrotate.conf`。

参 数:

<code>-?或--help</code>	显示帮助
<code>-d或--debug</code>	详细显示命令执行过程，便于debug或了解程序执行的情形。在debug模式中，没有任何记录文件或状态文件会被修改及变动
<code>-f或--force</code>	强行启动记录文件维护操作，即使logrotate命令认为不需要也一样。当刚在设置文件中加入新选项，或是手动删除了某些记录文件，却需要记录文件能够继续正确地维护并记录系统情形时，就可以使用这项参数
<code>-s <状态文件>或 --state=<状态文件></code>	使用指定的状态文件。如果有不同的用户采用不一样的设置文件，则可搭配这个参数，个别指定状态文件记录变动信息。默认的状态文件位于 <code>/var/lib</code> 目录里，文件名称为 <code>logrotate.status</code>

-v或--verbose	显示命令执行过程
--usage	显示命令基本用法

范 例:

① 进行记录文件维护操作，使用在/tmp目录下的logset.config设置文件，并把状态文件存放于/root目录里的110599.status文件：

```
# logrotate -s /root/110599.status /tmp/logset.config
```

② 使用/etc/logrotate.conf设置文件，强制执行记录文件维护工作：

```
# logrotate -f /etc/logrotate.conf
```

look

常用指数:



功能说明: 查询单词。

语 法: look [-adf] [-t <字尾字符串>] [字首字符串] [字典文件]

包 名 称: util-linux。

相关命令: ispell, spell。

补充说明: look命令用于英文单词的查询，仅需给予它欲查询的字首字符串，它会显示所有开头字符串符合该条件的单词。默认的字典文件是/usr/dict目录里的words文件，可以另行指定。

参 数:

-a	在某些UNIX版本的/usr/dict目录下有另一个字典文件 web2, 利用此参数可使用此字典来查询单词
-d	只比对英文字母和数字, 其余一概忽略不予比对
-f	忽略字符大小写的差别
-t <字尾字符串>	设置字尾字符串。只有同时符合字首与字尾字符串的单词, 才会显示出来

范 例:

列出所有字首为“aid”的英文单词:

```
# look aid
```

```
aid
```

```
Aida
```

```
aide
```

```
aided
```

```
Aides
```

```
aiding
```

```
aids
```

losetup (loop setup)

常用指数:



功能说明: 设置循环设备。

语 法: losetup [-d] [-e <加密方式>] [-o <平移数目>] [循环设备代号] [文件]

包 名 称: losetup。

补充说明: 循环设备可把文件虚拟成块设备 (Block Device), 借以模拟整个文件系统, 让用户得以将其视为硬盘、光驱或软驱等设备, 并加载当作目录来使用。losetup 命令用于文件、块设备和循环设备的连接及设置, 假设仅给予循环设备代号而没有其他参数, 则会显示该循环设备当前的情况。

参 数:

-d	卸除设备。终止该循环设备与文件或块设备的连接, 循环设备的代号通常是/dev/loop0~ /dev/loop7
-e <加密方式>	启动加密编码。编码后的数据具有相当程度的保密效果, 加密方式有三种: “NONE”、“XOR”和“DES”(Data Encryption Standard), 其中“NONE”是默认值, 即不启动加密编码, “XOR”的运算编码速度较快, 却容易被破解。“DES”具有良好的保密效果, 但需耗费较多的时间来运算
-o <平移数目>	设置数据平移的数目。当循环设备与文件或块设备相互连接时, 两者之间的数据可先平移后再连接。可以使用这项参数设置数据的平移量, 单位以 Byte 计算

范 例:

以下是循环设备的使用范例:

```
# losetup /dev/loop0 ← 查看/dev/loop0循环设备的情况
loop: can't get info on device /dev/loop0:没有这个设备或地址

# dd if=/dev/zero of=/file bs=1k count=100 ← 建立一个拥有100
                                                个块, 每个块大小
                                                为1KB的文件

...

# losetup /dev/loop0 /file ← 建立循环设备
# losetup /dev/loop0 ← 查看/dev/loop0循环设备的情况
/dev/loop0: [0301]:13 (/file) offset 0,no encryption
# mke2fs /dev/loop0 100
# mount -t ext2 /dev/loop0 /mnt/loop
# umount /mnt/loop
```

↑
该设备还没有建立


← 建立一个拥有100个块, 每个块大小为1KB的文件

┌
└ 如同一般的硬盘
执行格式化、加载、卸载等操作

```
# losetup -d /dev/loop0 ← 卸载/dev/loop0循环设备
# losetup /dev/loop0 ← 查看/dev/loop0循环设备的情况
loop: can't get info on device /dev/loop0:没有这个设备或地址
```

↑
该设备已经成功卸载

lpc (line printer control)

常用指数: 

功能说明: 控制打印机。

语法: lpc [? <参数>...] [status <队列名称>]

包名称: LPRng。

相关命令: lpd, lpq, lpr, lprm, tunelp。

补充说明: lpc命令提供系统管理者查询打印机的情况。

参数:

? <参数>或help<参数>	显示帮助。可以指定一个或多个参数,查看其帮助。 若不指定任何参数,则会列出所有可用的参数
status <队列名称>	显示打印机队列的打印操作的状态

范例:

① 执行lpc命令,会出现提示符号,可用稍后所介绍的操作命令控制打印机:

```
# lpc
```

② 查询现在打印机的状态:

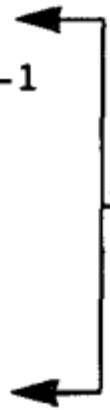
lpc status

hplj5:

```
printer is on device 'smb' speed -1
queuing is enabled
printing is enabled
no entries
daemon present
```

lp:

```
printer is on device 'parallel' speed -1
queuing is enabled
printing is enabled
2 entries
daemon present
```



此主机连接两台打印机

操作命令:

? <参数>或help <参数>	显示帮助。可以指定一个或多个参数查看其帮助。若不指定任何参数，则会列出所有可用的参数
exit或quit	离开lpc命令列模式
status <all>/<打印机名称>	显示本地主机的打印操作的状态。可用“all”设置全部打印机；或直接指定某打印机的名称

lpd (line printer daemon)

常用指数:

**功能说明:** 提供打印机排队常驻服务。**语法:** lpd [-FV] [-L <记录文件>] [-p <通信端口>]**包名称:** LPRng。**相关命令:** lpc, lpq, lpr, lprm, tunelp。

补充说明：lpd是提供打印机打印服务的程序，它以常驻服务的类型管理各项打印操作，包括队列与排队。一般来说，lpd命令都在系统启动时执行，它会读取/etc目录里的打印机设置文件printcap，并根据其设置提供打印服务。

参 数：

-F	一般执行lpd时，会以后台模式启动。若加上这个参数，则将以前台模式启动。通常用于debug
-L <记录文件>	指定记录错误信息及debug信息的记录文件
-p <通信端口>	若要以特别的通信端口启动，则可加上此参数
-V	显示版本信息

范 例：

启动打印机常驻服务：

```
# lpd
```

lpq (line printer queue)

常用指数：



功能说明：显示打印操作。

语 法：lpq [-al] [-P<打印机名称>] [工作编号...]

相关命令：lpc, lpd, lpr, lprm, tunelp。

包 名 称：LPRng、cups。

补充说明：lpq命令会查看打印队列里还有哪些工作正在排队中，并将其显示出来。也可直接指定工作编号或用户名称，列出该编号的操作情形，或是该用户的所有打印工作。单独执行lpq命令，它会列出全部的操作。

参 数:

-a	显示所有打印机的工作
-l	详细显示该打印操作的相关信息。除了打印顺序、拥有者、工作编号、文件名称和大小之外，还会显示打印临时文件的编号
-P<打印机名称>	显示该打印机的所有工作。假设没有指定打印机的名称，则会列出系统默认打印机的工作，该默认值由 PRINTER 环境变量来设置

范 例:

① 显示打印队列的内容，查看当前有哪些工作正等待打印：

```
# lpq
```

```
lp is ready and printing
```

Rank	Owner	Job	Files	Total Size
active	root	8	cronset	113字节
1st	root	9	SETI/README	3040字节
2nd	oracle	10	Xrootenv.0	767字节

② 显示打印机所有打印工作：

```
# lpq -l
```

```
lp is ready and printing
```

```
root: active          [job 008A78vLD3]
      cronset 113字节
```

```
root: 1st             [job 009AQEz1S7]
      SETI/README 3040字节
```


```
oracle: 2nd          [job 010Aa87Vnc]
      Xrootenv.0    767字节
```

打印临时
文件的编号

③ 显示打印机lp0打印队列的内容：

```
# lpq -P lp0
lp0 is ready and printing
Rank      Owner      Job    Files      Total Size
active    root       20     testfile    245字节
```

lpr (line printer)

常用指数：

功能说明：打印文件。

语法：lpr [-lpr] [-#<打印份数>] [-C <工作名称>] [-i<缩进列数>] [-P<打印机名称>] [文件...]

包名称：LPRng、cups。

相关命令：lpc, lpd, lpq, lprm, tunelp。

补充说明：lpr命令会把欲打印的文件先放到队列中排队等候，当系统与打印机完成连线或该打印机闲置时，就进行打印工作。若不指定任何文件名称，则lpr命令会从标准输入设备读取数据。

参数：

-#<打印份数>	设置打印份数。一般来说，打印服务器会将这项功能关闭，而改用复印机来达到相同的效果
-C <工作名称>、-J <工作名称>或-T <工作名称>	设置打印文件的名称。该名称会打印在摘要信息页
-l	将控制字符一并打印出来，且让分页控制字符无效
-p	使用pr命令将欲打印的文件格式化

-P<打印机名称>	指定打印机。假设没有指定打印机的名称，则会使用系统默认打印机来打印，该默认值由PRINTER环境变量或打印服务器来设置
-r	当欲打印的文件进入排队后，便将其删除。本参数无法和参数“-s”同时使用

范 例：

① 打印testfile文件到lp0打印机：

```
# lpr -P lp0 testfile
```

② 打印10份testfile文件：

```
# lpr -#10 testfile
```

③ 打印textfile文件，当打印的文件进入排队后，就将其删除：

```
# lpr -r testfile
```

lprm (line printer remove)

常用指数：



功能说明：删除打印工作。

语 法：lprm [-P<打印机名称>] [工作编号...]

包 名 称：LPRng、cups。

相关命令：lpc, lpd, lpq, lpr, tunelp。

补充说明: lprm命令会把正在队列里排队的打印操作删除,也可以指定用户名称、或工作编号删除打印操作。若不指定任何文件名称,则lprm命令将会删除当前阶级(Rank)标示为“active”的工作。给予的文件名为“-”,如果是一般用户,lprm命令会把属于该用户的打印工作取消。如果是系统管理者,则lprm命令将把全部的工作统统删除。

参 数:

-P<打印机名称>	指定打印机。假设没有指定打印机的名称,则会删除系统默认打印机的工作
-----------	-----------------------------------

范 例:

① 删除编号12的打印工作:

```
# lprm 12
```

② 删除lp0打印机的所有打印工作:

```
# lprm -P lp0
```

ls (list)

常用指数:



功能说明: 列出目录内容。

语 法: ls [-laAbBcCdDfFgGhHiklLmnNopqQrRsStuUvxX] [-I <范本样式>] [-T <跳格字数>] [-w <每行字符数>] [--block-size=<块大小>] [--color=<使用时机>] [--format=<列表格式>] [--full-time] [--help] [--indicator-style=<标注样式>] [--quoting-style=<引号样式>] [--show-control-chars] [--sort=<排序方式>] [--time=<时间戳>] [--version] [文件或目录...]

包名称: coreutils。

相关命令: dircolors, mdir。

补充说明: 执行ls命令可列出目录的内容, 包括文件和子目录的名称。它十分类似DOS的dir命令, 但功能更为强大。单独执行ls命令, 它会显示当前目录的内容。

参 数:

-l	每行仅显示一个文件或目录名称
-a或--all	显示所有文件和目录, 包括以“.”为名称开头字符的文件、当前目录“.”与上层目录“..”
-A或--almost-all	显示所有文件和目录, 包括以“.”为名称开头字符的文件, 但不含当前目录“.”与上层目录“..”
-b或--escape	显示脱离字符
-B或--ignore-backups	忽略备份文件和目录。不显示名称具有备份字尾字符串“~”号的文件或目录
-c	以更改时间 (change time) 排序, 显示文件和目录。配合参数“-l”会显示更改时间
-C	以由上至下、从左到右的直行方式显示文件和目录名称
-d或--directory	显示目录名称, 而非其内容
-D或--dired	用Emacs的模式产生文件和目录列表
-f	此参数的效果和同时指定“-aU”参数并关闭“-lst”参数的效果相同
-F或--classify	在执行文件、目录、Socket、符号链接、管道名称后面, 各自加上“*”、“/”、“=”、“@”、“ ”号
-g	此参数将忽略不予处理
-G或--no-group	不显示组名称。此参数需搭配“-l”参数使用, 其效果和指定“-o”参数相同
-h或--human-readable	用“k”、“M”、“G”来显示文件和目录的大小。此参数搭配“-l”、“-o”等参数使用

-H或--si	此参数的效果和指定“-h”参数类似，但计算单位是1 000字节，而非1 024字节
-i或--inode	显示文件和目录的inode编号
-I <范本样式>或 --ignore=<范本样式>	不显示符合范本样式的文件或目录名称
-k或--kilo字节	此参数的效果和指定“--block-size=1024”参数相同
-l	使用详细格式列表。加上这项参数之后，ls命令会将权限标示、硬链接数目、拥有者与组名称、文件或目录大小及变动时间一并列出
-L或--dereference	如遇到性质为符号链接的文件或目录，直接列出该连接所指向的原始文件或目录。除名称之外，其他诸如权限标示、硬链接数目、拥有者与组名称、文件或目录大小及变动时间等，都以所指向的原始文件或目录为准
-m	用“,”号区隔每个文件和目录的名称。以从左到右、由上至下的横列方式显示文件和目录名称，若配合参数“-l”或“-o”使用，将以放置在后面的参数为主
-n或--numeric-uid-gid	以用户识别码和组织识别码替代其名称。此参数需搭配“-l”、“-o”等参数使用
-N或--literal	直接列出文件和目录名称，包括控制字符。使用此参数ls命令会将控制字符视同普通字符，一并列出
-o	此参数的效果和指定“-l”参数类似，但不列出组名称或识别码
-p或--file-type	此参数的效果和指定“-F”参数类似，但不会在执行文件名称后面加上“*”号
-q或--hide-control-chars	用“?”号取代控制字符，列出文件和目录名称
-Q或--quote-name	把文件和目录名称以“”号标示起来
-r或--reverse	反向排序。用相反的顺序列出文件和目录名称
-R或--recursive	递归处理，将指定目录下的所有文件及子目录一并处理
-s或--size	显示文件和目录的大小，以块为单位
-S	用文件和目录的大小排序
-t	用文件和目录的变动时间（modification time）排序

-T <跳格字数>或 --tabsize=<跳格字数>	设置跳格字符所对应的空白字符数。默认值为8，可用这项参数改变它
-u	以最后存取时间排序，显示文件和目录。配合参数“-l”会显示存取时间
-U	列出文件和目录名称时不予排序
-v	文件和目录的名称列表以版本进行排序
-w <每行字符数>或 --width=<每行字符数>	设置每行的最大字符数
-x	以从左到右、由上至下的横列方式显示文件和目录名称
-X	以文件和目录的最后一个扩展名排序
--block-size=<块大小>	指定存放文件的块大小
--color<=使用时机>	在文件和目录清单加上色彩，便于区分各种类型。使用时机的默认值为“none”，即不显示任何色彩。仅加上这项参数而不指定使用时机，相当于设置该时机为“always”。至于“auto”只有在标准输出设备为TTY时，才能有效使用。色彩的变换需通过dircolors命令去设置
--format=<列表格式>	设置文件和目录的列表格式。格式设为“across”或“horizontal”、“commas”、“long”或“verbose”、“single-column”、“vertical”的效果，分别和指定“-x”、“-m”、“-l”、“-L”、“-C”参数相同
--full-time	列出完整的日期与时间。此参数需搭配“-l”、“-o”等参数使用，它会详细列出文件和目录的变动时间，包括周、秒及年份
--help	显示帮助
--indicator-style =<标注样式>	在文件和目录等名称后面加上标注，易于辨识该名称所属的类型为何。样式设为“classify”、“file-type”的效果，分别和指定“-F”、“-p”参数相同，默认值是“none”
--quoting-style= <引号样式>	把文件和目录名称以指定的引号样式标示起来。该样式可以设置为“literal”、“shell”、“shell-always”、“c”及“escape”
--show-control-chars	在文件和目录列表时，使用控制字符。此为默认值

<p>--sort=<排序方式></p>	<p>设置文件和目录列表的排序方式。方式设为“extension”、“none”、“size”、“time”、“version”、“status”、“atime”、“access”或“use”的效果，分别和指定“-X”、“-U”、“-S”、“-t”、“-v”、“-c”、“-u”参数相同</p>
<p>--time=<时间戳></p>	<p>用指定的时间戳取代变动时间。默认ls命令加上“-l”或“-o”等参数，所显示的是变动时间。可借由指定时间戳来改变它，时间戳可以是“atime”、“access”、“use”、“ctime”或“status”</p>
<p>--version</p>	<p>显示版本信息</p>

范 例：

① 列出当前目录所包含的文件和子目录的名称：

```
# ls
dir1 file1 file2 file3 file4~ file5
```

② 以每列一个文件或目录的方式，列出当前目录所包含的文件和子目录的名称：

```
# ls -l
dir1
file1
file2
file3
file4~
```

每行只列出一个文件或目录

③ 列出当前目录所有的文件和子目录的名称，并用双引号把文件或目录名称给标示起来：

```
# ls -Q
"dir1" "file1" "file2" "file3" "file4~"
```

- ④ 列出当前目录所有的文件和子目录的名称，并用“,”将文件或目录给区隔开来：

```
# ls -m
dir1, file1, file2, file3, file4~
```

- ⑤ 以反向的顺序，列出当前目录所包含的文件和子目录的名称：

```
# ls -r
file4~ file3 file2 file1 dir1
```

- ⑥ 列出当前目录所有的文件和子目录的名称，包含名称以“.”为起始字符的文件，但不包括“.”及“..”2个目录：

```
# ls -A
.fileA dir1 file1 file2 file3 file4~
```

↑
与上例比较，多列出这个文件

- ⑦ 列出当前目录所有的文件和子目录的名称，包含名称以“.”为起始字符的文件：

```
# ls -a
. .. .fileA dir1 file1 file2 file3 file4~
```

↑ ↑
与上例比较，多列出这2个目录

- ⑧ 除了备份文件和目录外，列出当前目录所有文件和子目录的名称：

```
# ls -B
dir1 file1 file2 file3 ← 此例没有列出file4~文件
```

⑨ 以详细格式列出当前目录所有的文件和子目录：

```
# ls -l
总计 1148
drwxr-xr-x 3 root  root  4096  6月 27 22:35 dir1
-rw----- 1 root  root  83855 6月 27 22:27 file1
lrwxrwxrwx 1 root  root   5      6月 27 22:23 file2 -> file1
-rwxr-xr-x 1 alkane alkane 156    6月 27 22:30 file3
-rw-r--r-- 1 root  root  1073152 6月 27 22:30 file4~
```

⑩ 除了组名称之外，以详细格式列出当前目录所有的文件和子目录：

```
# ls -o
总计 1148
drwxr-xr-x  3 root      4096  6月 27 22:35 dir1
-rw-----  1 root      83855 6月 27 22:27 file1
lrwxrwxrwx  1 root          5 6月 27 22:23 file2 -> file1
-rwxr-xr-x  1 alkane    156  6月 27 22:30 file3
-rw-r--r--  1 root    1073152 6月 27 22:30 file4~
```

↑
不列出组名称

⑪ 除了组名称之外，以详细格式列出当前目录所有的文件和子目录，并显示完整的日期与时间：

```
# ls -o --full-time
总计 1148
drwxr-xr-x 3 root  4096 2003-06-27 22:35:18.000000000 +0800 dir1
-rw----- 1 root 83855 2003-06-27 22:27:00.000000000 +0800 file1
lrwxrwxrwx 1 root   5 2003-06-27 22:23:51.000000000 +0800 file2 ->
file1
-rwxr-xr-x 1 alkane 156 2003-06-27 22:30:08.000000000 +0800 file3
-rw-r--r-- 1 root 1073152 2003-06-27 22:30:45.000000000 +0800 file4~
```

列出年、月、日、时、分、秒

⑫ 以详细格式并以可读性较高的方式，列出当前目录所有的文件和子目录：

```
# ls -hl
总计 1.1M
drwxr-xr-x 3 root root 4.0k 6月 27 22:35 dir1
-rw----- 1 root root 82k 6月 27 22:27 file1
lrwxrwxrwx 1 root root 5 6月 27 22:23 file2-> file1
-rwxr-xr-x 1 alkane alkane 156 6月 27 22:30 file3
-rw-r--r-- 1 root root 1.0M 6月 27 22:30 file4-
```

以“k”、“M”来显示文件或目录大小

⑬ 以详细格式列出当前目录所有的文件和子目录，并以用户识别码和组织识别码来代替其名称：

```
# ls -nl
总计 1148
drwxr-xr-x 3 0 0 4096 6月 27 22:35 dir1
-rw----- 1 0 0 83855 6月 27 22:27 file1
lrwxrwxrwx 1 0 0 5 6月 27 22:23 file2-> file1
-rwxr-xr-x 1 500 500 156 6月 27 22:30 file3
-rw-r--r-- 1 0 0 1073152 6月 27 22:30 file4-
```

改以识别码列出

⑭ 以详细格式列出当前目录所有的文件和子目录，如遇到为符号链接的文件或目录，则直接列出该连接所指向的原始文件或目录：

```
# ls -Ll
总计 1236
drwxr-xr-x 3 root root 4096 6月 27 22:35 dir1
-rw----- 1 root root 83855 6月 27 22:27 file1
-rw----- 1 root root 83855 6月 27 22:27 file2
-rwxr-xr-x 1 alkane alkane 156 6月 27 22:30 file3
-rw-r--r-- 1 root root 1073152 6月 27 22:30 file4
```

⑮ 列出当前目录所有的文件和子目录的名称，并在执行文件、目录、Socket、符号链接及管道名称后面加上特殊标记：

```
# ls -F
dir1/  file1  file2@  file3*  file4~
```

执行文件加上 “*”
符号链接加上 “@”
目录加上 “/”

⑯ 显示当前目录所有文件和子目录的 inode：

```
# ls -li
244075 dir1 164675 file1 164678 file2 164680 file3
164676 file4~
```

⑰ 以块为单位，显示当前目录所有文件和子目录的大小：

```
# ls -ls
总计 1148
 4 dir1 88 file1 0 file2 4 file3 1052 file4~
```

⑱ 以空间大小为顺序，列出当前目录所有文件和子目录的名称：

```
# ls -S
file4~ file1 dir1 file3 file2
```

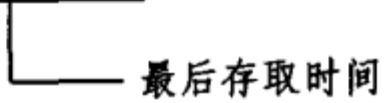
⑲ 以变动时间为顺序，并用详细的格式列出当前目录所有文件和子目录：

```
# ls -tl
总计 1148
drwxr-xr-x 3 root root 4096 6月 27 22:35 dir1
-rw-r--r-- 1 root root 1073152 6月 27 22:30 file4~
-rwxr-xr-x 1 alkane alkane 156 6月 27 22:30 file3
-rw----- 1 root root 83855 6月 27 22:27 file1
lrwxrwxrwx 1 root root 5 6月 27 22:23 file2 -
```

变动时间

② 以最后存取时间为顺序，并用详细的格式列出当前目录所有文件和子目录：

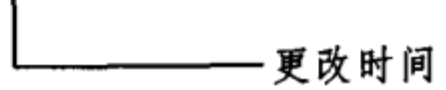
```
# ls -ul
总计 1148
lrwxrwxrwx 1 root root 5 6月 27 23:48 file2-> file1
-rwxr-xr-x 1 alkane alkane 156 6月 27 22:58 file3
-rw----- 1 root root 83855 6月 27 22:57 file1
drwxr-xr-x 3 root root 4096 6月 27 22:35 dir1
-rw-r--r-- 1 root root 1073152 6月 27 22:25 file4~
```



最后存取时间

② 以更改时间为顺序，并用详细的格式列出当前目录所有文件和子目录：

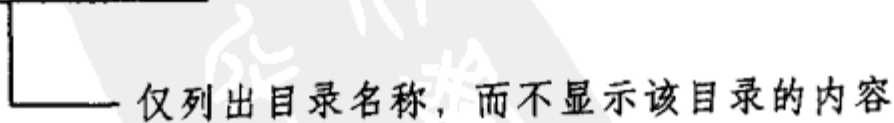
```
# ls -cl
总计 1148
-rwxr-xr-x 1 alkane alkane 156 6月 27 23:38 file3
-rw----- 1 root root 83855 6月 27 23:35 file1
drwxr-xr-x 3 root root 4096 6月 27 22:35 dir1
-rw-r--r-- 1 root root 1073152 6月 27 22:30 file4~
lrwxrwxrwx 1 root root 5 6月 27 22:23 file2 ->
file1
```



更改时间

② 某个目录里含有名称为dir1、dir2及dir3目录，还有名称为doc1与doc2的文件。今欲列出所有以字母d为名称字首的文件和目录，但无须列出目录的内容：

```
# ls -d d*
dir1 dir2 dir3 doc1 doc2
```



仅列出目录名称，而不显示该目录的内容

lsattr (list attribute)

常用指数:



功能说明: 显示文件属性。

语法: lsattr [-adlRvV] [文件或目录...]

包名称: e2fsprogs。

相关命令: chattr, mattrib。

补充说明: 用chattr命令改变文件或目录的属性, 可执行lsattr命令查询其属性。

参数:

-a	显示所有文件和目录, 包括以“.”为名称开头字符的文件、当前目录“.”与上层目录“..”
-d	显示目录名称, 而非其内容
-l	此参数当前没有任何作用
-R	递归处理, 将指定目录下的所有文件及子目录一起处理
-v	显示文件或目录版本
-V	显示版本信息

范例:

① 显示当前目录下所有文件的属性:

```
# lsattr
```


② 显示当前目录下所有文件及目录的属性：

```
# lsattr -a
```

③ 显示/home/metalgear 目录下的所有文件和子目录的属性：

```
# lsattr -R /home/metalgear
```

④ 显示 /home/metalgear 目录的属性：

```
# lsattr -d /home/metalgear
```

lsb_release

常用指数：



功能说明：显示 Linux 发行版本的相关信息。

语 法：lsb_release [-acdhirsv]

包名称：redhat-lsb (Red Hat Linux/Fedora Core), lsb-release (Mandrake Linux)。

相关命令：uname。

补充说明：现今主要的Linux发行版（如Red Hat Linux/Fedora Core、Mandrake Linux）都已支持LSB (Linux Standard Base)。通过此命令可查询当前使用Linux发行版的相关信息，如版本编号、发行厂商、代号、简要介绍等。



参 数:

-a或--all	显示所有的信息
-c或--codename	显示发行版的代号
-d或--description	显示发行版简介
-h或--help	显示帮助
-i或--id	显示发行版的名称
-r或--release	显示发行的版本号
-s或--short	显示简要的信息
-v或--version	显示版本

范 例:

① 显示LSB全部的信息:

```
# lsb_release -a
LSB Version:    1.3
Distributor ID: Fedora Core
Description:    Fedora Core release 3 (Heidelberg)
Release:        3
Codename:       (Heidelberg)
```

② 查询LSB版本:

```
# lsb_release -v
LSB Version:    1.3
```

lsmod (list modules)

常用指数: 

功能说明: 显示已载入系统的模块。

语 法: lsmod

包 名 称: modutils。

相关命令: insmod, depmod, modinfo, modprobe, rmmod。

补充说明: 执行lsmod命令, 会列出所有已载入系统的模块。Linux操作系统的内核具有模块化的特性, 因此在编译 (Compile) 内核时, 无需把全部的功能统统放入内核。可以将这些功能编译成一个个单独的模块, 待有需要时再分别载入它们。


参 数: 无。

范 例:

查看当前系统已经加载多少模块包:

```
# lsmod
Module                Size  Used by  Not tainted
ppp                    19916  0      (autoclean) (unused)
slhc                    4268   0      (autoclean) [ppp]
serail                 18388   1      (autoclean)
autofs                 8964   1      (autoclean)
ne                      6512   1      (autoclean)
8390                   6244   0      (autoclean) [ne]
unix                   10076  117    (autoclean) [ ]
└─┬──────────┬──────────┬──────────┘
  模块名称   模块大小  使用次数   相关模块
```

lynx

常用指数: **功能说明:** 浏览互联网 (World Wide Web)。

语 法: lynx [-accept_all_cookies] [-anonymous] [-assume_charset=<识别码>:<MIME 名称>] [-assume_local_charset=<识别码>:<MIME 名称>] [-assume_unrec_charset=<MIME 名称>] [-auth=<账号名称>:<密码>] [-base] [-blink] [-book] [-buried_news] [-cache=<缓存数据数>] [-case] [-cfg=<设置文件>] [-child] [-color] [-cookies] [-cookie_file=<cookie 文件>] [-core] [-crawl] [-debug_partial] [-display=<显示功能变量>] [-dump] [-editor=<文本编辑程序>] [-emacskeys] [-enable_scrollback] [-error_file=<记录文件>] [-exec] [-fileversions] [-force_html] [-force_secure] [-forms_options] [-from] [-ftp] [-get_data] [-head] [-help] [-hiddenlinks=<处理方式>] [-historical] [-homepage=<网页地址>] [-image_links] [-index=<网页地址>] [-ismap] [-link=<起始编号>] [-localhost] [-localexec] [-mime_header] [-minimal] [-newsch-unksize=<新闻数目>] [-newsmaxchunk=<新闻数目>] [-nobrowse] [-nocc] [-nocolor] [-noexec] [-nofilerferer] [-nolist] [-nolog] [-nopause] [-noprint] [-noredir] [-noreferer] [-nosocks] [-nostatus] [-number_links] [-partial] [-partial_thres<=显示列数>] [-pauth=<账号名称>:<密码>] [-popup] [-post_data] [-preparsed] [-print] [-pseudo_inlines] [-raw] [-realm] [-reload] [-restrictions<=功能选项>,...] [-resubmit_posts] [-rlogin] [-selective] [-show_cursor] [-soft_dquotes] [-source] [-stack_dump] [-startfile_ok] [-tagsoup] [-telnet] [-term=<终端类型>] [-tlog] [-trace] [-traversal] [-underscore] [-use_mouse] [-useragent=<用户名称>] [-validate] [-verbose] [-version] [-vikeys] [-width=<每列字符数>] [文件或目录或 URL]


包名称: lynx。

相关命令: httpd, wget。

补充说明: lynx是一个网页浏览程序, 它提供在纯文本模式下浏览网页。可以指定文件、目录或路径的名称, 以及URL (Uniform Resource Locator), 阅读HTML (Hypertext Markup Language) 格式的文件, 或是浏览互联网。lynx具有交互式操作界面, 可以轻易地利用方向键和热键来操控网页的浏览。若所给予的文件名为“-”, 则lynx命令会从标准输入设备读取数据。

参 数:

-accept_all_cookies	接受所有的cookies。cookies是浏览器用来记载某些网页数据的标示, 例如当您上网购物时, 放置在购物篮、购物车里的商品, 其数据可能就是临时在cookies中
-anonymous	使用匿名账号
-assume_charset= <识别码>:<MIME 名称>	遇到该识别码就使用指定的MIME (Multi-Purpose Internet Mail Extensions) 名称
-assume_local_charset=MIME <识别码>:<MIME名称>	遇到该识别码的本地文件, 就使用指定的名称
-assume_unrec_charset= <MIME 名称>	使用指定的MIME名称替代无法确认的字符集
-auth=<账号名称>:<密码>	设置登录保密网页的账号和密码
-base	在使用“-source”参数进行转储操作时, 先放置该URL的注释及BASE标签到欲转储的纯文本或HTML格式文件里
-blink	在彩色模式中, 强制使用高反差的后台色彩。终端必须支持本参数才能正常使用
-book	使用浏览器的书签 (Bookmark) 页作为起始页。当书签页无法使用或是内容空白时, lynx命令还是会使用默认的起始页

-buried_news	查找相关新闻组，然后将其转成新闻链接。不建议使用此参数进行转换操作，因为任何含有电子邮件地址或使用uuencode (UNIX to UNIX En code) 编码的新闻都可能变成一堆乱码
-cache=<缓存数据数>	设置要放入内存缓存的文件数目，默认值为10
-case	在查找字符串时，分辨字符的大小写
-cfg=<设置文件>	另行指定设置文件的名称。默认的设置文件名称为lynx.cfg
-child	在起始页中按下  键就离开lynx，且不存储文件
-color	在标准的后台色彩中强制使用彩色模式
-cookies	设置存放cookies文件的文件头
-cookie_file=<cookie文件>	指定用来存储cookies的文件名称
-core	当执行lynx命令发生错误时，将错误信息转储成文件记录下来，以便debug
-crawl	配合参数“-traversal”使用时，lynx会把每一页都输出成文件存储起来。配合参数“-dump”使用时，也会输出每个网页，不过是输出至标准输出设备
-debug_partial	增加信息的延迟秒数，便于debug
-display=<显示功能变量>	设置X Window程序在显示方面的功能变量
-dump	把第1个读入的文件转储至标准输出设备，然后离开lynx程序
-editor=<文本编辑程序>	设置欲使用的文本编辑程序
-emacskeys	使用类似emacs编辑程序的按键定义
-enable_scrollback	启用通信程序的回卷键。不建议使用本参数，因为可能会造成兼容性方面的问题
-error_file=<记录文件>	当浏览网页发生错误时，会把传回的HTTP (Hyper-Text Transfer Protocol) 错误码记录在指定的文件里
-exec	启动本地主机执行程序的功能。通常本参数不予使用
-fileversions	包含本地主机VMS (Virtual Memory System) 目录里所有版本的文件

-force_html	强制设置起始页为HTML 格式
-force_secure	强制在SSL (Secure Sockets Layer) 的cookies上设置保密识别标志
-forms_options	使用新式的选项选单
-from	使用From文件头的传送功能
-ftp	关闭ftp功能
-get_data	用Get的方式从标准输入设备读取数据, 再输出至标准输出设备
-head	送出要求文件头的信息
-help	显示帮助
-hiddenlinks=<处理方式>	设置如何处理隐藏的连接。处理方式有三种: "merge"、"listonly" 和 "ignore", 默认值为 "listonly"
-historical	用符号 ">" 或 "-->" 当作注释的结尾
-homepage=<网页地址>	另外指定首页, 和执行lynx命令时的起始页不同
-image_links	显示所有图像文件的超链接
-index=<网页地址>	指定该网页为默认的索引文件
-ismap	当客户端的MAP存在时, 就把ISMAP连接包含进来
-link=<起始编号>	使用 "-crawl" 参数抓取网页时, lynx会把存储的文件依次编号存储, 例如 lnk1.dat、lnk2.dat...依此类推。您可以用这个参数设置文件名称的起始编号, 如果起始编号设成10, 则文件名称将从lnk10.dat 开始依次编排
-localhost	仅允许在本地主机浏览网页, 关闭指向远程主机的URL
-localexec	只允许执行本地主机的程序
-mime_header	包含MIME的文件头并强制进行转储操作
-minimal	对有效注释做最低限度的分析
-newschunksize= <新闻数目>	设置新闻组列出的新闻件数
-newsmaxchunk= <新闻数目>	设置新闻组能够接收的最大件数
-nobrowse	不浏览目录内容
-nocc	在寄发复制的电子邮件时, 不向用户显示提示符号

-nocolor	关闭彩色模式的支持
-noexec	关闭执行本地主机的程序功能，此为默认值
-nofilereferer	在URL文件中，关闭Referer文件头的传送功能
-nolist	在转储操作中，关闭列出连接清单的功能
-nolog	关闭邮寄错误信息给文件拥有者的功能
-nopause	关闭状态栏显示错误信息时的强制暂停功能
-noprint	关闭打印功能
-noredir	关闭从某个连接重新导向另一个连接的功能
-noreferer	关闭Referer文件头的传送功能
-nosocks	不使用代理服务器的Socks传输协定
-nostatus	关闭杂项信息的显示功能
-number_links	强制计算超链接数目
-partial	下载网页数据时，可先显示已下载部分，而无需等到整页下载完毕
-partial_thres=<显示行数>	设置显示的行数，当部分显示的行数已达到这个数目时，就重新整理画面
-pauth=<账号名称>:<密码>	设置登录代理服务器的账号和密码
-popup	用弹出窗口选择列表的方式，来显示单一选项选单
-post_data	用Post的方式从标准输入设备读取数据，再输出至标准输出设备
-preparsed	配合“-source”参数查看源代码时，显示HTML格式的说明
-print	开启打印功能，此为默认值
-pseudo_inlines	当输入列之中没有ALT控制字符时，加入伪装的ALT控制字符
-raw	使用8位字符转换或中、日、韩文模式的起始字符集，作为默认值
-realm	限定URL存取的范围
-reload	但第一次读取某个网页时，更新代理服务器上的缓存数据

-restrictions=<功能选项>	关闭指定的功能选项。单独执行本参数而不附加任何功能选项，就会列出可使用的功能选项清单。同时指定多种功能，两两之间请用“,”号加以区隔
-resubmit_posts	强制重新读取已浏览过的网页数据，而不从缓存中查找
-rlogin	关闭rlogin命令的功能
-selective	欲浏览目录内容时，该目录必须含有名称为.www_browsable的文件才能读取目录的内容
-show_cursor	显示游标，当游标在画面右下角时不会隐藏起来
-soft_dquotes	修正旧版本Netscape和Mosaic浏览器里“>”号的错误
-source	此参数的效果和指定“-dump”参数类似，但输出为HTML格式，而非格式化的文本文件
-stack_dump	关闭SIGINT清理操作
-startfile_ok	配合参数“-validate”使用时，lynx将允许非HTTP的起始文件或首页
-tagsoup	在分析操作时，采用“TagSoup”取代“SortaSGML”字符串
-telnet	关闭telnet功能
-term=<终端类型>	设置终端的类型
-tlog	启动追踪记录功能
-trace	启动互联网的追踪模式
-traversal	浏览所有和首页有超链接的网页
-underscore	在转储操作时，使用底线格式
-use_mouse	开启X终端的鼠标支持
-useragent=<用户名称>	另行设置用户名称
-validate	仅接受使用HTTP的URL
-verbose	显示图像文件的文件名称及注释
-version	显示版本信息
-vikeys	使用类似vi编辑程序的按键定义
-width=<每行字符数>	设置在转储操作时，每行的最大字符数。默认值为80个字符

范 例:

① 浏览www.kde.org网页，可用稍后所介绍的操作命令来浏览网页：

```
# lynx www.kde.org
```

网页的
内容

可用的
操作键

```

KDE Homepage - Conquer your Desktop! (p1 of 8)
Skip to Content | Location: KDE Homepage Settings | Sitemap | Help
KDE Homepage Language: [Choose your location] Go

Skip to Link Menu
Conquer your Desktop!

KDE is a powerful Open Source graphical desktop environment for Unix
workstations. It combines ease of use, contemporary functionality, and
outstanding graphical design with the technological superiority of the
Unix operating system.
[ Announcements | News | Applications ]

Latest Announcements

KOffice 1.3 Beta 2 Released
On June 18th 2003, the KDE Project released the second beta version of
KOffice 1.3. It comes with a lot of bugfixes and a couple of new
(NORMAL LINK) Use right-arrow or (return) to activate.
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
Help Options Print G)o M)ain screen Q)uit -s=search [delete]=history list

```

② 执行lynx时，以书签当做起始页，可选择要进入书签中的哪一个网页：

```
# lynx -book
```

③ 执行lynx时，指定pico为文本编辑器，并进入默认的首页：

```
# lynx -editor=pico
```

④ 浏览www.linux.org网页，并指定当下载网页时，可先显示已下载的部分，不需等到整个网页都下载完毕才显示：

```
# lynx -partial www.linux.org
```

⑤ 浏览www.linux.org网页，当读取网页时，需先更新代理服务器上的数据：

```
# lynx -reload www.linux.org
```

⑥ 浏览www.linux.org网页，程序执行时接受所有的cookies：

```
# lynx -accept_all_cookies www.linux.org
```

⑦ 浏览www.linux.org网页，并指定缓存500文件：

```
# lynx -cache=500 www.linux.org
```

⑧ 浏览www.linux.org网页，当读取网页时，就算本地端已有此网页的缓存，仍强制重新载入此网页：

```
# lynx -resubmit_posts www.linux.org
```

⑨ 浏览www.linux.org网页，并指定关闭编辑器及电子邮件的功能：

```
# lynx -restrictions=editor,mail www.linux.org
```

⑩ 执行lynx 连接到FTP网站：

```
# lynx ftp://linux.cis.nctu.edu.tw/
```

操作命令：

↑ 或 8	移到此网页中的上一个连接
↓ 或 2	移到此网页中的下一个连接
→、6、Enter 或 Ctrl + J	进入当前链接的网页
0	可输入要进入到此网页的第几个链接

← 或 4	回到上一页
Page Up、-、9 B 或 Ctrl + B	往上移一页
Page Down、+、3 1、 Ctrl + F 或 Space	往下移一页
(往上移半页
)	往下移半页
Insert 或 Ctrl + P	往上移2行
Del 或 Ctrl + N	往下移2行
Home、7 或 Ctrl + a	移到当前网页的起始处
End、1 或 Ctrl + E	移到当前网页的结尾处
Ctrl + R	重新载入此页
m	回到首页
G	载入指定的网页
Z	立刻中止网络传输
q	停止载入此页
Q 或 Ctrl + D	强制停止载入此页
/	查找此网页的字符串
N	继续查找下一个字符串
Ctrl + a	将当前网页加入书签中
Ctrl + V	查看书签的链接
R	查看书签时，删除书签
Ctrl + H	查看历史清单
I	显示当前网页所有的链接清单
V	查看当前网页中已经参观过的链接清单
E	利用文本编辑器编辑当前的网页
P	打印、邮寄或存储当前画面
Ctrl + D	存储当前连接到本地电脑
Ctrl + t	切换是否记录用户浏览网页的过程
;	显示用户浏览网页的过程

=	显示当前网页的相关信息
c	写信给此网页的作者
G	编辑当前网页的地址
Ctrl + l 或 Ctrl + w	重新显示画面
O	显示并设置lynx环境设置
K	显示每个组合键操作说明
H 、 ?	显示帮助

ms 对下面提供的链接。有感兴趣的读者可以点击本文的一个链接。链接在
 登录登录后显示，请勿复制之。链接地址

地址人自办网本网并密家港	<地址> d
地址人自办网本网并密家港	<地址> e
地址人自办网本网并密家港	<地址> f
地址人自办网本网并密家港	<地址> g
地址人自办网本网并密家港	<地址> h
地址人自办网本网并密家港	<地址> i
地址人自办网本网并密家港	<地址> j
地址人自办网本网并密家港	<地址> k
地址人自办网本网并密家港	<地址> l
地址人自办网本网并密家港	<地址> m
地址人自办网本网并密家港	<地址> n
地址人自办网本网并密家港	<地址> o
地址人自办网本网并密家港	<地址> p
地址人自办网本网并密家港	<地址> q
地址人自办网本网并密家港	<地址> r
地址人自办网本网并密家港	<地址> s
地址人自办网本网并密家港	<地址> t
地址人自办网本网并密家港	<地址> u
地址人自办网本网并密家港	<地址> v
地址人自办网本网并密家港	<地址> w
地址人自办网本网并密家港	<地址> x
地址人自办网本网并密家港	<地址> y
地址人自办网本网并密家港	<地址> z



mail

常用指数: **功能说明:** E-mail管理程序。**语 法:** mail [-iInNv] [-b <地址>] [-c <地址>] [-s <邮件主题>] [收信人地址...] 或 mail [-iInNv] [-f <邮件文件>] 或 mail [-iInNv] [-u <用户账号>]**包 名 称:** mailx。**相关命令:** fetchmail, mailq, mutt, pine。**补充说明:** mail是一个文本模式的邮件管理程序。操作的界面不像elm或pine那么容易使用,但功能还算完整。**参 数:**

-b <地址>	指定密件副本的收信人地址
-c <地址>	指定副本的收信人地址
-f <邮件文件>	读取指定邮件文件(默认为mbox)中的邮件
-i	不显示控制台发出的信息
-I	使用互动模式
-n	程序启动时,不使用mail.rc文件中的设置
-N	执行mail命令后,不显示邮件的标题(标题中包含了寄件地址、寄件时间与日期,以及邮件主题)
-s <邮件主题>	指定邮件的主题。若主题中有空白字符,记得要在前后加上双引号(“”)
-u <用户账号>	读取指定用户的邮件
-v	执行时,显示详细的信息

范 例:

① 执行mail命令，可利用稍后所介绍的操作命令收发邮件：

```
# mail
```

② 寄信给terry@mail.flag.com.tw，副本寄给alkane@mail.flag.com.tw

```
# mail terry@mail.flag.com.tw ← 输入信件主题，按 Enter 键结束
Subject: Test mail ← 输入信件的正文
This is message body... ← 在新的一行按 Ctrl + D 键结束邮
Cc: alkane@mail.flag.com.tw ← 件的编辑，在此行可输入副本
                               收信人地址
```

③ 寄信给terry@mail.flag.com.tw及cirware@mail.flag.com.tw，并指定邮件主题、副本收信人及密件副本收信人：

```
# mail -s "Test mail" -c alkane@mail.flag.com.tw -b sphenoid@mail.flag.com.tw
```

邮件主题

副本收信人

密件副本收信人

```
terry@mail.flag.com.tw, cirware@mail.flag.com.tw
```

↑
当同时寄给一个以上的收信人时，用“，”区隔开来

④ 读取/var/spool/mail/alknae邮件文件的信件：

```
# mail -f /var/spool/mail/alkane
```

⑤ 读取sphenoid用户的邮件：

```
# mail -u sphenoid
```

⑥ 将已经写好的信件内容寄给tery@mail.flag.com.tw:

```
# mail -s "Test mail" tery@mail.flag.com.tw < letter
```

↑
将letter文件当成信件内容

操作命令:

数字键或\$	按数字键选择邮件。若要选择最后一封邮件，请使用\$
h	显示邮件标题的列表
p <邮件编号>或t <邮件编号>	显示指定的邮件。若邮件内容超过一页，可按空白键切换至下一页
n、+ 或 ████	显示下一封邮件的内容
-	显示上一封邮件的内容
s <邮件编号> <文件>	将指定的邮件存为文件
d <邮件编号>	删除指定的邮件。若要挽回被删除的邮件，只要在结束程序之前都可以利用“u”命令挽回删除的邮件。或者以x来结束程序，则之前所输入的删除邮件命令将不会作用
u <邮件编号>	恢复被删除的邮件
m <收信人地址>	寄送新邮件
R <邮件编号>	恢复邮件给寄信人
r <邮件编号>	恢复邮件给寄信人，与该封邮件的其他收信人
! <命令>	执行指定的命令
cd <目录>	切换目录
fo <邮件目录或文件>	显示当前的邮件目录或文件（不加选项时），或切换至指定的邮件目录或文件。“%”表示用户邮件的系统文件，“&”表示收件文件（mbox）
q	将读过的邮件移至mbox，并删除已标示删除的邮件，然后结束程序
x	相较于“q”命令，此命令并不对邮件做任何更改即结束程序
?	显示操作命令的帮助

M

mailq (mail queue)

常用指数:



功能说明: 显示待发送邮件的列表。

语法: mailq [-v]

包名称: sendmail (Red Hat Linux/Fedora Core), postfix (Mandrake Linux)。

相关命令: fetchmail, mutt, pine。

补充说明: mailq可列出待发送邮件的列表, 包括邮件ID、邮件大小、邮件存储时间、寄信人、收信人, 以及邮件无法寄出的原因, 提供管理员参考。

参数:

-v	显示较详细的信息
----	----------

范例:

① 列出待发送邮件列表:

mailq

```

                Mail Queue (3 requests)
--Q-ID--  -- Size--  ---Q-Time---  -----Sender/Recipient--
QAA00514   9        Fri Jun 27 16:19   daemon
                (host map:lookup   (mail.flag.com.tw):deferred)
                ad@mail.flag.com.tw

QAA00504   9        Fri Jun 27 16:17   yolande
                (host map:lookup   (mail.flag.com.tw):deferred)
                haunt@mail.flag.com.tw

PAA00346   9        Fri Jun 27 15:00   root
                (host map:lookup   (mail.flag.com.tw):deferred)
                les@mail.flag.com.tw

```



M

make

常用指数: 

功能说明: 编译系统内核或模块。

语法: `make [-deiknpqrsStvw] [-C <目录>] [-f <文件>] [-I <目录>] [-j <工作编号>] [-l <平均负载>] [-o <文件>] [-W <目的文件>] [目的文件...]`

包名称: make。

相关命令: depmod, lilo, mkinitrd, patch, rpm, uname。

补充说明: make 用来编译或建立执行文件，或是更新软件组件。make 在执行时会去读取 makefile 中的步骤与设置。makefile 为 script 文件。基本上，makefile 会依次调用前置处理器、编译器 (cc 或 gcc)，以及链接器，将程序源代码逐步转为可执行文件。

参数:

-C <目录>	执行前，先切换至指定的目录
-d	执行时，显示与排错相关的信息
-e	优先使用环境变量。若不使用“-e”参数，则make会优先使用makefile中所定义的变量
-f <文件>	指定makefile文件。若不使用此参数，程序会依次去寻找GNUmakefile、makefile与Makefile等文件为makefile
-i	执行时，忽略所有的错误信息
-I <目录>	指定查找makefile的目录
-j <工作编号>	指定在执行make时，同时执行的工作编号
-k	尽可能完成所有的编译工作。例如，在编译某一模块失败时，仍然继续编译相关的其他模块

-l <平均负载>	只有在系统低于指定的平均负载时 (load average), 才会执行新的工作。若只输入“-l”参数, 则会取消上次设置的平均负载
-n	显示要执行的命令, 但实际上不会执行
-o <文件>	不要编译指定的文件
-p	显示makefile中的相关规则与参数, 然后按正常步骤编译
-q	不执行任何命令, 仅显示返回值
-r	不使用内建的编译规则
-s	执行时不显示任何信息
-S	取消“-k”参数的作用。这通常只会用在递归的make循环。例如, 外层循环设置了“-k”参数, 若内层循环不要使用“-k”, 则可在内层循环中设置“-S”参数
-t	执行时不会重建目的文件, 但会修改相关的版本信息, 使它如同已经重新编译过
-v	显示版本信息
-w	在编译前后, 分别显示工作目录。此参数能提供更详细的信息, 以防排错
-W <目的文件>	模拟重建目的文件, 但不会实际更改目的文件。若与“-n”参数一起使用, 可模拟整个编译的过程

范 例:

① 以下为实际编译kernel的程序:

```
# cd /usr/src/linux      ← 切换到放置kernel源代码的目录
# make config           ← 以互动方式让你选择编译的选项
    ↑
这是makefile的参数
...
# make dep              ← 链接程序码与函数库
...
# make clean            ← 删除不必要的模块和文件
...
# make bzImage          ← 开始编译系统内核
...
```

```
# make modules          ← 开始编译外挂模块
...
# make modules_install ← 安装编译完成的模块
...
# depmod -a            ← 分析模块相依关系，完成模块安装
```

man (manual page)

常用指数: 

功能说明: 在线查询命令。

语法: man [-acdfhKtwW] [-C <设置文件>] [-M <路径>] [-p <预处理程序>] [-P <浏览方式>] [-S <小节列表>] [小节名称] [帮助主题...]

包名称: man。

相关命令: help, info, less, manpath。

补充说明: Linux提供了许多帮助。这些帮助文件必须经过排版后才能阅读。man就是用来排版及显示帮助文件的工具程序。第一次显示某一帮助主题时，man会找出帮助文件并加以排版。排版后的帮助文件存成名称为“cat”+“原始文件名”的文件，且放在/var/cache/man (Mandrake Linux为/var/catman) 目录下。下一次再查询此主题时，man便会直接显示已排版过的帮助文件，以节省时间。

参数:

-a	man默认在找到第一个符合的帮助文件后就会停止查找。使用此选项可强迫man显示所有符合的帮助文件
-c	即使已存在排版过的帮助文件（即以前曾查询过此主题），仍会再执行一次排版的动作。此参数适用于已排版过的帮助文件损毁而需重建，或屏幕的行列数已经改变

-C <设置文件>	指定man的环境设置文件，默认的文件为/etc/man.config
-d	仅显示排错信息，不显示帮助内容
-f	显示系统命令与工具程序的简单帮助
-h	显示man的语法与参数帮助
-K	在所有的帮助文件中，查找包含关键字的帮助文件。此查找结果的响应可能会很慢，可配合“-s”参数来查找特定小节
-M <路径>	指定帮助文件的查找路径。若不指定此参数，则man会使用MANPATH环境变量的设置
-p <预处理程序>	指定要排版帮助文件前的处理程序，共有5个选项：“e (eqn)”、“g (grap)”、“p (pic)”、“t (tbl)”和“v (vgrind)”
-P <浏览方式>	指定浏览的方式。若不指定此参数，则使用MANPAGER或PAGER环境变量中的设置。默认的显示方式为/usr/bin/less -is
-S <小节列表>	<p>所有的帮助主题都分别隶属于不同的小节 (section)。小节的划分如下：</p> <ul style="list-style-type: none"> 小节1：用户命令 小节2：系统调用 小节3：程序库调用 小节4：设备 小节5：文件格式 小节6：游戏 小节7：杂项 小节8：系统命令 小节9：kernel内部命令 小节n：Tcl或Tk命令 <p>有些命令或程序可能有一个以上的主题，位于不同的小节。因此，可以在<小节列表>指定man 查找小节的顺序，列表中小节之间必须以冒号 (:) 隔开，例如，1:8:2:3:n</p>
-t	以/usr/bin/groff - Tps - mandoc来排版帮助文件
-w或--path	仅显示帮助主题的文件位置，并不显示主题内容
-W	功能与“-w”相同，各个文件分行显示。可利用信息转向，让其他命令或程序进一步处理

范 例:

① 显示vim的帮助文件:

```
# man vim
```

② 查找并显示所有小节中man的帮助文件:

```
# man -a man
```

③ 显示patch帮助文件, 并指定以more命令来浏览此文件:

```
# man -P /bin/more patch
```

④ 指定以 /home/alkane/man.config 设置文件来显示find帮助文件:

```
# man -C /home/alkane/man.config find
```

⑤ 显示umount命令的简要帮助:

```
# man -f umount
```

```
umount [mount]      (8) - umount file systems
```

```
umount [mount]      (2) - mount and unmount filesystems
```

⑥ 显示kill的帮助文件, 当查找帮助文件时先寻找第2小节, 然后再寻找小节1:

```
# man -S 2:1 kill
```

先查找小节2, 再查找小节1, 但不查找其他小节

⑦ 显示pine帮助文件的位置:

```
# man -w pine
/usr/man/man1/pine.1
```

⑧ 查找并显示含有raid字符串的帮助文件:

```
# man -K raid
/usr/man/man8/mdadm.8.gz? [ynq]
```

← 当找到此文件时, 按 **Y** 键可阅读此说明文件, 按 **N** 键不观看说明, 按 **Q** 键则停止继续查找

...

⑨ 将chown命令的帮助文件转换成chown.txt文本文件:

```
# man chown | col -b > chown.txt
```

操作命令: 当使用默认的浏览方式时, 其操作命令帮助如下:

E 、 Enter + E 、 Ctrl + N 或 Enter	往下移一行
Y 、 Ctrl + Y 、 K 、 Ctrl + K 或 Ctrl + P	往上移一行
F 、 Ctrl + F 、 Ctrl + V 或 Space	往下移一页
B 、 Ctrl + B 或 Esc V	往上移一页
D 或 Ctrl + D	往上移半页
U 或 Ctrl + U	往下移半页
G 、 < 或 Esc V	移至文件起始处
G 、 > 或 Esc V	移至文件结束处

Esc) 或 →	向右移8列
Esc (或 ←	向左移8列
/ <字符串>	往后查找指定的字符串
? <字符串>	往前查找指定的字符串
N	查找下一个字符串
n	反方向查找字符串
Esc U	取消查找的反白标示
m <字母>	可指定一个字母，以记忆当前的位置
\ <字母>	回到此字母指定的位置。其中“^”表示文件起始处，“\$”表示文件结束处
Q 、 ; Q 或 z z	结束程序
R 、 Ctrl + R 或 Ctrl + I	重新显示画面
H	显示操作帮助

manpath (manual path)

常用指数:



功能说明: 显示说明文件的查找路径。

语 法: manpath

包 名 称: man。

相关命令: man。

补充说明: 当执行man程序时，它会查找说明文件所在的路径，以打开适当的说明文件。寻找的路径可在设置文件/etc/man.config中设置。而利用manpath可查询说明文件的查找路径。

参 数: 无。

范 例:

显示说明文件的查找路径:

```
# manpath
/usr/kerberos/man:/usr/local/share/man:/usr/share/man:/usr/X11R6/man
```

mattrib (MS-DOS attribute)

常用指数:



功能说明: 更改或显示 MS-DOS 文件的属性。

语 法: mattrib [+/-][ahrs][-X] [文件...] 或 mattrib [-X] [文件...]

包 名 称: mtools。

相关命令: chattr, lsattr。

补充说明: mattrib为mtools工具命令, 模拟MS-DOS的attrib命令, 可更改MS-DOS文件的属性。

使用第一种语法时, 可更改MS-DOS文件属性。使用第二种语法时, 则可显示MS-DOS文件当前的属性。

参 数:

+/-	“+”表示打开指定的文件属性。“-”表示关闭指定的文件属性
a	备份属性。使备份程序可用来判断文件是否已经备份过
h	隐藏属性。在DOS下用dir命令查看文件时, 将看不到打开此属性的文件
r	只读属性。打开此属性后, 在DOS下无法修改或删除此文件
s	系统属性。打开此属性, 表示文件为MS-DOS操作系统的系统文件

-/	列出指定目录及子目录下所有文件的属性
-X	显示文件属性时，中间不插入空格。这是为了方便将结果直接用在script文件中

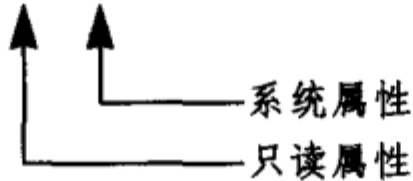
范 例：

① 显示MS-DOS文件dosfile的属性：

```
# attrib dosfile
```

② 将MS-DOS的dosfile文件设为具有系统及只读的属性：

```
# attrib +r +s dosfile
```



③ 关闭MS-DOS文件dosfile1及dosfile2的隐藏属性：

```
# attrib -h dosfile1 dosfile2
```

④ 列出dosdir目录及子目录下所有文件的属性：

```
# attrib -/ A:/dosdir
```

mbadblocks

常用指数：



(MS-DOS bad blocks)

功能说明： 检查MS-DOS文件系统的磁盘是否有损坏的扇区。

语 法： mbadblocks [磁盘代号]

包名称: mtools。

相关命令: badblocks。

补充说明: mbadblocks为mtools工具命令，可用来扫描MS-DOS文件系统的磁盘，并标示出损坏的扇区。磁盘代号后面需加冒号 (:)。

参数: 无。

范 例:

检查A驱中的磁盘:

```
# mbadblocks A:
```

```
plain_io: Input/output error
```

```
Bad cluster 2829 found ←找到损毁的扇区
```

mc (midnight commander)

常用指数:



功能说明: 菜单式的文件管理程序。

语 法: mc [-abcdfhkPstuUVx] [-C <参数>] [-l <文件>] [-v <文件>]
[目录]

包名称: mc。

补充说明: mc是用来管理文件的程序，它具有图形化的操作界面，可利用各个功能键轻松地操控文件或目录的一般操作，如复制、删除、移动、建立等等。

参 数:

-a	当mc程序画线时不用绘图字符画线
-b	使用单色模式显示
-c	使用彩色模式显示
-C <参数>	指定显示的颜色
-d	不使用鼠标
-f	显示mc函数库所在的目录
-h	显示帮助
-k	重设softkeys成默认值。当使用HP终端而功能键失去作用时, 请使用此参数
-l <文件>	在指定文件中存储ftpfs对话框的内容
-P	程序结束时, 列出最后的工作目录
-s	用慢速的终端模式显示, 在该模式下将减少大量的绘图及文本显示
-t	使用TEMPCAP变量设置终端, 而不使用默认值
-u	不用当前的shell程序
-U	用当前的shell程序
-v <文件>	使用mc的内部编辑器来查看指定的文件
-V	显示版本信息
-x	指定以xterm模式显示

范 例:

① 执行mc会出现以下操作画面, 可利用稍后所介绍的操作命令进行操作:

```
# mc
```


功能菜单，可按 **F9** 键
切换到此菜单，再用方
向键及 **Enter** 键执行功能

左边窗口所在
的目录名称

右边窗口所在
的目录名称

光标当前所
在文件或目
录的名称

操作小技巧

可在此处
输入命令

当前各个功能键
所表示的功能

另一窗口当前所在
的文件或目录名称

名字	大小	修改时间	名字	大小	修改时间
/.gstreamer-0.8	4096	8月 2 2006	/.gstreamer-0.8	4096	8月 2 2006
/.kde	4096	12月 12 2006	/.kde	4096	12月 12 2006
/.mc	4096	12月 21 10:29	/.mc	4096	12月 21 10:29
/.metacity	4096	8月 2 2006	/.metacity	4096	8月 2 2006
/.mozilla	4096	8月 21 2006	/.mozilla	4096	8月 21 2006
/.nautilus	4096	8月 2 2006	/.nautilus	4096	8月 2 2006
/.qt	4096	12月 12 2006	/.qt	4096	12月 12 2006
/.ssh	4096	9月 13 08:55	/.ssh	4096	9月 13 08:55
/.vmware	4096	12月 12 2006	/.vmware	4096	12月 12 2006
/.vnc	4096	8月 17 2006	/.vnc	4096	8月 17 2006
/Desktop	4096	9月 14 2006	/Desktop	4096	9月 14 2006
/Mail	4096	12月 21 10:36	/Mail	4096	12月 21 10:36
./ICEauthority	3238	12月 17 09:02	./ICEauthority	3238	12月 17 09:02
./Xauthority	493	12月 17 09:09	./Xauthority	493	12月 17 09:09
./bash_history	9845	12月 21 10:35	./bash_history	9845	12月 21 10:35
./bash_logout	24	9月 23 2004	./bash_logout	24	9月 23 2004
./bash_profile	191	9月 23 2004	./bash_profile	191	9月 23 2004
./bashrc	176	9月 23 2004	./bashrc	176	9月 23 2004
./cshrc	100	9月 23 2004	./cshrc	100	9月 23 2004

② 执行mc，并将工作目录切换至/etc目录：

```
# mc /etc
```

③ 利用mc的内部编辑器查看 textfile 文件：

```
# mc -v textfile
```

④ 使用单色模式执行 mc：

```
# mc -b
```

操作命令：以下列出mc较常用的操作按键，其中 **F1~F10** 随着执行命令的不同而有不一样的功能，下面列出的是刚进入mc时的功能：

↑ 或 Ctrl + P	将光标往上移
↓ 或 Ctrl + N	将光标往下移
Enter	若光标所在的是一个目录，则进入此目录
Tab	切换工作窗口
Page Up 或 Esc V	将光标往上移一页
Page Down 或 Ctrl + V	将光标往下移一页
Home	将光标移至当前窗口第1个文件或目录
End	将光标移至当前窗口最后1个文件或目录
F1	显示帮助
F2	针对文件或目录进行常用的操作
F3	查看文件的内容
F4	利用文本编辑器编辑文件
F5	复制文件或目录，会要求输入目的文件或目录的位置
F6	移动文件或目录，或是更改名称，会要求输入目的文件或目录，或是新的名称
F7	建立新目录，会要求输入目录名称
F8	删除文件或目录
F9	拉下上方的功能菜单
F10	离开mc程序
Ctrl + X I	建立文件的硬链接
Ctrl + X S	建立文件的符号链接
Ctrl + C	改变当前所在的目录位置
.	选取文件，可输入要选取的条件。如输入“text*”表示选取文件名称以text起始的文件
- 或 /	取消选取的文件，可输入要取消的条件。如输入“text*”表示取消选取文件名称以text起始的文件

mcd (MS-DOS change directory)

常用指数:



功能说明: 在MS-DOS文件系统中切换工作目录。

语 法: `mcd [目录名称]`

包 名 称: `mtools`。

相关命令: `cd`。

补充说明: `mcd`为`mtools`工具命令，可在MS-DOS文件系统中切换工作目录。若不加任何参数，则显示当前所在的磁盘与工作目录。

参 数: 无。

范 例:

将工作目录切换至`dosdir`:

```
# mcd dosdir
```

mcopy (MS-DOS copy)

常用指数:



功能说明: 复制MS-DOS文件。

语 法: `mcopy [-bmnPQt/] [源文件...] [目的文件或目录]`

包 名 称: `mtools`。

相关命令: `cp`。



补充说明： mcopy为mtools工具命令，可在MS-DOS文件系统中复制文件；或是在MS-DOS与Linux的文件系统之间复制文件。mcopy会根据MS-DOS磁盘代号（例如，“A:”）出现的位置来判断如何复制文件。

参 数：

-b	批次模式。若一次复制许多文件时，“-b”参数可提升复制的效率
-m	将目的文件修改时间设成源文件的修改时间
-n	覆盖其他文件时，不需进行确认而直接覆盖
-p	将目的文件属性设成源文件的属性
-Q	复制多个文件时，若发生错误，则立即结束程序
-t	转换文本文件。MS-DOS文本文件中的换行符号包含了两个字符（换行以及光标移至最前端），而Linux文本文件的换行符号只有一个字符。利用此参数，可在复制时将文本文件做适当的转换，以便在Linux中显示时不会出现乱码
-/	复制子目录以及其中的所有文件

范 例：

① 利用批次模式，将A:\的dosfile1及dosfile2等两个文件复制到现行目录下：

```
# mcopy -b a:/dosfile1 a:/dosfile2 .
```

② 将A:\dosfile1文件复制到当前目录下，若已存在同样文件名的文件，不需确认直接覆盖文件：

```
# mcopy -n a:/dosfile1 .
```

③ 将A:\dosfile1文件复制到当前目录下，并将源文件修改时间设为目的文件的修改时间：

```

# mdir a:
Volume in drive A is Linux (abbr=LINUX      )
Volume Serial Number is 1903-313E
Directory for A:/
dosfile          83855 2003-07-03  23:19  dosfile

```

↑
dosfile的修改时间为23:19

...

```
# mcopy a:/dosfile1 . ← 先以默认的方式复制
```

```
# ls -l
总计 88
```

```
-rw-r--r--  1 root  root  83855  7月 3  23:21  dosfile
```

↑
目的文件的修改时间为复制此文件的时间

```
# mcopy -m a:/dosfile . ← 加上-m参数，再一次复制此文件
```

```
# ls -l
总计 88
```

```
-rw-r--r--  1 root  root  83855  7月 3  23:19  dosfile
```

↑
目的文件的修改时间与源文件一致

④ 将MS-DOS软盘中dosdir目录下的全部文件复制到Linux的/root/temp目录下:

```
# mcopy -/ A:/dosdir /root/temp
```

↑ ↑
 若不指定目的目录，则复制到当前的目录
复制目录中所有的子目录与文件

⑤ 将linuxfile文件从Linux磁盘复制到MS-DOS文件系统的C:\temp目录中:

M

```
# mcopy linuxfile C:/temp
# mdir C:/temp
Volume in drive C has no label
Volume Serial Number is 12E2-265A
Directory for C:/temp
.          <DIR>      2003-07-03  12:14
..         <DIR>      2003-07-03  12:14
CARTOOL  ZIP    1577603 2003-07-03  16:07  Cartool.zip
linuxf-1          11285 2003-07-03  14:36  linuxfile
↑
  4 files                1 588 888 bytes
                        779 157 504 bytes free
```

长文件名会自动转换成符合MS-DOS 8.3格式的短文件名

md5sum (MD5 checksum)

常用指数:



功能说明: 计算与检查MD5函数值。

语法: md5sum [-bt] [--help] [--string <字符串>] [文件...]或
md5sum [-bt] [-c <文件>] [--status] [--help] [文件]

包名称: coreutils。

相关命令: cksum, sum。

补充说明: MD5为当前最常用来检查文件正确性的函数，可用来计算数据的校验和 (checksum) 值以验证数据正确与否。MD5函数的用途很广，例如，在发送E-mail前可先用MD5函数计算出邮件内文的总和值 (长度为128位)，然后将此结果放在邮件内一起寄出。当对方收到这封邮件后，同样以MD5函数计算出邮件内文的校验和值，以此对比附在邮件中的校验和值。若两者不相同，表示邮件已被改变。

参 数:

-b或--binary	将所有输入文件都视为二进制文件，此参数在UNIX系统是无效的
-c <文件>或--check <文件>	检查文件的正确性。请在<文件>选项中指定原来利用md5sum所输出的文件
-t 或 --text	将所有输入文件视为文本文件，此参数在UNIX系统是无效的
--help	显示帮助
--string <字符串>	针对所给的字符串来计算MD5函数值
--status	只用状态码告知执行结果的正确与否，而不显示任何信息

范 例:

① 计算出文件file MD5函数值，再加以验证:

```
# md5sum file > checksumfile ← 将计算出的MD5函数值存入文件
# cat checksumfile           ← 看一下此文件的内容
22dae9855d792da9f7acb68cbc3f7a9f file
    ↑                               ↑
    这就是128位的MD5函数值       原始文件的名称
```

```
# md5sum -c checksumfile ← 检查时原始文件也必须放在同一个目录下
```

```
file: OK
```

② 计算指定字符串的MD5函数值:

```
# md5sum --string "Happy New Year"
6d713031cd29f69c679de72c234e45aa "Happy New Year"
```


mdel (MS-DOS delete)

常用指数: 

功能说明: 删除MS-DOS文件。

语 法: `mdel [文件...]`

包 名 称: `mtools`。

相关命令: `rm`。

补充说明: `mdel`为`mtools`工具命令，模拟MS-DOS的`del`命令，可删除MS-DOS文件系统中的文件。

参 数: 无。

范 例:

删除`dosfile1`与`dosfile2`文件:

```
# mdel dosfile1 dosfile2
```

mdeltree (MS-DOS deltree)

常用指数: 

功能说明: 删除 MS-DOS 目录。

语 法: `mdeltree [目录...]`

包 名 称: `mtools`。

相关命令: `rmdir`, `mmd`, `mrd`。



补充说明: mdel为mtools工具命令，模拟MS-DOS的deltree命令，可删除MS-DOS文件系统中的目录，及目录下所有子目录与文件。

参 数: 无。

范 例:

删除dosdir目录，及其中的所有子目录与文件：

```
# mdeltree dosdir
```

mdir (MS-DOS directory)

常用指数:



功能说明: 显示MS-DOS目录。

语 法: mdir [-/afwX] [目录...]

包 名 称: mtools。

相关命令: ls。

补充说明: mdir为mtools工具命令，模拟MS-DOS的dir命令，可显示MS-DOS文件系统中的目录内容。

参 数:

-/	显示目录下所有子目录与文件
-a	显示所有文件，包含隐藏文件
-f	不显示磁盘所剩余的可用空间。由于在计算FAT可用空间时，需要不少时间。因此使用“-f”参数可减少执行时间
-w	仅显示目录或文件名称，并以横排方式呈现，以便一次能显示较多的目录或文件
-X	仅显示目录下所有子目录与文件的完整路径，不显示其他信息。每个目录或文件均以一行显示

范 例:

① 显示 dosdir 目录下所有子目录与文件:

```
# mdir -/ dosdir
Volume in drive A has no label
Volume Serial Number is 19D5-2147
Directory for A:/dosdir

.           <DIR>      2003-07-03  21:52
..          <DIR>      2003-07-03  21:52
tyrell     <DIR>      2003-07-03  22:11  tyrell
           3 files           0 bytes
Directory for A:/dosdir/tyrell
.           <DIR>      2003-07-03  22:11
..          <DIR>      2003-07-03  22:11
RACHEL     515 2003-07-03  22:11  rachel
DECKARD    515 2003-07-03  22:11  deckard
           4 files           1 030 bytes
Total files listed:
           7 files           1 030 bytes
                           1 142 272 bytes free
```

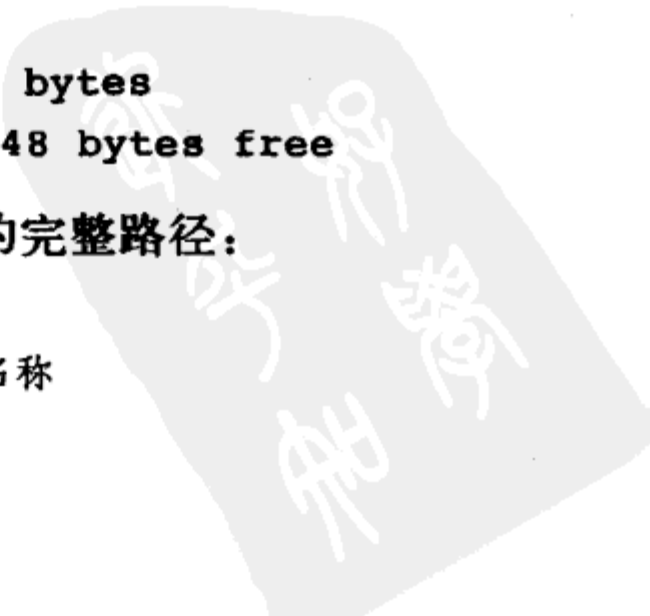
② 以横向的方式显示c:\backup目录下的子目录与文件:

```
# mdir -w c:/backup
Volume in drive C has no label
Directory for C:/backup
[.]           [...]           [OFFICE]           [NEW]
[ZIP]
           5 files           0 bytes
                           2 057 576 448 bytes free
```



③ 显示c:\backup目录下所有子目录及文件的完整路径:

```
# mdir -x c:/backup
C:/backup/Office/ ← 只显示路径名称
C:/backup/New/
...
```




④ 显示c:\下所有目录与文件，包含隐藏文件：

```
# mdir -a c:/
```

⑤ 显示c:\dir1及c:\dir2下所有的子目录与文件，但不显示磁盘剩余的可用空间：

```
# mdir -f c:/dir1 c:/dir2
```

mdu (MS-DOS disk used)

常用指数：

功能说明：显示MS-DOS目录所占用的磁盘空间。

语法：mdu [-as] [目录]

包名称：mtools。

相关命令：du, minfo。

补充说明：mdu为mtools工具命令，可显示MS-DOS文件系统中目录（包含子目录与文件）所占用的磁盘空间。

mdu显示的单位为丛集。若要知道丛集的大小，请用minfo命令。

参数：

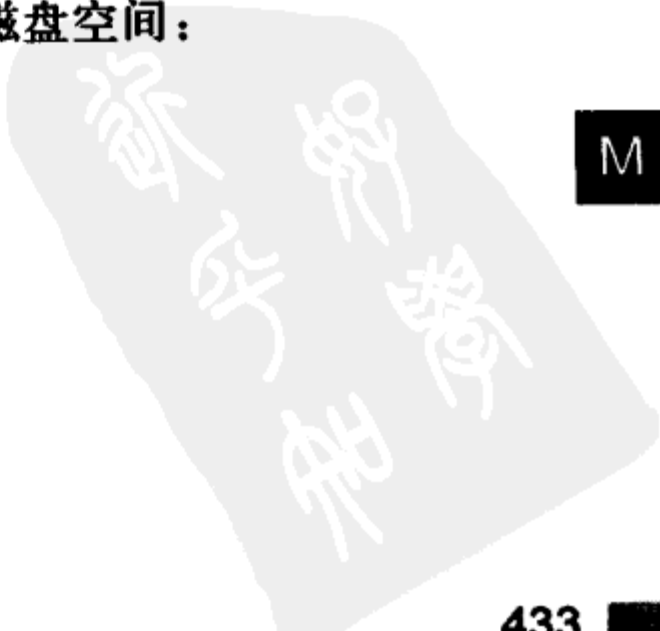
-a	显示每个文件及整个目录所占用的空间
-s	仅显示整个目录所占用的空间

范 例：

① 显示a:\dosdir目录以及每个文件所占用的磁盘空间：

```
# mdu -a A:/dosdir
A:/dosdir/dosfile1 60
A:/dosdir/dosfile2 33
A:/dosdir 93
```


M



② 仅显示a:\dosdir及a:\dosdir1整个目录所占用的空间:

```
# md5 -s A:/dosdir A:/dosdir1
A:/dosdir 53
A:/dosdir1 93
```

mesg (messages)

常用指数: 

功能说明: 设置终端的写入权限。

语法: mesg [ny]

相关命令: talk, wall, who, write, ytalk。

补充说明: 将mesg设置y(yes)时, 其他用户可利用write命令, 将信息直接显示在自己的屏幕上; 或利用talk命令要求跟你交谈。若不加参数, 则mesg会显示当前的设置状态。

参数:

n	不允许其他用户, 将信息直接显示在屏幕上
y	允许其他用户, 将信息直接显示在屏幕上

范例:


① 显示终端当前的写入权限:

```
# mesg
is n
```

② 允许其他用户能将信息显示在自己的屏幕上:

```
# mesg y
```

metamail

常用指数: 

功能说明: 负责处理非文本E-mail的程序。

语 法: metamail [-bBdehpPqrRTwxyz] [-c <邮件内容类型>] [-f <寄信人地址>] [-m <邮件管理程序名称>] [-s <邮件主题>] [文件名]

包 名 称: metamail (Mandrake Linux)。

相关命令: mail, mutt, pine。


补充说明: 这是Mandrake Linux才有的命令。当E-mail管理程序（例如elm）遇到非ISO-8859-1标准文本的邮件内容，而不知道要如何处理的时候（例如，声音或影像等多媒体的数据）就调用metamail。metamail会根据mailcap文件中的设置以适当的程序来处理这些非文本的数据。

这样的设计可方便管理员修改邮件程序支持的类型。也就是说，虽然系统中有许多种E-mail管理程序，但只要这些程序在收到非文本数据时都调用metamail，管理员只需修改mailcap中的设置，即可指定要以何种程序来处理该类型的数据。

metamail主要是以邮件标头（header）中的“Content-type”栏位来判断数据的类型。因此，其他像新闻组的程序，只要拥有“Content-type”的设计，都可调用metamail来处理非文本数据。

若不加任何参数，metamail会默认所处理的邮件为RFC 822格式（Internet邮件的标准）的邮件。若要处理的邮件不符合RFC 822的标准，则必须使用合适的参数。

参 数:

-b	邮件只有内文，没有标头。此参数必须与“-c”配合使用
-B	在背景显示信息。此参数不能与“-p”及“-P”一起使用
-c <邮件内容类型>	不要读取邮件标头的“Content-type”栏，而去读取<邮件内容类型>中的设置，以决定要调用哪个程序来处理邮件信息
-d	metamail在调用其他程序显示信息前，不需向用户确认
-e	删除内文前端的空白行。此参数可用来处理MH格式的邮件文件
-f <寄信人地址>	不要读取邮件表头中的寄信人地址，而去读取选项中的设置
-h	使用metamail来打印邮件时，必须设置此参数，以执行mailcap中的打印设置
-m <邮件管理程序名称>	记录调用metamail的邮件管理程序名称。此项信息主要是提供给metamail所要调用的程序
-p	将显示内容送给more命令处理之后，再送到输出设备。因此可逐页显示邮件内容
-P	与“-p”参数功能类似，但会在信息结束，显示“Press RETURN to go on”字符串，以等待用户按下  键
-q	metamail在调用其他程序来显示信息前，不用在屏幕上显示相关的信息
-r	metamail默认无法让root用户执行。使用此参数可让root从命令行来执行metamail
-R	显示信息前先清除屏幕
-s <邮件主题>	不要读取邮件表头中的邮件主题，而去读取选项中的设置
-T	若要在一个新的终端模拟视窗中重新启动metamail，则metamail必须以递归模式来调用程序本身，这时便需要使用“-T”参数
-w	不要读取mailcap中的设置，而直接处理邮件，并将数据（不管是非ISO-8859-1字符组或是二进制数据）存入文件中
-x	不要以终端模式来执行，即不使用互动模式
-y	抽出邮件内文中MIME格式的部分
-z	程序结束时将输入的文件删除

范 例:

- ① 利用mail将一封附有图文件的邮件存为jpgmail (在mail中, 执行s[邮件编号][文件]), 再以metamail来处理jpgmail:

```
# metamail -r jpgmail  ← root用户必须加上-r参数
Date: Tue, 15 Jul 2003 23:44:02 +0800 (CST)
From: root <root@free.flag.com.tw>
To: wyw <wyw@free.flag.com.tw>
Subject: OceanGears.jpg
```

} 邮件的标头

my wallpaper

Content-Description:

This message contains 'image/jpeg'-format data.

↑
邮件的Content-type设置

Do you want to view it using the 'ee' command (y/n) [y] ? y

---Executing: ee

↑
设会以“电眼”程序来显示图文件

- ② 查看包含附加文件的jpgmail, 结束后并将此文件删除:

```
# metamail -z jpgmail
```

- ③ 查看包含附加文件的mpegmail, 并直接指定此邮件的内容类型为mpeg:

```
# metamail -c video/mpeg mpegmail
```

- ④ jpeg是一个只有内文, 而没有表头的邮件, 查看此文件, 并直接指定此邮件内容类型为jpeg文件:

```
# metamail -b -c image/jpeg jpeg
```

⑤ 存储包含附加文件的jpegmail内容:

```
# metamail -w jpgmail
Date: Tue, 15 Jul 2003 23:44:02 +0800 (CST)
From: root <root@free.flag.com.tw>
To: wyw <wyw@free.flag.com.tw>
Subject: OceanGears.jpg
```

```
This message contains 'text/plain'-format data.
Please enter the name of a file to which the data should
be written
```

```
(Default: /tmp/mm.mRKXjo) > mail.txt ← 输入要存储内文的
文件名称
```

```
Wrote file mail.txt
Content-Description:
```

```
This message contains 'image/jpeg'-format data.
Please enter the name of a file to which the data should
be written
```

```
(Default: OceanGears.jpg) > mail.jpg ← 输入要存储附加文
件的文件名称
```

```
Wrote file mail.jpg
```

⑥ 查看包含附加文件的jpgmail, 当显示内容时逐页显示:

```
# metamail -p jpgmail
```

⑦ 查看包含附加文件的jpgmail, 当调用其他程序显示附加文件时,
不需向用户确认:

```
# metamail -d jpgmail
```


⑧ 查看包含附加文件的jpgmail, 但不在前台显示信息:

```
# metamail -B jpgmail
```

⑨ 查看包含附加文件的jpgmail, 显示前先清除屏幕画面:

```
# metamail -R jpgmail
```

mformat (MS-DOS format)

常用指数: 

功能说明: 对MS-DOS文件系统的磁盘进行格式化。

语法: mformat [-l aCFkX] [-0 <数据传输率>] [-2 <扇区数>] [-A <数据传输率>] [-B <开机区文件>] [-c <丛集大小>] [-h <磁头数>] [-H <隐藏扇区数>] [-M <扇区大小>] [-n <序号>] [-r <根目录大小>] [-S <扇区大小>] [-t <柱面数>] [磁盘代号]

包名称: mtools。

相关命令: minfo, mkfs, mkdosfs。

补充说明: mformat为mtools工具命令，模拟MS-DOS的format命令，可将指定的软盘或硬盘分区格式化为MS-DOS文件系统。在格式化软盘时，除非是一些很早或很特别的软盘，否则一般只要指定磁盘代号即可，不需要再加上任何额外的参数。

参数:

-0 <数据传输率>	指定第0磁道的数据传输率
-1	不使用2MB格式
-2 <扇区数>	指定在第0磁道的第0个磁头的扇区数，也就是所谓的2M格式。若是扇区大小大于一般值的512字节，建议加上此参数
-a	指定产生Atari格式的序号
-A <数据传输率>	指定第0磁道以外所有磁道的数据传输率
-B <开机区文件>	从指定的文件来建立开机区
-c <丛集大小>	指定丛集大小，单位为扇区
-C	建立软盘MS-DOS文件系统的映像文件
-F	以FAT32来格式化软盘

-h <磁头数>	指定磁头的数目
-H <隐藏扇区数>	指定隐藏扇区的数目
-k	尽可能地保留原有的开机区
-M <扇区大小>	指定MS-DOS文件系统所使用的扇区大小，默认应该与硬件扇区大小相同，单位为字节
-n <序号>	指定序号
-r <根目录大小>	指定根目录的大小，单位为扇区
-S <扇区大小>	指定硬件扇区大小。实际的扇区大小是由<扇区大小>加上7后，再作为2的幂方所得的数字。例如，<扇区大小>设为1，则实际的扇区大小等于 $2^{(7+1)} = 256$ 字节
-t <柱面数>	指定柱面数目
-X	格式化XDF磁盘

范 例：

① 格式化A:磁盘中的软盘：

```
# mformat A:
```

② 格式化单面的软盘：

```
# mformat -c 1 A:
```

minfo (MS-DOS information)

常用指数：



功能说明：显示MS-DOS文件系统的各项参数。

语 法：minfo [-v] [磁盘代号]

包 名 称：mtools。

相关命令：mformat。

补充说明: minfo为mtools工具命令, 可显示MS-DOS文件系统磁盘的各项参数, 包括扇区数、磁头数、柱面数等等。

参 数:

-v	除了一般信息外, 并显示启动扇区的内容 (以十六进制格式来表示)
----	----------------------------------

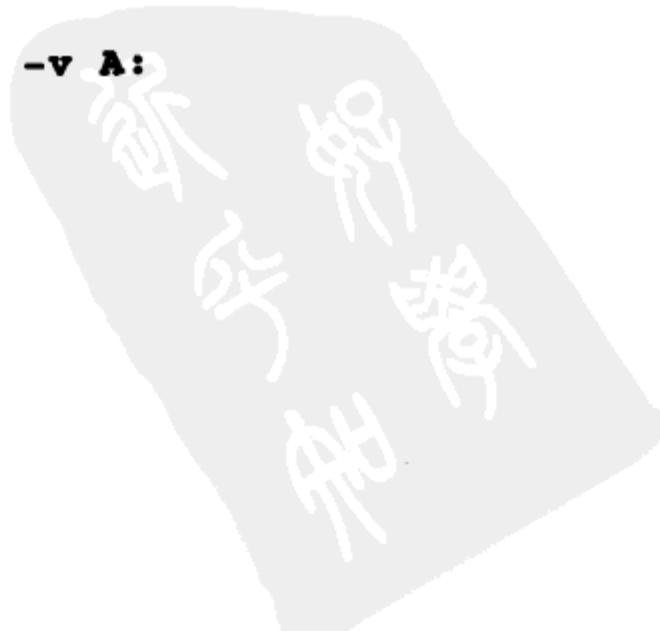
范 例:

① 显示软盘中MS-DOS文件系统的各项参数:

```
# minfo A:
device information:
=====
filename="/dev/fd0"
sectors per track: 18
heads: 2
cylinders:80
mformat command line: mformat -t 80 -h 2 -s 18 a:
bootsector information
=====
banner:")L^P<IHC"
sector size: 512 bytes
cluster size: 1 sectors
reserved (boot) sectors: 1
fats: 2
...
```

② 显示软盘中MS-DOS文件系统的各项参数, 同时显示启动扇区的内容:

```
# minfo -v A:
```



M

mingetty (minimal getty)

常用指数:



功能说明: 精简版的 getty。

语 法: mingetty [--long-hostname] [--noclear] [tty]

包 名 称: mingetty。

相关命令: login。

补充说明: mingetty适用于本机上的登录程序。Red Hat Linux/Fedora Core及Mandrake Linux默认使用mingetty来登录 (Slackware则是使用agetty)。

参 数:

--long-hostname	显示完整的主机名称
--noclear	在询问登录的用户名称之前, 不要清除屏幕画面

范 例:

① 登录程序的设置存储在/etc/inittab中。可以在此设置值, 然后重新登录看看有什么变化。以Red Hat Linux为例, 在inittab中有关mingetty的tty1设置如下:

```
1:2345:respawn:/sbin/mingetty tty1
```

此时登录的信息是像这个样子:

```
Red Hat Linux release 9(Shrike)
Kernel 2.4.20-8 on an i586
vito login:
```

若是将inittab中的tty1设置改成：


```
1:2345:respawn:/sbin/mingetty --long-hostname tty1
```

则登录的信息会变成这个样子：

```
Red Hat Linux release 9(Shrike)
Kernel 2.4.20-8 on an i586
vito.flag.com.tw login:
```

↑
显示完整的主机名称

minicom

常用指数：

(minimal communication program)

功能说明：数据机通信程序。

语 法： minicom [-8lmMostz] [-a <on 或 off>] [-c <on 或 off>] [-C <撷取文件>] [-d <编号>] [-p <模拟终端>] [-S <script文件>] [设置文件]

包名称： minicom。

相关命令： dip, pppd, pppstats。

补充说明： minicom是一个相当受欢迎的PPP拨号连接程序。minicom提供了类似功能表的操作模式，远比单纯的命令行模式（例如Kermit）容易操作。



参 数:

- 8	不要修改任何8位编码的字符
-a <on或off>	设置终端属性。默认值为“-a on”，若使用序列式终端，则必须设为“-a off”
-c <on或off>	设置彩色模式。默认值为“-c off”，也就是不使用彩色模式
-C <擷取文件>	指定擷取文件，并在启动时打开擷取功能
-d <编号>	启动后直接拨号。选项中的<编号>为电话号码簿（Dialing directory）中的记录编号
-l	不会将所有的字符都转成ASCII码
-m	以 Alt 或Meta键作为命令键（默认的命令键为 Ctrl + a ）
-M	与“-m”参数类似。若Meta键将字符码的第8个位设为high bit时，则必须使用“-M”参数
-o	不要初始化数据机。如果在前一次结束程序时未重设数据机，重新启动程序时即可使用此参数，省略初始化的动作
-p <模拟终端>	使用伪终端（Pseudo terminal）
-s	执行前，先设置minicom的环境，此处设置值默认会存储在/etc/minirc.dfl设置文件中
-S <script文件>	在启动时，执行指定的script文件
-t	设置终端的类型
-z	在终端上显示状态列
[设置文件]	指定minicom设置文件。默认的设置文件为minirc.dfl

范 例:

① 使用minicom拨号连接:

```
# minicom
```

按 **Ctrl + a** 键，再按 **z** 键，会出现以下画面，可执行常用的功能:

```

Welco
OPTIO
Compi
Press
                                Minicom Command Summary
                                Commands can be called by CTRL-A <key>

                                Main Functions                                Other Functions
Dialing directory..D  run script (Go)....G  Clear Screen.....C
Send files.....S    Receive files.....R  cOnfigure Minicom..O
comm Parameters...P  Add linefeed.....A  Suspend minicom...J
Capture on/off....L  Hangup.....H        eXit and reset....X
send break.....F    initialize Modem...M  Quit with no reset.Q
Terminal settings..T  run Kermit.....K    Cursor key mode...I
lineWrap on/off...W  local Echo on/off..E  Help screen.....Z
                                scroll Back.....B

                                Select function or press Enter for none. █

                                Written by Miquel van Smoorenburg 1991-1995
                                Some additions by Jukka Lahtinen 1997-2000
                                il8n by Arnaldo Carvalho de Melo 1998

CTRL-A Z for help | 38400 8N1 | NOR | Minicom 2.00.0 | VT102 | Offline

```

② 启动minicom后直接拨电话号码簿中编号2号的电话号码:

```
# minicom -d 2
```

③ 执行minicom拨号连接, 并启动截取功能, 将执行过程中屏幕所显示的信息存储至minicom_cap文件中:

```
# minicom -C minicom_cap
```

④ 执行minicom拨号连接前, 先设置执行环境:

```
# minicom -s
```

⑤ 执行minicom拨号连接时, 不要初始化数据机:

```
# minicom -o
```

⑥ 以彩色模式执行minicom拨号连接:

```
# minicom -c on
```

⑦ 执行minicom拨号连接, 并以 **Alt** 键取代 **Ctrl** 键:

```
# minicom -m
```

操作命令：进入minicom之后请按 **Ctrl + a** 键，再按 **z** 键，即可显示命令列表与相关帮助。以下帮助常用的操作命令（命令没有大小写的区别）：

A	按 Enter 键时，默认只会将光标移至行首。若执行A命令，除了移至行首还会换行。再执行一次A命令，即可切回原来的状态
B	产生滚动缓冲区。因此可以使用 Page Up 、 Page Down 以及方向键来滚动屏幕画面
C	清除屏幕画面
D	显示电话号码簿
E	设置是否要在本机显示信息
F	对数据机送出中断 (break) 信号
G	执行登录的script
H	挂断电话
I	指定方向键所传送的字符符号
J	跳到shell
K	清除屏幕画面，并执行Kermit程序
L	启动或关闭撷取功能。启动撷取功能时，所有在屏幕上显示的信息都会存储至撷取文件中
M	送出数据机初始化的参数
O	进入minicom设置画面
P	设置与传输有关的参数，包括bps、同位检查等等
Q	结束minicom程序，但不重设数据机
R	接收文件
S	传送文件
T	选择Ansi或VT100模式
W	设置是否要换行
X	结束minicom程序，并重设数据机
Z	显示帮助画面

mkbootdisk

常用指数: 

功能说明: 建立当前系统的启动盘。

语 法: `mkbootdisk [-v] [--device <设备>] [--noprompt] [--version]`
[kernel 版本]

包名称: `mkbootdisk`。

相关命令: `mkinitrd`。

补充说明: `mkbootdisk`可建立当前系统的启动盘。`mkbootdisk`会根据当前系统的设置,使得利用软盘启动时,载入必要的设备与文件系统。

参 数:

<code>--device <设备></code>	指定设备。默认为/dev/fd0
<code>--noprompt</code>	不会提示用户插入软盘
<code>-v</code> 或 <code>--verbose</code>	执行时显示详细的信息
<code>--version</code>	显示版本信息

范 例:


① 制作启动盘:

```
# mkbootdisk --verbose 2.4.20-8
```

↑ 显示详细的信息 | kernel 的版本

```
Insert a disk in /dev/fd0. Any information on the disk  
will be lost. ← 软盘中所有的数据将会全部清除
```

```
Press <Enter> to continue or ^C to abort: ←
```

插入软盘后, 请按  键

M

```
Formatting /tmp/mkbootdisk.S4o3aw... done. ← 格式化软盘  
Copying /boot/vmlinuz-2.4.20-8... done. ← 复制kernel  
Copying /boot/initrd-2.4.20-8.img... done. ← 复制ramdisk  
映像文件
```

```
Configuring bootloader... done.  
读入了 20+0 个小节  
输出了 20+0 个小节
```

设置启动环境

② 在/dev/fd0磁盘建立启动盘:

```
# mkbootdisk --device /dev/fd0 2.4.20-8
```

③ 建立启动盘, 建立时不需提示用户插入软盘:

```
# mkbootdisk --noprompt 2.4.20-8
```

mkdir (make directories)

常用指数:



功能说明: 建立目录。

语法: mkdir [-p] [-m <目录属性>] [--help] [--verbose] [--version]
[目录名称...]

包名称: coreutils。

相关命令: mmd, rmdir。

补充说明: mkdir可建立目录并同时设置目录的权限。



参 数:

-m <目录属性>或 --mode <目录属性>	建立目录时，同时设置目录的权限。权限的设置法与 chmod 命令相同
-p 或 --parents	若所要建立目录的上层目录当前尚未建立，则会一起 建立上层目录
--help	显示帮助
--verbose	执行时显示详细的信息
--version	显示版本信息

范 例:

① 建立名称为public的目录，并让全部人都有rwx的权限：

```
# mkdir -m 777 public
```

② 建立/root/parentdir/childdir目录，当前 /root 下没有任何目录：

```
# mkdir -p /root/parentdir/childdir
```

```
# ls -lR
```

```
..:
```

```
总计 1
```

```
drwxr-xr-x  3 root root  1024 7月  7 18:23 parentdir ←
```

```
parentdir:
```

```
总计 1
```

```
drwxr-xr-x  2 root root  1024 7月  7 18:23 childdir ←
```

```
parentdir/childdir:
```

```
总计 0
```

同时建立/root/parentdir以及
/root/parentdir/childdir两个目录

M

mkdosfs (make DOS file system)

常用指数:

**功能说明:** 建立MS-DOS文件系统。**语法:** mkdosfs [-cv] [-f <FAT 数目>] [-F <FAT 记录的单位>] [-i <磁盘序号>] [-l <文件名>] [-m <信息文件>] [-n <标签>] [-r <根目录项目数>] [-s <扇区数>] [设备名称] [块数]**包名称:** dosfstools。**相关命令:** badblocks, mformat, mkfs。**补充说明:** mkdosfs可在Linux下将磁盘格式化为MS-DOS文件系统的格式。

使用时可省略指定[块数]参数, mkdosfs会自动设置适当的块数。

参数:

-c	检查是否有损毁的块
-f <FAT 数目>	指定FAT的数目, 当前支持1与2两种选项, 默认值为2。如果选择两份FAT, 其中一份FAT可作为备份之用
-F <FAT 记录的单位>	指定FAT记录的单位为12或16位。若不指定, mkdosfs会自动选择适当的大小
-i <磁盘序号>	设置文件系统的磁盘序号。此序号长度为32位, 并以十六进制表示。若不指定, 则依据当时的时间来产生磁盘序号
-l <文件名>	从指定的文件中, 读取文件系统中损毁块的信息。可利用badblocks命令找出损毁的块, 并存储在文件中
-m <信息文件>	若以此文件系统开机, 而却因操作系统没有正常安装, 则会显示选项中所指定的信息文件内容。信息文件的最大为418字节。若是在选项中以“-”来取代文件名, 则会从键盘读取信息

-n <标签>	设置文件系统的磁盘标签名称（长度为11个字符）
-r <根目录项目数>	设置根目录最多能记载的项目数（指根目录下的文件数目与子目录数目的总和）。若设备为软盘，此默认值为112或224；若设备为硬盘，则默认值为512
-s <扇区数>	指定每个丛集所包含的扇区数。指定的扇区数必须为2的幂方
-v	执行时显示详细的信息

范 例：

① 在软盘上建立MS-DOS文件系统，并检查有无损毁的块：

```
# mkdosfs -c /dev/fd0
```

② 在/dev/hdb1分区建立MS-DOS文件系统，并指定标签名称为 win95，
执行时显示详细的信息：

```
# mkdosfs -n win95 -v /dev/hdb1
```

```
mkdosfs 2.8 (28 Feb 2001)
```

```
/dev/hdb1 has 64 heads and 63 sectors per track,  
logical sector size is 512,
```

```
using 0xf8 media descriptor, with 4124672 sectors;
```

```
file system has 2 16-bit FATs and 64 sectors per cluster.
```

```
FAT size is 252 sectors, and provides 64439 clusters.
```

```
Root directory contains 512 slots.
```

```
Volume ID is 391f75c8, volume label win95.
```

③ 在/dev/hdb1分区建立MS-DOS文件系统，并从hdb1_badblocks文件中
读取此分区损毁的块列表：

```
# mkdosfs -l hdb1_badblocks /dev/hdb1
```

④ 在/dev/hda2分区建立MS-DOS文件系统，并指定开机错误信息：

```
# mkdosfs -m - /dev/hda2
```

└─ 从键盘输入开机错误信息

```
mkdosfs 2.8 (28 Feb 2001)
```

```
Sorry, can't find operating system!
```

← 输入开机错误时要显示的信息，按 **Enter** + **Ctrl** 键结束输入

mke2fs (make ext2 file system)

常用指数：



功能说明： 建立ext2文件系统。

语法： mke2fs [-cFjMqrSvV] [-b <块大小>] [-f <不连续段大小>] [-i <字节>] [-N <inode数>] [-l <文件>] [-L <标签>] [-m <百分比值>] [-R =<块数>] [设备名称] [块数]

相关命令： badblocks, e2fsck, mkfs, mkfs.ext2。

补充说明： mke2fs可建立Linux的ext2文件系统。使用时可省略指定[块数]参数，mke2fs能自动设置适当的块数。

参数：

-b <块大小>	指定块大小，单位为字节
-c	检查是否有损毁的块
-f <不连续段大小>	指定不连续段 (Fragment) 的大小，单位为字节
-F	不管指定的设备为何，强制执行mke2fs
-i <字节>	指定“字节/inode”的比例。也就是每个inode指向空间的大小。默认值为4 096字节，最小不可低于1 024字节
-j	建立ext3日志式文件系统

-N <inode数>	指定要建立的inode数目。若此参数的设置与其他参数的设置冲突，则mke2fs会依此处的设置来建立文件系统
-l <文件>	从指定的文件中，读取文件系统中损毁块的信息。可利用badblocks命令找出损毁的块，并存储在文件中
-L <标签>	设置文件系统的标签名称
-m <百分比值>	指定给管理员保留块的比例，默认为5%
-M	记录最近一次加载的目录
-q	执行时不显示任何信息
-r	指定要建立的ext2文件系统版本
-R =<块数>	设置磁盘阵列参数。当前仅支持RAID 0（数据分散存储在各个磁盘中，不具容错能力）。=<块数>指定磁盘阵列中，数据存储在磁盘中的最小单位
-S	仅写入superblock与group descriptors，而不更改inodetable、inode bitmap以及block bitmap。此参数的功能在于修复损毁的superblock与group descriptors
-v	执行时显示详细的信息
-V	显示版本信息

范 例：

① 在/dev/hdb1分区建立ext3文件系统，并指定块单位为2 048个字节：

```
# mke2fs -b 2048 -j /dev/hdb1
```

② 在/dev/hdb1分区建立ext2文件系统，并检查有无损毁的块：

```
# mke2fs -c /dev/hdb1
```

- ③ 在/dev/hda5分区建立ext2文件系统，并指定标签名称为MyLinux，执行时显示详细的信息：

```
# mke2fs -v -L MyLinux /dev/hda5
mke2fs 1.32 (09-Nov-2002)
Filesystem label=MyLinux
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
538560 inodes, 1076347 blocks
53817 blocks (5.00%) reserved for the super user
First data block=0
33 block groups
32768 blocks per group, 32768 fragments per group
16320 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Writing inode tables: done
Writing superblocks and filesystem accounting
information: done
```

This filesystem will be automatically checked every 38 mounts or 180 days, whichever comes first. Use tune2fs -c or -i to override.

- ④ 在/dev/hdb1分区建立ext2文件系统，并从hdb1_badblocks文件中读取此分区损毁的块列表：

```
# mke2fs -l hdb1_badblocks /dev/hdb1
```

M

- ⑤ 在/dev/hdb1分区建立ext3文件系统，并指定保留给管理员8%块的容量：


```
# mke2fs -m 8 -j /dev/hdb1
```

⑥ 修复/dev/hdb1文件系统的superblock与group descriptors:

```
# mke2fs -S /dev/hdb1
```

```
# e2fsck /dev/hdb1 ←修复后再以e2fsck检查一下有没有问题
```

mkfs (make file system)

常用指数: 

功能说明: 建立各种文件系统。

语法: mkfs [fs-options] [-vV] [-t <文件系统类型>] [设备名称]
[块数]

包名称: util-linux。

相关命令: mkdosfs, mke2fs, mkfs.minix。

补充说明: mkfs本身并不执行建立文件系统的工作,而是去调用相关的程序来执行。例如,若在“-t”参数中指定ext2,则mkfs会调用mke2fs来建立文件系统。

使用时如省略指定[块数]参数, mkfs会自动设置适当的块数。

参数:

fs-options	<p>指定建立文件系统时的参数。会因文件系统的不同有不同的参数,比较常见的有:</p> <ul style="list-style-type: none"> -c 检查是否有损毁的块 -l <文件> 从指定的文件中,读取文件系统中损毁块的信息。可利用badblocks命令找出损毁的块,并存储在文件中 -v 执行时显示详细的信息
------------	---

-t <文件系统类型>	指定要建立何种文件系统。常用的选项如下： minix Linux最早期所使用的文件系统 ext2 当前Linux常用的文件系统（默认值） ext3 新一代的Linux日志文件系统 msdos FAT的文件系统
-v	显示版本信息与详细的使用方法
-V	显示简要的使用方法

范 例：

① 在/dev/hdb1分区建立MS-DOS文件系统，并检查有无损毁的块：

```
# mkfs -t msdos -c /dev/hdb1
```

建立MS-DOS文件系统 检查有无损毁块


② 在/dev/hdb1分区建立ext2文件系统，并从hdb1_badblocks文件中读取此分区损毁块的列表：

```
# mkfs -t ext2 -l hdb1_badblocks /dev/hdb1
```

③ 在/dev/hdb1分区建立ext2文件系统，执行时显示详细的信息：


```
# mkfs -t ext2 -v /dev/hdb1
mke2fs 1.32 (09-Nov-2002)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
538560 inodes, 1076347 blocks
...
```

mkfs.ext3

常用指数: 

功能说明: 与mke2fs相同。另外还有一个mkfs.ext2命令, 功能与mkfs.ext3相同。

mkfs.minix

常用指数: 

功能说明: 建立 Minix 文件系统。

语 法: mkfs.minix [-cv] [-i <inode 数目>] [-l <文件>] [-n <文件名长度>] [设备名称] [块数]

包 名 称: util-linux (Mandrake Linux)。

相关命令: badblocks, mkfs。

补充说明: 此为Mandrake Linux才有的命令。mkfs.minix可建立Minix文件系统。Minix文件系统是Linux早期所使用的文件系统。

参 数:

-c	检查是否有损毁的块
-i <inode 数目>	指定文件系统的inode总数
-l <文件>	从指定的文件中, 读取文件系统中损毁块的信息。可利用badblocks命令找出损毁的块, 并存储到文件中
-n <文件名长度>	指定文件名称长度的上限。有14与30两种选项, 默认为30
-v	建立第2版的Minix文件系统

范 例:

① 在/dev/hdb1分区建立Minix文件系统:

```
# mkfs.minix /dev/hdb1
```

② 在/dev/hdb1分区建立Minix文件系统, 并检查有无损毁的块:

```
# mkfs.minix -c /dev/hdb1
```

③ 在/dev/hdb1分区建立Minix文件系统, 并从hdb1_badblocks文件中读取此分区损毁块的列表:

```
# mkfs.minix -l hdb1_badblocks /dev/hdb1
```

④ 在/dev/hdb1分区建立Minix文件系统, 并指定文件名称长度上限为14个字符:

```
# mkfs.minix -n 14 /dev/hdb1
```

⑤ 在/dev/hdb1分区建立第2版的Minix文件系统:

```
# mkfs.minix -v /dev/hdb1
```


mkfs.msdos

常用指数:



功能说明: 与mkdosfs相同。

mkinitrd

常用指数: 

功能说明: 建立要载入ramdisk的映像文件。

语 法: mkinitrd [-fv] [--omit-raid-modules] [--omit-scsi-modules]
 [--preload=<模块名称>] [--version] [--with=<模块名称>]
 [映像文件] [Kernel 版本]

包 名 称: mkinitrd。

相关命令: lilo, make, mkbootdisk, patch, rpm。

补充说明: mkinitrd可建立映像文件, 以供Linux开机时载入ramdisk。映像文件中包含了一些必要的模块(例如SCSI模块), 使kernel能够正常地存取Linux所在的文件系统。操作系统载入程序(例如lilo)会负责将映像文件载入ramdisk中。

参 数:

-f	若指定的映像文件名称与现有文件重复, 则覆盖现有的文件
-v	执行时显示详细的信息
--omit-raid-modules	不要载入RAID模块
--omit-scsi-modules	不要载入SCSI模块
--preload=<模块名称>	指定要载入的模块。指定的模块会在SCSI模块之前先行载入ramdisk
--with=<模块名称>	指定要载入的模块。指定的模块会在SCSI模块之后才载入ramdisk
--version	显示版本信息

范 例:

① 制作映像文件:

```
# mkinitrd /boot/initrd.img 2.4.20-8
```

映像文件名称 kernel的版本

② 制作映像文件时, 若指定的映像文件与现有文件重复, 直接覆盖文件:

```
# mkinitrd -f /boot/initrd.img 2.4.20-8
```

③ 制作映像文件时, 指定不要载入 RAID 模块:

```
# mkinitrd --omit-raid-modules /boot/initrd.img 2.4.20-8
```

④ 制作映像文件时, 指定不要载入SCSI模块:

```
# mkinitrd --omit-scsi-modules /boot/initrd.img 2.4.20-8
```

mkisofs

常用指数:



功能说明: 建立ISO 9660映像文件。

语 法: mkisofs [-dDfLNrRTvz] [-A <应用程序 ID>] [-abstract <摘要文件>] [-b <开机映像文件>] [-biblio <ISBN 文件>] [-c <开机文件名称>] [-C <盘区编号, 扇区编号>] [-copyright <版权信息文件>] [-hide <目录或文件名>] [-hide-joliet <目录或文件名>] [-log-file <记录文件>] [-m <目录或文件名>] [-M <映像文件>] [-o <映像文件>] [-p <数据处理人>] [-P <光盘发行人>] [-print-size] [-quiet]

[-sysid <系统 ID>][-V <光盘 ID>] [-volset <卷册集 ID>]
 [-volset-size <光盘总数>] [-volset-seqno <卷册序号>] [-x
 <目录>] [目录或文件]

包名称: mkisofs。

相关命令: mount。

补充说明: mkisofs可将指定的目录与文件做成ISO 9660格式的映像文件，以供刻录光盘。mkisofs并无控制刻录机界面的能力，必须借由其他刻录程序才能将映像文件刻录至CD-R光盘。

mkisofs也支持Rock Ridge Extensions。Rock Ridge Extension可使用ISO 9660中未定义的列，以增加可存放的文件名长度及其他信息，让ISO 9660的文件名称更接近Unix /Linux的文件名称。不过，为了兼容于MS-DOS系统，mkisofs默认会自动将长文件名转换为8.3的MS-DOS文件格式。

制作好的映像文件可利用mount命令加载，以便检查映像文件是否正常。

参 数:

-A <应用程序ID>	指定光盘的应用程序ID（长度为128字符）。应用程序ID会写入光盘Primary Volume Descriptor的Application ID栏
-abstract <摘要文件>	指定摘要文件的文件名。摘要文件位于光盘根目录下，文件中说明了光盘的内容。此参数会写入光盘Primary Volume Descriptor的Abstract File ID栏
-b <开机映像文件>	指定在制作可开机光盘（El Torito）时，所需的开机映像文件
-biblio <ISBN 文件>	指定ISBN（International Standard Book Number）文件的文件名，ISBN文件位于光盘根目录下，记录光盘的ISBN。此参数会写入光盘Primary Volume Descriptor的Bibliographic File ID栏

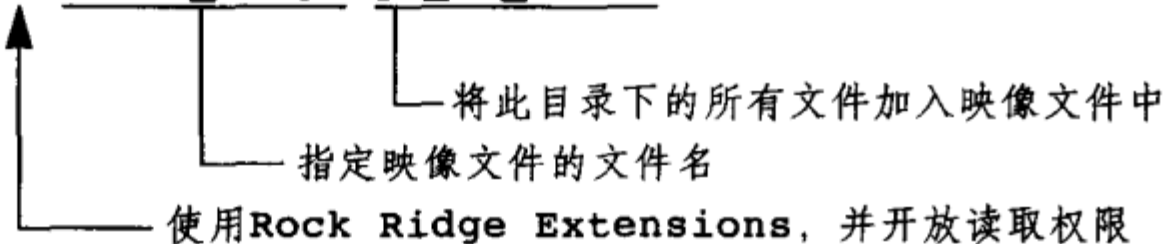
-c <开机文件名称>	制作可开机光盘时，mkisofs会将开机映像文件中的全部内容做成一个文件。“-c”参数即用来指定这个文件在光盘中的目录与文件名。设置时，记得不要和光盘中的其他文件名重复
-C <盘区编号, 盘区编号>	将许多节区 (session) 合成一个映像文件时，必须使用此参数。选项中的第一个盘区 (sector) 编号是指最后一个节区的第一个盘区。选项中的第二个盘区编号是指新加节区的第一个盘区。“-C”参数应配合“-M”参数一起使用
-copyright<版权信息文件>	指定版权信息文件的文件名。版权信息文件位于光盘根目录下，文件中记录了光盘的版权信息。此参数会写入光盘Primary Volume Descriptor的Copyright File ID栏
-d	省略文件后的句号
-D	ISO 9660最多只能处理8层的目录，超过8层的部分(称作Deep Directories)，RRIP会自动将它们配置成ISO 9660兼容的格式。使用“-D”参数可关闭此功能
-f	若遇到符号链接，默认会将此符号链接放入映像文件中。如加上此参数，则把符号链接指向的文件放入映像文件，而不保留符号链接的关系
-hide <目录或文件名>	使指定的目录或文件在ISO 9660或Rock Ridge Extensions的系统中隐藏
-hide-joliet<目录或文件名>	使指定的目录或文件在Joliet系统中隐藏
-l	可使用ISO 9660 32字符长度的文件名。不过烧出来的光盘可能无法在MS-DOS下正常存取
-L	允许文件名的第一个字符为句点(.)。烧出来的光盘可能无法在MS-DOS下阅读
-log-file <记录文件>	在执行过程中若有错误信息，默认会显示在屏幕上。使用此参数可将错误信息存储至指定的文件中
-m <目录或文件名>	指定的目录或文件名将不会放入映像文件中
-M <映像文件>	与指定的映像文件合并。此参数必须与“-C”参数一起使用
-N	省略ISO 9660文件中的版本信息
-o <映像文件>	指定映像文件的名称

-p <数据处理人>	记录光盘的数据处理人（长度为128字符）。此参数会写入光盘Primary Volume Descriptor的Data Preparer ID栏
-P <光盘发行人>	记录光盘的发行人（长度为128字符）。此参数会写入光盘Primary Volume Descriptor的Publisher ID栏
-print-size	显示预估的文件系统大小。也就是指定的目录与文件，刻录在光盘上的大小。使用此参数可避免制作出太大的映像文件，而超出光盘实际的容量
-quiet	执行时不显示任何信息
-r	使用Rock Ridge Extensions，并开放全部文件的读取权限。一般都会使用此参数
-R	使用Rock Ridge Extensions
-sysid <系统ID>	指定光盘的系统ID（长度为32位）。系统ID会写入光盘Primary Volume Descriptor的System ID栏
-T	建立文件名的转换表，适用于不支持Rock Ridge Extensions的系统上
-v	执行时显示详细的信息
-V <光盘ID>	记录光盘的卷册（Volume）识别码（长度为128位）。此参数会写入光盘Primary Volume Descriptor的Volume ID栏
-volset <卷册集ID>	指定光盘的卷册集ID（Volume Set ID，长度为128位）。卷册集ID会写入光盘Primary Volume Descriptor的Volume Set ID栏
-volset-size <光盘总数>	指定卷册集所包含的光盘张数（长度为4个位）。此参数会写入光盘Primary Volume Descriptor的Volume Set Size栏
-volset-seqno <卷册序号>	指定光盘片在卷册集中的编号（长度为4个位）。此参数会写入光盘Primary Volume Descriptor的Volume Sequence Number栏，此参数使用的位置不得早于“-volset-size”参数
-x <目录>	指定的目录将不会放入映像文件中
-z	建立透明性（Transparent）压缩文件的SUSP记录，此记录当前只在Alpha机器上的Linux有效

范 例:

① 制作cd_image映像文件:

```
# mkisofs -r -o cd_image go_to_burn/
```



```
Total extents actually written = 2834
Total translation table size: 0
Total rockridge attributes bytes: 47983
Total directory bytes: 190464
Path table size(bytes): 1236
Max brk space used 3e000
2834 extents written (5 Mb)
# mount -t iso9660 -o ro,loop=/dev/loop0 cd_image /mnt/cdrom
```

制作映像文件过程中产生的信息

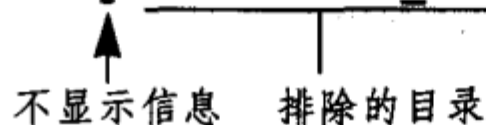
将映像文件载入/mnt/cdrom中, 查看是否正常

② 制作cd_image映像文件, 并指定不将exclude_file文件放入映像文件中:

```
# mkisofs -m exclude_file -r -o cd_image go_to_burn/
```

③ 制作cd_image映像文件, 并指定不将exclude_dir目录放入映像文件中, 执行时不显示任何信息:

```
# mkisofs -q -x exclude_dir -r -o cd_image go_to_burn/
```



M

④ 制作cd_image映像文件, 若遇到符号链接, 则将其指向的文件放入映像文件, 而不保留符号链接的关系:

```
# mkisofs -f -r -o cd_image go_to_burn/
```


- ⑤ 制作cd_image映像文件时，将hide_file文件设为隐藏，并指定数据处理人为Jacket Chen：

```
# mkisofs -p "Jacket Chen" -hide hide_file -r -o cd_image go_to_burn/
```

|
|
 数据处理人 将此文件设为隐藏

- ⑥ 制作cd_image映像文件，并指定一共有4张光盘，此张光盘是第2片：

```
# mkisofs -volset-size 4 -volset-seqno 2 -r -o cd_image go_to_burn/
```

|
|
 总张数 此光盘的编号数

- ⑦ 制作cd_image映像文件，并将信息记录到mkiso.log文件中：

```
# mkisofs -log-file mkiso.log -r -o cd_image go_to_burn/
```

- ⑧ 制作cd_image映像文件，并指定光盘发行人为Terry Chen：

```
# mkisofs -P "Terry Chen" -r -o cd_image go_to_burn/
```

- ⑨ 制作cd_image映像文件，并指定摘要文件的文件名为about_cd：

```
# mkisofs -abstract about_cd -r -o cd_image go_to_burn/
```

- ⑩ 预估制作映像文件的容量大小：

```
# mkisofs -print-size -r go_to_burn/
```

- ⑪ 制作cd_image映像文件，但不开放全部文件的读取权限：

```
# mkisofs -R -o cd_image go_to_burn/
```

mkraid (make RAID)

常用指数:



功能说明: 初始化磁盘阵列。

语 法: mkraid [-foV] [-c <设置文件>] [磁盘阵列设备]

包 名 称: raidtools。

相关命令: raidstart, raidstop。

补充说明: 磁盘阵列 (redundant array of independent disks) 可以将多个磁盘组成一个逻辑上的磁盘。在操作上, 我们会觉得跟使用单一磁盘没什么不同, 但实际上, 磁盘阵列是将数据分别存储在不同的磁盘上。借由磁盘阵列, 用户可获得以下4点好处:

- **增进磁盘的存取效率:** 由于数据是同步地写入不同的磁盘, 所以可提高存取效率。
- **可整合多个磁盘:** 用户可将多个容量较小的磁盘组成一个容量较大的磁盘。
- **有较高的安全性:** 磁盘阵列允许用户加上备用磁盘, 当某个磁盘不能正常运作时, 磁盘阵列会立刻启动备用磁盘, 而使得整个磁盘阵列仍能继续正常运作。
- **具容错能力:** 磁盘阵列可将错误修正码放在不同的磁盘上, 当存储的数据发生问题时, 即可利用此修正码来恢复原始的数据。

Linux当前支持线性 (Linear)、RAID-0、RAID-1、RAID-4和RAID-5等磁盘阵列的运作运作模式。

要使用磁盘阵列, 需先设置好/etc/raidtab设置文件, 然后再执行mkraid命令初始化此磁盘阵列。当完成初始化动作后, 可查看/proc/mdstat记录文件查看磁盘阵列当前的执行状态。

参 数:

-c <设置文件>或 --configfile <设置文件>	使用指定的设置文件来初始化磁盘阵列，而不用默认的/etc/raidtab设置文件
-f 或--force	即使磁盘内已有数据，仍强制地初始化磁盘阵列
-h 或--help	显示帮助
-o 或--upgrade	此参数是将旧版本的磁盘阵列更新成新版的磁盘阵列。除非曾经安装过旧版的磁盘阵列，否则不需使用此参数
-V或--version	显示版本信息

范 例:

① 初始化/dev/md0磁盘阵列设备，然后格式化及加载此磁盘设备:

```
# mkraid /dev/md0          ← 初始化磁盘阵列
handling MD device /dev/md0
analyzing super-block
# mke2fs /dev/md0          ← 格式此磁盘阵列设备
# mount -t ext2 /dev/md0 /mnt/md ← 载入到/mnt/md目录中
```

② 初始化myraidtab设置文件中/dev/md0磁盘阵列设备:

```
# mkraid -c myraidtab /dev/md0
```

③ 初始化/dev/md0磁盘阵列设备，并删除此设备原有的数据:

```
# mkraid --force /dev/md0
```

mkswap (make swap)

常用指数:



功能说明: 设置交换区 (swap area)。

语 法: mkswap [-cf] [-v0] [-v1] [设备名称或文件] [交换区大小]

包名称: util-linux。

相关命令: fdisk, swapon, swapoff。

补充说明: mkswap可将磁盘分区或文件设为Linux的交换区。

Linux支持新旧两种型式的交换区。旧式交换区会记录交换区中损毁的块。由于设计架构上的缘故，旧式交换区的上限为128 MB。新式交换区舍弃了记录损毁块的功能，这是由于分区若已出现损毁的块，大概也没人敢拿来当交换区了。舍弃此项功能后，新式交换区的上限可达到2 GB。而交换区大小的上限值会依CPU类型不同而异。

参 数:

-c	建立交换区前，先检查是否有损毁的块
-f	在SPARC电脑上建立交换区时，要加上此参数
-v0	建立旧式交换区，若分区的容量不超过128 MB，此为默认值
-v1	建立新式交换区，若分区的容量超过128 MB，此为默认值
[交换区大小]	指定交换区的大小，单位为1 024字节。此参数通常可省略，mkswap会利用整个分区或整个文件来建立交换区

范 例:

① 设置 /dev/hdb1 为新式交换区:

```
# mkswap -v1 /dev/hdb1
```

```
Setting up swap space version 1, size = 123342 KB
```

↑ “1” 表示新式交换区

此交换区容量为123342KB

```
# swapon /dev/hdb1
```

← 建立好交换区之后，再将它启动

② 设置/dev/hdb1为旧式交换区：

```
# mkswap -v0 /dev/hdb1
```

```
Setting up swapspace version 0, size = 123342KB
```

交换区的容量为123342KB

“0”表示旧式交换区

③ 设置/dev/hdb1为交换区，并指定其容量为1 000 000 Kbytes：

```
# mkswap /dev/hdb1 1000000
```

④ 设置/dev/hdb1为交换区，建立前并先检查是否有损毁的块：

```
# mkswap -c /dev/hdb1
```

mktemp (make temporary file)

常用指数：



功能说明：建立临时文件。

语法：mktemp [-qu] [文件名参数]

包名称：mktemp。

补充说明：mktemp可建立一个临时文件，供shell script使用。此临时文件有独特的文件名，以防黑客猜出文件名而加以利用。

参数：

-q	执行时若发生错误，不会显示任何信息。通常是在执行shell script时，若不想在屏幕上显示错误信息，即可使用此参数
-u	临时文件会在mktemp结束前先行删除
[文件名参数]	文件名参数必须是以“自定义名称.XXXXXX”的格式。例如，temp.XXXXXX。mktemp会根据文件名参数，建立一个独特文件名的临时文件

M

范 例:

建立临时文件:

```
# mktmp temp.XXXXXX ← 这类说明通常是放在shell script中
temp.ISkwVM          ← 建好的临时文件
```

mlabel (make label)

常用指数:



功能说明: 显示或设置MS-DOS磁盘的标签名称。

语 法: mlabel [-csvV] [磁盘代号] [标签名称]

包 名 称: mtools。

相关命令: mformat, mkdosfs。

补充说明: mlabel为mtools工具命令, 模拟MS-DOS的label命令, 可显示或设置MS-DOS磁盘的标签名称。

若仅指定磁盘代号, mlabel会以互动的方式让用户设置软盘标签。

参 数:

-c	清除标签名称
-s	显示标签名称
-v	执行时显示详细的信息
-V	显示版本信息

范 例:

① 显示a:磁盘片的标签名称:

```
# mlabel -s a:
Volume label is win95
```

② 将a:磁盘的标签名称改为win98:

```
# mlabel a:win98
```

③ 以交谈式的方式更改a:磁盘的标签名称:

```
# mlabel a:
Volume label is WIN98
Enter the new volume label : win2000
```

④ 清除a:磁盘片的标签名称:

```
# mlabel -c a:      ← 清除标签名称
# mlabel -s a:      ← 显示当前的标签名称
Volume has no label ← 标签名称已经被清除
```

mmd (MS-DOS make directory)

常用指数:



功能说明: 在MS-DOS文件系统中建立目录。

语法: mmd [目录...]

包名称: mtools。

相关命令: mkdir, mdeltree, mrd。

补充说明: mmd为mtools工具命令, 模拟MS-DOS的md命令, 可在MS-DOS的文件系统中建立目录。


参数: 无。

范例:

建立名称为sebastian的目录:

```
# mmd A:/Sebastian
```


mmount (MS-DOS mount)

常用指数: 

功能说明: 加载MS-DOS文件系统。

语 法: mmount [磁盘代号] [mount 参数]

包 名 称: mtools。

相关命令: mount, umount。

补充说明: mmount为mtools工具命令, 可根据[mount 参数]中的设置, 将软盘内容加载到Linux的目录中。若不指定[mount参数], 则mmount会根据/etc/fstab中的设置来加载目录。

参 数: [mount 参数]的用法请参考mount命令。

范 例:

① 将A:磁盘中的MS-DOS软盘, 加载到/mnt/floppy目录下:

```
# mmount A: -t msdos /dev/fd0 /mnt/floppy
```

mount参数

```
# ls /mnt/floppy
```

```
dosfile1  dosfile2 ← 磁盘中的内容已载入到此目录
```

② 依据/etc/fstab的设置, 加载A:的文件系统:

```
# mmount A:
```

mmove (MS-DOS move)

常用指数: 

功能说明: 在MS-DOS文件系统中, 移动文件或目录, 或更改名称。

语 法: mmove [源文件或目录...] [目的文件或目录]

包 名 称: mtools。

相关命令: mv。

补充说明: mmove为mtools工具命令, 模拟MS-DOS的move命令, 可在MS-DOS文件系统中移动现有的文件或目录, 或是更改现有文件或目录的名称。

参 数: 无。

范 例:


① 将dosfile1文件改名为dosfile2:

```
# mmove dosfile1 dosfile2
```

② 将dosfile移至dosdir目录下:

```
# mmove dosfile A:/dosdir
```

modinfo (module information)

常用指数: 

功能说明: 显示kernel模块的信息。

语 法: modinfo [-adhpV] [模块文件]

包 名 称: modutils。



相关命令: depmod, insmod, lsmod, modprobe, rmmod。

补充说明: modinfo会查看kernel模块的物件文件, 以显示该模块的相关信息。

参 数:

-a或--author	显示模块开发人员
-d或--description	显示模块的帮助
-h或--help	显示帮助
-p或--parameters	显示模块所支持的参数
-V或--version	显示版本信息

范 例:

① 显示指定模块的帮助:

```
# modinfo -d 8139too.o
```



这是网卡的模块

```
"RealTek RTL-8139 Fast Ethernet driver"
```

② 显示模块的开发人员:

```
# modinfo -a nfs.o
```

```
"Olaf Kirch <okir@monad.swb.de>"
```

③ 显示模块所支持的参数:

```
# modinfo -p ne.o
```



modprobe (module probe) 常用指数:

功能说明: 自动处理可载入模块。

语 法: modprobe [-acdlrtvV] [--help] [模块文件] [符号名称=符号值]

包 名 称: modutils。

相关命令: depmod, insmod, lsmod, modinfo, rmmod。

补充说明: modprobe可载入指定的个别模块,或是载入一组相依的模块。modprobe会根据depmod所产生的模块相依关系,决定要载入哪些模块。若在载入程序中发生错误,则modprobe会卸除整组的模块。

参 数:

-a或--all	载入全部的模块
-c或--show-conf	显示所有模块的设置信息
-d或--debug	使用排错模式
-l或--list	显示可用的模块
-r或--remove	卸除闲置不用的模块
-t或--type	指定模块类型
-v或--verbose	执行时显示详细的信息
-V或--version	显示版本信息
--help	显示帮助

范 例:

① 列出cdrom类型所有可用的模块:

```
# modprobe -l -t cdrom  
/lib/modules/2.4.20-8/kernel/drivers/cdrom/cdrom.o
```

② 显示所有模块的设置信息:

```
# modprobe -c
```

③ 载入指定的appletalk模块:

```
# modprobe appletalk
```

④ 载入ne模块, 并指定irq符号名称的值为3:

```
# modprobe ne irq=3
```

⑤ 卸除闲置不用的sb模块:

```
# modprobe -r sb
```

⑥ 载入ipv4模块类型中的ipip一个模块:

```
# modprobe -t /lib/modules/2.4.20-8/kernel/net/ipv4/ipip
```

⑦ 载入net模块类型的所有模块:

```
# modprobe -t /lib/modules/2.4.20-8/kernel/drivers/net/*
```



more

常用指数:



功能说明: 使文件能逐页地显示。

语 法: more [-dlfpsu] [-<行数>] [+<字符串>] [+<行数>][文件]

包 名 称: util-linux。

相关命令: less。

补充说明: more可将文件内容显示于屏幕上，并在画面下方列出当前显示的百分比。

参 数:

-d	在画面下方显示“Press space to continue, ‘q’ to quit.”，用户若按错键，则显示“Press ‘h’ for instructions.”
-l	more默认在碰^L（送纸字符）时会暂停。使用“-l”参数可取消此功能
-f	使程序在计算行数时，以实际的行数来计算，而非使用自动换行后的行数
-p	在显示每页内容时，不采用滚动画面的方式，而是先将屏幕清除干净，然后再显示该页的内容
-c	与“-p”类似，不采用卷动画面的方式，而是先显示内容，然后清除留在屏幕上的其他数据
-s	若出现连续的空白行，则将其合并成一行
-u	在有些文本文件中，有的字符有下底线，使用此参数则不显示文本的底线
-<行数>	指定每次要显示的行数
+<字符串>	在文件中查找选项中指定的字符串，然后显示字符串所在该页的内容
+<行数>	从指定的行数开始显示

范 例:

① 显示textfile文本文件，可利用稍后介绍的操作命令来浏览文件：

more textfile

② 以每次10行的方式显示textfile文本文件:

more -10 textfile

③ 指定从第25行开始显示textfile文本文件:

more +25 textfile

④ 在textfile文本文件中查找“zhora”字符串, 然后从该处开始显示文件内容:

more +/zhora textfile

⑤ 显示textfile文本文件, 若该文件有连续的空白行, 则仅以1行显示:

more -s textfile

⑥ 显示textfile文本文件, 但不显示文本的底线:

more -u textfile

⑦ 显示textfile文本文件, 当按■键计算行数时, 以实际的行数计算:

more -f textfile

⑧ 显示textfile文本文件, 并在画面下方显示“Press space to continue, 'q' to quit”的提示符号:

more -d textfile

操作命令:

Space 或 z	往下移 “-<行数>” 所指定的行数。若无指定此参数，则往下移1页
D 或 Ctrl + D	往下移11行
Enter	往下移1行
s	跳过后1行不显示
F	跳过 “-<行数>” 所指定的行数不显示。若无指定此参数，则跳过1页
B 或 Ctrl + B	往上移 “-<行数>” 所指定的行数。若无指定此参数，则往上移1页
/ <字符串>	往后查找指定的字符串
N	继续查找下一个字符串
=	显示当前所在的行数
: F	显示文件名称，及当前所在的行数
! <命令> 或 : ! <命令>	执行指定的命令
.	执行上次执行的命令
q 或 Q	结束程序
H	显示操作帮助

mount

常用指数:



功能说明: 加载文件系统。

语 法: mount [-afFhnrVw] [-L <标签>] [-o <选项>] [-t <文件系统类型>] [设备名称] [加载点]

包名称: mount。

相关命令: mmount, umount。

补充说明： mount可将指定设备中的文件系统加载到Linux的目录下（也就是加载点）。管理员可在参数中直接指定设备名称与文件系统类型。对于一些经常使用的设备，也可在/etc/fstab文件中设置，以方便加载。

mount在加载设备时会记录信息在/etc/mtab文件中。卸除设备时（umount命令），则将记录清除。

参 数：

-a	加载/etc/fstab中设置的所有设备
-f	模拟加载，但不会实际去加载设备。可与“-v”参数一起使用，查看mount的执行过程
-F	要跟“-a”参数一起使用。所有在/etc/fstab中设置的设备会同时间被加载，并没有任何先后顺序关系，所以可以加快执行速度
-h	显示帮助
-L <标签>	加载文件系统标签为<标签>的设备
-n	不要将加载的信息记录在/etc/mtab文件中
-o <选项>	<p>指定加载文件系统时的选项。有些选项也可在/etc/fstab中使用。可用选项帮助如下：</p> <p>async 以非同步的方式执行文件系统的输入输出动作</p> <p>atime 每次存取时都更新inode的存取时间，此为默认值，取消选项为noatime</p> <p>auto 必须在/etc/fstab文件中指定此选项。执行“-a”参数时，会加载设置为auto的设备，取消选项为noauto</p> <p>defaults 使用默认的选项。默认的选项为rw、suid、dev、exec、auto、nouser与async</p> <p>dev 可解读文件系统上的字符或块设备，取消选项为nodev</p> <p>exec 可执行二进制文件，取消选项为noexec</p> <p>noatime 每次存取时，不用更新inode的存取时间</p>

-o <选项>	<p>noauto 无法使用“-a”参数来加载</p> <p>nodev 不要解读文件系统上的字符或块设备</p> <p>noexec 无法执行二进制文件</p> <p>nosuid 关闭set-user-identifier (设置用ID) 与set-group-identifer (设置组ID) 设置位</p> <p>nouser 使一般用户 (非管理员) 无法执行加载动作, 此为默认值</p> <p>remount 重新加载设备。通常用于改变设备的设置状态, 例如从只读模式改成可读写模式</p> <p>ro 以只读模式加载</p> <p>rw 以可读写模式加载</p> <p>suid 打开set-user-identifier (设置用户ID) 与set-group-identifer (设置组ID) 设置位, 取消选项为nosuid</p> <p>sync 以同步的方式执行文件系统的输入输出动作</p> <p>user 可让一般用户加载设备</p>
-r	以只读的方式加载设备
-t <文件系统类型>	<p>指定设备的文件系统类型。常用的选项帮助如下:</p> <p>minix Linux最早使用的文件系统</p> <p>ext2 Linux 当前最常用的文件系统</p> <p>msdos MS-DOS的FAT</p> <p>vfat Win 95 / 98的VFAT</p> <p>nfs 网络文件系统</p> <p>iso9660 CD-ROM光盘片的标准文件系统</p> <p>ntfs Windows NT 4.0/2000的文件系统</p> <p>hpfs OS/2文件系统。Windows NT 3.51之前的版本也使用HPFS</p> <p>auto 自动检测文件系统</p> <p>若不加载指定的文件系统, 可在上述选项前面加上“no”</p>
-v	执行时显示详细的信息
-V	显示版本信息
-w	以可读写模式加载设备, 此为默认值

范 例:

① 加载/dev/hdb1磁盘的VFAT文件系统:

```
# mount -t vfat /dev/hdb1 /mnt/vfat
```

指定载入的
是vfat文件系统

设备名称

载入点

② 以只读模式加载/dev/hdb2磁盘的ext2文件系统:

```
# mount -t ext2 -o ro /dev/hdb2 /mnt/ext2
```

只读模式

③ 加载软盘的MS-DOS文件系统:

```
# mount -t msdos /dev/fd0 /mnt/floppy
```

④ 加载光驱 (需先放入所要读取的光盘):

```
# mount -t iso9660 /dev/cdrom /mnt/cdrom
```

```
mount: block device /dev/cdrom is write-protected,  
mounting read-only
```

⑤ 以自动检测文件系统的方式加载/dev/hdb3设备, 执行时并显示详细的信息:

```
# mount -v -t auto /dev/hdb3 /mnt/dir
```

自动检测文件系统

执行时显示详细信息

```
mount: you didn't specify a filesystem type for /dev/hdb3  
I will try type ext3
```

```
/dev/hdb3 on /mnt/dir type ext3 (rw)
```

检测结果, 载入的是ext3文件系统

⑥ 加载/dev/hdb2磁盘的ext2文件系统，但不允许执行此设备的执行文件：

```
# mount -t ext2 -o noexec /dev/hdb2 /mnt/ext2
```

不可以执行执行文件

⑦ 模拟加载/dev/hdb4磁盘的ntfs文件系统：

```
# mount -f -t ntfs /dev/hdb4 /mnt/ntfs
```

⑧ 加载/etc/fstab设置文件中的所有设备：

```
# mount -a
```

⑨ 以同步的方式加载/etc/fstab设置文件中的所有设备：

```
# mount -F -a
```

⑩ 加载/etc/fstab设置文件中除了文件系统类型为vfat的其他所有设备：

```
# mount -a -t novfat
```

在文件系统名称前面加上“no”，
表示不载入此类型的文件系统

⑪ 加载/etc/fstab设置文件中，文件系统类型为minix的设备：

```
# mount -a -t minix
```

⑫ /etc/fstab设置中已经有以下的设置：

```
/dev/hdb3            /mnt/minix            minix    defaults    0 0
```



如果要加载此设备上的文件系统，可执行以下命令：

```
# mount /dev/hdb3
```

或执行以下命令亦可：

```
# mount /mnt/minix
```

⑬ 列出当前加载的vfat文件系统：

```
# mount -t vfat
```

```
/dev/hda1 on /mnt/vfat type vfat (rw)
```

mouseconfig

常用指数：



(mouse configure)

功能说明：设置鼠标相关参数。

语 法：mouseconfig

包名称：redhat-config-mouse (Red Hat Linux/Fedora Core 1), system-config-mouse (Fedora Core 2&3)。

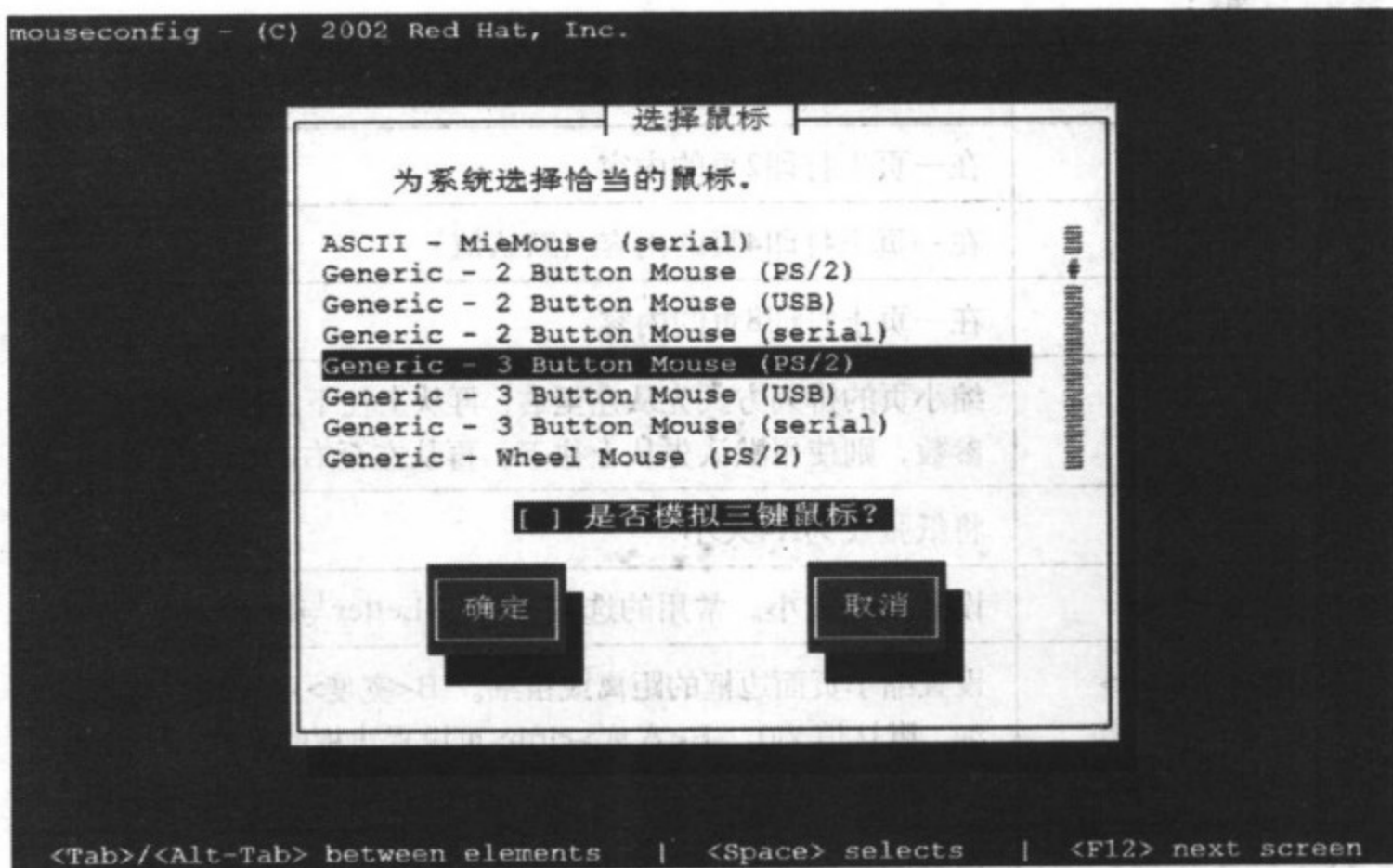
补充说明：以互动方式设置鼠标。

参 数：无。

范 例:

① 执行mouseconfig, 会出现以下操作画面, 可利用 **Tab**、**Alt + Tab**、**F10**、空白键及方向键来设置鼠标:

```
# mouseconfig
```



mpage (multiple pages)

常用指数:



功能说明: 在PostScript打印机上, 将许多页面合并成一页来打印。

语 法: mpage [-1248aAcEfHloOrRStTUv] [-b <纸张大小>] [-B <宽度><lrtb>] [-C <字符码文件>] [-d <a 或 p>] [-D <日期时间格式>] [-F <字体类型>] [-h <表头>] [-I <缩进>] [-j <打印范围>] [-L <行数>] [-m <宽度><lrtb>] [-M <宽度><lrtb>] [-P <打印机>] [-W <宽度>] [-X<表头>] [-z <打印命令>] [文件...]

M

包名称: mpage。

相关命令: enscript。

补充说明: mpage可读取文本文件或PostScript文件, 将许多页的内容浓缩成一页来打印。

参 数:

-1	在一页上打印1页的内容
-2	在一页上打印2页的内容
-4	在一页上打印4页的内容 (默认值)
-8	在一页上打印8页的内容
-a	缩小页的排列方式先从左至右、再从上往下。若不指定“-a”参数, 则使用默认先从上往下、再从左至右的排列方式
-A	将纸张设为A4大小
-b <纸张大小>	设置纸张大小。常用的选项有A4、Letter与Legal等
-B <宽度><lrtb>	设置缩小页面边框的距离或粗细。-B<宽度>可设置边框的粗细, 默认值为0。-B<宽度><lrtb>可设置边框的距离, l、r、t、b分别表示边框的左、右、上、下四条线, 默认的距离为0
-c	在打印多个文件时, 默认不同的文件会分页打印。若要让不同文件能在同一页上连起来打印, 请使用此参数
-C <字符码文件>	指定字符码编码的文件。若不指定“-C”参数, 则使用默认的Latin-1字符码
-d <a或p>	强制将文件视作ASCII文件 (选项设为a), 或PostScript文件 (选项设为p)
-D<日期时间格式>	设置打印表头的时间格式。设置格式与strftime()函数相同
-E	以4页为一个单位, 但在一页上只打印第2页与第3页的内容 (相较于“-O”参数)
-f	若句子太长超出页面范围 (指缩小以后的页面), 则自动换行
-F <字体类型>	指定字体类型, 默认字体为Courier
-h <表头>	指定表头的内容。此参数必须与“-H”参数一起使用

-H	在缩小页面的每一页上加上表头。默认加上的表头（不使用“-h”参数）包含了文件上次修改日期、文件名与页码
-I <缩进>	指定缩进的宽度，单位为字符数
-j <打印范围>	指定要打印的范围。<打印范围>的格式为<起始页面-结束页面%隔页>。例如，-j 5-10%1表示要打印第5页至第10页之间的每一页（含第5页与第10页）。-j 2%2（省略结束页面）表示要打印第2页至最后一页之间的偶数页
-l	指定横式（55行，每行132个字符）打印。若不指定“-l”参数，则使用默认的直式（66行，每行80个字符）打印
-L <行数>	指定缩小页面中，每页的行数
-m <宽度><lrtb>	设置纸张的边界。宽度的单位为点数，l、r、t、b分别表示左、右、上、下边界。默认的边界为20点，若仅使用“-m”参数，但不加上宽度，则将边界设为40点
-M <宽度><lrtb>	指定缩小页中的打印边界。选项的格式与“-m”参数相同。默认的边界为8点，若仅使用“-M”参数，但不加上宽度，则将边界设为4点
-o	在缩小页面的四周不要加上边框。若不使用此参数，则默认会加上边框
-O	以4页为一个单位，但在一页上只打印第1页与第4页的内容。请注意此参数是英文字母大写的“O”，不是阿拉伯数字的“0”
-P <打印机>	指定打印机设备
-r	从最后一页开始打印
-R	改变缩小页面的排列方式，第1页从左下角开始排起
-S	在缩小页面时，默认X轴与Y轴的缩小比例是一样的。使用“-S”参数可允许X轴与Y轴有不同的缩小比例，以便节省打印空间
-t	指定双面打印。若不指定“-t”参数，则默认使用单面打印。打印机须具备双面打印功能，才能使用此参数
-T	指定在双面打印时，每隔多少页就翻页。此参数必须配合“-t”参数一起使用
-U	将纸张设为Letter大小
-v	在屏幕上显示总共打印的张数
-W <宽度>	指定缩小页面中，每行的长度，单位为字符数

-X<表头>	在打印纸张的中间上方加上表头。若未指定<表头>，则在左上方加上文件名，右上方加上页码。请注意，使用时在<表头>选项与-X之间不可加上空白字符
-z <打印命令>	指定要送出的打印命令，默认的命令为lpr

范 例：

① 以每张纸2页的方式打印printdoc文件：

```
# mpage -2 printdoc
```

② 从最后一页开始打印printdoc文件：

```
# mpage -r printdoc
```

③ 以每张纸8页的方式打印printdoc文件，并且指定不要在每页加上边框：

```
# mpage -8 -o printdoc
```

↑ ↑
 └─ 每页不加上边框
 ↑
 每张纸打印8页

④ 以较节省空间的方式打印printdoc文件，并指定使用Times-Roma字体类型：

```
# mpage -s -f Times-Roma printdoc
```

↑ |
 └─ 使用的字体类型
 ↑
 以较节省空间的方式

⑤ 打印printdoc文件，在每页加上表头，并指定表头的时间格式为“月/日/年”：

```
# mpage -H -D %m/%d/%y printdoc
```

↑ |
 └─ 指定表头的时间格式
 ↑
 加上表头

M

- ⑥ 打印printdoc文件，在每页加上表头，并指定表头内容为“Blade Runner”：

```
# mpage -H -h "Blade Runner" printdoc
```

指定表头的内容

- ⑦ 打印printdoc文件到lp1打印机，并指定每张缩小页的边框宽度为2点：

```
# mpage -B2 -P lp1 printdoc
```

缩小页的边框宽度

打印到lp1打印机

- ⑧ 打印printdoc时，并指定纸张的左边界为25点，右边界为20点，上下边界为35点：

```
# mpage -m25 120r 35tb printdoc
```

上下边界

右边界

左边界

- ⑨ 打印printdoc文件，每个缩小页的边界为6点，并指定从左下角开始打印第1页：

```
# mpage -M6 -R printdoc
```

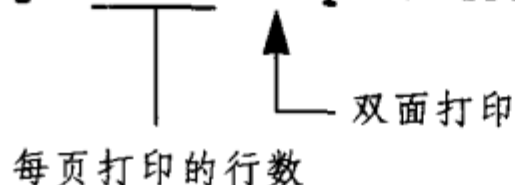
每页的边界 从左下角开始打印

- ⑩ 打印printdoc文件，并指定每行打印70个字符：

```
# mpage -W 70 printdoc
```

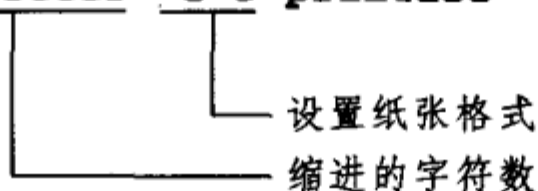
⑩ 打印printdoc文件，指定每页打印 60 行，并采用双面打印：

```
# mpage -L 60 -t printdoc
```



⑪ 打印printdoc文件，并指定纸张为Letter的尺寸，每行缩进5个字符：

```
# mpage -b Letter -I 5 printdoc
```



⑫ 打印printdoc文件，以4页为单位，但在一页上只打印该单位的第1页及第4页：

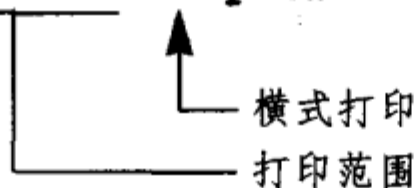
```
# mpage -O printdoc
```

⑬ 打印printdoc文件，以4页为单位，但在一页上只打印该单位的第2页及第3页：

```
# mpage -E printdoc
```

⑭ 打印printdoc文件的第3页至第7页，并指定使用横式打印：

```
# mpage -j 3-7&1 -l printdo
```



⑮ 打印printdoc1及printdoc2文件，并允许两个文件能在同一页上连着打印：

```
# mpage -c printdoc1 printdoc2
```

mpartition (MS-DOS partition) 常用指数:

功能说明: 建立或删除MS-DOS的分区。

语 法: mpartition [-acdfIprv] [-b <扇区数>] [-h <磁头数>] [-l <扇区数>] [-s <扇区数>] [-t <柱面数>] [磁盘代号]

包名称: mtools。

相关命令: mformat。

补充说明: mpartition为mtools工具命令，可建立或删除磁盘分区。通常用来建立Jaz或Zip软盘的分区。

参 数:

-a	将分区设置为可开机分区
-b <扇区数>	建立分区时，指定要从第几个扇区开始建立分区。若不指定，则从第1个扇区，或紧接着上一个分区开始建立
-c	建立分区
-d	将分区设置为无法开机的分区
-f	强制地修改分区而不管检查时发生的错误信息
-h <磁头数>	建立分区时，指定分区的磁头数
-l	删除全部的分区
-l <扇区数>	建立分区时，指定分区的容量大小，单位为扇区数。若不指定，则以磁头数、柱面数及每个磁道的扇区数来计算容量。以上几个参数若都不指定，mpartition考虑了磁盘及已存在的分区大小之后，会尽可能地建立最大容量
-p	当要重新建立分区时，显示命令行
-r	删除分区
-s <扇区数>	建立分区时，指定每个磁道的扇区数

-t <柱面数>	建立分区时，指定分区的柱面数
-v	与“-p”参数一起使用，若没有同时下达修改分区的命令，则显示当前分区的状态。否则显示修改后的分区状态

范 例：

建立E:软盘的分区，并设成可开机的分区：

```
# mpartition -a -c E:
```



mrd (MS-DOS remove directory)

常用指数：

功能说明：删除MS-DOS文件系统中的目录。

语 法：mrd [目录...]

包 名 称：mtools。

相关命令：mdeltree, mmd。

补充说明：mrd为mtools工具命令，模拟MS-DOS的rd命令，可删除MS-DOS的目录。若目录尚有其他文件或子目录，则无法用mrd来删除，必须使用mdeltree命令。

参 数：无。

范 例：

删除dosdir目录：

```
# mrd A:/dosdir
```



mren (MS-DOS rename)

常用指数:



功能说明: 更改MS-DOS文件或目录的名称, 或移动文件或目录。

语 法: `mren [源文件或目录...] [目的文件或目录]`

包 名 称: mtools。

相关命令: mv。

补充说明: mren为mtools工具命令, 模拟MS-DOS的ren命令, 可更改MS-DOS文件或目录名称。除此之外, mren也可移动文件或目录, 但仅限于在同一设备内 (也就是同一个磁盘代号)。

参 数: 无。

范 例:

① 将dosfile1文件改名为dosfile2:

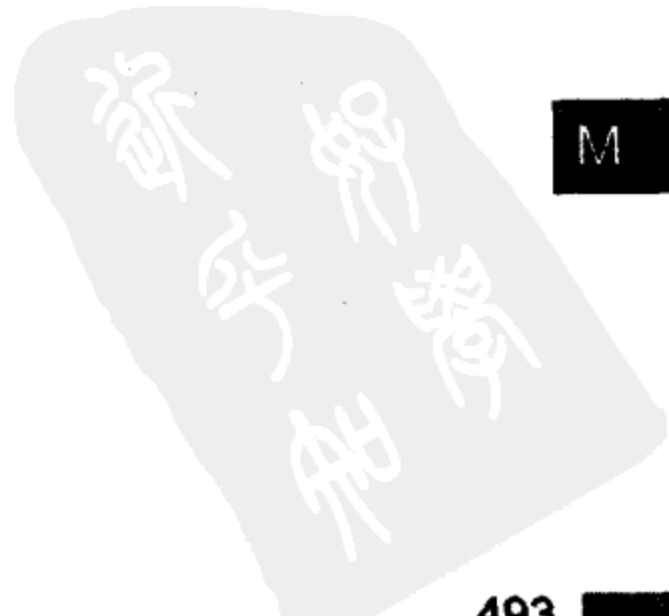
```
# mren dosfile1 dosfile2
```

② 将dosdir1目录改名为dosdir2, 并移至A:/目录下:

```
# mren dosdir1 A:/dosdir2
```

③ 将dosfile1与dosfile2文件, 移至A:/temp目录下:

```
# mren dosfile1 dosfile2 A:/temp
```



mshowfat

常用指数: 

(MS-DOS show FAT)

功能说明: 显示MS-DOS文件在FAT中的记录。

语 法: mshowfat [文件...]

包 名 称: mtools。

补充说明: mshowfat为mtools工具命令，可显示MS-DOS文件在FAT中的记录编号。

参 数: 无。

范 例:

显示c:/dosdir/msdos.txt在FAT中的记录:

```
# mshowfat c:/dosdir/msdos.txt  
C:/dosdir/msdos.txt <329>
```

mtools (MS-DOS tools)

常用指数: 

功能说明: 显示mtools支持的命令。

语 法: mtools

包 名 称: mtools。

相关命令: mattrib, mbadblocks, mcd, mcopy, mdel, mdeltree, mdir, mdu, mformat, minfo, mlabel, mmd, mmount, mpartition, mrd, mmove, mren, mshowfat, mtoolstest, mtype, mzip。

补充说明： mtools为MS-DOS文件系统的工具程序，可模拟许多MS-DOS的命令。这些命令都是mtools的符号链接，因此会有一些共同的特性。以下将说明mtools命令的共同特性，至于每个命令的功能，请参阅个别命令的说明。

mtools的设置文件为/etc/mtools.conf，默认会将A:磁盘代号指向/dev/fd0设备，并将mtools的工作目录设为A:/。若系统中有MS-DOS的分区，也可在mtools.conf中加入以下语句：

```
drive c: file="/dev/hda4"
```

如此，mtools命令即可透过C:磁盘代号来存取/dev/hda4上的MS-DOS文件系统（不必使用mount命令将设备加载到Linux目录）。

使用mtools时，若须指定目录，可用Linux惯用的“/”或MS-DOS惯用的“\”。不过，“\”在使用上有较多的限制，因此仍然建议使用“/”。

mtools当前支持VFAT（Windows 95/98的文件系统），可自动将Linux的长文件名转换为VFAT所支持的长文件名，并产生8.3格式的短文件名。

mtools除了针对各别命令提供各种参数外，也有一些共同参数，适用于大部分的mtools命令。以下说明这些共同参数：

参 数：

-a	长文件名重复时（指定的目的文件的长文件名与现有文件的长文件名重复），自动更改目的文件的长文件名。更改的方式是在长文件名后面加上“-t”，t为从1开始算起的整数
-A	短文件名重复但长文件名不同时，自动更改目的文件的短文件名。更改的方式是在短文件名后面加上“~t”，t为从1开始算起的整数
-o	长文件名重复时，将目的文件覆盖现有的文件

-O	短文件名重复但长文件名不同时，将目的文件覆盖现有的文件
-r	长文件名重复时，要求用户更改目的文件的长文件名
-R	短文件名重复但长文件名不同时，要求用户更改目的文件的短文件名
-s	长文件名重复时，则不处理该目的文件
-S	短文件名重复但长文件名不同时，则不处理该目的文件
-v	执行时显示详细的帮助
-V	显示版本信息

范 例：

列出mtools所有的命令（依mtools版本而有所差异，有些命令可能不支持）：

mtools

Supported commands:

mattrib, mbadblocks, mcat, mcd, mcopy, mdel, mdeltree, mdir
 mdoctorfat, mdu, mformat, minfo, mlabel, mmd, mmount,
 mpartition
 mrd, mread, mmove, mren, mshowfat, mtoolstest, mtype, mwrite
 mzip

mtoolstest (MS-DOS tools test)

常用指数：



功能说明：测试并显示 mtools 的相关设置。

语 法：mtoolstest

包 名 称：mtools。

相关命令：mtools。

M



补充说明: mtoolstest为mtools工具命令, 可读取与分析mtools的设置文件, 并在屏幕上显示结果。

参 数: 无。

范 例:

测试mtools的设置:

```
# mtoolstest
```

```
drive J:
```

```
#fn=0 mode=0 builtin
file="/dev/sdb4" fat_bits=16
tracks=0 heads=0 sectors=0 hidden=0
offset=0x0
partition=0
mformat_only
```

```
drive Z:
```

```
#fn=0 mode=0 builtin
file="/dev/sdb4" fat_bits=16
tracks=0 heads=0 sectors=0 hidden=0
offset=0x0
partition=0
mformat_only
```

```
drive X:
```

```
#fn=0 mode=0 builtin
file="#DISPLAY" fat_bits=0
tracks=0 heads=0 sectors=0 hidden=0
offset=0x0
partition=0
```

```
...
```

mtype (MS-DOS type)

常用指数: 

功能说明: 显示MS-DOS文件的内容。

语 法: mtype [-st] [文件...]

包 名 称: mtools。

相关命令: cat。

补充说明: mtype为mtools工具命令, 模拟MS-DOS的type命令, 可显示MS-DOS文件的内容。

参 数:


-s	去除8位字符码集的第一个位 (High bit), 使它兼容于7位的ASCII
-t	将MS-DOS文本文件中的“换行+光标移至行首”字符转成Linux的换行字符

范 例:

显示MS-DOS文件内容:

```
# mtype -t msdos.txt:
```

mutt

常用指数: 

功能说明: E-mail管理程序。

语 法: mutt [-hnpRvxz] [-a <文件>] [-b <地址>] [-c <地址>] [-f <邮件文件>] [-F <设置文件>] [-H <邮件草稿>] [-i <文件>] [-m <类型>] [-s <主题>] [邮件地址]

包名称: mtools。

相关命令: fetchmail, mail, mailq, metamail, pine。

补充说明: mutt是一个文本模式的邮件管理程序, 提供了全屏幕的操作界面。

mutt是由elm衍生而来, 两者相当类似, 但mutt功能较强, 也比较容易自定义功能。mutt程序的相关设置存储在用户设置文件(.muttrc文件)以及程序设置文件(/etc/Muttrc)中。在这些设置文件中, 可指定 mutt 所使用的文本编辑程序、邮件文件目录、PGP (一种应用在E-mail的加密协议) 目录等参数。

参 数:

-a <文件>	在邮件中加上附加文件
-b <地址>	指定密件副本的收信人地址
-c <地址>	指定副本的收信人地址
-f <邮件文件>	指定要载入的邮件文件
-F <设置文件>	指定mutt程序的设置文件, 而不读取默认的.muttrc文件
-h	显示帮助
-H <邮件草稿>	将指定的邮件草稿 (包含邮件表头与内容) 送出
-i <文件>	将指定文件插入邮件内文中
-m <类型>	指定默认的邮件信箱类型
-n	不要去读取程序设置文件 (/etc/Muttrc)
-p	在mutt中编辑完邮件后, 而不想将邮件立即送出, 可将该邮件暂缓寄出 (postpone)。若要恢复之前暂缓的邮件, 请使用本参数
-R	以只读的模式打开邮件文件, 在此模式下不得删除邮件
-s <主题>	指定邮件的主题
-v	显示mutt的版本信息, 及当初编译此文件时所给予的参数
-x	模拟mailx (一种E-mail管理程序) 的编辑方式
-z	若邮件文件中没有邮件, 即不启动mutt

范 例:

① 执行mutt会出现以下画面，可利用稍后所介绍的操作命令收发邮件：

mutt

可执行的操作命令

```

q:離開 d:刪除 u:反刪除 s:儲存 m:信件 r:回覆 g:群組 ?:求助
64 O + May 09 Disk.Usage.Moni ( 3) Low disk space warning
65 O + May 13 Disk.Usage.Moni ( 3) Low disk space warning
66 O F May 13 To root@webmail ( 61) LogWatch for webmail.flag.com.tw
67 O F May 13 To root@webmail ( 8) Anacron job 'cron.daily'
68 O + May 13 Disk.Usage.Moni ( 3) Low disk space warning
69 O + May 23 Disk.Usage.Moni ( 3) Low disk space warning
70 O + Jul 04 Disk.Usage.Moni ( 3) Low disk space warning
71 O + Jul 04 Disk.Usage.Moni ( 3) Low disk space warning
72 O F Jul 04 To root@webmail ( 33) LogWatch for webmail.flag.com.tw
73 O F Jul 04 To root@webmail ( 8) Anacron job 'cron.daily'
74 O + Jul 04 Disk.Usage.Moni ( 3) Low disk space warning
75 O + Jul 04 Disk.Usage.Moni ( 3) Low disk space warning
76 O + Jul 04 Disk.Usage.Moni ( 3) Low disk space warning
77 + Jul 04 Mail Delivery S ( 48) Returned mail: see transcript for details
78 N + Jul 04 Disk.Usage.Moni ( 3) Low disk space warning
79 N + Jul 04 Disk.Usage.Moni ( 3) Low disk space warning
80 N + Jul 04 Disk.Usage.Moni ( 3) Low disk space warning
81 N + Jul 05 Disk.Usage.Moni ( 3) Low disk space warning
82 N F Jul 05 To root@webmail ( 90) LogWatch for webmail.flag.com.tw
83 N F Jul 05 To root@webmail ( 8) Anacron job 'cron.daily'
84 N + Jul 05 Disk.Usage.Moni ( 3) Low disk space warning
-----Mutt: /var/spool/mail/root Msgs:89 New:12 Old:76 Ssk -----(date/date)-(94%)
    
```

信件列表

读取的邮件
文件位置

总邮件数
还未读取的邮件数

全部邮件所占用的磁盘空间

② 将事先编辑好的邮件内容寄出：

```
# mutt -s Hello alkane@mail.flag.com.tw < content.txt
```

邮件主题

收件人地址

事先编写好的邮件

③ 将文本文件插入要送出的邮件中：

```
# mutt -i textfile -s 'Test mutt' terry@mail.flag.com.tw
```

指定要插入的文本文件

由于邮件主题有空格，
所以前后要加上引号

接着mutt会执行vi编辑器可供继续编辑邮件内容，此时指定插入的文本文件内容会设成邮件内容。

④ 寄发邮件给terry@mail.flag.com.tw，并指定副本及密件副本收信人：

```
# mutt -b alkane@mail.flag.com.tw -c sphenoid@mail.flag.com.tw
```

密件副本收信人

副本收信人

```
terry@mail.flag.com.tw
```

⑤ 寄发邮件给terry@mail.flag.com.tw，并在此封信夹入attach_file附加文件：

```
# mutt -a attach_file terry@mail.flag.com.tw
```

⑥ 执行mutt，并读取指定的/var/spool/mail/alkane邮件文件：

```
# mutt -f /var/spool/mail/alkane
```

⑦ 执行mutt将暂缓寄出的邮件寄出：

```
# mutt -p
```

⑧ 以只读的模式执行mutt：

```
# mutt -R
```

⑨ 执行mutt，若没有新邮件则不启动程序：

```
# mutt -z
```



操作命令： 执行mutt之后，画面上方会显示操作命令，以下介绍常用的操作命令：

I ~ 9	选择邮件
a	建立寄件人的别名
B	将此封信寄给其他人
c	打开另一个邮件信箱（邮件文件）
D	删除邮件。若要挽回被删除的邮件，只要在结束程序之前，都可以利用u命令挽回删除的邮件
E	编辑当前邮件，再重新寄出此信件
F	转寄邮件
G	恢复邮件给所有收信人
H	显示邮件的表头
J 、 ↓ 或 Page Up	移至下一封未被删除的邮件
K 、 ↑ 或 Page Down	移至上一封未被删除的邮件
l	只显示符合指定条件的邮件
m	寄送新邮件
O	将信件排序
p	打印当前邮件
Q	将读过的邮件移到mbox，将标示删除的邮件予以删除，然后结束程序
R	恢复邮件给寄信人
s	将指定的邮件存为文件
U	挽回被删除的邮件
V	显示MIME附件
w	设置当前邮件的状态标示
?	显示操作命令的帮助
Enter	读取邮件

mv (move files)常用指数: **功能说明:** 移动或更名现有的文件或目录。**语 法:** mv [-bfuiv] [-S <附加字尾>] [--help] [--version] [源文件或目录...] [目的文件或目录]**包 名 称:** coreutils。**相关命令:** mmove, mren。**补充说明:** mv可移动文件或目录，或是更改文件或目录的名称。**参 数:**

-b或--backup	若需覆盖文件，则覆盖前先行备份。备份文件默认的命名方式是在文件名后加上“~”
-f 或 --force	若[目的文件或目录]与现有的文件或目录重复，则直接覆盖现有的文件或目录
-i 或 --interactive	覆盖前先行询问用户
-S <附加字尾> 或 --suffix=<附加字尾>	与“-b”参数一起使用，可指定备份文件的所要附加的字尾
-u 或 --update	在移动或更改文件名时，若目的文件已存在，且其文件日期比源文件新（或相同），则不覆盖目的文件
-v 或 --verbose	执行时显示详细的信息
--help	显示帮助
--version	显示版本信息

范 例:

① 将file1文件改名为file2:

mv file1 file2

② 将file1及file2文件移动到dir目录下:

```
# mv file1 file2 dir
```

③ 将file文件移动到dir目录下, 若文件已有重复, 则直接覆盖文件:

```
# mv -f file dir
```

④ 将file1文件改名成file2, 若需覆盖文件, 则覆盖前先备份文件, 并以默认的方式命名备份文件:

```
# mv -b file1 file2
```

```
# ls
```

```
file2  file2~
```

↑
备份文件

⑤ 将file1文件改名成file2, 若需覆盖文件, 则覆盖前先备份文件, 并以原始文件名加上_bkup的方式来命名备份文件:

```
# mv -b -s _bkup file1 file2
```

```
# ls
```


```
file2  file2_bkup
```

↑
备份文件

⑥ 将file1文件改名成file2, 若需覆盖文件, 只有源文件的日期比目的文件新才覆盖文件:

```
# mv -u file1 file2
```

mzip

常用指数: 

功能说明: Zip/Jaz磁盘控制命令。

语 法: mzip [-efpqrwx]

包名称: mtools。

补充说明: mzip为mtools工具命令，可设置Zip或Jaz磁盘的保护模式，以及执行退出软盘的动作。

参 数:


-e	退出软盘
-f	与“-e”参数一起使用，不管是否已经加载软盘中的文件系统，一律强制退出软盘
-p	设置软盘的写入密码
-q	显示当前的状态
-r	将软盘设为写保护状态
-u	退出软盘以前，暂时解除软盘的保护状态
-w	将软盘设为可写入状态
-x	设置软盘使用密码保护

范 例:

将Zip软盘设为可写入的状态:

```
# mzip -w
```

nc

常用指数: **功能说明:** 连接与监听TCP/UDP通信端口。**语法:** nc [-hlnruz] [-g <网关>...] [-G <指向器数目>] [-i <延迟秒数>] [-o <输出文件>] [-p <通信端口>] [-s <源地址>] [-v...] [-w <超时秒数>] [主机名称] [通信端口...]**包名称:** nc。**相关命令:** netstat。**补充说明:** 该命令可用于打开TCP连接、传送UDP包、监听TCP与UDP通信端口，以及扫描各通信端口等。**参数:**

-g <网关>	设定来源路由跳 (Hop) 通信网关 (Gateway), 最多可设定8个
-G <指向器数目>	设定来源路由指向器, 其数值为4的倍数
-h	显示帮助
-i <延迟秒数>	设定间隔的时间, 以便传送信息及扫描通信端口, 单位以秒计算
-l	使用监听模式, 管控传入的数据
-n	直接使用IP地址, 而不通过名称服务器
-o <输出文件>	指定文件名称, 把往来传输的数据以十六进制码输出成该文件存储
-p <通信端口>	设定本地主机使用的通信端口
-r	随机指定本地与远程主机的通信端口
-s <源地址>	设定本地主机送出包的IP地址
-u	使用UDP (User Datagram Protocol) 传输协议
-v	显示命令执行过程。如重复指定这项参数多次, 则会显示更为详细的相关信息
-w <超时秒数>	设定等待连接的时间, 单位以秒计算, 超过此时间则为超时
-z	使用0输入/输出模式, 只在扫描通信端口时使用

范 例:

连接到192.168.0.135的21端口,以取得相关信息:

```
# nc 192.168.0.135 21
220 ProFTPD 1.2.7 Server (ProFTPD Default Installation)...
```

ncftp

常用指数: 

(nc file transfer protocol)

功能说明: 传输文件。

语 法: ncftp [-u <账号名称>] [主机名称或 IP 地址]

包 名 称: ncftp。

相关命令: ftp, ncftpget, ncftpput。

补充说明: ftp让用户得以下载存放于远程主机的文件,也能将文件上传至远程主机放置。NcFTP是文本模式 ftp 程序的佼佼者,它具备多样特色,包括显示传输速率、下载进度、自动续传、标注书签、可通过防火墙和代理服务器等等。

参 数:

-u <账号名称>	设定登录ftp服务器的账号,不使用匿名登录
-----------	-----------------------

范 例:

① 以匿名方式登录ftp.flag.com.tw服务器,可用稍后所介绍的操作命令来上传或下载文件:

```
# ncftp ftp.flag.com.tw
```

② 以指定的alkane账号登录ftp.flag.com.tw服务器：

```
# ncftp -u alkane ftp.flag.com.tw
```

操作命令：

ascii	将传输数据的模式设成ASCII模式
bgget <文件名称>	以后台的方式下载远程的文件
bgput <文件名称>	以后台的方式上传本地端的文件
bgstart	以后台的方式传输当前正在下载或下传的文件
binary	将传输数据的模式设成二进制模式
bookmark	将此次连接存储到书签中。此书签将会被存储成 \$HOME/ncftp/bookmarks 文件
bookmarks	列出书签的内容。可供编辑书签，或连接到书签中的ftp服务器
cat <文本文件>	显示远程文本文件的内容
cd <目录名称>	切换到远程指定的目录
chmod<文件权限> <文件名称>	改变远程文件的权限
close	结束此次连接但不结束ftp程序
debug <排错等级>	启动程序的排错模式
dir <目录名称>	列出远程目录的内容
get <文件名称>	下载远程的文件
help <操作命令>	列出操作命令的简要说明。若不指定<操作命令>参数，则列出所有可用的操作命令
jobs	列出当前正在后台执行的工作
lcd <目录名称>	切换本地端所在的目录
lchmod<文件权限> <文件名称>	改变本地端文件的权限
lls <目录名称>	列出本地端目录的内容
mkdir <目录名称>	在本地端建立新的目录

lookup <域名或IP地址>	查询主机的域名或IP地址。如输入的是域名，则列出其IP地址。若输入的是IP地址，则列出对应的域名
lpage <文件名称>	一页一页地显示本地端的文本文件
lpwd	列出本地端当前所在的目录位置
lrename<来源文件> <目的目录或文件>	将本地端来源文件改名为目的文件名称，或搬移到目的目录
lrm <文件名称>	删除本地端的文件
lrmdir <目录名称>	删除本地端的目录
ls <目录名称>	列出远程目录的内容
mkdir <目录名称>	在远程建立目录
open <主机地址>	与指定ftp服务器建立ftp连接
page <文件名称>	一页一页地显示远程的文本文件
pdir <目录名称>	一页一页地列出远程目录的内容
pls	一页一页地列出远程目录的内容
put <文件名称>	上传本地端的文件
pwd	列出远程当前所在的目录位置
quit	结束此次连接并结束ftp程序
quote <命令>	传送指定的ftp命令给ftp服务器
rename<来源文件> <目的目录或文件>	将远程来源文件改名为目的文件名称，或搬移到目的目录
rhelP	列出ftp服务器所支援的ftp命令
rm <文件名称>	删除远程的文件
rmdir <目录名称>	删除远程的目录
set <变量名称> <变量值>	设定环境变量
show <变量名称>	显示环境变量的变量值，如指定“all”则列出所有环境变量的变量值
site <命令>	传送给ftp服务器与服务器相关的命令
umask <权限掩码>	指定建立文件时默认的权限掩码
version	显示版本信息

ncftpget

常用指数：**功能说明：**下载文件。

语 法：ncftpget [-aAbFRTvVzZ] [-B <缓冲区大小>] [-d <记录文件>] [-f <设定文件>] [-p <密码>] [-P <通信端口>] [-r <连接次数>] [-t <超时秒数>] [-u <账号名称>] [-DD] [主机名称] [本地目录] [远程路径与文件名称...]或ncftpget [-aAbFRTvVzZ] [-B <缓冲区大小>] [-d <记录文件>] [-f <设定文件>] [-p <密码>] [-P <通信端口>] [-r <连接次数>] [-t <超时秒数>] [-u <账号名称>] [-DD] [ftp://<主机名称>/<远程路径名称>/<远程文件名称>]

包名称：ncftp。**相关命令：**ncftp, ncftpput。**补充说明：**ncftpget是NcFTP的公用程序，让用户可以直接使用参数、执行命令，下载文件。**参 数：**

-a	使用ASCII传输类型，而非默认的二进制类型
-A	下载文件时，若该文件名称已经存在，则将下载数据附加到已存在文件的后面，而非将其覆盖
-b	在后台进行下载操作
-B <缓冲区大小>	设定接收缓冲区的大小
-d <记录文件>	指定记录文件，记录ftp连接、传输过程，以便于排错
-DD	成功下载文件之后，就把它从远程主机上移除
-f <设定文件>	读取指定的设定文件内的主机名称、账号名称和密码，进行登录操作

-F	使用PASV (Passive) 模式连接数据, 只有在搭配安全系统 (如防火墙) 而遭遇问题时, 才附加这项参数
-p <密码>	设定登录ftp站点的密码, 不使用电子邮件作为密码
-P <通信端口>	指定通信端口, 不使用ftp服务默认的第21通信端口
-r <连接次数>	设定重复尝试连接的次数。当网络断线时, ncftpget可自动重新连接, 继续下载文件
-R	递归处理, 将指定目录下的所有文件及子目录一并处理
-t <超时秒数>	设定等待服务器响应的时间, 单位以秒计算, 超过此时间则为超时
-T	在递归处理模式中, 不使用TAR的格式下载
-u <账号名称>	设定登录ftp站点的账号, 不使用匿名登录
-v	显示下载进度, 此为默认值
-V	不显示下载进度
-z	打开自动续传功能, 此为默认值
-Z	关闭自动续传功能

范 例:

① 下载远程ftp服务器的file文件:

```
# ncftpget ftp://ftp.flag.com.tw/download/file
file:                1.20 kB        1.86 MB/s
  ↑                ↑                ↑
  下载的文件名称   文件大小        传输速率
```

② 下载远程ftp服务器的file文件到/home/sphenoid目录中:

```
# ncftpget ftp.flag.com.tw /home/sphenoid /download/file
```

③ 利用alkane账号及sec13579密码下载远程ftp服务器的alkane_file文件:

```
# ncftpget -u alkane -p sec13579 ftp://ftp.flag.com.tw/alkane_file
```

④ 下载远程ftp服务器/download/dir1目录下的所有子目录及文件:

```
# ncftpget -R ftp://ftp.flag.com.tw/download/dir1
```

⑤ 以ASCII传输模式下载远程ftp服务器的file文件:

```
# ncftpget -a ftp://ftp.flag.com.tw/download/file
```

⑥ 下载远程ftp服务器的file文件, 若连接不成功, 则最多尝试连接10次:

```
# ncftpget -r 10 ftp://ftp.flag.com.tw/download/file
```

⑦ 在后台下载远程ftp服务器的file文件:

```
# ncftpget -b ftp://ftp.flag.com.tw/download/file  
+ Spooled; writing locally as ./file.
```

⑧ 下载远程ftp服务器的file文件, 当下载完成后, 则删除该文件:

```
# ncftpget -DD ftp://ftp.flag.com.tw/download/file
```

⑨ 下载远程ftp服务器的file文件, 并设定超时秒数为60秒:

```
# ncftpget -t 60 ftp://ftp.flag.com.tw/download/file
```

⑩ 下载远程ftp服务器的file文件, 传输时不显示下载进度:

```
# ncftpget -V ftp://ftp.flag.com.tw/download/file
```



① 下载远程ftp服务器的file文件，若本地端已有相同的文件名称，则将下载的文件附加在本地端文件之后：

```
# ncftpget -A ftp://ftp.flag.com.tw/download/file
```

② 下载远程ftp服务器的file文件，并将传输过程记录在download.log文件中：

```
# ncftpget -d download.log ftp://ftp.flag.com.tw/download/file
```

ncftpput

常用指数：

功能说明：上传文件。

语 法：ncftpput [-aAbcFmRvVyzZ] [-B <缓冲区大小>] [-d <记录文件>] [-e <记录文件>] [-f <设定文件>] [-p <密码>] [-P <通信端口>] [-r <连接次数>] [-S <临时字尾字符串>] [-t <超时秒数>] [-T <临时字首字符串>] [-u <账号名称>] [-U <权限设定>] [-DD] [主机名称] [远程目录] [本地路径与文件名称...]

包名称：ncftp。

相关命令：ncftp, ncftpget。

补充说明：ncftpput是NcFTP的公用程序，可以直接使用参数、执行命令，上传文件。



参 数:

-a	使用ASCII传输类型，而非默认的二进制类型
-A	上传文件时，若该文件名称已经存在，则将上传数据附加到已存在文件的后面，而非将其覆盖
-b	在后台进行上传操作
-B <缓冲区大小>	设定传送缓冲区的大小
-c	从标准输入设备读取数据，然后再把所得到的数据上传至远程主机
-d <记录文件>	指定记录文件，记录ftp连接、传输过程，便于排错
-DD	成功上传文件之后，就把它从本地主机上移除
-e <记录文件>	指定记录文件，记录ftp连接、传输过程的错误信息
-f <设定文件>	读取指定的设定文件内的主机名称、账号名称和密码，进行登录操作
-F	使用PASV (Passive) 模式连接数据，只有在搭配安全系统（如防火墙）而遭遇问题时，才附加这项参数
-m	上传整个目录前，先尝试建立目的目录
-p <密码>	设定登录ftp站点的密码，不使用电子邮件信箱作为密码
-P <通信端口>	指定通信端口，不使用ftp服务默认的第21通信端口
-r <连接次数>	设定重复尝试连接的次数。当网络断线时，ncftpget可自动重新连接，继续上传文件
-R	递归处理，将指定目录下的所有文件及子目录一起处理
-S <临时字尾字符串>	上传临时存放的文件时，在该文件名称之字尾加上指定的字符串
-t <超时秒数>	设定等待服务器响应的时间，单位以秒计算，超过此时间则为超时
-T <临时字首字符串>	上传临时存放的文件时，在该文件名称的字首加上指定的字符串
-u <账号名称>	设定登录ftp站点的账号，不使用匿名登录
-U <权限设定>	设定上传的文件的权限。采用数字代号与umask的方式，例如权限设定为117，则上传后该文件在主机内的权限数字代号为660

-v	显示上传进度, 此为默认值
-V	不显示上传进度
-y	尝试使用远程主机的时间戳记
-z	打开自动续传功能
-Z	关闭自动续传功能, 此为默认值

范 例:

① 将file文件上传到远程ftp服务器:

```
# ncftpput ftp.flag.com.tw upload/ file
```

↑ ↑
 远程要存放的目录 本地端要上传的文件

```
file:                                    1.20 kB  891.93 kB/s
  ↑                                    ↑                    ↑
  上传的文件名称                      文件大小            传输速率
```

② 使用指定的账号及密码, 将file文件上传远程的ftp服务器:

```
# ncftpput -u alkane -p sec13579 ftp.flag.com.tw upload/ file
```

③ 将dir整个目录上传到远程ftp服务器:

```
# ncftpput -R ftp.flag.com.tw upload/ dir
```

④ 将file文件上传到远程ftp服务器, 上传完成后, 删除本地端的此文件:

```
# ncftpput -DD ftp.flag.com.tw upload/ file
```

⑤ 以ASCII模式将file文件上传到远程ftp服务器：

```
# ncftpput -a ftp.flag.com.tw upload/ file
```

⑥ 以后台的方式将file文件上传到远程ftp服务器：

```
# ncftpput -b ftp.flag.com.tw upload/ file
```

⑦ 将file文件上传到远程ftp服务器，并启动续传的功能：

```
# ncftpput -z ftp.flag.com.tw upload/ file
```

⑧ 将file文件上传到远程ftp服务器，并设定超时秒数为60秒：

```
# ncftpput -t 60 ftp.flag.com.tw upload/ file
```

⑨ 将file文件上传到远程ftp服务器，并设定尝试连接的次数为10次：

```
# ncftpput -r 10 ftp.flag.com.tw upload/ file
```

⑩ 将file文件上传到远程ftp服务器，若服务器上已有相同名称的文件，则将上传的文件附加在原来文件的后面：

```
# ncftpput -A ftp.flag.com.tw upload/ file
```

⑪ 将file文件上传到远程ftp服务器，上传时不显示传输进度：

```
# ncftpput -v ftp.flag.com.tw upload/ file
```

⑫ 将file文件上传到远程ftp服务器，并指定将传输过程记录到upload.log文件中：

```
# ncftpput -d upload.log ftp.flag.com.tw upload/ file
```

netconfig

常用指数:



功能说明: 设置网络环境。

语 法: netconfig

包 名 称: netconfig (Red Hat Linux/Fedora Core)。

相关命令: ifconfig, netstat。

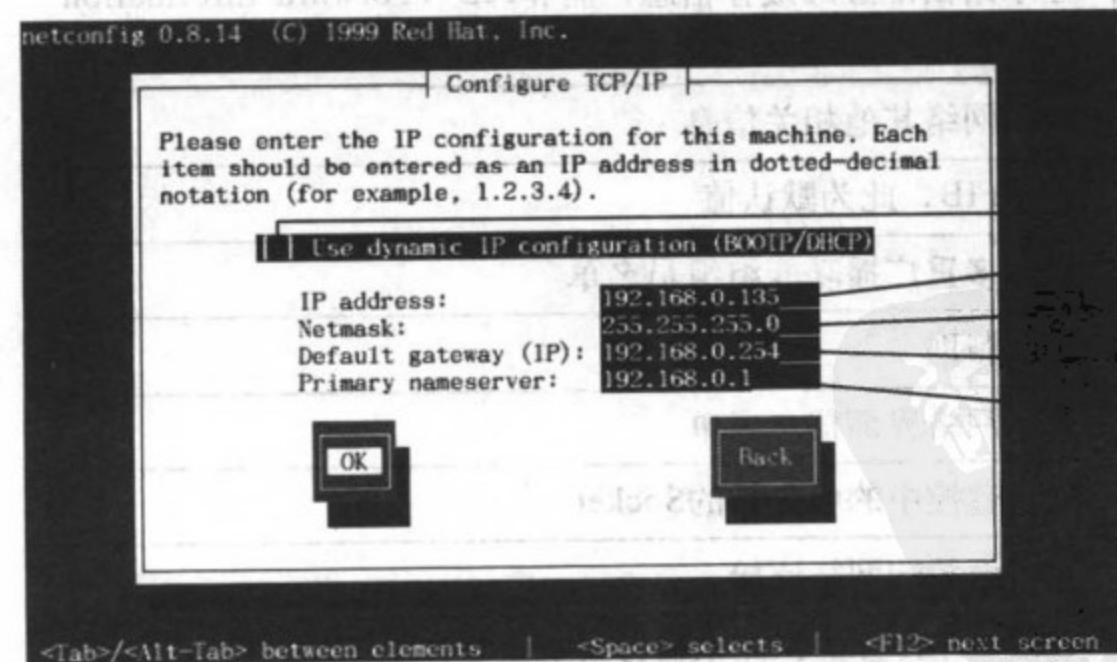
补充说明: 这是在某些Linux发行版如Red Hat Linux/Fedora Core、Slackware Linux内附的程序 (Mandrake Linux并没有此程序), 它具有互动式的问答界面, 让用户轻易完成网络环境的设定。通过netconfig程序可设定本地主机IP地址、网络掩码、网关以及域名服务器, 并决定是否要用动态分配的方式来设定IP地址。

参 数: 无。

范 例:

使用netconfig程序设置网络环境, 会出现以下操作画面, 可以利用 **Tab**、**Alt + Tab** 及空格键来设定:

netconfig



选取此处, 则采用动态分配的方式来决定IP地址

输入IP地址

输入网络掩码地址

输入网关地址

输入域名服务器地址

N

netstat (network status)

常用指数:



功能说明: 显示网络状态。

语 法: netstat [-acCeFghilMnNoprstuvVwx] [-A <网络类型>] [--ip]

包名称: net-tools。

相关命令: ifconfig, netconfig, ping。

补充说明: 利用netstat命令可让用户得知整个Linux系统的网络情况。假设没有指定任何参数给netstat命令, 则效果和指定参数“-F”相同。

参 数:

-a 或 --all	显示所有连接中的Socket
-A <网络类型>或 --<网络类型>	列出该网络类型连接中的相关地址。网络类型默认为“inet”, 它是DARPA (Defense Advanced Research Projects Agency) 网络, 即现在的 Internet网络。可以另行指定“ax25”、“netrom”、“ipx”、“ddp”与“unix”, 分别表示AMPR AX.25 (Amateur Radio AX.25 Level 2)、AMPR NET/ROM、IPX (Internetwork Packet Exchange)、AppleTalk DDP (Datagram Delivery Protocol) 及UNIX传输协议
-c或--continuous	持续列出网络状态
-C或--cache	显示路由配置的缓存信息, 而非FIB (Forward Information Base)
-e或--extend	显示网络其他相关信息
-F或--fib	显示FIB, 此为默认值
-g或--groups	显示多重广播功能组组员名单
-h或--help	显示帮助
-i或--interfaces	显示网络界面信息表单
-l或--listening	显示监控中的服务器的Socket
-M或--masquerade	显示伪装的网络连接
-n或--numeric	直接使用IP地址, 而不通过域名服务器

N、--netlink或--symbolic	显示网络硬件外围设备的符号连接名称
-o或--timers	显示计时器
-p或--programs	显示正在使用Socket的程序识别码和程序名称
-r或--route	显示Routing Table
-s或--statistics	显示网络工作信息统计表
-t或--tcp	显示TCP传输协议的连接状况
-u或--udp	显示UDP传输协议的连接状况
-v或--verbose	显示命令执行过程
-V或--version	显示版本信息
-w或--raw	显示RAW传输协议的连接状况
-x或--unix	此参数的效果和指定“-A unix”参数相同
--ip或--inet	此参数的效果和指定“-A inet”参数相同

范 例：

① 观察TCP传输协议的连接状况：

```
# netstat -t
```

```
Active Internet connections (w/o servers)
```

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 rhlinux.flag.c:ftp-data 205.47.203.53:1172 TIME_WAIT
tcp      0      0 rhlinux.flag.com.t:1140 210.208.142.15:www ESTABLISHED
tcp      0      0 rhlinux.flag.com.t:1133 210.208.142.15:www ESTABLISHED
tcp      0      0 rhlinux.flag.com.tw:ftp 205.47.203.53:1165 ESTABLISHED
tcp      0      0 rhlinux.flag.com.t:1128 netnews.hinet.net:nntp ESTABLISHED
```

↑
通信协议

↑ ↑ ↑
|
└───┬───┘
存放在接收与输出
队列中的数据量

↑
本机的域名
和Port编号

↑
连接中的远程主机，
分别以ftp、nntp、
http等方式连接

↑
连接状态

② 查看当前连接中的所有Socket:

```
# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0    285 localhost.localdom:1194 WWW.ym.edu.tw:www      STABLISHED
tcp      0     0 localhost.localdom:1191 vwww-mv1.netscape.c:ww TIME_WAIT
tcp      0     0 localhost.localdom:1190 210.200.129.142:www    TIME_WAIT
tcp      0     0 *:6000                **                      LISTEN
...
```

③ 显示网络连接状态，列出时显示IP地址而不使用域名:

```
# netstat -n
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0     0 203.74.205.36:113     203.74.205.5:3570     CLOSE
tcp      0     0 203.74.205.36:1213   203.74.205.5:110     ESTABLISHED
```

↑
列出IP地址

④ 列出监控中的服务器Socket:

```
# netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0     0 *:6000                **                      LISTEN
tcp      0     0 *:smtp                **                      LISTEN
...
```

⑤ 观察网卡传送、接收包状况:

```
# netstat -i
Kernel Interface table
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0 1500 0 52163 0 0 0 5302 0 0 0 BRU
lo 3924 0 503 0 0 0 503 0 0 0 LRU
```

↑
网络界面
(Interface)

网卡正确接收包的数量，以及
发生错误、流失、碰撞的总数

网卡正确传送包的数量，以及
发生错误、流失、碰撞的总数

⑥ 查看Routing Table的状态:

netstat -r

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS	Window	irttI	face
203.74.205.36	*	255.255.255.255	UH	0	0	0	eth0
203.74.205.0	*	255.255.255.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	203.74.205.1	0.0.0.0	UG	0	0	0	eth0

↑
网关↑
掩码U表示启动状态, H表示
Host, G表示Gateway

↑ Windows Size

↑ Initial Round

↑ Trip Time

以Byte为单位, 设定包经过路由器的大小限制

⑦ 查看Routing Table的状态, 并显示其他相关的信息:

netstat -er

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
203.74.205.36	*	255.255.255.255	UH	0	0	0	eth0
203.74.205.0	*	255.255.255.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	203.74.205.1	0.0.0.0	UG	0	0	0	eth0

⑧ 列出正在使用Socket的程序名称及识别码:

netstat -p

Active Internet connections (w/o servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
-------	--------	--------	---------------	-----------------	-------	------------------


```

tcp 1 0 localhost.localdom:1284 webmail.seed.net.tw:www
CLOSE_WAIT
979/netscape-commun
  ↑
程序识别码  程序名称
...

```

⑨ 显示网络连接状况的统计表：

netstat -s

Ip:

6930 total packets received	←全部收到的包数
0 forwarded	←被转送的包数
0 incoming packets discarded	←被丢弃的包数
2929 incoming packets delivered	←收到后再传送出去的包数
4214 requests sent out	←送出request的次数

...

⑩ 持续不断地列出网络连接状态：

netstat -c

⑪ 显示网络连接状态，并加上连接的计时器：


netstat -o

⑫ 显示UDP传输协议的连接状况：

netstat -u



newgrp

常用指数: 

功能说明: 登录另一个组。

语 法: newgrp [组名称]

包 名 称: util-linux (Red Hat Linux/Fedora Core), shadow-utils (Mandrake Linux)。

相关命令: login。

补充说明: newgrp命令类似login命令，但它是以相同的账号、另一个组名称，再次登录系统。欲使用newgrp命令切换组，您必须也是该组的用户，否则将无法登录指定的组。单一用户要同时隶属多个组，利用交替 (Alternate) 用户的设定。若不指定组名称，则newgrp命令会登录该用户名称的默认组。

参 数: 无。

范 例:

用户helen的默认组是staff，同时也属于组manager的用户 (交替用户) 在登录系统之后，欲切换至manager组：

```
$ newgrp manager
```



如不指定组名称，则newgrp会再用staff组登录系统

nice

常用指数:   

功能说明: 设置优先级。

语 法: nice [-n <优先等级>] [--help] [--version] [执行命令]

包名称: coreutils。

相关命令: renice。

补充说明: nice命令可以改变程序执行的优先级。通常执行某个命令时，若无特别搭配nice命令，则其优先级为0。如果搭配nice命令却没有设定优先级，则默认为10，即不增反降。

参 数:

-n <优先等级>、-<优先等级>或--adjustment=<优先等级>	设定欲执行的命令的优先权等级。等级的范围从-20~19，其中-20最高，19最低，只有系统管理者可以设定负数的等级
--help	显示帮助
--version	显示版本信息

范 例:

用不同的方式执行同一个程序，看看其优先顺序有何不同：

- # joe & ← 不搭配nice命令执行
- # nice joe & ← 搭配nice命令但不设定优先级
- # nice -19 joe & ← 设定程序的优先级为19
- # nice --20 joe & ← 设定程序的优先级为-20
- # ps -l ← 观察程序执行的情况

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
100	S	0	575	1	0	60	0	-	534	wait4	tty4	00:00:00	login
100	S	0	722	575	0	67	0	-	435	wait4	tty4	00:00:00	bash
000	T	0	788	722	0	63	0	-	482	do_sig	tty4	00:00:00	joe
000	T	0	789	722	0	72	10	-	482	do_sig	tty4	00:00:00	joe
000	T	0	790	722	0	79	19	-	482	do_sig	tty4	00:00:00	joe
100	T	0	791	722	0	50	-20	-	482	do_sig	tty4	00:00:00	joe
100	R0	792	722	0	73	0	-	636	-	tty4		00:00:00	ps

↑
此栏列出每个程序的优先权高低

nntpget

常用指数:



功能说明: 下载新闻组文章。

语法: nntpget [-ov] [-d <组类型清单>] [-f <文件或目录>] [-n <组名称清单>] [-t <时间日期>] [-u <文件或目录>] [新闻组服务器]

包名称: inn。

相关命令: getlist, slrn。

补充说明: 执行nntpget命令将从指定的新闻组服务器下载文章。nntpget命令会从标准输入设备读取信息识别码，而下载的新闻文章则呈现至标准输出设备。

参数:


-d <组类型清单>	指定新闻组类型为distribution的清单。本参数配合“-f”或“-t”参数使用，默认不使用distribution清单
-f <文件或目录>	下载其异动时间较指定文件或目录的异动时间，更接近现在的新闻文章。通常nntpget命令是以文章的信息识别码来分辨下载哪些文章
-n <组名称清单>	指定新闻组名称的清单。本参数配合“-f”或“-t”参数使用，默认值为“*”号
-o	下载本地主机所没有的新闻文章。本参数搭配innd常驻服务使用
-t <时间日期>	指定时间与日期，较该日期时间更新的新闻文章都会被下载
-u <文件或目录>	这个参数的效果和指定“-f”参数类似，但它在成功下载文章之后，会去更改指定文件或目录的异动时间，便于下次再度作为参考
-v	当命令执行时，把每篇文章的信息识别码都送到标准输出。本参数配合参数“-o”使用

范 例:

从news.hinet.net服务器的tw.bbs.comp.linux新闻组中下载5月18日以后的文章:

```
# /usr/lib/news/bin/nntpget -n tw.bbs.comp.linux -t 0518  
news.hinet.net
```

nohup (no hangup)

常用指数: 

功能说明: 可在退出后继续执行程序。

语 法: nohup [执行命令] 或nohup [--help] [--version]

包 名 称: coreutils。

相关命令: logout。

补充说明: nohup命令能让用户在退出系统之后, 继续于后台中执行设定的命令。注意nohup命令并不会自动把欲执行的命令放到后台里执行, 因此必须配合shell的特殊字符“&”使用。

参 数:

--help	显示帮助
--version	显示版本信息

范 例：

用ncftpget搭配nohup命令，在退出系统之后继续下载文件：

```
# nohup ncftpget ftp.wowo.com . /valkyrie_profile/fray.mov
```

↑
提供ftp服务的站点名称

↑
文件下载后存储在当前的目录中

↑
欲下载的文件名称及其存放路径

ntsysv

常用指数：



功能说明：设置系统的各种服务。

语 法：ntsysv [--back] [--level <等级代号>]

包 名 称：ntsysv (Red Hat Linux/Fedora Core)。

相关命令：chkconfig, runlevel, telinit。

补充说明：此为Red Hat Linux/Fedora Core才有的命令。这是Red Hat公司遵循GPL规则所开发的程序，它具有交互式操作界面，可让用户轻易地打开、关闭操作系统在每个执行等级中，所要执行的系统服务。

Linux操作系统将执行等级分成以下7种：

执行等级	执行的状态
0	关机（请不要切换到此等级）
1	单人用户模式的文字界面
2	多人用户模式的文字界面，但不具有网络文件系统（NFS）功能
3	多人用户模式的文字界面，具有全部的网络功能
4	某些发行版的Linux使用此等级进入X Window

5	某些发行版的Linux则使用此等级进入X Window
6	重新开机

参 数:

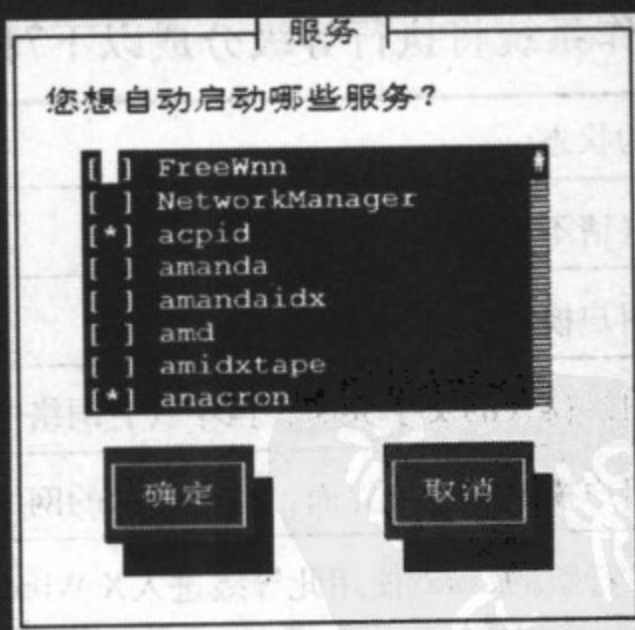
--back	在互动式界面里, 显示上一步按钮, 而非取消按钮
--level <等级代号>	在指定的执行等级中, 决定要打开或关闭哪些系统服务。等级代号的范围在0~6之间, 用户可以一次指定多个等级, 例如“235”表示同时设定等级2、3、5的系统服务

范 例:

- ① 使用ntsysv程序设置操作系统启动时欲执行的服务, 会出现以下画面, 可利用 **Tab**、**Alt + Tab** 及空格键选择要执行的服务。也可以按下 **F1** 键得知当前服务的简要说明:

```
# ntsysv
```

```
ntsysv 1.3.13.2 - (C) 2000-2001 Red Hat, Inc.
```



按 <F1> 可获得关于某项服务的详情。

② 设定在3和5执行等级所要启动的服务：

```
# ntsysv --level 35
```

③ 设定操作系统启动时要启动的服务，并指定在互动式界面中，显示上一步按钮，而非取消按钮：

```
# ntsysv --back
```



od (octal dump)

常用指数:

**功能说明:** 输出文件内容。**语 法:** od [-abcdfhilovx] [-A <内码单位>] [-j <字符数目>] [-N <字符数目>] [-s<字符串字符数>] [-t <输出格式>] [-w<每列字符数>] [--help] [--version] [文件...]**包 名 称:** coreutils。**补充说明:** od命令会读取所给予的文件的内容,并将其内容以八进制内码呈现出来。**参 数:**

-a	此参数的效果和指定“-t a”参数相同
-A <内码单位>或--address-radix=<内码单位>	选择要以什么单位计算内码。该基数可以是“d”、“o”、“x”和“n”,分别表示十进制、八进制、十六进制数来计算内码,以及不显示。默认以八进制为单位来计算内码
-b	此参数的效果和指定“-t oC”参数相同
-c	此参数的效果和指定“-t c”参数相同
-d	此参数的效果和指定“-t u2”参数相同
-f	此参数的效果和指定“-t fF”参数相同
-h	此参数的效果和指定“-t x2”参数相同
-i	此参数的效果和指定“-t d2”参数相同
-j <字符数目>或--skip-bytes=<字符数目>	略过设定的字符数目。使用这个参数,od命令会从数据开头算起,先略过设定的字符数目,再输出数据。假设在指定的字符数之前加上“0x”或“0X”字符串,则以十六进制计算;加上“0”字符,则用八进制计算。在指定的字符数目之后加上“b”、“k”和“m”字符,则分别表示乘上512、1 024及1 048 576
-l	此参数的效果和指定“-t d4”参数相同

-N <字符数目>或--read-bytes=<字符数目>	输出到设定的字符数目为止。使用这个参数，od命令会从数据开头算起，然后输出数据到所设定的字符数目为止。字符数目的设定方法和参数“-j”相同
-o	此参数的效果和指定“-t o2”参数相同
-s<字符串字符数>或--strings=<字符串字符数>	只显示符合指定字符数目的字符串，且该字符串必须以NULL控制字符作结尾。若使用此参数却没有设定字符数，将使用默认值3
-t<输出格式>或--format=<输出格式>	设定输出格式。输出格式包括“a”、“c”、“d”、“f”、“o”、“u”以及“x”，各自表示字符名称、ASCII字符、十进制（包含负数，signed decimal）、浮点、八进制、十进制（不包含负数，unsigned decimal）和十六进制格式。可以同时指定多个输出格式，od命令会在每一列依次显示不同的格式。除“a”、“c”之外，可在“d”、“f”、“o”、“u”、“x”后面以十进制数字设定数据大小，单位以Byte计算。也能用C语言的格式规定数据长度，“d”、“o”、“u”、“x”后面可搭配“C”、“S”、“I”、“L”，分别表示字符、短整数、整数和长整数，“f”配合“F”、“D”、“L”，表示浮点精度、双精度与长双精度
-v或--output-duplicates	输出时不省略重复的数据。默认od命令会检查输出的数据，如果发现某连续两行或两行以上的内容完全相同，则只会显示一行，并在该行的后面加上“*”号，标示重复的内容已被省略。加上此参数之后，od命令将不会省略内容重复的数据
-w<每行字符数>或--width=<每行字符数>	设定每列的最大字符数。默认值为每行输出16个字符的内码，可用这项参数去改变它。若使用此参数却没有设定字符数目，将使用设定值32
-x	此参数的效果和指定“-t x2”参数相同
--help	显示帮助
--version	显示版本信息

范 例：

① 以八进制为计算单位显示file文件的内容：

```
# od file
```

```
0000000 020040 020040 060515 066151 020040 041440 066557 060555
0000020 062156 005163 020164 066474 071545 060563 062547 066040
```

↑
文件的偏移量

└───┬───┘
文件的内容

...

② 以十六进制为计算单位显示file文件的内容：

```
# od -A x file
000000 020040 020040 060515 066151 020040 041440 066557 060555
000010 062156 005163 020164 066474 071545 060563 062547 066040
```

与上例比较，文件内容不变，但偏移量则有所不同

...

③ 以十进制为计算单位显示file文件的内容：

```
# od -A d file
0000000 020040 020040 060515 066151 020040 041440 066557 060555
0000016 062156 005163 020164 066474 071545 060563 062547 066040
```

...

④ 显示file文件的内容，但不列出偏移量的值：

```
# od -A n file
020040 020040 060515 066151 020040 041440 066557 060555
062156 005163 020164 066474 071545 060563 062547 066040
```



并没有显示偏移量

...

⑤ 以每列8个字符来显示file文件的内容：

```
# od -w8 file
0000000 020040 020040 060515 066151 ← 每行显示的字符只
0000010 020040 041440 066557 060555 有前面例子的一半
```

...



⑥ 以字符名称格式显示file文件的内容:

```
# od -t a file
0000000 sp sp sp sp M a i l sp sp sp C o m m a
          ↑           ↑
        空白字符    "M" 字符
0000020 n d s nl t sp < m e s s a g e sp l
          ↑
        换行字符
...
```

⑦ 以ASCII格式显示file文件的内容:

```
# od -t c file
0000000          M a i l          C o m m a
0000020 n d s \n t < m e s s a g e l
          ↑
        换行字符
...
```

⑧ 同时以十进制、十六进制格式显示file文件的内容:

```
# od -t ux file
          ↑
        同时指定2种格式
0000000 538976288 1818845517 1126178848 1634561391 ← 十进制格式
20202020 6c69614d 43202020 616d6d6f ← 十六进制格式
0000020 175334510 1832657012 1634956133 1814062439
0a73646e 6d3c2074 61737365 6c206567
...
```

⑨ 从第40个字符（以八进制计算）开始显示file文件的内容：

```
# od -j 040 file
0000040 071551 037164 020040 020040 020040 020040 020040 020040
0000060 020040 020040 074564 062560 066440 071545 060563 062547
...
```

⑩ 以十进制为计算单位显示file文件的内容，显示时略过前面的1024字节：

```
# od -A d -j 2b file
          ↑
          略过2x512 bytes
0001024 067440 020162 071565 071145 067040 066541 071545 071412
0001040 070145 071141 072141 062145 061012 020171 070163 061541
...
```

⑪ 只显示file文件前面10个字符的内容（以八进制计算）：

```
# od -N 10 file
0000000 020040 020040 060515 066151 020040
0000012
```

⑫ file1文件中有许多重复的字符，先用默认的方式显示文件内容，再以输出时不省略重复数据的方式显示文件内容：

```
# od file1 ← 以默认的方式显示
0000000 062564 072163 053440 064562 072164 071145 005156 030461
0000020 030461 030461 030461 030461 030461 030461 030461 030461
*
          ← 重复的列默认会以“*”显示
0000100 005061 062146 062146 062146 063541 072161 072162 000012
0000117
```

od -v file1 ←输出时不省略重复字符

```
0000000 062564 072163 053440 064562 072164 071145 005156 030461
0000020 030461 030461 030461 030461 030461 030461 030461 030461
0000040 030461 030461 030461 030461 030461 030461 030461 030461
0000060 030461 030461 030461 030461 030461 030461 030461 030461
0000100 005061 062146 062146 062146 063541 072161 072162 000012
0000117
```

显示出全部的字符

open

常用指数:



功能说明: 打开虚拟终端执行命令。

语法: open [-lsuvw] [-c <终端编号>] [-- 执行命令]

包名称: open。

相关命令: login。

补充说明: open命令会自动寻找第1个可使用的虚拟终端，然后在该终端执行指定的命令。凡是位于“--”号之后的字符串皆视为欲执行的命令，因此用户可以依照一般方式执行该命令，包括参数的设定。

参数:

-c <终端编号>	指定终端的编号。默认open命令会自动寻找第1个可使用的终端，用户可利用此参数直接设定第1个可使用的终端
-l	将所执行的命令当成登录的shell
-s	切换到执行命令的终端。当开始执行所给予的命令时，就自动将画面切换到该终端
-u	侦测当前终端的拥有者，然后以该拥有者的账号登录新的终端。本参数不会执行所指定的命令，因此也不该搭配“-c”、“-l”参数使用

-v	显示命令执行过程
-w	等待命令执行完毕。使用这项参数后，在指定命令执行完毕之前，原先的终端将无法继续执行其他命令。若搭配参数“-s”使用，则在命令执行完毕时，会自动切回原来的终端，即执行open命令的终端

范 例：

① 寻找第1个可使用的虚拟终端执行ftp命令，执行时显示执行过程：

```
# open -v -- ftp
```

open: using VT /dev/tty8 ← 使用第8号虚拟终端

② 使用第10号虚拟终端执行ftp命令：

```
# open -c 10 -- ftp
```

③ 寻找第1个可使用的虚拟终端执行ftp命令，执行时同时切换到该终端：

```
# open -s -- ftp
```

④ 寻找第1个可使用的虚拟终端执行ftp命令，执行时同时切换到该终端，执行完后再切回原来的终端：

```
# open -sw -- ftp
```

⑤ 寻找第1个可使用的虚拟终端执行bash shell：

```
# open -l bash
```

⑥ 以当前的账号登录到第1个可用的虚拟终端，同时切换到该终端：

```
# open -su
```



passwd (password)

常用指数:



功能说明: 设置密码。

语法: passwd [-dklS] [-u <-f>] [用户名称]

包名称: passwd。

相关命令: useradd, usermod。

补充说明: passwd命令可让用户更改自己的密码，而系统管理员则能用来管理系统用户的密码。只有管理员可以指定用户名称，一般用户只能变更自己的密码。

参数:

-d	删除密码。本参数仅有系统管理员方能使用
-f	强制执行。当要解开上锁的账号时，若该账号原本没有密码，则passwd命令会拒绝删除字首符号“!”。加上本参数后，passwd会随机产生一组密码，强行解开锁住的账号。此参数配合“-u”参数使用
-k	设定只有在密码逾期失效后，方能更新
-l	锁住账号。使用这项参数之后，passwd命令会在加密过的密码字符串加上符号“!”字首，让该账号的密码失效，无法登录系统。本参数仅有系统管理员方能使用
-S	列出密码的相关信息。本参数仅有系统管理员方能使用
-u	解开已上锁的账号。使用这项参数之后，passwd命令会把加上的“!”号字首删除，恢复该账号的密码，让指定的用户得以登录系统。本参数仅有系统管理员方能使用

资源知识

范 例:

① 变更自己的密码:

\$ passwd

Changing password for user jacket.

Changing password for jacket

(current) UNIX password: ← 需先输入原本的密码

New UNIX password: ← 输入新的密码, 为了安全起见, 所输入的密码并不会在屏幕上显示

Retype new UNIX password: ← 再一次输入密码, 以便确认二次所输入的密码都是一样的

passwd: all authentication tokens updated successfully.

② 变更jacket用户的密码:

passwd jacket

③ 删除 jacket 用户的密码:

passwd -d jacket

Removing password for user jacket. ← 已经删除密码

passwd: Success

④ 列出jacket用户密码的相关信息:

passwd -s jacket

Password set, MD5 encryption.

使用MD5 (Message Digest) 编码方式,
可使用高达256字符的密码



⑤ 锁住jacket用户的账号：

```
# passwd -l jacket
```

Locking password for user jacket. ← 已经锁住密码

```
passwd: Success
```

⑥ 解开锁住的jacket用户密码：

```
# passwd -u jacket
```

Unlocking password for user jacket. ← 已经解锁密码

```
passwd: Success
```

paste

常用指数：



功能说明：合并文件的行。

语法：paste [-s] [-d <间隔字符>] [--help] [--version] [文件...]

包名称：coreutils。

相关命令：colrm。

补充说明：paste命令会把每个文件以行对行的方式，一行行地加以合并。隶属于不同文件的行的内容，彼此间以Tab字符间隔。每个合并之后的行，用新增行控制字符作结尾，然后输出至标准输出设备。若不指定任何文件名称，或是所给予的文件名为“-”，则paste命令会从标准输入设备读取数据。

参数：

<p>-d <间隔字符>或 --delimiters=<间隔字符></p>	<p>用指定的间隔字符取代Tab字符。默认paste会用Tab字符区隔不同文件的行的内容，可以借由此参数另行指定</p>
---	--

-s或--serial	序列进行而非平行处理。paste命令默认会平行处理每一个文件，即从第1个文件的第1行开始，依次处理第2个文件的第1行、第3个文件的第1行等，待所有文件的第1行都处理完毕之后，再由第1个文件的第2行开始。使用本参数后，paste命令会依次处理每个文件的每一行，当一个文件从第1行至最后一行全部处理完毕之后，才会进入下个文件的第1行，如此依次进行
--help	显示帮助
--version	显示版本信息

范 例：

① 以默认的方式合并file1及file2等两个文件的行：

```
# cat file1          ← 查看file1文件的内容
line1_in_file1
line2_in_file1
line4_in_file1
# cat file2          ← 查看file2文件的内容
line1_in_file2
line3_in_file2
# paste file1 file2 ← 合并这两个文件
line1_in_file1 line1_in_file2
line2_in_file1 line3_in_file2
line4_in_file1
```

┌─── 合并的结果

② 同上例的file1及file2文件，合并这两个文件，并指定以“%”符号作为列与列之间的区隔字符：

```
# paste -d % file1 file2
line1_in_file1%line1_in_file2
line2_in_file1%line3_in_file2
line4_in_file1%
```


③ 同第1例的file1及file2文件，但以序列的方式合并这两个文件：

```
# paste -s file1 file2
line1_in_file1      line2_in_file1      line4_in_file1
line1_in_file2      line3_in_file2
```

④ 合并file1、file2及file3文件，并将结果存成merge_file：

```
# paste file1 file2 file3 > merge_file
```

patch

常用指数：

功能说明：修补文件。

语 法： patch [-bceEflnNRstTuvZ] [-B <备份字首字符串>] [-d <工作目录>] [-D <标示符号>] [-F <鉴别列数>] [-g <控制数值>] [-i <修补文件>] [-o <输出文件>] [-p<剥离层级>] [-r <拒绝文件>] [-V <备份方式>] [-Y <备份字首字符串>] [-z <备份字尾字符串>] [--backup-if-mismatch] [--binary] [--dry-run] [--help] [--nobackup-if-mismatch] [--verbose] [原始文件 <修补文件>] 或 patch [-p<剥离层级>] < [修补文件]

包名称： patch。

相关命令： make, mkinitrd, rpm, uname。

补充说明： patch命令让用户利用设定修补文件的方式，修改、更新原始文件。假如一次仅修改一个文件，可直接在命令行中下达命令依次执行。如果配合修补文件的方式，则能一次修补大批文件，这也是Linux系统内核的升级方法之一。

参 数:

-b或--backup	备份每一个原始文件
-B <备份字首字符串>或 --prefix=<备份字首字符串>	设定文件备份时, 附加在文件名称前面的字首字符串, 该字符串可以是路径名称。例如修补操作使用的路径与文件名为src/linux/fs/super.c, 加上“/backup/”字符串后, 文件super.c会备份于/backup/src/linux/fs目录里
-c或--context	把修补数据解释成关联性的差异
-d<工作目录>或--directory= <工作目录>	设定工作目录。在修补操作一开始执行时, 就会先切换到指定的目录下
-D<标示符号>或--ifdef= <标示符号>	用指定的符号把改变的地方标示出来
-e或--ed	把修补数据解释成ed命令可用的叙述文件
-E或--remove-empty-files	若修补过后输出的文件的内容是一片空白, 则删除该文件
-f或--force	此参数的效果和指定“-t”参数类似, 但会假设修补数据的版本为新版本
-F <鉴别行数> 或--fuzz <鉴别行数>	设定鉴别行数的最大值。本参数仅在diff命令处理关联注解, 让修补操作略过注解行的内容时使用。注意鉴别行数设定过大, 将可能造成文件修补失败, 默认值为2, 一般不应超过3
-g <控制数值> 或--get=<控制数值>	设定以RCS (Revision Control System) 或SCCS (SourceCode Control System) 控制修补操作。如果该数值为正数, 修补操作将以RCS为准, 读取修补文件的数据; 若为0, 修补操作将忽略RCS或SCCS; 为负数时, 修补操作会询问用户应到何处取得数据
-i <修补文件> 或--input = <修补文件>	读取指定的修补文件。若所给予的文件名为“-”, 则patch命令会从标准输入设备读取数据
-l或--ignore-whitespace	忽略修补数据与输入数据的Tab、空格字符
-n或--normal	把修补数据解释成一般性的差异
-N或--forward	忽略修补的数据比原始文件的版本更旧, 或该版本的修补数据已使用过
-o <输出文件>或--output= <输出文件>	设定输出文件的名称, 修补过的文件会以该名称存放

-p<剥离层级>或--strip=<剥离层级>	设定欲剥离几层路径名称。通常在修补文件内指定的文件名称，其前面都有加上绝对路径，可利用这个参数删除路径名称。举例来说，当路径与文件名称为/usr/src/linux/fs/super.c时，每设定1个层级就能删除1个“/”号及其之前的路径名称。当剥离层级设为1时，修补文件所用的路径及文件名称就是usr/src/linux/fs/super.c，剥离层级设成3，则路径、文件名称为 linux/fs/super.c，依此类推。不设定此项参数时，仅会使用文件名称 super.c 而不会指定路径名称，所以其路径默认为现行的目录下，或是用参数“-d”指定的目录
-r <拒绝文件>或--reject-file=<拒绝文件>	设定存储拒绝修补相关信息的文件名称，默认的文件名称为rej
-R或--reverse	假设修补数据是由新旧文件交换位置而产生
-s或--quiet或--silent	不显示命令执行过程，除非发生错误
-t或--batch	自动略过错误，不询问任何问题。假如版本错误便假设修补数据的版本为旧版本
-T或--set-time	此参数的效果和指定“-Z”参数类似，但以本地时间为主
-u或--unified	把修补数据解释成一致化的差异
-v或--version	显示版本信息
-V<备份方式>或--version-control=<备份方式>	用“-b”参数备份目的文件后，备份文件的字尾会被加上一个备份字符串，这个字符串不仅可用“-z”参数变更，当使用“-V”参数指定不同备份方式时，也会产生不同字尾的备份字符串。备份方式指定为“numbered”（或“t”），将使用备份编号，在字尾会加上“~1~”字符串，且数字编号依次递增。备份方式指定为“simple”（或“never”），将使用简单备份，默认的备份字尾字符串是“~”，也能通过参数“-z”、“-B”和“-Y”，分别设定字尾、字首及文件名字首的字符串。备份方式指定为“existing”（或“nil”），将使用现行方式：程序会先检查是否存在着备份编号，若有则采用备份编号，若无则采用简单备份，此为默认值

-Y <备份字首字符串> 或-- basename-prefix=<备份字首 字符串>	设定文件备份时，附加在文件基本名称开头的字 首字符串
-z <备份字尾字符串>或-- suffix=<备份字尾字符串>	此参数的效果和指定“-B”参数类似，差别在于 修补操作使用的路径与文件名若为src/linux/ fs/super.c，加上“backup/”字符串后，文件 super.c会备份于/src/linux/fs/backup目录里
-Z或--set-utc	把修补过的文件异动、存取时间设为UTC
--backup-if-mismatch	在修补数据不完全吻合，且没有刻意指定要备 份文件时，才备份文件。除非设有环境变量 POSIXLY_CORRECT，否则此为默认值
--binary	以二进制模式读写数据，而不通过标准输出设备。 本参数在相容于POSIX标准的操作系统下无效
--dry-run	不执行命令，仅列出实际执行会进行的动作
--help	显示帮助
--nobackup-if-mismatch	在修补数据不完全吻合，且没有刻意指定要备 份文件时，不要备份文件。若设有环境变量 POSIXLY_CORRECT，则此为默认值
--verbose	详细显示命令的执行过程

范 例：

① 把Linux操作系统2.4.20的系统内核以修补方式更新成2.4.21：

```
# gzip -cd /tmp/patch-2.4.21.gz | patch -p0
```

系统核心修补文件的存放目录

② 以linux-2.4.21-openssl-0.9.6b-mppe.patch.gz修补文件修补系统核心:

```
# gzip -cd linux-2.4.21-openssl-0.9.6b-mppe.patch.gz |
patch -p0
```

③ 以patch_file修补文件修补original_file文件:

```
# patch original_file patch_file
```

④ 以patch_file修补文件修补original_file文件, 并指定/home/alkane为工作目录:

```
# patch -d /home/alkane original_file patch_file
```

⑤ 以patch_file修补文件修补original_file文件, 并备份原始文件:

```
# patch -b original_file patch_file
```

pcxtoppm

常用指数:



(pcx to portable pixmap)

功能说明: 转换图像文件。

语法: pcxtoppm [-stdpalette] [PCX 图像文件]

包名称: netpbm-progs (Red Hat Linux/Fedora Core), netpbm (Mandrake Linux)。

补充说明: 执行pcxtoppm命令, 读取指定的PCX格式图像文件(默认的扩展文件名为.pcx), 并把该文件转成Portable Pixmap(默认的扩展文件名为.ppm)格式后, 输出到标准输出设备。

参数:

stdpalette	当颜色数仅16色或更少时, 使用系统默认的色彩配置表
------------	----------------------------

P

范 例:

① 把图像文件mypicture1.pcx转成Portable Pixmap格式:

```
# pxtoppm mypicture1.pcx > mypicture1.ppm
```

② 将颜色数仅为16色的mypicture2.pcx转成Portable Pixmap格式:

```
# pxtoppm -stdpalette mypicture2.pcx > mypicture2.ppm
```

pfbtops

常用指数:



(postscript type 1 font binary to postscript type 1 font ascii)

功能说明: 转换字体文件。

语 法: pfbtops [-v] [PFB 字体文件]

包 名 称: groff。

补充说明: 执行pfbtops命令, 它将读取指定的Type 1二进制格式字体文件(默认的扩展文件名为.pfb), 并把该文件转成 Type 1 ASCII格式字体文件(默认的扩展文件名为.pfa)后, 输出到标准输出设备。若不指定任何文件名称, 则pfbtops命令会从标准输入设备读取数据。

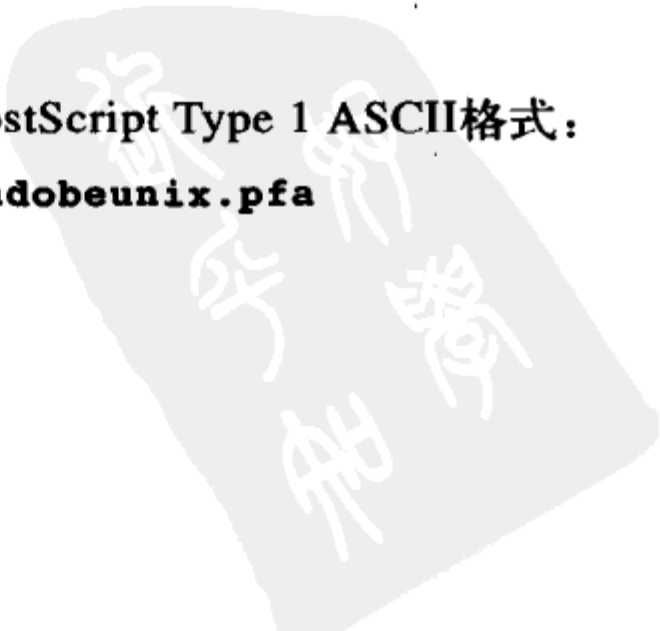
参 数:

-v	显示版本信息
----	--------

范 例:

把字体文件adobedos.pfb转成PostScript Type 1 ASCII格式:

```
# pfbtops adobedos.pfb > adobeunix.pfa
```



pico(pine composer)

常用指数: 

功能说明: 编辑文本文件。

语 法: pico [-bdefghjkmqtvwxz] [-n<间隔秒数>] [-o <工作目录>]
[-r<编辑页宽>] [-s <拼字检查器>] [+<行数编号>] [文件]

包 名 称: pine (Red Hat Linux)。

相关命令: pine, vim。

补充说明: pico是一个简单易用的文本编辑程序 (Red Hat Linux才有此程序), 它起源于处理电子邮件和新闻组的程序pine。如果觉得Linux默认的文本编辑程序vi实在太难用, 不妨试试具备帮助、热键及互动式操作界面的pico文本编辑程序, 轻松利用方向键和各种热键编辑文本文件。

参 数:

-b	打开置换 (Replace) 的功能。使用本参数后, 当执行查找命令时, 会增加置换的选项以供选择, 用户可利用它查找并置换字符串
-d	打开删除 (Delete) 的功能。pico默认按下 Delete 键的效果和 +Backspace 键相同, 加上本参数后, 按下 Delete 键将会清除光标所在位置的字符, 而非像 +Backspace 键一样, 清除光标左方字符
-e	使用完整的文件名称
-f	支持键盘上的 F1 、 F2 等功能键。此参数仅在和UW (University of Washington) 加强型NCSA (National Center for Supercomputing Applications) 的telnet连接时, 才会有效
-g	显示光标。在文件浏览模式里, 默认光标会一直停留在左下方, 附加这个参数将让光标跟着移动

-h	显示帮助
-j	打开切换 (Goto) 的功能。使用本参数后, 当在文件浏览模式时, 会增加切换的选项以供选择, 可利用它直接指定欲浏览的目录
-k	默认pico在使用剪切 (Cut) 命令时, 会把光标所在行的内容全部删除。使用这项参数之后, 执行剪切命令只会删除光标右方 (包含光标所在位置) 的所有字符
-m	打开鼠标支持的功能, 可用鼠标点选命令列表, 直接执行命令。此参数仅在X Window的虚拟终端窗口中有效
-n<间隔秒数>	设定多久检查一次新邮件。可使用这个参数去设定检查电子邮件的间隔时间, 单位以秒计算, 默认值为180秒, 最少不得低于30秒
-o <工作目录>	设定工作目录, 只有位于该目录下的文件或子目录得以被打开。用户将无法切换到该目录的上层目录
-q	忽略默认值, 优先使用termcap和terminfo的定义。termcap为一个设定文件, 位于/etc目录中, 定义各类终端的特性。terminfo为一个目录, 位于/usr/share目录下, 包含各种终端的相关信息
-r<编辑页宽>	设定编辑文件的页宽。pico命令的默认页宽为72列, 可用这项参数改变页宽
-s <拼字检查器>	另外指定拼字检查器
-t	启动工具模式。当通过别的程序执行pico时, 可加上这个参数让pico在存储文件和离开程序时, 不会询问用户
-v	启动阅读模式, 用户将只能查看, 但无法编辑文件内容
-w	关闭自动换行, 通过这个参数可以编辑内容很长的行
-x	关闭画面下方的命令列表
-z	让pico可被 Ctrl+C + Ctrl+Z 中断, 临时在后台操作中
+<行数编号>	执行pico命令进入编辑模式时, 从指定的行数开始编辑。例如起始行数设定为5, 光标一开始会位于欲编辑的文件的第5行。默认值为1, 即从第1行开始编辑

范 例:

① 编辑mytext文本文件，可利用稍后所介绍的操作命令编辑文本：

```
# pico mytext
```

载入文件的名称

```

UW PICO(tm) 4.2 (File: mytext)
#
# Based upon the NCSA server configuration files originally by Rob McCool.
#
# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs-2.0/> for detailed information about
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# The configuration directives are grouped into three basic sections:
# 1. Directives that control the operation of the Apache server process as a
# whole (the 'global environment').
# 2. Directives that define the parameters of the 'main' or 'default' server,
# which responds to requests that aren't handled by a virtual host.
# These directives also provide default values for the settings
# of all virtual hosts.
| Read 1068 lines |
^G Get Help ^O WriteOut ^R Read File ^Y Prev Pg ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where is ^V Next Pg ^U UnCut Text ^T To Spell

```

编辑区

可操作的组合键

② 编辑mytext文本文件，并将光标置于第10行：

```
# pico +10 mytext
```

③ 在X Window的虚拟终端里，执行pico命令编辑mytext文本文件，并启动鼠标功能，使得可用鼠标移动光标及选择所要的功能：

```
# pico -m mytext
```


④ 编辑mytext文本文件，并指定只能打开/tmp目录中的文件：

```
# pico -o /tmp mytext
```

⑤ 建立新的文本文件，并设定页宽为78列：

```
# pico -r 78
```

⑥ 建立新的文本文件，并取消当某行的长度过长时，自动换行的功能：

```
# pico -w
```

⑦ 编辑mytext文本文件，并打开执行查找命令时置换的功能：

```
# pico -b mytext
```

⑧ 编辑mytext文本文件，并启动按■键的功能：

```
# pico -d mytext
```

⑨ 编辑mytext文本文件，并启动工具模式，使得当存储文件及结束程序时，不询问用户：

```
# pico -t mytext
```

⑩ 编辑mytext文本文件，并关闭画面下方的操作命令表，使得有较大的文本显示区域：

```
# pico -x mytext
```

⑪ 以只读模式打开mytext文本文件：

```
# pico -v mytext
```

⑫ 编辑mytext文本文件，并设定每30秒检查一次是否有新的电子邮件：


```
# pico -n30 mytext
```

操作命令：

Ctrl + F	往前移一个字符
Ctrl + b	往后移一个字符
Ctrl + N	往下移一行
Ctrl + P	往上移一行
Ctrl + a	移至此行的起始处
Ctrl + E	移至此行的结尾处
Ctrl + V	往下移一页
Ctrl + Y	往上移一页
Ctrl + d	删除光标所在的字符
Ctrl + K	删除光标所在的行
Ctrl + i	在当前光标所在的位置，插入一个tab
Ctrl + R	在当前光标所在的位置，插入指定的文件内的文本
Ctrl + W	查找字符串
Ctrl + J	重整光标当前所在的段落
Ctrl + U	恢复重整的段落或还原删除的行
Ctrl + t	检查拼字的正确性
Ctrl + C	显示光标当前所在的位置

Ctrl + L	重新显示画面
Ctrl + G	显示帮助
Ctrl + O	存储文件
Ctrl + X	结束程序

pictoppm

常用指数: 

(picture to portable pixmap)

功能说明: 转换图像文件。

语 法: pictoppm [-fontdir <设定文件>] [-fullres] [-noheader] [-quickdraw] [-verbose] [PICT 图像文件]

包 名 称: netpbm-progs (Red Hat Linux), netpbm (Mandrake Linux)。

补充说明: 执行pictoppm命令, 读取指定的PICT格式图像文件(默认的扩展文件名为.pict), 并把该文件转成Portable Pixmap(默认的扩展文件名为.ppm)格式后, 输出到标准输出设备。

参 数:

-fontdir <设定文件>	指定字体的设定文件。PICT图像文件可使用BDF字体, 这些字体能用设定文件来指定编号、名称、样式等等。有关设定BDF字体的信息, 详见Macintosh电脑的相关文件
-fullres	强制以最高分辨率输出PICT图像文件的内容, 避免数据流失
-noheader	不忽略PICT图像文件开头的512字节数据
-quickdraw	使用快速绘制。本参数会关闭某些支持PostScript打印机的转译操作
-verbose	显示命令执行过程

范 例:

① 把图像文件mypicture1.pict转成Portable Pixmap格式:

```
# picttoppm mypicture1.pict > mypicture1.ppm
```

② 以最高分辨率的方式将mypicture2.pict图像文件转成Portable Pixmap格式:

```
# picttoppm -fullres mypicture2.pict > mypicture2.pict
```

pine

常用指数: 

(a program for Internet News and E-mail)

功能说明: 收发电子邮件、浏览新闻组。

语 法: pine [-hikorz] [-attach <附加文件>] [-attach_and_delete <附加文件>] [-attachlist <附加清单>] [-c <邮件编号>] [-conf] [-create_lu <地址簿> <排序法>] [-f <收件夹>] [-F <文件>] [-I <临时热键>] [-n <邮件编号>] [-nr] [-p <环境文件>] [-pinerc <输出文件>] [-P <设定文件>] [-sort <排序法> </reverse>] [-url <URL>] [-<功能选项>=<设定值>] [用户名称或电子邮件地址...]

包 名 称: pine (Red Hat Linux)。

相关命令: fetchmail, mail, mailq, metamail, mutt。

补充说明: pine是一个邮件处理程序 (Red Hat Linux才有此程序), 包括一般电子邮件和新闻组的文章。pine也支持MIME格式, 可让用户将MIME的物件存储成文件, 甚至指定外

挂程序直接打开物件。pine程序具有帮助、热键及互动式操作界面，轻松利用方向键和各种热键收发电子邮件、浏览新闻组。在执行pine命令时加上用户名称，会进入电子邮件编辑模式，而该用户的电子邮件地址则以默认的网域为准。如果要同时寄信给多人，也可指定多个用户或电子邮件地址，便可轻易地将一封电子邮件发给多人。

参 数：

-attach <附加文件>	把指定的文件附加在信件里，一并寄出
-attach_and_delete <附加文件>	此参数的效果和指定“-attach”参数类似，但在信件寄出后，会将指定的文件删除
-attachlist<附加清单>	设定一份文件清单，把该清单内的文件全部附加到信件中，附加寄出
-c <邮件编号>	直接打开指定编号的邮件。不同收件夹里的邮件各有其编号，收件夹默认为INBOX，可借由参数“-f”另行设定
-conf	产生pine.conf的范例文件，并输出到标准输出设备
-create_lu <地址簿> <排序法>	将地址簿（Address Book）以指定的排序法排序，产生.addressbook.lu索引文件。地址簿的默认文件名为.addressbook，和索引文件都位于每个用户的专属目录中。排序法共分成“dont-sort”、“nickname”、“fullname”、“nickname-with-lists-ast”和“fullname-with-lists-last”等5种
-f <收件夹>	打开指定的接收信件文件夹，而非默认的INBOX收件夹
-F <文件>	直接浏览指定文件的内容
-h	显示帮助
-i	直接进入收件夹
-l <临时热键>	设定进入pine就要执行的命令，该命令以热键的按键指定。例如临时热键设定为“？”，执行pine命令之后就会紧接着执行帮助（Help）的命令，其余依此类推。同时指定多个命令，两两之间用“，”号区隔
-k	支持键盘上的 F1、F2 等功能键，而非使用一般热键
-n <邮件编号>	设定现在的邮件编号

-nr	使用UWIN的特殊模式
-o	设定收件夹为只读
-p <环境文件>	另外指定pine的环境文件。环境文件提供每位pine的用户设定相关信息，默认的文件名称为.pinerc，存放在用户的专属目录下，用户可利用此参数改变它
-pinerc <输出文件>	读取.pinerc环境文件的内容并输出成文件存储起来，必须指定输出文件的名称
-P <设定文件>	另外指定pine的设定文件。设定文件提供所有pine的用户设定相关信息，默认的文件名称为pine.conf，存放在/etc目录下，可利用此参数改变它
-r	展示模式。使用本参数后，pine仅局限于展示各项功能，将无法发信给其他用户，只能把信寄给自己，也无法存储任何改变过的设定
-sort <排序法>	把收件夹内信件依指定的方式排序。排序的方法有</reverse>“arrival”、“subject”、“from”、“date”、“orderedsubj”与“reverse”，默认值为“arrival”。用户可在排序法的后面加上“/reverse”参数，使用逆向排序，排序法“arrival”加上“/reverse”就等于排序法“reverse”，但是不可以用“reverse”排序法再加上“/reverse”参数
-url <URL>	直接打开所给予的URL。本参数不可与“-f”、“-F”及“-attach”参数同时使用
-z	让pine可被 Alt + Z 中断，临时在后台操作中
-<功能选项>=<设定值>	暂时指定各项功能的设定值。例如设定功能选项为“user-domain”、设定值为“ganbale.net”，则pine就会用ganbale.net作为默认的网域名称，而忽略环境文件或设定文件内的设定

范 例：

- ① 执行pine命令，会出现以下画面，可利用画面下方所显示的组合键来收发邮件及浏览新闻组：

```
# pine
```



```

PINE 4.44  MAIN MENU                               Folder: INBOX 44 Messages

?  HELP - Get help using Pine
C  COMPOSE MESSAGE - Compose and send a message
I  MESSAGE INDEX - View messages in current folder
L  FOLDER LIST - Select a folder to view
A  ADDRESS BOOK - Update address book
S  SETUP - Configure Pine Options
Q  QUIT - Leave the Pine program

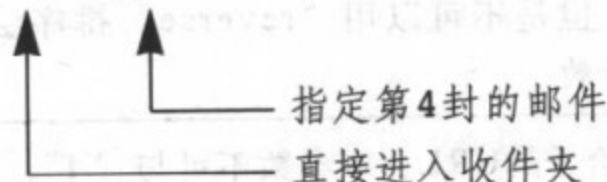
Copyright 1989-2002. PINE is a trademark of the University of Washington.

? Help OTHER CMDS [ListFldrs] P PrevCmd R RelNotes
0 K KBlock N NextCmd
    
```

当前可使用的组合键

② 执行pine命令，执行时直接进入收件夹，并将光标移到第4封邮件上：

```
# pine -i -n4
```



③ 执行pine命令，并立即进入地址簿菜单：

```
# pine -I a
```

进入pine后由程序自动执行 **a** 键所表示的功能

④ 执行pine命令，并指定接收/var/spool/mail/alkane收件夹的邮件：

```
# pine -f /var/spool/mail/alkane
```

⑤ 执行pine命令，并指定将信件夹内的邮件依邮件主题排列：

```
# pine -sort subject
```

⑥ 执行pine命令，并指定将信件夹内的邮件按照寄件者名称排序：

```
# pine -sort from
```

⑦ 执行pine命令，并指定将信件夹内的邮件按照收到信件的时间由新至旧排列：

```
# pine -sort arrival/reverse
```

反向排序

⑧ 编写新的邮件，并在此信件中附加myfile文件：

```
# pine -attach myfile
```

⑨ 编写新的邮件，并在此信件中附加myfile文件，寄出信件后则删除此文件：

```
# pine -attach_and_delete myfile
```

⑩ 执行pine命令，并指定读取mypine.conf设定文件：

```
# pine -P mypine.conf
```

⑪ 以展示模式执行pine命令：

```
# pine -r
```



ping

常用指数:



功能说明: 侦测主机。

语 法: ping [-dfnqrRv] [-c <完成次数>] [-i <间隔秒数>] [-I <网络接口>] [-l <前置载入>] [-p <范本样式>] [-s <包大小>] [-t <存活时间>] [主机名称或 IP 地址]

包 名 称: iputils。

相关命令: ifconfig, netstat, traceout。

补充说明: 执行ping命令，它会使用ICMP（Internet Control Message Protocol）传输协议，发出要求响应的信息，若远程主机的网络功能没有问题，就会响应该信息，因而得知该主机运作正常。

参 数:

-c <完成次数>	设定完成要求响应的次数。ping命令会反复发出信息，直到达成设定的次数为止。每发出1个要求且收到1个响应，才算完成1次
-d	使用Socket的SO_DEBUG功能
-f	极限侦测。不待收到响应就全速发出要求信息，每秒最高可达数百次之多，以便统计包的漏失、错误比率。每发出1个要求信息，就会显示1个“.”号。由于本参数将造成网络的严重负担，故仅有系统管理员方能使用
-i <间隔秒数>	指定收发信息的间隔时间，单位以秒计算，默认值为1秒。本参数不相容于“-f”参数
-I <网络接口>	使用指定的网络接口送出包。一般而言，第1张网卡的界面代号通常是eth0，可执行ifconfig命令查询网络接口代号
-l <前置载入>	设定在送出要求信息之前，先行送出的包。只有系统管理员才能使用这个参数

-n	只输出数值，ping将不会尝试去寻找主机地址的符号名称
-p <范本样式>	设定填满包的范本样式。最多可指定16 Bytes的范本样式，该样式必须用十六进制设定
-q	不显示命令执行过程，开头和结尾的相关信息除外
-r	忽略普通的Routing Table，直接将包送到远程主机上。如果该主机并非在本局域网内，则会传回错误信息
-R	记录路由过程。在要求信息的包里加上RECORD_ROUTE的功能，显示包经过的路由过程。注意IP文件头仅够容纳9个路由信息，而且许多主机会忽略或放弃该项功能
-s <包大小>	设定包的大小。默认值是56 Bytes，加上8 Bytes的ICMP文件头成为64 Bytes。假如设定大小为100 Bytes，则送出的包应该是108 Bytes
-t <存活数值>	设定存活数值TTL (Time-to-Live) 的大小，范围从1~255。每个IP包都会拥有这项数值，以便决定是否应放弃该包。当包经过路由器，TTL数值就会被扣减，直到数值为0时，路由器就会放弃这个包，且发出ICMP的信息通知送出该包的主机
-v	详细显示命令的执行过程，包括非响应信息的其他信息都会被显示出来

范 例：

① 侦测www.flag.com.tw主机的网络功能是否正常运行：

```
# ping www.flag.com.tw
```

```
PING www.flag.com.tw (210.62.132.10) 56(84) bytes of data.
64 bytes from c10.h210062132.is.net.tw (210.62.132.10):
icmp_seq=1 ...
64 bytes from c10.h210062132.is.net.tw (210.62.132.10):
icmp_seq=2 ...
64 bytes from c10.h210062132.is.net.tw (210.62.132.10):
icmp_seq=3 ...
```

← 按 [Ctrl] + [C] 键可中断显示

```
--- flag.com.tw ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time
2020ms
```

```
rtt min/avg/max/mdev = 93.315/95.450/97.916/1.892 ms
```


② 侦测210.62.132.10主机的网络功能是否正常运作，并指定送出的要求信息完成3次响应：

```
# ping -c 3 210.62.132.10
```

```
PING 210.62.132.10 (210.62.132.10) 56(84) bytes of data.
```

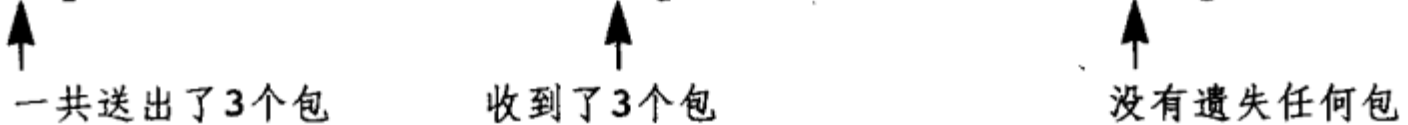
侦测的过程

```
64 bytes from 210.62.132.10:icmp_seq=0 ttl=57 time=1386.5 ms
64 bytes from 210.62.132.10:icmp_seq=1 ttl=57 time=776.5 ms
64 bytes from 210.62.132.10:icmp_seq=2 ttl=57 time=1764.1 ms
```

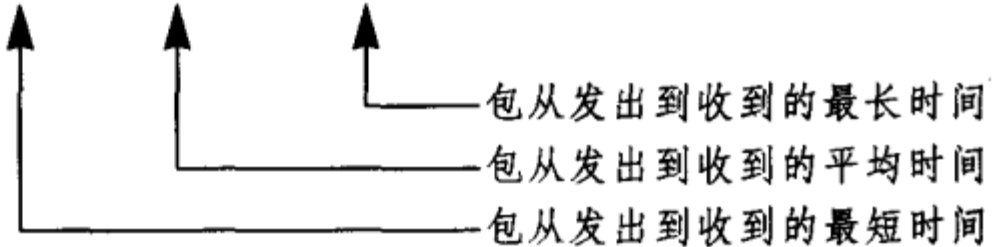


```
--- 210.62.132.10 ping statistics ---
```

```
3 packets transmitted, 3 packets received, 0% packet loss
```



```
rtt min/avg/max = 776.5/1309.0/1764.1 ms
```



③ 侦测www.flag.com.tw主机的网络功能是否正常运作，执行时不显示执行过程：

```
# ping -q www.flag.com.tw
```

```
PING www.flag.com.tw (210.62.132.10) 56(84) bytes of data.
```

← 不显示执行过程

```
--- www.flag.com.tw ping statistics ---
```

```
7 packets transmitted, 7 received, 0% packet loss, time 6053ms
```

```
rtt min/avg/max/mdev = 49.268/78.379/229.931/62.153 ms
```

④ 侦测www.flag.com.tw主机的网络功能是否正常运作，并指定每2秒钟送出一次侦测信息：

```
# ping -i 2 www.flag.com.tw
```



- ⑤ 侦测www.flag.com.tw主机的网络功能是否正常运行，并指定每个包的大小为512字节：

```
# ping -s 504 www.flag.com.tw
PING www.flag.com.tw (210.62.132.10) 504(532) bytes of data.
512 bytes from c10.h210062132.is.net.tw (210.62.132.10):
icmp_seq=1
  ↑
 包大小
...
```

- ⑥ 侦测www.flag.com.tw主机的网络功能是否正常运行，并指定TTL为64：

```
# ping -t 64 www.flag.com.tw
```

- ⑦ 指定使用第2张网卡来侦测www.flag.com.tw主机的网络功能是否正常运行：

```
# ping -I eth1 www.flag.com.tw
```

↑
eth1 为第2张网卡 (eth0为第1张)

- ⑧ 侦测本地端主机的网络功能是否正常运行：

```
# ping 127.0.0.1
```

pjtopppm

常用指数：



(paintjet to portable pixmap)

功能说明：转换打印文件。

语 法： pjtopppm [PaintJet 打印文件]

包 名 称： netpbm-progs (Red Hat Linux/Fedora Core) , netpbm (Mandrake Linux)。

P

补充说明：执行pjtoppm命令，读取指定的PaintJet打印机产生的扫描文件，并把该文件转成Portable Pixmap（默认的扩展文件名为.ppm）格式后，输出到标准输出设备。


参 数：无。

范 例：

把打印文件fromhpprinter转成Portable Pixmap格式：

```
# pjtoppm fromhpprinter > nowitsapixmap.ppm
```

playmidi

常用指数：

功能说明：播放音乐文件。

语 法：playmidi [-48adefgIkMrvwz] [-A <频道掩码>] [-c <频道掩码>] [-C <混音层级>] [-D <外围设备代号>] [-E <频道掩码>] [-F <频道掩码>] [-G <频道掩码>] [-i <频道掩码>] [-m <频道掩码>] [-p <频道编号,><乐器种类>...] [-P <频道编号>] [-R <初始化层级>] [-t <节拍速度>] [-V <频道编号,><播放速度>...] [-x <频道编号>] [MIDI 音乐文件...]

相关命令：sndconfig, xplaymidi。

包 名 称：playmidi (Mandrake Linux)。

补充说明：这是Mandrake Linux才有的程序。MIDI (Music Instrument Digital Interface) 是乐器和电脑之间的标准沟通界面，它记录着每个播放频道所对应的乐器种类与节拍速度等信息。利用playmidi命令，用户可以播放MIDI音乐文件。若不给予任何文件名称或参数，则会显示说明。

参 数:

- 4	输出至OPL-3音源晶片
- 8	在UltraSound声卡上, 强制使用8位元取样
- a	输出至AWE32声卡
-A <频道掩码>	把指定的频道以AWE32声卡播放
-c <频道掩码>	仅播放指定的频道, 不播放其他频道的音乐。频道掩码用十六进制指定
-C <混音层级>	设定混音初始化的层级, 范围从0~127
- d	不播放任何打击乐器, 包括鼓声
-D <外围设备代号>	使用指定的MIDI设备播放音乐
- e	输出至外部MIDI音源器
-E <频道掩码>	把指定的频道以外部的MIDI音源器播放
-f	输出至FM (Frequency Modulation) 音源
-F <频道掩码>	把指定的频道以FM音源播放
- g	输出至UltraSound声卡
-G <频道掩码>	把指定的频道以UltraSound音源卡播放
-i <频道掩码>	忽略指定的频道, 不播放该频道的音乐。频道掩码用十六进制设定
-l	列出GM乐器种类的编号对照表
- k	删除所有正在使用/dev/sequencer的程序
-m <频道掩码>	在指定的频道播放打击乐器。频道掩码用十六进制设定
-M	把MT-32的频道对照表转换成GM (General MIDI) 的频道对照表
-p <频道编号,> <乐器种类>	用指定的乐器播放设定的频道。若没有设定频道, 则会用该乐器播放所有的频道。乐器种类的范围从1~128
-P <频道编号>	在指定的频道播放所有打击乐器
- r	播放音乐时, 以ANSI绘图字符即时显示相关信息
-R <初始化层级>	设定reverb初始化的层级, 范围从0~127
-t <节拍速度>	设定播放时的节拍速度
- v	显示命令执行过程
-V <频道编号,> <播放速度>	设定频道的播放速度。若没有指定频道, 则所有的频道都会以设定的速度为准

-w	等待所有正在使用/dev/sequencer的程序
-x <频道编号>	在可播放的频道掩码中，排除指定的频道
-z	忽略所有的频道。本参数仅供特别的应用程序使用

范 例：

① 在AWE32声卡上播放music.midi音乐文件：

```
# playmidi -a music.midi
```

```
Playmidi 2.4 Copyright (C) 1994-1997 Nathan I. Laredo,
AWE32 by Takashi Iwai
```

```
This is free software with ABSOLUTELY NO WARRANTY.
```

```
For details please see the file COPYING.
```

② 使用外部的音源器播放music.midi音乐文件：

```
# playmidi -e music.midi
```

③ 列出GM乐器所有种类的编号对照表：

```
# playmidi -I
```

```
Playmidi 2.4 Copyright (C) 1994-1997 Nathan I. Laredo,
AWE32 by Takashi Iwai
```

```
This is free software with ABSOLUTELY NO WARRANTY.
```

```
For details please see the file COPYING.
```

```
Gravis Ultrasound Program Info:
```

```
1 acpiano 2 britepno 3 synpiano 4 honky      5 epiano1
6 epiano2 7 hrpschrd 8 clavinet 9 celeste 10 glocken
11 musicbox 12 vibes
```

...



- ④ 在AWE32声卡上播放music.midi音乐文件，但不播放任何打击乐器的声音：

```
# playmidi -a -d music.midi
```

- ⑤ 在AWE32声卡上，指定以第10频道播放music.midi音乐文件：

```
# playmidi -a -P10 music.midi
```

- ⑥ 在AWE32声卡上播放music.midi音乐文件，并指定不使用第5频道播放：

```
# playmidi -a -x5 music.midi
```

- ⑦ 在AWE32声卡上，以82的混音层级播放music.midi音乐文件：

```
# playmidi -a C82 music.midi
```

- ⑧ 在AWE32声卡上，指定以第1至8的频道播放music.midi音乐文件：

```
# playmidi -a -E00ff music.midi
```

- ⑨ 在AWE32声卡上，以10的节拍速度播放music.mid音乐文件：

```
# playmidi -a -t10 music.midi
```

- ⑩ 在AWE32声卡上播放music.midi音乐文件，把第3、7频道分别指定成第126、45号乐器，并将第10、15频道的速度设定为96：

```
# playmidi -p 3,126,7,45 -v 10,96,15,96 -a music.midi
```

连续设定频道的播放速度，组别之间用“,”号区隔
连续设定频道使用的乐器，组别之间用“,”号区隔

popd (pop directory)

常用指数:



功能说明: 删除目录堆栈中的记录。

语 法: popd [-/+n]

包 名 称: bash。

相关命令: dirs, pushd。

补充说明: popd命令可删除目录堆栈中的记录。也可使用dirs命令显示目录堆栈中的记录，使用pushd命令将目录加入目录堆栈中。

若不加任何参数，则先删除目录堆栈中最上层的目录，再将现行工作目录切换成新目录堆栈中最上层的目录。

参 数:

-n	删除从右边算起第n笔的目录 (n从0开始)
+n	删除从左边算起第n笔的目录 (n从0开始)

范 例:

① 显示目录堆栈中的记录，然后删除最上层的目录:

```
# dirs          ← 显示目录堆栈中的记录
/home /etc /usr /var /mnt
# pwd          ← 显示当前工作目录
/home
# popd         ← 删除最上层的目录
/etc /usr /var /mnt
# pwd         ← 显示删除后的目录堆栈
/etc          ← 显示当前工作目录
# pwd         ← 工作目录已经切换成新目录堆栈
/etc          ← 中最上层的目录
```


② 显示目录堆栈中的记录，然后删除目录堆栈中从右边算起第1笔目录：

```
# dirs ← 显示目录堆栈中的记录
```

```
/etc /usr /var /mnt
```

此目录为从右边算起的第1笔目录

```
# popd -1
```

```
/etc /usr /mnt ← /var目录已经被删除
```

③ 显示目录堆栈中的记录，然后删除目录堆栈中从左边算起第1笔目录：

```
# dirs ← 显示目录堆栈中的记录
```

```
/boot /etc /usr /mnt
```

此目录为从左边算起的第1笔目录

```
# popd +1
```

```
/boot /usr /mnt ← /etc目录已经被删除
```

pppd

常用指数：



(point to point protocol daemon)

功能说明：提供PPP连接常驻服务。

语法：pppd [外围设备代号] [连接速度] [auth] [cdtrcts] [connect <script 文件>] [crtsets] [defaultroute] [disconnect <script 文件>] [file <设定文件>] [idle <逾时秒数>] [logfile <记录文件>] [modem] [mru <包大小>] [mtu <包大小>] [netmask <网络掩码>] [noaccomp] [noauth] [noccp] [nocrtsets] [nodefaultroute] [nodtrcts] [nolog] [nopcomp] [novj] [novjccomp] [refuse-chap] [refuse-pap] [require-chap] [require-pap] [xonxoff] [<本地 IP 地址>:<远程 IP 地址>]

包名称：PPP。

相关命令: minicom, pppstats。

补充说明: pppd常驻服务让电脑之间能以PPP通信协议相互连接。它包括三个部分: 序列连接数据的包装、LCP (Link Control Protocol) 与NCP (Network Control Protocol)。通常我们并不直接执行pppd, 而是通过其他程序调用pppd完成PPP通信协议的建立, 例如netcfg、linuxconf、gnome-ppp、kppp或xisp等等。

外围设备一般多指调制解调器 (MODEM, MOdulator/DEModulator) 的串行端口, 旧代号为cua1或cua2 (位于/dev目录下, 相当于DOS中的com1和com2), 新代号则是ttyS0或ttyS1等, 有些程序会建立外围设备代号的符号链接/dev/modem。

连接速度多设成调制解调器标示速度的4倍, 即14 400 bps的调制解调器的速度为57 600, 28 800 bps以上则设为115 200, 连接速度的设定和调制解调器彼此间的传输协议、压缩方式有关, 例如MNP (Microcom Networking Protocol) Class 1~10、v.32/bis、v.34/bis、v.42/bis以及 v.90等, 相关信息详见调制解调器或电话网络方面的数据。

参 数:

auth	要求对等的连接端先提出认证数据, 否则将不允许传送或接收网络包。假设系统已经有内定的Route, 则本参数为默认值
cdtrcts	使用非标准的硬件流量控制, 例如DTR (Data Terminal Ready) /CTS (Clear To Send), 控制串行端口的数据流量

connect <script 文件>	指定在拨号连接、建立PPP通信协议后，所要执行的Script文件或执行文件
crtsets	使用硬件流量控制，例如RTS (Ready To Send) /CTS，控制串行端口的数据流量
defaultroute	把内定的Route加到Routing Table，在PPP连接中断后，加入的Route就会被删除。本参数会覆盖“default-troute”
disconnect <script 文件>	指定在连接中断、停止PPP通信协议后，所要执行的Script文件或执行文件
file <设定文件>	从指定的文件中读取pppd的参数设定
idle <逾时秒数>	设定网络连接的闲置时间，单位以秒计算，超过此时间则为逾时，pppd命令会切断连接。所谓网络闲置，就是表示某段时间内没有接收或送出任何数据包
logfile <记录文件>	把pppd命令执行过程以附加的方式（非覆盖方式）记录到指定的文件中
modem	使用调制解调器控制网络连接，此为默认值。pppd命令在使用序列外围设备时，会等待调制解调器传来的CD (Carrier Detect) 信号
mru <包大小>	设定MRU (Maximum Receive Unit)，单位以Byte计算。最小值为128，默认值为1 500，如果网络连接速度很慢，可设为296（包括40 Bytes的TCP/IP文件头及256 Bytes的数据）
mtu <包大小>	设定最大传输单位 (MTU, Maximum Transmit Unit)，单位以Byte计算。除非对等连接端通过 MRU 设定要求较小的包，否则送出的包大小不会超过本设定值
netmask <网络掩码>	设定网络掩码，例如C级网络的掩码可设为“255.255.255.0”
noaccomp	关闭传送与接收的地址控制压缩
noauth	不要求对等的连接端先提出认证数据。本参数具有优先级
noccp	关闭CCP (Compression Control Protocol) 的沟通操作
nocrtsets	关闭RTS/CTS串行端口硬件流量控制

nodefaultroute	关闭“defaultroute”的功能
nodtrcts	关闭DTR/CTS串行端口硬件流量控制
nolog	不记录pppd命令执行的过程。本参数会取消“logfile”参数的功能
nopcomp	关闭传送与接收的传输协议压缩沟通操作
novj	关闭传送与接收的VJ (Van Jacobson) 形式TCP/IP文件头压缩
novjccomp	在VJ形式的TCP/IP文件头压缩里，关闭连接ID压缩功能
refuse-chap	不接受使用CHAP (Challenge Handshake Authentication Protocol) 认证的连接
refuse-pap	不接受使用PAP (Password Authentication Protocol) 认证的连接
require-chap	要求连接对方使用CHAP认证
require-pap	要求连接对方使用PAP认证
xonxoff	使用软件流量控制 (XON/XOFF) 控制串行端口的数据流量
<本地IP 地址> <远程IP地址>	设定本地主机与远程主机的IP地址，本参数可忽略不予<远程IP地址>指定

范 例：

① 执行pppd命令启动PPP网络连接，若欲通过这种文本模式的手动方式拨号上网，请先使用拨号程序（如minicom）建立拨号连接，离开该程序后再以pppd建立PPP网络连接：

```
# pppd noauth      执行后PPP连接数据会出现于终端画面上
~.}#R!}!}!} }4}"& } } }&}&}'"-.#R! & } } }"}& } } }&}'"
```

← 出现一堆乱码，表示PPP通信协议开始送出数据
 由于并无拨号连接，pppd在尝试连接无效之后，
 将自动结束回到命令提示符号

② 执行PPP连接常驻服务，并指定最大传输单位为512字节：

```
# pppd noauth mtu 512
```


③ 执行PPP连接常驻服务，并指定网络掩码为255.255.255.0：

```
# pppd noauth netmask 255.255.255.0
```

④ 执行ppp连接常驻服务，并启动软件流量控制（XON/XOFF）的方法来
控制串行端口的数据流量：

```
# pppd noauth xonxoff
```

pppstats

常用指数：

(point to point protocol status)

功能说明：显示PPP连接状态。

语法：pppstats [-arv] [-c <执行次数>] [-w <间隔秒数>] [网络接口]

包名称：ppp。

相关命令：minicom, pppd。

补充说明：利用pppstats命令可让用户得知PPP连接网络的相关信息，
假设没有指定网络接口，则其默认的界面为ppp0。



参 数:

-a	显示绝对统计值。如不加上这个参数，pppstats命令会从最后一次回报状态起，计算传输状况的统计值。加上本参数后，就会由PPP一开始连接时计算
-c <执行次数>	设定回报状况的次数。如果附加参数“-w”，却没有搭配这项参数使用，则pppstats将会不断地回报网络状态，直到中断它为止。默认的执行次数为1
-r	显示包压缩比率的统计值
-v	显示VJ TCP文件头的压缩效率统计值
-w <间隔秒数>	设定显示统计信息的间隔时间，单位以秒计算。默认值为5秒

范 例:

① 执行pppstats命令显示PPP网络连接状态:

pppstats

IN	PACK	VJCOMP	VJUNC	VJERR	OUT	PACK	VJCOMP	VJUNC	NON-VJ
0	0	0	0	0	450	10	0	0	10
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
总接收到的位组		收到压缩文件头的TCP包数		收到不正确且压缩文件头的TCP包数		总传送的包数		传送未压缩文件头的TCP包数	
	总收到的包数		收到未压缩文件头的TCP包数		总传送的位组		传送压缩文件头的TCP包数		传送非TCP的包数

② 显示PPP连接状态，并且每3秒更新一次:

pppstats -w 3

③ 显示PPP连接状态，每10秒更新一次，一共回报6次：

```
# pppstats -w 10 -c 6
```

④ 显示PPP连接状态，并列出行包压缩比率的统计值：

```
# pppstats -r
```

⑤ 显示PPP连接状态，并列出行包相对统计值：

```
# pppstats -d
```

⑥ 显示PPP连接状态，并列出行包VJTCP文件头压缩效率的统计值：

```
# pppstats -v
```

pr (print file)

常用指数：



功能说明：将文件格式化编排，以便打印。

语法：pr [-acdfJmrtTv] [-e <Tab字符><Tab宽度>] [-h <文件头字符串>] [-i <Tab字符><Tab宽度>] [-l <每页行数>] [-n <间隔字符><编号位数>] [-N <行数编号>] [-o <偏移量>] [-s <间隔字符串>] [-w <每行字符数>] [+<打印首页><:打印尾页>] [-<每页栏位数>] [--help] [--version] [文件...]

包名称：coreutils。



补充说明： pr命令会读取所给予的文件之内容，并将其内容编排后，输出至标准输出设备。若不指定任何文件名称，或是所给予的文件名为“-”，则pr命令会从标准输入设备读取数据。

参 数：

-a	把竖栏转为横栏。默认每个栏的方向都是由上往下，可利用这项参数将其方向转成从左到右。本参数必须搭配“-<每页栏位数>”参数使用
-c	用“^”符号加上英文本母，显示可打印的控制字符。其他无法打印的控制字符，则以八进制的倒退字符显示
-d	每列都插入一列空白列
-e <Tab字符><Tab宽度>	设定读取数据时的Tab字符和其宽度。用户可利用本参数另行指定Tab字符，或修改它的Tab宽度。假如不指定Tab宽度，则默认值为8
-f或-F	在每页的最后，加上FF (Form Feed) 控制字符，而非新增列控制字符
-h <文件头字符串>	设定文件头字符串，取代原本的文件名称。将文件头字符串设定为“ ”时，则会显示空白列
-i <Tab字符><Tab宽度>	此参数的效果和指定“-e”参数类似，不过是设定输出数据时的Tab字符和其宽度
-J	把文件的内容统统合并成一行。此参数搭配“-m”、“-w”、“-<每页栏位数>”参数使用，可避免文件内容因超过栏宽或页宽，而发生遭截除的状况
-l <每页行数>	设定每一页的总列数。默认每页有66行，其中内容占有56行，页首、页尾各占 10行。当设定的行数小于10 的时候，页首、页尾都会被忽略不予处理，其效果类似参数“-t”或“-T”
-m	同时打印多个文件。每个文件占1个栏，超过栏位宽度的部分将被截除。可借由参数“-j”的功能避免这种现象

-n<间隔字符><编号位数>	列出文件内容时，在每一行的开头标上编号。默认号码从1开始依次累加，位数可达5位数字。间隔字符默认是Tab字符，用来区隔行数编号与文件内容，可另行指定其他字符
-N <行数编号>	设定第1页的第1行的行数编号，其后的行数编号以此为准，依次累加
-o <偏移量>	设定左边界向右移动的偏移量。例如偏移量设为3，则输出时内容会向右偏移3个字符，默认值为0。这项参数不会影响-w参数指定的每行字符数，即每一行的真正字符总数，等于偏移量加上每行字符数的总和
-r	取消文件无法打开时的警告信息
-s “<间隔字符串>”	设定用来区隔栏间的内容的字符串。默认是空格字符，也就是“ ”。若使用本参数却不指定间隔字符串，其效果等于“ ”，即栏与栏之间没有区隔开
-t	取消每一页的页首与页尾
-T	此参数的效果和指定“-t”参数类似，但会消除每页结尾的FF控制字符（假设该控制字符存在）
-v	无法打印的控制字符，使用八进制的倒退字符显示
-w <每行字符数>	设定每行的最大字符数。默认值为每行72个字符，可用这项参数去改变它。超过宽度的部分将被截除，搭配参数“-j”使用可避免这种现象
+<打印起始页><:打印结束页>	打印指定页的内容。默认是从第1页开始打印，假如没有指定打印结束页，则会打印至最后一页
-<每页栏位数>	设定每页要分成多少栏。默认是每页1栏
--help	显示帮助
--version	显示版本信息

范 例：

① 以默认的方式重新编排textfile文本文件，并将结果输出到打印机：

```
# pr textfile | lpr
```

- ② 以每页60行、每行78字符的格式重新编排textfile文本文件，并将结果存储成output文件：

```
# pr -l 60 -w 78 textfile > output
```

- ③ 以每页2栏的格式重新编排textfile文本文件，并将结果存储成output文件：

```
# pr -2 textfile > output
```

```
# cat output ← 显示结果
```

```
2003-07-08 14:40          text          1
  Pico is designed to be a si The bottom two lines list t
  layout very similar to the
  top of the display shows pi Each character typed is aut
```

以每页2栏的方式编排

...

- ④ 重新编排textfile及textfile1等两个文本文件，并将两个文件各以1个栏显示，最后则将结果存储成output文件：

```
# pr -m textfile textfile1 > output
```

```
# cat output
```

```
2003-07-08 15:05          1
  Pico is designed to be a si ps gives a snapshot of the
  layout very similar to the a repetitive update of this
  top of the display shows pi documents the /proc-based ve
```

textfile文件的内容

textfile1文件的内容

...

- ⑤ 以左边界为5个字符的方式重新编排textfile文本文件，并将结果输出到打印机：

```
# pr -o 5 textfile | lpr
```

⑥ 重新编排textfile文本文件，并在每列之间插入一列空白列，最后将结果输出到打印机：

```
# pr -d textfile | lpr
```

⑦ 重新编排textfile文本文件，并指定自定义的文件头字符串，最后将结果输出到打印机：

```
# pr -h "For Love of the Linux" textfile > output
```

指定的文件头字符串

⑧ 重新编排textfile文本文件，并在每列加上流水号，最后将结果输出到打印机：

```
# pr -n textfile | lpr
```

⑨ 重新编排textfile文本文件，并在每列加上流水号，流水号从101开始累加，最后将结果输出到打印机：

```
# pr -n -N 101 textfile | lpr
```

⑩ 以不显示页首及页尾的方式重新编排textfile文本文件，并将结果输出到打印机：

```
# pr -t textfile | lpr
```

⑪ 重新编排textfile文本文件，并指定打印第2页至第5页：

```
# pr +2:5 textfile | lpr
```




⑫ 重新编排textfile文本文件，并从第3页开始打印到最后1页：

```
# pr +3 textfile | lpr
```

↑
因为没有指定结束页，故会打印到最后1页

procinfo

常用指数：

(process information)

功能说明：显示系统状态。

语 法：procinfo [-abdDfhimsSv] [-F<输出文件>] [-n<间隔秒数>]

包 名 称：procinfo。

相关命令：free, ps, tload, top, w。

补充说明：procinfo命令从/proc目录里读取系统相关数据，将数据妥善整理过后输出到标准输出设备。

参 数：

-a	显示所有信息
-b	显示磁盘设备的块数目，而非存取数目
-d	显示系统信息每秒间的变化差额，而非总和的数值
-D	此参数的效果和指定“-d”参数类似，但内存和交换文件（虚拟内存）的信息为总和数值
-f	进入全画面的交互式操作界面
-F<输出文件>	把信息状态输出至文件存储起来，而非默认的标准输出设备。输出的文件必须是一个已经存在的文件，也可以将数据导向外围设备代号，比如“/dev/tty9”，则结果会显示在第9个终端的画面上
-h	显示帮助

-i	显示完整的IRQ (Interrupt Request) 列表
-m	显示系统模块和外围设备等相关信息
-n<间隔秒数>	设定全画面互动模式的信息更新速度, 单位以秒计算。本参数必须配合“-f”参数使用。间隔秒数默认为5秒, 系统管理员可设成0, 持续更新信息, 此时procinfo的优先级将是最高等级, 即-20
-s	显示系统的内存、磁盘空间、IRQ和DMA (Direct Memory Access) 等信息, 此为默认值
-S	搭配参数“-d”或“-D”使用时, 每秒都会更新信息, 不论是否有使用参数“-n”
-v	显示版本信息

范 例:

① 单独执行procinfo命令, 查询系统状态:

procinfo

```
Linux 2.4.20-8 (bhcompile@stripples)(gcc 3.2.2 20030222 ) #1 ...
```

显示系统核心版本、用户及主机名称和域名

```
Memory:  Total    Used    Free Shared Buffers  Cached
Mem:      117956 108144  9812      0   14472  67280
Swap:     522104  61424 460680
```

显示实体与虚拟内存 (交换文件) 的使用情况

```
Bootup:  Fri Jul 4 14:37:28 2003 Load average:0.00 0.00 0.00/78 19480
```

标示系统启动的日期时间

列出过去1、5、15分钟内的系统平均负载, 正在进行中与所有执行程序的数目, 以及方才执行的程序的识别码



```

user : 0:19:47.73      0.4% page in : 3030377 disk 1: 447539r 290188w
nice : 0:00:12.27     0.0% page out: 3644871
system: 0:07:36.97   0.1% swap in : 21132
idle  :3d 20:42:28.57 99.5% swap out: 216938
uptime:3d 21:10:05.53      context : 8706307

```

执行一般用户程序、调整优先级的用户程序，系统程序、闲置时间，及系统启动总时间

显示被系统核心调用、切换的磁盘块数目，及系统启动后数据的总切换数目

磁盘存取过的次数、包括读取与写入

```

irq 0:33540554 timer      irq 6:      2
irq 1:      678 keyboard  irq 8:      1 rtc
irq 2:      0 cascade [4] irq 12:     449 PS/2 Mouse
irq 3: 620013 eth0        irq 14:733092 ide0
irq 4:      3              irq 15:      1 ide1
irq 5:      0 usb-uhci

```

} IRQ 信息

② 以全画面互动模式显示系统状态，可利用稍后所介绍的操作命令，切换所查看的系统信息：

```
# procinfo -f
```

③ 以全画面互动模式显示系统状态，并指定每3秒更新信息一次：

```
# procinfo -f -n3
```

④ 以全画面互动模式显示系统状态，显示时统计出系统信息每秒差额的变化值：

```
# procinfo -d
```

⑤ 以全画面互动模式显示系统状态，除了内存跟交换文件的信息以总和数值显示外，其他部分以每秒的差额显示：

```
# procinfo -D
```


⑥ 显示所有系统状态的信息：

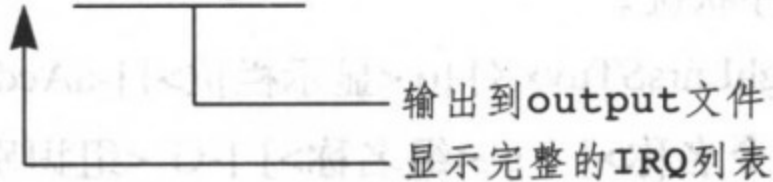
```
# procinfo -a
```

⑦ 显示系统状态，并指定以块数为单位来显示磁盘设备：

```
# procinfo -b
```

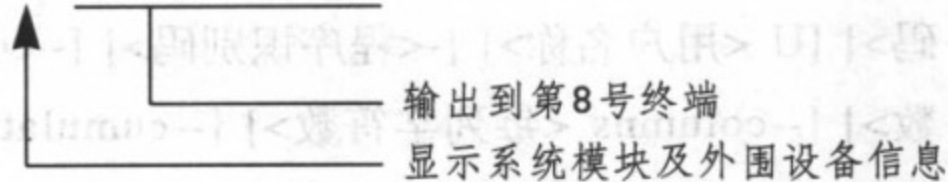
⑧ 列出完整的IRQ列表，并将结果存成output文件（该文件需已存在）：

```
# procinfo -i -F output
```



⑨ 列出系统模块及外围设备的相关信息，并将结果输出第8号终端：

```
# procinfo -m -F /dev/tty8
```



操作命令：

a	显示所有的信息
b	切换磁盘设备以块数或存取数目显示
d	显示系统每秒的差额变化值
D	除了内存跟交换文件的信息以总和数值显示外，其他部分显示每秒的差额
i	切换是否显示完整的IRQ列表
m	显示系统模块和外围设备等信息
q	结束程序
s	当按了 a 或 m 键之后，可按此键回到原画面
S	切换是否每秒钟更新系统信息

■	当按了■或■键之后,可按此键回到原画面
■+■	重新显示画面

ps(process status)

常用指数: 

功能说明: 报告程序状况。

语 法: ps [acefghLnrsSTuvxX] [o <显示栏位>] [-aAcdefHjlmNVwy] [-C <命令名称>] [-g <组名称>] [-G <组识别码>] [-p <程序识别码>] [p <程序识别码>] [-s <阶段操作>] [-t <终端编号>] [t <终端编号>] [-u <用户识别码>] [-U <用户识别码>] [U <用户名称>] [-<程序识别码>] [--cols <每列字符数>] [--columns <每列字符数>] [--cumulative] [--forest] [--headers] [--help] [--info] [--lines <显示列数>] [--no-headers] [--group <组名称>] [--Group <组识别码>] [--pid <程序识别码>] [--rows <显示列数>] --sid <阶段操作>] [--tty <终端编号>] [--user <用户名称>] [--User <用户识别码>] [--version] [--width <每行字符数>]

包名称: procps。

相关命令: bg, fg, free, jobs, kill, pstree, tload, top, w。

补充说明: ps是用来报告程序执行状况的命令,可以搭配kill命令随时中断、删除不必要的程序。

参 数:

-a	显示所有终端下执行的程序,除了session leader及不属任何终端的程序之外
a	显示现行终端下的所有程序,包括其他用户的程序
-A	显示所有的程序
-c	显示CLS和PRI栏位

c	列出程序时，显示每个程序真正的命令名称，而不包含路径、参数，或是常驻服务的标示
-C <命令名称>	指定执行命令的名称，并列出该命令的程序状况
-d	显示所有的程序，但不包括session leader的程序
-e	此参数的效果和指定“-A”参数相同
e	列出程序时，显示每个程序所使用的环境变量
-f	显示UID、PPID、C与STIME栏位
f	用ASCII字符显示树状结构，表达程序间的相互关系
-g <组名称>	此参数的效果和指定“-G”参数相同，但也能使用阶段操作领导者的名称来指定
g	显示现行终端下的所有程序，包括group leader的程序
-G <组识别码>	列出属于该组的程序状况，也可使用组名称来指定
h	不显示标题栏
-H	显示树状结构，表达程序间的相互关系
-j或j	采用工作控制的格式显示程序状况
-l或l	采用详细的格式来显示程序状况
L	列出栏位的相关信息
-m或m	显示所有的进程
n	以数字来表示USER和WCHAN栏位
-N	显示所有的程序，除了执行ps命令终端下的程序之外
o <显示栏位>	以自定的栏位格式来显示程序的状态，可显示的栏位有% C表% CPU栏位、% G表GROUP栏位、% P表PPID栏位、% U表USER栏位、% a及% c表COMMAND栏位、% g表RGROUP栏位、% n表NI栏位、% p表PID栏位、% r表PGID栏位、% t表ELAPSED栏位、% u表PUSER栏位、% x表TIME栏位、% y表TTY栏位、% z表VSZ栏位
-p<程序识别码>	指定程序识别码，并列出该程序的状况
p <程序识别码>或 -<程序识别码>	此参数的效果和指定“-p”参数相同，只在列表格式方面稍有差异。这项参数可以省略，直接指定程序识别码即可
r	只列出现行终端正在执行中的程序
-s <阶段操作>	指定阶段操作的程序识别码，并列出隶属该阶段操作的程序状况

s	采用程序信号的格式显示程序状况
S	列出程序时，包括已死亡的子程序数据
-t <终端编号>	指定终端编号，并列出于该终端的程序状况
t <终端编号>	此参数的效果和指定“-t”参数相同，只在列表格式方面稍有差异
T	显示现行终端下的所有程序
-u <用户识别码>	此参数的效果和指定“-U”参数相同
u	以用户为主的格式来显示程序状况
-U <用户识别码>	列出属于该用户的程序状况，也可使用用户名称来指定
U <用户名称>	列出属于该用户的程序状况
v	采用虚拟内存的格式显示程序状况
-V或V	显示版本信息
-w或w	采用宽阔的格式来显示程序状况
x	显示所有程序，不以终端来区分
X	采用旧式的Linux i386登录格式显示程序状况
-y	配合参数“-l”使用时，不显示F (Flag) 栏位，并以RSS栏位取代ADDR栏位
-<程序识别码>	此参数的效果和指定“p”参数相同
--cols <每列字符数>	设定每列的最大字符数
--columns<每列字符数>	此参数的效果和指定“--cols”参数相同
--cumulative	此参数的效果和指定“S”参数相同
--forest	此参数的效果和指定“f”参数相同
--headers	重复显示标题栏
--help	显示帮助
--info	显示排错信息
--lines <显示行数>	设定显示画面的行数
--no-headers	此参数的效果和指定“h”参数相同，只在列表格式方面稍有差异
--group <组名称>	此参数的效果和指定“-G”参数相同
--Group <组识别码>	此参数的效果和指定“-G”参数相同

--pid <程序识别码>	此参数的效果和指定“-p”参数相同
--rows <显示列数>	此参数的效果和指定“--lines”参数相同
--sid <阶段操作>	此参数的效果和指定“-s”参数相同
--tty <终端编号>	此参数的效果和指定“-t”参数相同
--user <用户名称>	此参数的效果和指定“-U”参数相同
--User <用户识别码>	此参数的效果和指定“-U”参数相同
--version	此参数的效果和指定“-V”参数相同
--width <每行字符数>	此参数的效果和指定“-cols”参数相同

范 例:

① 查询自己正在执行的程序:

ps

PID	TTY	TIME	CMD
879	ttyp0	00:00:00	bash
885	ttyp0	00:00:00	ps

↑ 正在执行的程序或命令
 ↑ 该程序所消耗的CPU时间
 ↑ 使用的终端代号
 ↑ 每个程序都会拥有一个识别码, 以供系统辨识及处理

② 列出自己正在执行的所有程序:

ps x

PID	TTY	STAT	TIME	COMMAND
1	?	S	0:04	init
...				
622	tty1	S	0:00	/sbin/mingetty tty1
624	tty3	S	0:00	/sbin/mingetty tty3
627	tty6	S	0:00	/sbin/mingetty tty6
628	?	S	0:00	/usr/bin/gdm -nodaemon

↑ 包含了自己在所有终端所执行的程序

③ 查询用户名称为oracle正在执行的程序:

```
# ps -U oracle
  PID   TTY     TIME    CMD
 7519   tty4    00:00:00  bash
 7532   tty4    00:00:00  joe
```

④ 查询用户识别码为500的用户正在执行的程序:

```
# ps -U 500    ← 指定用户识别码
```

⑤ 查询第5号终端正在执行的程序:

```
# ps -t 5
  PID  TTY     TIME    CMD
 626   tty5    00:00:00  login
 918   tty5    00:00:00  bash
 933   tty5    00:00:00  joe
```

显示属于第5号终端的程序

⑥ 以不同的格式显示第5号终端正在执行的程序:

```
# ps t 5
  PID  TTY     STAT    TIME    COMMAND
 626   tty5    S        0:00    login -- alkane
 918   tty5    S        0:00    -bash
 933   tty5    S        0:00    joe
```

程序的执行状态, "S" 表示闲置中的程序

⑦ 显示当前终端正在执行的程序:



ps T

PID	TTY	STAT	TIME	COMMAND
988	tty6	S	0:00	login -- sphenoid
991	tty6	S	0:00	-bash
1006	tty6	R	0:00	ps T

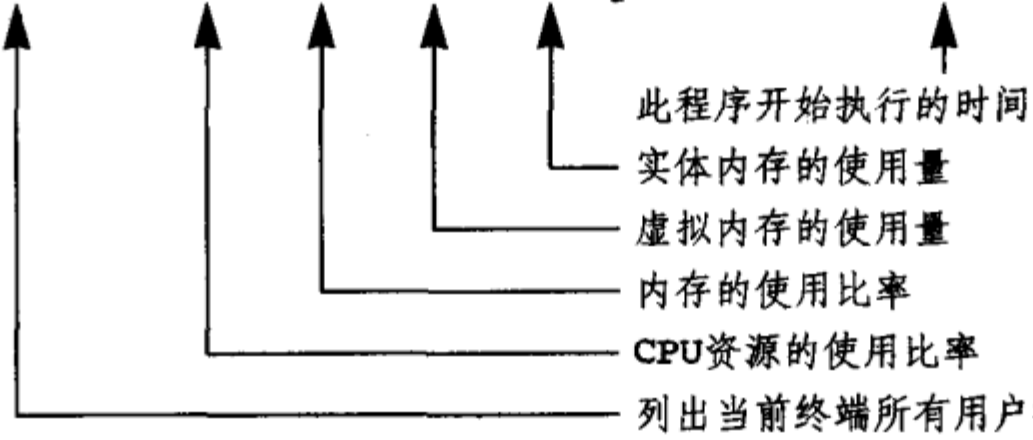


"R" 表示此程序为可执行状态

⑧ 列出当前终端包括其他用户所执行的程序，并以用户为主的格式显示：

ps a u

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	807	0.0	0.8	3068	1116	pts/2	S	09:20	0:00	-bash
root	1095	0.0	0.6	2908	860	pts/2	S	10:11	0:00	su -l alkane
alkane	1096	0.0	0.8	3020	1060	pts/2	S	10:11	0:00	-bash
alkane	1149	0.0	0.6	2508	860	pts/2	R	10:29	0:00	ps a u



⑨ 列出所有终端包括其他用户所执行的程序，并以用户为主的格式显示，但不列出session leader的程序及不属任何终端程序：

ps -a u

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
sphenoid	991	0.0	0.8	3028	1068	tty5	S	09:50	0:00	-bash
root	1050	0.0	0.8	3060	1100	pts/2	S	10:09	0:00	-bash
root	1095	0.0	0.6	2908	860	pts/2	S	10:11	0:00	su -l alkane
alkane	1096	0.0	0.8	3020	1060	pts/2	S	10:11	0:00	-bash
root	1121	0.0	0.3	1648	456	pts/0	S	10:25	0:00	sh./etscape



与上例比较，多列出了在所有终端所执行的程序



⑩ 列出所有的程序，并以用户为主的格式显示：

```
# ps -A u
USER PID %CPU %MEM  VSZ  RSS  TTY  STAT  START  TIME  COMMAND
root   1  0.0  0.3 1104  460  ?   S     09:12  0:04  init
root   2  0.0  0.0   0    0  ?   SW    09:12  0:00  [kflushd]
root   3  0.0  0.0   0    0  ?   SW    09:12  0:02  [kupdate]
```



与上例比较，列出了不属于任何终端的程序

⑪ 查询整个系统所有程序的执行情形：

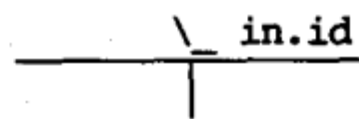
```
# ps aux
```

⑫ 查询整个系统所有执行的程序中，名称包含“httpd”字符串的程序：

```
# ps aux | grep httpd
root   487 0.0  1.6 5408 2136 ?   S    09:13  0:00  /usr/sbin/httpd
apache 492 0.0  0.8 5572 1108 ?   S    09:13  0:00  /usr/sbin/httpd
...
```

⑬ 查询整个系统所有执行的程序，并以树状结构表达出程序间的关系：

```
# ps auxf
USER  PID %CPU %MEM  VSZ  RSS  TTY  STAT  START  TIME  COMMAND
root   1  0.0  0.3 1104  460  ?   S     09:12  0:04  init
...
root   401 0.0  0.3 1124  436  ?   S     09:13  0:00  inetd
nobody 756 0.0  0.5 1400  664  ?   S     09:15  0:00  \_ in.identd -e
root   757 0.0  0.5 1400  664  ?   S     09:15  0:00  \_ in.identd
root   758 0.0  0.5 1400  664  ?   S     09:15  0:00  \_ in.id
```



以树状结构列出程序间的关系

⑭ 以详细的格式列出自己正在执行的程序：

```
# ps -l
  F S UID PID  PPID C  PRI  NI ADDR SZ  WCHAN TTY  TIME  CMD
000 S 0   1237 1235 0  73   0   -   768 wait4 pts/2 00:00:00 bash
100 R 0   1275 1237 0  78   0   -   637 -      pts/2 00:00:00 ps
```

⑮ 以程序信号格式列出自己正在执行的程序：

```
# ps s
UID PID  PENDING  BLOCKED  IGNORED  CAUGHT  STAT TTY  TIME COMMAND
0   1237 00000000 00010000 00384004 0f813efb S   pts/2 0:00 -bash
0   1304 00000000 00000000 00000000 73fbfaf9 R   pts/2 0:00 ps s
```

⑯ 查询自己正在执行的程序，列出程序时，显示每个程序所使用的环境变量：

```
# ps e
PID TTY  STAT TIME COMMAND
1237 pts/2 S      0:00 -bash PWD=/root XAUTHORITY=/root/.Xauthority
1323 pts/2 R      0:00 ps e PWD=/root COLORFGBG=default;default;0
```

列出了程序执行时所用的环境变量

⑰ 查询自己正在执行的程序，显示时不列出标题栏：

```
# ps h
1237 pts/2 S      0:00 -bash
1330 pts/2 R      0:00 ps h
```

← 不显示标题栏

⑱ 以自定义的格式列出自己正在执行的程序：

```
# ps o "%U %p %y %C %x %c"
      ↑   ↑   ↑   ↑   ↑   ↑
      USER PID CPU TIME COMMAND
```

p

```
USER  PID  TTY    %CPU  TIME    COMMAND
root  1237  pts/2  0.0   00:00:00  bash
root  1436  pts/2  0.0   00:00:00  ps
```

⑨ 查询程序识别码第1165号程序的执行状态:

```
# ps -p 1165
  PID TTY          TIME CMD
 1165 pts/1    00:00:00 man
```


⑩ 以不同的格式列出程序识别码第1165号程序的执行状态:

```
# ps p 1165      ← 执行“ps 1165”也有同样的结果
  PID TTY          STAT TIME  COMMAND
 1165 pts/1    S           0:00  man ps
```

⑪ 查询执行less命令程序的状态:

```
# ps -C less
  PID TTY          TIME CMD
 1301 pts/1    00:00:00 less
```

pstree(process status tree)

常用指数: 

功能说明: 以树状图显示程序。

语 法: pstree [-acGhlnpuUV] [-H <程序识别码>] [<程序识别码>/
<用户名称>]

包名称: psmisc。

相关命令: ps。

补充说明： pstree命令用ASCII字符显示树状结构，清楚地表达程序间的相互关系。如果不指定程序识别码或用户名称，则会把系统启动时的第1个程序视为基层，并显示之后的所有程序。若指定用户名称，便会以隶属该用户的第1个程序当作基层，然后显示该用户的所有程序。假设某个程序反复执行数次，pstree会将其显示精简化，例如：

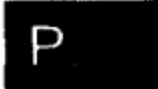
```
init-+-apmd
  |-asclock_applet
  ...
  |-mingetty
  |-mingetty
  |-mingetty
  |-mingetty
  ...
```

上面的树状图将会被简化成下列的样子：

```
init-+-apmd
  |-asclock_applet
  ...
  |-4*[mingetty] ← 简化后的样子
  ...
```

参 数：

-a	显示每个程序的完整命令，包含路径、参数，或是常驻服务标示
-c	不使用精简标示法
-G	使用VT100终端的列绘图字符
-h	列出树状图时，特别标明现在执行的程序
-H <程序识别码>	此参数的效果和指定“-h”参数类似，但特别标明指定的程序
-l	采用长列格式显示树状图。默认当树状图超过画面宽度又没有设定显示宽度，或是输出至非TTY输出设备每行却超过132字符时，其超过的部分将被截除



-n	用程序识别码排序。默认是以程序名称来排序
-p	显示程序识别码
-u	显示用户名称。使用本参数后，当某个子程序的拥有者不同于其父程序的拥有者时，就会特别标出子程序的用户名称
-U	使用UTF-8 (Unicode) 列绘图字符 (line drawing characters)
-V	显示版本信息

范 例:

① 以树状图显示执行的程序:

```
# pstree
init-+-apmd
|-atd
|-crond
|-gdm-+-X
|   `--gdm---Default---kwm-+-kbgndwm
|
|               |-kfm-+-kedit
|
|               | |-rxvt.bin---bash---man---sh
|
...

```

② 以树状图显示当前执行的程序，并列出行程序的完整命令:

```
# pstree -a
init
|-apmd -p 10 -w 5 -W
|-atd
|-crond
|-gdm -nodaemon
...

```

—— 列出执行程序的完整命令



③ 以树状图显示执行的程序，并特别标明当前正在执行的程序：

```
# pstree -h
init-+-apmd
|-atd
|-crond
|-gdm-+-x
|   `--gdm---Default---kwm-+-kbgndwm
|       |
|       |--kfm-+-kedit
...

```

粗体表示正在执行的程序
↓

④ 以树状图显示执行的程序，并特别标明程序识别码为547的程序：

```
# pstree -H 547
init-+-apmd
|-atd
|-crond
|-gdm-+-x
|   `--gdm---Default---kwm-+-kbgndwm
...

```

← 程序识别码547的父程序init也以粗体表示
← 程序识别码547的gdm程序，以粗体表示

⑤ 以树状图显示执行的程序，指定以程序识别码进行排序，并同时显示识别码编号：

```
# pstree -n -p
init(1)-+-kflushd(2)
|   |--kupdate(3)
|   |--kpiod(4)
...

```

↑ 显示程序识别码
↑ 以程序识别码排序
↑ 程序的识别码



⑥ 以树状图显示执行的程序，并显示用户名称：

```
# pstree -u
init-+-apmd
    |-atd
    ...
    |-inetd-+-in.telnetd---login---bash(benny)
    |           ^-in.telnetd---login---bash(irchen)
                                     ↑
                                     用户名称
    ...
```


⑦ 以树状图显示执行的程序，显示时不使用精简标示法：

```
# pstree -c
```

⑧ 以树状图显示执行的程序，并指定使用VT100终端的列绘图字符显示树状结构：

```
# pstree -G
```

pushd (push directory)

常用指数：

功能说明：加入目录堆栈中的记录。

语 法：pushd [目录] 或 pushd [-/+n]

包 名 称：bash。

相关命令：dirs, popd。

补充说明：push命令可将目录加入目录堆栈中，或更换目录堆栈中的顺序。也可使用dirs命令显示目录堆栈中的记录；popd命令删除目录堆栈中的记录。

若不加任何参数，则将目录堆栈中最上层2个目录的顺序对调。

参 数:

-n	将从右边算起第n笔的目录 (n从0开始)，移至目录堆栈的最上层
+n	将从左边算起第n笔的目录 (n从0开始)，移至目录堆栈的最上层

范 例:

① 显示目录堆栈中的记录，然后加入 /home 目录：

```
# dirs          ← 显示目录堆栈中的记录
/mnt /usr /boot
# pushd /home   ← 将/home目录加入目录堆栈中
/home /mnt /usr /boot ← 显示加入后的目录堆栈
    ↑
    新加入的/home目录
```

② 显示目录堆栈中的记录，然后将目录堆栈中最上层2个目录的顺序对调：

```
# dirs
/home /mnt /usr /boot
# pushd
/mnt /home /usr /boot
    ↑
    已经对调最上层的2个目录
```

③ 显示目录堆栈中的记录，然后将目录堆栈中从右边算起第1笔目录，移至目录堆栈的最上层：


```
# dirs          ← 显示目录堆栈中的记录
/mnt /home /usr /boot
    ↑
    此目录为从右边算起的第1笔目录
# pushd -1
/usr /boot /mnt /home
    ↑
    /usr目录已经被移至最上层
```



④ 显示目录堆栈中的记录,然后将目录堆栈中从左边算起第1笔目录,移至目录堆栈的最上层:

```
# dirs
/usr /boot /mnt /home ←显示目录堆栈中的记录
      ↑
      此目录为从左边算起的第1笔目录
# pushd +1
/boot /mnt /home /usr
      ↑
      /boot目录已经被移至最上层
```

pwconv

常用指数: 

(password convert to shadow password)

功能说明: 打开用户的影子密码。

语 法: pwconv

包 名 称: shadow-utils。

相关命令: grpconv, grpuncov, pwuncov。

补充说明: Linux系统里的用户和组密码,分别存放在名称为passwd和group的文件中,这两个文件位于/etc目录下。因系统运作所需,任何人都得以读取它们,造成安全上的漏洞。影子密码将文件内的密码改存在/etc目录下的shadow和gshadow文件内,只允许系统管理员读取,同时把原密码置换为“x”字符,有效地强化了系统的安全性。影子密码的功能可随时打开或关闭,用户只需执行pwconv命令就能打开用户影子密码。

参 数: 无。

范 例:

打开用户影子密码，加强安全防护：

```
# pwconv
```

pwd

常用指数: 

(print working directory)

功能说明: 显示工作目录。

语 法: pwd [--help] [--version]

包 名 称: coreutils。

补充说明: 执行pwd命令可立刻得知用户当前所在的工作目录的绝对路径名称。

参 数:

--help	显示帮助
--version	显示版本信息

范 例:

显示现行的工作目录名称：

```
# pwd
/home/anbes ← 现在的工作目录
# cd /tmp
# pwd
/tmp ← 切换到/tmp目录
      ← 工作目录变成/tmp
```



pwunconv

常用指数: 

(password unconvert from shadow password)

功能说明: 关闭用户的影子密码。

语 法: pwunconv

包 名 称: shadow-utils。

相关命令: grpconv, grpuncov, pwconv。

补充说明: 执行pwunconv命令可关闭用户影子密码, 它会把密码从/etc/shadow文件内回存到/etc/passwd文件里。


参 数: 无。

范 例:

关闭用户影子密码, 方便直接编辑passwd文件建立、删除用户账号:

```
# pwunconv
```

qrttoppm

常用指数: 

(qrt to portable pixmap)

功能说明: 转换QRT输出文件。**语 法:** qrttoppm [QRT输出文件]**包 名 称:** netpbm-progs (Red Hat Linux/Fedora Core), netpbm (Mandrake Linux)。**补充说明:** 执行qrttoppm命令, 读取指定的QRT光迹追踪程序产生的文件, 并把该文件转成 Portable Pixmap (默认最后的扩展文件名为 .ppm) 格式后, 输出到标准输出设备。**参 数:** 无。**范 例:**

把扫描文件 fromraytracer 转成 Portable Pixmap 格式:

```
# qrttoppm fromraytracer > nowitsapixmap.ppm
```

quota

常用指数: **功能说明:** 显示磁盘已使用的空间与限制。**语 法:** quota [-quvV] [用户名称...]或quota [-gqvV] [组名称...]**包 名 称:** quota。**相关命令:** edquota, quotacheck, quotaoff, quotaon, repquota。**补充说明:** 执行quota命令, 可查询磁盘空间的限制, 并得知已使用多少空间。

参 数:

-g	列出组的磁盘空间限制
-q	简明列表。只列出超过限制的部分
-u	列出用户的磁盘空间限制。此为默认值
-v	显示该用户或组, 在所有载入文件系统的磁盘空间限制
-V	显示版本信息

范 例:

① 执行quota命令, 检查alkane用户磁盘空间的使用情形与限制:

```
# quota alkane
```

```
Disk quotas for user alkane (uid 500):
```

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/hda3	220	15360	20480		59	5000	5500	

文件系统 已使用的 磁盘空 磁盘空 文件数 已使用 文件 文件 磁盘空
设备代号 磁盘空间 间限制 间上限 目剩余 的文件 数目 数目 间剩余
宽限期 数目 限制 上限 宽限期

② 查询所有组磁盘空间的使用情形与限制:

```
# quota -g
```

```
Disk quotas for group root (gid 0): none
```

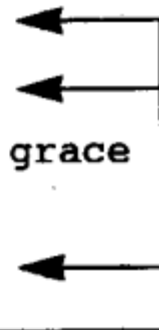
```
Disk quotas for group bin (gid 1):
```

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/hda3	100	25000	27000		4	70		

```
Disk quotas for group daemon (gid 2):
```

```
...
```

会列出所有的组



③ 查询所有组在已载入的文件系统中，磁盘空间的使用情形与限制：

```
# quota -gv
Disk quotas for group root (gid 0):
Filesystem blocks quota limit grace files quota limit grace
/dev/hda3 1719252 0 0 95501 0 0
/dev/hdb1 20 0 0 2 0 0
Disk quotas for group bin (gid 1):
Filesystem blocks quota limit grace files quota limit grace
/dev/hda3 100 25000 27000 4 7000 7200 ←
/dev/hdb1 0 13240 18000 0 5250 5700 ←
...
```

列出了所有载入的文件系统

④ 查询 sphenoid 组磁盘空间使用情形与限制：

```
# quota -g sphenoid
```

⑤ 查询 sphenoid 组在所有已载入文件系统中，磁盘空间的使用情形与限制：

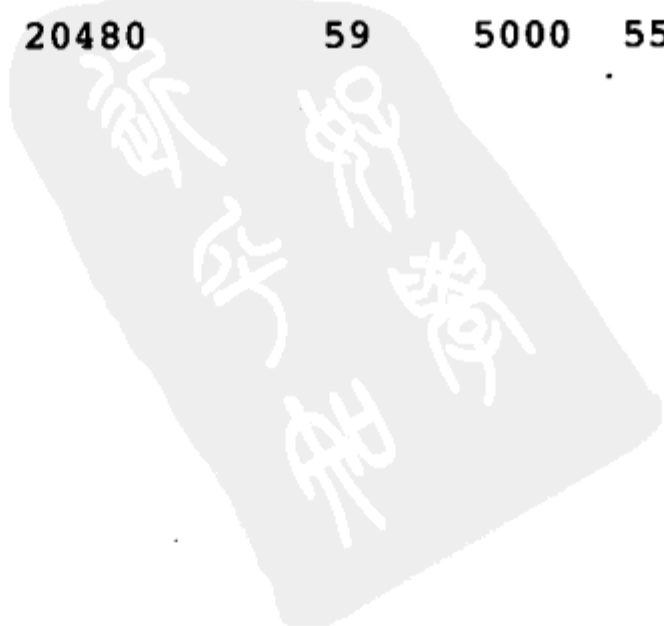
```
# quota -gv sphenoid
```

⑥ 查询 sphenoid 组超过磁盘空间限制的部分：

```
# quota -gq sphenoid
```

⑦ 查询自己磁盘空间使用情形与限制（执行“quota -u”也有同样的结果）：

```
# quota ← 不加上用户名称或组名称
Disk quotas for user alkane (uid 500):
Filesystem blocks quota limit grace files quota limit grace
/dev/hda3 220 15360 20480 59 5000 5500
```



q

⑧ 查询自己在所有已载入的文件系统中, 磁盘空间的使用情形与限制:

```
$ quota -uv
```

```
Disk quotas for user alkane (uid 500):
```

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/hda3	220	15360	20480		59	5000	5500	←
/dev/hdb1	0	15560	20600		0	5200	5700	←

列出了所有载入的文件系统

⑨ 查询 cirware 用户磁盘空间的使用情形与限制:

```
# quota -u cirware
```

⑩ 查询 cirware 及 jacket 用户在已载入的文件系统中, 磁盘空间的使用情形与限制:

```
# quota -uv cirware jacket
```

⑪ 查询 cirware 用户超过磁盘空间限制的部分:

```
# quota -uq cirware
```

quotacheck

常用指数: 

功能说明: 检查磁盘的使用空间与限制。

语 法: quotacheck [-dgRuy] [文件系统...] 或 quotacheck [-acdgmuv]

包 名 称: quota。

相关命令：edquota, quota, quotaoff, quotaon, repquota。

补充说明：执行quotacheck命令，扫描载入系统的分区，并在各分区的文件系统根目录（一个分区载入/tmp/harddisk目录，其文件系统根目录就是/tmp/harddisk）下产生quota.user和quota.group文件，设定用户和组的磁盘空间限制，也可直接指定文件系统进行扫描。quotacheck命令应仅限于系统管理者方可执行，因其他用户不见得具有足够的权限，能够读取指定的文件系统的所有目录。

注意quotacheck扫描的时间和磁盘空间的使用率有关，假设一个已使用90%的分区需要2分钟进行扫描，则在相同分区大小的前提下，一个使用率为45%的分区，仅不到1分钟就能完成扫描操作。

参 数：

-a	扫描在/etc/fstab文件里，有加入quota设定的分区
-c	不读取已存在的aquota数据库，重新扫描硬盘并存储
-d	详细显示命令执行过程，以便于排错或了解程序执行的情形
-g	扫描磁盘空间时，计算每个组织别码所占用的目录和文件数目，并建立quota.group文件
-m	强制执行命令
-R	排除根目录（绝对路径）所在的分区。本参数必须配合“-a”参数使用
-u	扫描磁盘空间时，计算每个用户识别码所占用的目录和文件数目，并建立quota.user文件。此为默认值
-v	显示命令执行过程

范 例:

- ① 依据/etc/fstab文件, 为所有分区建立aquota.user文件, 执行时显示命令的执行过程:

```
# quotacheck -avumc
```

```
quotacheck: Scanning /dev/hda2 [/] done
```

```
quotacheck: Checked 20401 directories and 294718 files —
                          建立/dev/hda2分区的aquota.user
```

```
quotacheck: Scanning /dev/hda3 [/mnt/download] done
```

```
quotacheck: Checked 10049 directories and 155718 files —
                          建立/dev/hda3分区的aquota.user
```

- ② 依据/etc/fstab文件, 为所有分区建立aquota.group文件, 执行时显示命令的执行过程:

```
# quotacheck -avgmc
```

```
quotacheck: Scanning /dev/hda2 [/] done
```

```
quotacheck: Checked 20402 directories and 294720 files —
                          建立/dev/hda2分区的aquota.group
```

```
quotacheck: Scanning /dev/hda3 [/mnt/download] done
```

```
quotacheck: Checked 10049 directories and 155718 files —
                          建立/dev/hda3分区的aquota.group
```

- ③ 为/dev/hda2分区建立aquota.user及aquota.group文件, 执行时显示命令的执行过程:

```
# quotacheck -vugmc /dev/hda2
```

```
quotacheck: Scanning /dev/hda2 [/] done
```

```
quotacheck: Checked 20401 directories and 294718 files
```

④ 依据/etc/fstab文件，除了根目录所在的分区外，建立所有分区的 quota.user文件，执行时显示命令的执行过程：

```
# quotacheck -aRvcm
```

```
quotacheck: Scanning /dev/hda3 [/mnt/download] done
```

```
quotacheck: Checked 10049 directories and 155718 files
```

仅在载入点不是根目录的分区建立aquota.user

quotaoff

常用指数：

功能说明：关闭磁盘空间限制。

语 法：quotaoff [-guv] [文件系统...] 或 quotaoff [-aguv]

包 名 称：quota。

相关命令：edquota, quota, quotacheck, quotaon, repquota。

补充说明：执行 quotaoff 命令可关闭用户和组的磁盘空间限制。

参 数：

-a	关闭在/etc/fstab 文件里有加入 quota 设定的分区的空间限制
-g	关闭组的磁盘空间限制
-u	关闭用户的磁盘空间限制, 此为默认值
-v	显示命令执行过程

范 例:

① 关闭所有分区的磁盘空间限制，执行时显示命令执行过程：

```
# quotaoff -av
/dev/hda2 [/]: group quotas turned off
                        关闭 /dev/hda2 的组磁盘空间限制
/dev/hda2 [/]: user quotas turned off
                        关闭 /dev/hda2 的用户磁盘空间限制
/dev/hda3 [/mnt/download]: group quotas turned off
                        关闭 /dev/hda3 的组磁盘空间限制
/dev/hda3 [/mnt/download]: user quotas turned off
                        关闭 /dev/hda3 的用户磁盘空间限制
```

② 关闭所有分区的组磁盘空间限制，执行时显示命令执行过程：

```
# quotaoff -agv
/dev/hda2 [/]: group quotas turned off
/dev/hda3 [/mnt/download]: group quotas turned off
                        只关闭组磁盘空间限制
```

③ 关闭所有分区的用户磁盘空间限制，执行时显示命令执行过程：

```
# quotaoff -auv
/dev/hda2 [/]: user quotas turned off
/dev/hda3 [/mnt/smb]: user quotas turned off
                        只关闭用户磁盘空间限制
```

④ 关闭/dev/hda3分区的用户磁盘空间限制，执行时显示命令执行过程：

```
# quotaoff -uv /dev/hda3
/dev/hda3 [/mnt/download]: user quotas turned off
```

quotaon

常用指数: 

功能说明: 打开磁盘空间限制。

语法: quotaon [-guv] [文件系统...] 或 quotaon [-aguv]

包名称: quota。

相关命令: edquota, quota, quotacheck, quotaoff, repquota。

补充说明: 执行quotaon命令可打开用户和组的磁盘空间限制, 各分区的文件系统根目录必须有aquota.user和aquota.group设定文件。

参数:

-a	打开在 /etc/fstab 文件里有加入 quota 设定之分区的空间限制
-g	打开组的磁盘空间限制
-u	打开用户的磁盘空间限制, 此为默认值
-v	显示命令执行过程

范例:

① 打开所有分区的磁盘空间限制, 执行时显示命令执行过程:

```
# quotaon -av
```

```
/dev/hda2 [/]: group quotas turned on
```

← 打开 /dev/hda2 的组磁盘空间限制

```
/dev/hda2 [/]: user quotas turned on
```

← 打开 /dev/hda2 的用户磁盘空间限制

```
/dev/hda3 [/mnt/download]: group quotas turned on
```

← 打开 /dev/hda3 的组磁盘空间限制

```
/dev/hda3 [/mnt/download]: user quotas turned on
```

← 打开 /dev/hda3 的用户磁盘空间限制

q

② 打开所有分区的组磁盘空间限制, 执行时显示命令执行过程:

```
# quotaon -agv
```

```
/dev/hda2 [/]: group quotas turned on
```

```
/dev/hda3 [/mnt/smb]: group quotas turned on
```

} 只打开组磁
盘空间限制

③ 打开所有分区的用户磁盘空间限制, 执行时显示命令执行过程:

```
# quotaon -auv
```

```
/dev/hda2 [/]: user quotas turned on
```

```
/dev/hda3 [/mnt/smb]: user quotas turned on
```

} 只打开用户磁
盘空间限制

④ 打开 /dev/hda3分区的组磁盘空间限制, 执行时显示命令执行过程:

```
# quotaon -gv /dev/hda3
```

```
/dev/hda3: group quotas turned on
```

raidstart(RAID start)

常用指数:



功能说明: 启动磁盘阵列。

语 法: `raidstart [-ahV] [-c <设定文件>] [磁盘阵列设备]`

包 名 称: `raidtools`。

相关命令: `mkraid, raidstop`。

补充说明: 磁盘阵列 (redundant array of independent disks) 是将多个磁盘组成一个逻辑上的磁盘。在操作上, 我们会觉得跟使用单一磁盘一样, 但在实际的运作中, 磁盘阵列是将数据分别存储在不同的磁盘上。

利用磁盘阵列具有增进访问效率、可整合多个磁盘、具有高安全性、容错能力等好处。

`raidstart`命令可用来重新启动停止中的磁盘阵列设备, 用户也可查看`/proc/mdstat`记录文件查看磁盘阵列当前的执行状态。

参 数:

<code>-a</code> 或 <code>-all</code>	启动设定文件中所有磁盘阵列设备
<code>-c <设定文件></code> 或 <code>--configfile <设定文件></code>	使用指定的设定文件, 而不使用默认的 <code>/etc/raidtab</code> 设定文件
<code>-h</code> 或 <code>--help</code>	显示帮助
<code>-V</code> 或 <code>--version</code>	显示版本信息

范 例:

① 启动 `/dev/md0` 磁盘阵列:

```
# raidstart /dev/md0
```

② 启动 myraidtab 设定文件中 /dev/md0 磁盘阵列设备:

```
# raidstart -c myraidtab /dev/md0
```

③ 启动默认设定文件中所有的磁盘阵列设备:

```
# raidstart -a
```

raidstop(RAID stop)

常用指数:



功能说明: 停止运作中的磁盘阵列。

语法: raidstop [-ahV] [-c <设定文件>] [磁盘阵列设备]

包名称: raidtools。

相关命令: mkraid, raidstart。

补充说明: 当启动磁盘阵列后, 可利用此命令停止运作中的磁盘阵列, 用户也可查看 /proc/mdstat 记录文件查看磁盘阵列当前的执行状态。

参数:

-a 或 -all	停止执行设定文件中所有的磁盘阵列设备
-c <设定文件> 或 --configfile <设定文件>	使用指定的设定文件, 而不使用默认的/etc/raidtab 设定文件
-h 或 --help	显示帮助
-V 或 --version	显示版本信息

范 例:

① 停止运作中的 /dev/md0 磁盘阵列设备:

```
# raidstop /dev/md0
```

② 停止执行 myraidtab 设定文件中 /dev/md0 磁盘阵列设备:

```
# raidstop -c myraidtab /dev/md0
```

③ 停止执行默认设定文件中所有的磁盘阵列设备:

```
# raidstop -a
```

rasttopnm

常用指数:



(raster to portable anymap)

功能说明: 转换图像文件。

语 法: rasttopnm [Raster 图像文件]

包 名 称: netpbm-progs (Red Hat Linux/Fedora Core), netpbm (Mandrake Linux)。

补充说明: 执行rasttopnm命令, 读取指定的Raster格式图像文件, 并把该文件转成Portable Anymap (默认最后的扩展文件名为.pnm) 格式后, 输出到标准输出设备。Anymap的格式可以是Bitmap、Graymap或Pixmap, 视输入的文件格式而定。

参 数: 无。

范 例:

把图像文件 mypicture 转成 Portable Anymap 格式:

```
# rasttopnm mypicture > mypicture.pnm
```

rcp(remote copy)

常用指数:



功能说明: 远程复制文件或目录。

语 法: rcp [-pr] [来源文件或目录] [目的文件或目录]或rcp [-pr]
[来源文件或目录...] [目的目录]

包 名 称: rsh。

相关命令: rlogin, rsh。

补充说明: rcp命令用在远程复制文件或目录,如同时指定两个以上的文件或目录,且最后的目的地是一个已经存在的目录,则它会把前面指定的所有文件或目录复制到该目录中。远程路径与文件名称之前,必须指定远程主机的名称。例如存放在主机sw.flash.com的 /tmp目录里的文件remotefile,其文件名称应设成“sw.flash.com:/tmp/remotefile”。假如不用当前的账号登录该主机,便需再加上欲使用的账号名称,假设账号为anya,则文件名称为“anya@sw.flash.com:/tmp/remotefile”。注意rcp命令并不会提示用户输入密码,必须使用无须密码就能登录的账号。

参 数:

-p	保留来源文件或目录的属性,包括拥有者、所属组、权限与时间
-r	递归处理,将指定目录下的文件与子目录一并处理

范 例:

- ① 利用rcp命令，以irchen账号登录到linux.flag.com.tw主机，再将该主机上的/tmp/file文件复制到本地主机的/var/tmp目录内：

```
# rcp irchen@linux.flag.com.tw:/tmp/file /var/tmp
```

|
|
|
|

账号名称
主机地址
远程文件
本地端目录

- ② 利用rcp命令，以irchen账号登录到linux.flag.com.tw主机，再将该主机上的/usr/tmp目录中的所有文件包括子目录，统统复制到现在的目录里：

```
# rcp -r irchen@linux.flag.com.tw:/usr/tmp .
```

- ③ 利用rcp命令，以irchen账号登录到linux.flag.com.tw主机，再将该主机上的用户根目录下的chen_file文件复制到本地主机的/var/tmp目录内，复制后保留原来文件的属性：

```
# rcp -p irchen@linux.flag.com.tw:chen_file /var/tmp
```

不使用绝对路径，则表示用户根目录下的文件

rdate(receive date)

常用指数：



功能说明：显示其他主机的日期与时间。

语 法：rdate [-ps] [主机名称或 IP 地址...]

包 名 称：rdate。

相关命令： clock, date, hwclock, timeconfig。

补充说明： 执行rdate命令，向其他主机询问系统时间并显示出来。该命令遵循RFC (Request For Comments)868规范，传输日期和时间的信息。系统管理者可通过这个命令校对、调整本地主机的系统时间。

参 数：

-p	显示远程主机的日期与时间。此为默认值
-s	把从远程主机收到的日期和时间存回到本地主机的系统时间

范 例：


① 显示 time.odin.org 主机的日期与时间：

```
# rdate time.odin.org
```

② 从 time.odin.org 主机下载日期与时间的信息，并以其为标准来调整系统时间：

```
# rdate -s time.odin.org
```

reboot

常用指数：

功能说明： 重新开机。

语 法： reboot [-dfinw]

包 名 称： SysVinit。

相关命令： halt, shutdown。

补充说明：执行reboot命令可让系统停止运作，并重新开机。

参 数：

-d	重开机时不把数据写入记录文件 /var/tmp/wtmp。本参数具有“-n”参数的效果
-f	强制重新开机，不调用 shutdown 命令的功能
-i	在重开机之前，先关闭所有网络接口设备
-n	重开机之前不检查是否有未结束的程序
-w	仅作测试，并不真正地将系统重新开机，只会把重新开机的数据写入 /var/log目录下的wtmp记录文件

范 例：

① 将系统重新开机：

```
# reboot
```

② 先关闭所有的网络接口设备后，再重新开机：

```
# reboot -i
```

③ 不真正地将系统重新开机，仅将重新开机的信息写入 wtmp 记录文件：

```
# reboot -w
```

renice

常用指数：

功能说明：调整优先级。

语 法：renice [优先等级] [-g <程序组名称>...] [-p <程序识别码>...] [-u <用户名称>...]

包名称: util-linux。

相关命令: nice, top。

补充说明: renice命令可重新调整程序执行的优先级。默认是以程序识别码指定程序调整其优先级，也可指定程序组或用户名称调整优先级，一并修改所有隶属于该程序组或用户的程序优先级。等级范围从-20（最高优先级）~19（最低优先级），只有系统管理者得以改变其他用户程序的优先级，也仅有系统管理者可以设定负数等级。

参 数:

-g <程序组名称>	使用程序组名称, 修改所有隶属于该程序组的程序优先级
-p <程序识别码>	改变该程序的优先级等级, 此参数为默认值
-u <用户名称>	指定用户名称, 修改所有隶属于该用户的程序优先级

范 例:

① 将程序识别码为13575的程序优先级设成-10:

```
# renice -10 13575
```

↑ ↑ 参数“-p”可以省略，直接指定程序识别码即可
 ↑ 优先级的表示法和 nice 命令不同，相差一个“-”号

② 把属于用户gabe和lian的程序优先级统统改为5:

```
# renice 5 -u gabe lian
```

```
500: old priority 2, new priority 5
```


```
501: old priority 2, new priority 5
```

↑ ↑ ↑
 用户识别码 原本的优先级 新设定的优先级

③ 把属于 jacket 组程序的优先级改成 2:

```
# renice 2 -g jacket
506: old priority 0, new priority 2
```

repquota (report quota)

常用指数: 

功能说明: 检查磁盘空间限制的状态。

语法: repquota [-guv] [文件系统...] 或 repquota [-aguv]

包名称: quota。

相关命令: edquota, quota, quotacheck, quotaoff, quotaon。

补充说明: 执行repquota命令, 可报告磁盘空间限制的状况, 清楚得知每位用户或每个组已使用多少空间。

参数:

-a	列出在 /etc/fstab 文件里有加入 quota 设定的分区的使用状况, 包括用户和组
-g	列出所有组的磁盘空间限制
-u	列出所有用户的磁盘空间限制。此为默认值
-v	显示该用户或组的所有空间限制

范 例:

① 查询系统上所有用户和组的磁盘空间限制:

repquota -a *** Report for user quotas on device /dev/hda2

组与用 户名称	存放空间 的限制				文件数量 的限制			
	Block limits				File limits			
User	used	soft	hard	grace	used	soft	hard	grace
root	-- 1146362	0	0	78601	0	0		
bin	-- 247	102400	153600	9	5000	5500		
..								
squid	-- 2	0	0		2	0	0	
gabe	-- 1946	10240	15360		390	1000	550	
lian	-- 51	15360	20480	47	700	750		
oinstall	+-- 10373	10240	20480	7days	106	150	300	

↑
存放空间宽限期↑
文件数量宽限期, 若超过会出现在这一栏

后面出现“+”号时, 表示超过文件数量限制

前面出现“+”号时, 表示超过存放空间限制

② 查询系统上所有用户的磁盘空间限制:

repquota -au

③ 查询系统上所有组的磁盘空间限制:

repquota -ag

④ 查询 /dev/hdb1 分区上所有用户与组的磁盘空间限制:

repquota -ug /dev/hdb1

R

resize

常用指数: 

功能说明: 设定 X Window 终端窗口大小。

语 法: `resize [-cu] [-s <行数> <列数>]`

包 名 称: XFree86。

补充说明: 执行resize命令可设定X Window的xterm终端窗口大小。

参 数:

-c	就算用户环境并非 C Shell, 也用 C Shell 命令改变窗口大小
-s<行数><列数 >	设定xterm 终端窗口的垂直高度和水平宽度
-u	就算用户环境并非Bourne Shell, 也用Bourne Shell命令改变窗口大小

范 例:

① 列出当前xterm终端窗口大小的设定值:

```
# resize
COLUMNS=80;
LINES=24;
export COLUMNS LINES;
```

② 把xterm终端的窗口设为50行高、132字符宽:

```
# resize -s 50 132
```

restore

常用指数：

功能说明：还原 (Restore) 由转储 (Dump) 操作所备份下来的文件或整个文件系统 (一个分区)。

语 法：restore [-cCvy] [-b <块大小>] [-D <文件系统>] [-f <备份文件>] [-s <文件编号>] 或 restore [-chimvy] [-b <块大小>] [-f <备份文件>] [-s <文件编号>] 或 restore [-crvy] [-b <块大小>] [-f <备份文件>] [-s <文件编号>] 或 restore [-cRvy] [-b <块大小>] [-f <备份文件>] [-s <文件编号>] 或 restore [-chtvy] [-b <块大小>] [-f <备份文件>] [-s <文件编号>] [文件...] 或 restore [-chmvxy] [-b <块大小>] [-f <备份文件>] [-s <文件编号>] [文件...]

相关命令：dump。

补充说明：restore命令所进行的操作和dump命令相反，转储操作可用来备份文件，而还原操作则是写回这些已备份的文件。

参 数：

-b <块大小>	设定块大小, 单位是 Byte。若不指定块大小, restore 会自行决定
-c	不检查输出操作的备份格式, 仅准许读取使用旧格式的备份文件
-C	使用比对模式, 将备份的文件与当前文件相互比对
-D <文件系统>	允许用户指定文件系统的名称
-f <备份文件>	从指定的文件中读取备份数据, 进行还原操作。也可指定周边设备代号, 如磁带机或磁盘等等。若所给予的文件名为 “-”, 则 restore 命令会从标准输入设备读取数据。假设文件名称以网络形式指定, 例如 “remote.host:backup” 或 “bugs@remote.host:backup”, 则会通过 rmt 命令从远程主机上读取数据

-h	仅解出目录,而不包括与该目录相关的所有文件
-i	使用互动模式,在进行还原操作时,restore 命令将依次询问用户
-m	解开符合指定的 inode 编号的文件或目录,而非采用文件名称指定
-r	进行还原操作
-R	全面还原文件系统时,检查应从何处开始进行。如果备份的数据超过一卷磁带,restore 命令将要求用户依先后顺序放入正确的磁带
-s <文件编号>	当备份数据超过一卷磁带时,可以指定备份文件的编号
-t	指定文件名称,若该文件已存在备份文件中,则列出它们的名称。如不指定任何文件名称,restore 命令会列出备份文件中的根目录(相对路径),并包含其下全部的文件和子目录
-v	显示命令执行过程
-x	设定文件名称且从指定的存储媒体里读入它们,若该文件已存在备份文件中,则将其还原至文件系统内。如不指定任何文件名称,restore 命令会还原备份文件中的根目录(相对路径),并包含其下全部的文件和子目录
-y	不询问任何问题,一律以同意回答并继续执行命令

范 例:

将 totalrecall 备份文件的数据还原至整个文件系统:

```
# restore -r -f totalrecall
```

rgrep(recursive grep)

常用指数:



功能说明: 递归查找文件里符合条件的字符串。

语 法: rgrep [-?BcDFhHilnNrv] [-R <范本样式>] [-W <列长度>] [-x <扩展文件名>] [--help] [--version] [范本样式] [文件或目录...]

包名称: rgrep (Mandrake Linux)。

相关命令: egrep, fgrep, grep。

补充说明: 这是Mandrake Linux才有的命令。rgrep命令的功用和grep命令类似,可查找内容包含指定的范本样式的文件,如果发现某文件的内容符合所指定的范本样式,默认rgrep命令会把含有范本样式的那一行显示出来。rgrep本身就具备递归处理的能力,可以查找指定目录下的所有子目录,效率较grep更佳。

参 数:

-?	显示范本样式与范例的说明
-B	忽略二进制的文件。若rgrep命令找到的文件其数据是二进制格式而非文字文件时,则略过该文件不予处理
-c	计算符合范本样式的列数。使用本参数将不会显示合乎条件之列的内容,而会显示每个文件里共有几行符合范本样式。搭配参数“-v”使用,则会显示每个文件里共有几行不符合范本样式
-D	排错模式,只列出命令查找的目录清单,而不会读取文件内容
-F	当遇到符号链接时,rgrep默认是忽略不予处理。加上本参数后,rgrep命令就会读取该连接所指向的原始文件的内容
-h	特别将符合范本样式的字符串标示出来
-H	只列出符合范本样式的字符串,而非显示整列的内容
-i	忽略字符大小写差别。包括范本样式和指定要查找的文件名称
-l	列出文件内容符合指定范本样式的文件名称
-n	在显示符合范本样式的那一行之前,标示出该行的行数编号
-N	不要递归处理
-r	递归处理,将指定目录下的所有文件及子目录一并处理
-R <范本样式>	此参数的效果和指定“-r”参数类似,但只处理符合范本样式的文件名称的文件

-v	反转查找。寻找并显示所有不包含指定范本样式的行的内容。举例而言,假设范本样式为“1481010”,使用此参数之后,在查找范围内所有文件内容中,每个不包含“1481010”字符串的行都会被显示出来
-W<每行字符数>	限制每行最多显示的字符数
-x <扩展文件名>	只处理符合指定扩展文件名的文件名称的文件
--help	显示帮助
--version	显示版本信息

范 例:

- ① 对当前目录下所有扩展文件名为.txt的文件内容寻找句子中包含“RAID”字符串的文件:

```
# rgrep RAID *.txt
file1.txt: -- one of the RAID developers --, the linux-raid mailing
file1.txt: The reason this HOWTO was written though a Software-RAID
```

↑
包含字符串的文件名称

↑
包含字符串该行的内容

...

若执行以下的命令也有同样的结果:

```
# rgrep -x txt RAID .
```

↑ 寻找当前目录下的文件

↑ 只处理扩展文件名为 txt 的文件

- ② 对当前目录下所有扩展文件名为.txt的文件内容寻找句子中包含“RAID”字符串的文件,显示时在每行前面标示该行的行数编号:

```
# rgrep -n RAID *.txt
file1.txt:3:  -- one of the RAID developers --, the
linux-raid mailing
file1.txt:6:  The reason this HOWTO was written though a
Software-RAID
    ↑
    行数编号
...
```

- ③ 对当前目录下所有扩展文件名为.txt的文件内容寻找句子中包含“RAID”字符串的文件，并特别标示出此字符串：

```
# rgrep -h RAID *.txt
file1.txt:  -- one of the RAID developers --, the linux-
raid mailing file1.txt:  The reason this HOWTO was
written though a Software-RAID
    ↑
    以粗体表示该字符串
...
```

- ④ 对当前目录下所有扩展文件名为.txt的文件内容寻找句子中包含“RAID”字符串的文件，显示时只列出该字符串，而不显示整行的内容：

```
# rgrep -H RAID *.txt
file1.txt:RAID
file1.txt:RAID
...
```

- ⑤ 对当前目录下所有扩展文件名为.txt的文件内容寻找句子中包含“RAID”字符串的文件，寻找时忽略大小写的差别：

```
# rgrep -i RAID *.txt
file1.txt:  -- one of the RAID developers --, the linux-
raid mailing
file1.txt:  raid@vger.rutgers.edu) and various other people.
    |
```

小写的RAID仍被视为符合条件

⑥ 对当前目录下所有扩展文件名为.txt的文件内容寻找句子中不包含“RAID”字符串的文件：

```
# rgrep -v RAID *.txt
file1.txt: This howto is written by Jakob OEstergaard based on a
file1.txt: emails between the author and Ingo Molnar(mingo@chiara.
```

↑
文件名称

└──┘
这些列不包含此字符串

...

⑦ 对当前目录下所有扩展文件名为.txt的文件内容寻找句子中包含“RAID”字符串的文件，若找到则显示其文件名，但不显示包含字符串的该行：

```
# rgrep -l RAID *.txt
file1.txt
file2.txt
```

└──┘ 包含该字符串的文件

⑧ 对当前目录下所有扩展文件名为.txt的文件计算包含“RAID”字符串的总列数：

```
# rgrep -c RAID *.txt
file1.txt:109 ← file1.txt 文件中有 109 列包含此字符串
file2.txt:2 ← file2.txt 文件中有 2 列包含此字符串
```

⑨ 对当前目录下所有扩展文件名为.txt的文件，包括符号链接的文件，计算包含“RAID”字符串的总行数：

```
# ls -l *.txt ← 列出当前目录下所有扩展文件名为 .txt 的文件
-rw-r--r-- 1 root root 31430 May 29 15:58 file1.txt
-rw-r--r-- 1 root root 5070 May 29 5:48 file2.txt
lrwxrwxrwx 1 root root 5 May 29 16:39
symbolic_file.txt -> file
```

此文件是一个符号链接



rgrep -c RAID *.txt ← 以默认的方式计算 "RAID" 出现的列数

file1.txt:109

file2.txt:2

rgrep -c -F RAID *.txt

↑
处理符号链接的文件

file1.txt:109

file2.txt:2

symbolic_file.txt:3 ← 列出符号链接文件的结果

⑩ 对当前目录下所有扩展文件名为.txt的文件计算不包含 "RAID" 字符串的总列数:

rgrep -c -v RAID *.txt

file1.txt:830 ← file1.txt 文件中有 830 行不包含此字符串

file2.txt:119 ← file2.txt 文件中有 119 行不包含此字符串

⑪ 对当前目录下包含所有子目录的文件内容寻找句子中包含 "RAID" 字符串的文件:


rgrep -r RAID .

⑫ 对当前目录下包含所有子目录, 所有扩展文件名为.txt的文件内容寻找句子中包含 "RAID" 字符串的文件:

rgrep -R '*.txt' RAID .

只处理符合此条件的文件

rlogin(remote login)

常用指数: 

功能说明: 远程登录。

语 法: rlogin [-8EL] [-e <脱离字符>] [-l <用户名称>] [主机名称
或 IP 地址]

包 名 称: rsh。

相关命令: rcp, rsh。

补充说明: 执行rlogin命令打开终端阶段操作, 并登录远程主机。

参 数:

-8	允许输入8位字符数据
-e <脱离字符>	设定脱离 (Escape) 字符。默认值为“~”, 可通过此项参数更改
-E	滤除脱离字符。搭配参数“-8”使用时, 本参数将提供完整透明 (Transparent) 模式的连接
-l <用户名称>	指定要登录远程主机的用户名称。假设没有使用这项参数, 则 rlogin 会询问用户欲使用的登录账号
-L	使用 litout 模式进行远程登录阶段操作

范 例:

① 以当前账号登录到linux.flag.com.tw主机:

```
# rlogin linux.flag.com.tw
```

```
Password: ← 输入账号密码
```

```
Last login: Sun Jul 13 15:26:47 from 203.74.205.36
```

```
#
```

② 以irchen账号登录到linux.flag.com.tw主机:

```
# rlogin -l irchen linux.flag.com.tw
```

R

rm(remove)

常用指数:



功能说明: 删除文件或目录。

语 法: `rm [-dfirv] [--help] [--version] [文件或目录...]`

包 名 称: coreutils。

相关命令: `mdel`。

补充说明: 执行`rm`命令可删除文件或目录，如欲删除目录，必须加上“-r”参数，否则默认仅会删除文件。

参 数:

-d 或 --directory	直接把欲删除的目录的硬链接数目删成0，移除该目录。本方法仅适用于特殊时机，它将造成该目录下的文件失去连接，导致必须执行 <code>fsck</code> 命令检查磁盘
-f 或 --force	强制删除文件或目录。本参数将会忽略放在它前面的“-i”参数
-i 或 --interactive	删除已有文件或目录之前先询问用户。本参数将会忽略放在它前面的“-f”参数
-r、-R或--recursive	递归处理，将指定目录下的所有文件及子目录一并处理
-v 或 --verbose	显示命令执行过程
--help	显示帮助
--version	显示版本信息

范 例:

① 删除当前目录中所有文件和子目录:

```
# rm -r *
```



② 使用互动模式删除当前目录中所有扩展文件名为.txt的文件，并显示命令执行过程：

```
# rm -iv *.txt
```

```
rm: 是否移除普通文件 'aa.txt' ? n ← 删除文件之前先询问用户，回答
rm: 是否移除普通文件 'ab.txt' ? y   "y" 或 "Y" 就删除
已移除 'ab.txt'                       该文件，不想删除则回答 "n"
                                         或 "N"，也可随时按
```

```
rm: 是否移除普通文件 'ac.txt' ?      █ + █ 中断删除文件
...
```

③ 当两参数的功能相互冲突时，后一个参数的效果会盖过前一个：

```
# rm -if *.txt 在参数 "-i" 之后加上 "-f" 参数，将导致参数 "-i" 失效
```

rmmdir(remove directory)

常用指数：

功能说明：删除目录。

语法：rmmdir [-p] [--help] [--ignore-fail-on-non-empty] [--verbose] [--version] [目录...]

包名称：coreutils。

相关命令：mdeltree。

补充说明：当有空目录要删除时，可使用rmmdir命令。若所给予的目录非空目录，则会出现错误信息。

参数：

-p 或 --parents	删除指定目录之后，若该目录的上层目录已变成空目录，则将其一并删除
--help	显示帮助
--ignore-fail-on-non-empty	忽略非空目录的错误信息
--verbose	显示命令执行过程
--version	显示版本信息

R

范 例:

① 删除genuine空目录:

```
# rmdir genuine
```

② 在目录parent_dir中除了子目录child_dir之外别无他物, 若chile_dir为空目录, 我们可让rmdir命令在移除空目录child_dir时, 自动将已经变成空目录的parent_dir一并删除:

```
# rmdir -p --verbose parent_dir/child_dir
rmdir: removing directory, parent_dir/child_dir
rmdir: removing directory, parent_dir
```

←
执行命令时不可
在parent_dir
目录内, 否则将
无法移除该目录

rmmod (remove module)

常用指数:



功能说明: 删除模块。

语 法: rmmod [-as] [模块名称...]

包 名 称: modutils。

相关命令: depmod, insmod, lsmod, modinfo, modprobe。

补充说明: 执行rmmod命令可删除不要的模块。Linux操作系统的内核具有模块化的特性, 因此在编译内核时, 无须把全部的功能统统放入内核。可以将这些功能编译成一个个单独的模块, 待有需要时再分别载入。

参 数:

-a	删除所有当前不需要的模块
-s	把信息输出至 syslog 常驻服务程序, 而非终端界面

范 例:**① 删除系统中不必要的模块:**

```
# lsmod                                     ← 查询已载入的模块名称
Module      Size  Used  by Not tainted
parport_pc  19076  1      (autoclean)
lp          8996   0      (autoclean)
parport     37056  1      (autoclean) [parport_pc lp]
autofs     13268  0      (autoclean) (unused)
8139too    18088  1
mii        3976   0      [8139too]
microcode  4668   0      (autoclean)
...
# rmmod lp   ← 删除 lp 模块
```

② 删除当前不使用的模块:

```
# rmmod -a
```

rmt(remote magnetic tape)

常用指数:



功能说明: 远程磁带传输协议模块。

语 法: rmt

包 名 称: rmt。

补充说明: 使用rmt命令, 用户可通过IPC (InterProcess Communication) 连接, 远程操控磁带机的输出和还原操作。

参 数: 无。

R

范 例:

利用rmt命令操控远程的磁带机:

```
# rmt
```

route

常用指数: 

功能说明: 管理与显示路由表。

语 法: route [-CeFhmvV] 或 route [-v] add [-net 或 -host] 目的地
地址 [netmask <网络掩码>] 或 [gw <网关地址>] 或
route [-v] del [-net 或 -host] 目的地 [gw <网关地址>]
或 [netmask <网络掩码>]

补充说明: 使用此命令可动态改变内核中的路由表 (routing table), 例如可以指定某台主机或是某个网段的封包该由哪个网络界面出去。

参 数:

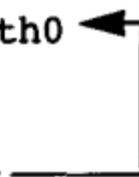
add	新增一项路由记录
del	删除一项路由记录
gw <网关地址>	指定网关的 IP 地址
netmask <网络掩码>	指定主机或网段的网络掩码
- C	列出内核中的路由缓存
- e	以 netstat 命令的排版格式显示路由表
- F	访问内核里的 FIB (Forwarding Information Base) 路由表
-h 或 --help	显示简易的帮助文件
-n	以数字模式显示路由表
-v	以详细模式显示
-V 或 --version	查询当前的版本
-host	指定目的地址为主机
-net	指定目的地址为网段

范 例:

① 列出系统内核的路由表:

```
# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.0.0 * 255.255.255.0 U 0 0 0 eth0
169.254.0.0 * 255.255.0.0 U 0 0 0 eth0
127.0.0.0 * 255.0.0.0 U 0 0 0 lo
default 192.168.0.3 0.0.0.0 UG 0 0 0 eth0
```

当前的网关



② 删除路由表中的默认网关:

```
# route del -net 0.0.0.0 gw 192.168.0.3 ← 删除默认网关
```

```
# route ← 列出路由表
```

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.0.0 * 255.255.255.0 U 0 0 0 eth0
169.254.0.0 * 255.255.0.0 U 0 0 0 eth0
127.0.0.0 * 255.0.0.0 U 0 0 0 lo
```

← 原本的网关已经被删除了

③ 增加新的网关到路由表中:

```
# route add -net 0.0.0.0 gw 192.168.0.1 ← 新增一条网关记录
```

```
# route ← 检查路由表
```

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.0.0 * 255.255.255.0 U 0 0 0 eth0
169.254.0.0 * 255.255.0.0 U 0 0 0 eth0
127.0.0.0 * 255.0.0.0 U 0 0 0 lo
default 192.168.0.1 0.0.0.0 UG 0 0 0 eth0
```

网关已新增, IP 地址为 192.168.0.1



rpm

常用指数: 

(Red Hat package manager)

功能说明: 管理包 (Package)。

语 法: rpm [-acdhilqRsvV] [-e <包文件>] [-f <文件>+] [-F <包文件>] [-i <包文件>] [-p <包文件>+] [-U <包文件>] [-vv] [--addsign <包文件>+] [--allfiles] [--allmatches] [--badreloc] [--changelog] [--checksig <包文件>+] [--clean] [--dbpath <数据库目录>] [--dump] [--excludedocs] [--excludepath <排除目录>] [--force] [--ftp proxy <主机名称或 IP 地址>] [--ftpport <通信端口>] [--help] [--http proxy <主机名称或 IP 地址>] [--httpport <通信端口>] [--ignorearch] [--ignoreeos] [--ignoresize] [--includedocs] [--initdb] [--justdb] [--nodeps] [--nofiles] [--nomd5] [--noorder] [--noscripts] [--notriggers] [--oldpackage] [--percent] [--pipe <执行命令>] [--prefix <目的目录>] [--provides] [--queryformat <文件头格式>] [--querytags] [--rcfile <设定文件>] [--rebuildidb] [--recompile <包文件>] [--relocate <原目录>=<新目录>] [--replacefiles] [--replacepkgs] [--requires] [--resign <包文件>+] [--rmsource] [--rmsource <文件>] [--root <根目录>] [--scripts] [--setperms] [--setugids] [--shortcircuit] [--test] [--triggeredby <包文件>] [--triggers] [--verify] [--version] [--whatprovides <功能特性>] [--whatrequires <功能特性>]

包 名 称: rpm。

补充说明： rpm (Red Hat Package Manager) 原本是 Red Hat Linux 发行版专门用来管理Linux各项包的程序，由于它遵循GPL规则且功能强大方便，因而广受欢迎，逐渐被其他发行版采用。RPM包管理方式的出现，让Linux的软件易于安装、升级，间接提升了Linux的适用度。

参 数：

-a	查询所有的包
-c	列出配置设定文件
-d	列出文件
-e<包文件>或--erase <包文件>	删除指定的包
-f <文件>+	查询拥有指定文件的包
-F <包文件>	更新指定的包文件。在此模式下，会检查所指定的包是否比系统中原有的包来得新。如果属实，rpm 命令会自动更新指定的包。反之，则不更新包
-h 或 --hash	包安装时列出标记
-i	显示包的相关信息
-i<包文件>或--install<包文件>	安装指定的包文件
-l	显示包的文件列表
-p <包文件>+	查询指定的 RPM 包文件
-q	使用询问模式
-R	显示包的关联性信息
-s	显示文件状态
-U<包文件>或--upgrade <包文件>	升级指定的包文件
-v	显示命令执行过程
-V	检查文件正确性

-vv	详细显示命令执行过程, 以便于排错
--addsign <包文件>+	在指定的包里加上新的签名认证
--allfiles	安装所有文件
--allmatches	移除符合指定的包所包含的文件
--badreloc	发生错误时, 重新配置文件
--changelog	显示包的异动记录
--checksig <包文件>+	检验该包的签名认证
--clean	完成包的包装后, 移除包装过程中所建立的目录
--dbpath <数据库目录>	设定欲存放 RPM 数据库的目录
--dump	显示每个文件的验证信息。此参数配合“-l”参数使用
--excludedocs	安装包时, 不要安装文件
--excludepath <排除目录>	忽略在指定目录里的所有文件
--force	强行替换包或文件
--ftp proxy<主机名称或 IP> 地址	指定 FTP 代理服务器
--ftpport <通信端口>	设定 FTP 服务器或代理服务器使用的通信端口
--help	显示帮助
--http proxy<主机名称或 IP>地址	指定 HTTP 代理服务器
--httpport <通信端口>	设定 HTTP 服务器或代理服务器使用的通信端口
--ignorearch	不验证包文件的结构正确性
--ignoreos	不验证包文件的操作系统
--ignoresize	安装前不检查磁盘空间是否足够
--includedocs	安装包时, 一并安装文件
--initdb	确认有正确的数据库可以使用
--justdb	更新数据库, 但不变动任何文件

--nodeps	不验证包文件的相互关联性
--nofiles	不验证文件的属性
--nomd5	不使用 MD5 编码演算确认文件的大小与正确性
--noorder	不重新编排包的安装顺序, 以便满足其彼此间的关联性
--noscripts	不执行任何安装 script 文件
--notriggers	不执行该包包装内的任何 script 文件
--oldpackage	升级成旧版本的包
--percent	安装包时显示完成进度的百分比
--pipe <执行命令>	建立管道, 把输出结果转为该执行命令的输入数据
--prefix <目的目录>	若重新配置文件, 就把文件放到指定的目录下
--provides	查询该包所提供的相容度
--queryformat <文件头格式>	设定文件头的表示方法。文件头格式分为“name”、“version”和“description”3种, 本参数需配合“-i”参数使用
--querytags	列出可用于文件头格式的标签
--rcfile <设定文件>	使用指定的设定文件。默认的设定文件位于 /etc 或用户专属目录里, 文件名称为 rpmrc 或 .rpmrc, 可用此参数改变它
--rebulddb	以现有的数据库为主, 重建一份数据库
--relocate<原目录>=<新目录>	把本来会放到原目录下的文件改放至新目录
--replacefiles	强行替换文件
--replacepkgs	强行替换包
--requires	查询该包所需要的相容度
--resign <包文件>+	移除现有认证, 重新产生签名认证
--rmsource	完成包的包装后, 移除原始码
--rmsource <文件>	移除原始码和指定的文件

<code>--root <根目录></code>	设定欲当作根目录的目录
<code>--scripts</code>	列出安装包的Script的变数
<code>--setperms</code>	设定文件的权限
<code>--setugids</code>	设定文件的拥有者和所属组
<code>--short-circuit</code>	直接略过指定完成阶段的步骤
<code>--test</code>	仅作测试, 并不真正安装包
<code>--triggeredby <包文件></code>	查询该包的包装者
<code>--triggers</code>	展示包文件内的包装 Script
<code>--verify</code>	此参数的效果和指定“-q”参数相同
<code>--version</code>	显示版本信息
<code>--whatprovides <功能特性></code>	查询该包对指定的功能特性所提供的相容度
<code>--whatrequires<功能特性></code>	查询该包对指定的功能特性所需要的相容度

范 例:

① 将 `kxicq-0.3.1-1.i386.rpm` 包安装到系统上:

```
# rpm -i kxicq-0.3.1-1.i386.rpm
warning: kxicq-0.3.1-1.i386.rpm: V3 RSA/MD5 signature:
          NOKEY, key ID 67ad71d5
          该包不含签章,可忽略此警告信息
```

② 将 `kxicq-0.3.1-1.i386.rpm` 包安装到系统上, 执行时显示命令执行过程:

```
# rpm -iv kxicq-0.3.1-1.i386.rpm
warning: kxicq-0.3.1-1.i386.rpm: V3 RSA/MD5 signature:
          NOKEY, key ID 67ad71d5
Preparing packages for installation...
```

③ 将 kxicq-0.3.1-1.i386.rpm 包安装到系统上, 执行时显示更详细的命令执行过程:

```
# rpm -ivv kxicq-0.3.1-1.i386.rpm
D: ===== kxicq-0.3.1-1.i386.rpm
D: Expected size: 223791 = lead(96)+sigs(268)+pad(4)+data(223423)
...
```

④ 将 kxicq 包安装到系统上, 执行时显示命令执行过程及安装过程:

```
# rpm -ivh kxicq-0.3.1-1.i386.rpm
warning: kxicq-0.3.1-1.i386.rpm: V3 RSA/MD5 signature:
        NOKEY, key ID 67ad71d5
Preparing... ##### [100%]
```

会以 “#” 表示安装过程

⑤ 将 kxicq-0.3.1-1.i386.rpm 包安装到系统上, 安装时并显示进度的百分比:

```
# rpm -i --percent kxicq-0.3.1-1.i386.rpm
warning: kxicq-0.3.1-1.i386.rpm: V3 RSA/MD5 signature:
        NOKEY, key ID 67ad71d5

%% 0.000000
%% 13.190522
...
%% 99.974488
%% 100.000000
```

显示安装的进度

⑥ 将 kxicq-0.3.1-1.i386.rpm 包安装到系统上, 若此包已经安装过, 则强行替换包:

```
# rpm -ivh --replacepks kxicq-0.3.1-1.i386.rpm
```

⑦ 将 `kxicq-0.3.1-1.i386.rpm` 包安装到系统上, 若此包的某些文件已经安装过, 则强行替换文件:

```
# rpm -ivh --replacefiles kxicq-0.3.1-1.i386.rpm
```

⑧ 以模拟方式安装 `kxicq-0.3.1-1.i386.rpm` 包, 但不实际安装包:

```
# rpm -i --test kxicq-0.3.1-1.i386.rpm
```

⑨ 升级 `kxicq-0.3.1-1.i386.rpm` 包:

```
# rpm -U kxicq-0.3.1-1.i386.rpm
```

⑩ 升级 `kxicq-0.3.1-1.i386.rpm` 包, 执行时显示命令执行过程及安装过程:

```
# rpm -Uvh kxicq-0.3.1-1.i386.rpm
```

```
kxicq #####
```

↑
会出现“#”字号

⑪ 升级 `kxicq-0.3.1-1.i386.rpm` 包, 升级时不考虑版本的新旧:

```
# rpm -Uvh --oldpackage kxicq-0.3.1-1.i386.rpm
```

⑫ 更新 `kxicq-0.3.1-1.i386.rpm` 包, 执行时显示命令执行过程及安装过程:

```
# rpm -Fvh kxicq-0.3.1-1.i386.rpm
```



⑬ 从系统中移除 kxicq 包:

```
# rpm -e kxicq
```

↑
当移除包时, 不需完整的文件名称

⑭ 从系统中移除 kxicq 包, 移除时不考虑包间的相依性, 即强制移除包:

```
# rpm -e --nodeps kxicq
```

⑮ 检查 kxicq 包的正确性:

```
# rpm -V kxicq
```

⑯ 检查所有已经安装包的正确性:

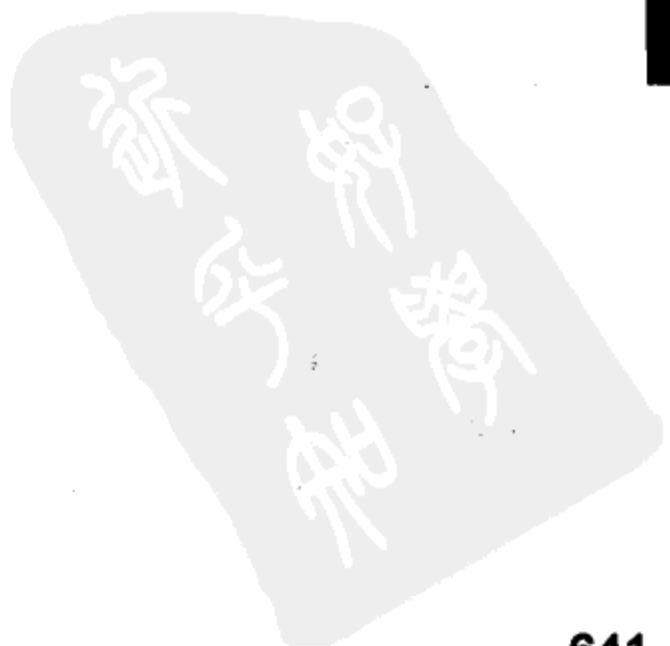
```
# rpm -Va  
S.5....T c /etc/hotplug/usb.usermap  
S.5....T c /etc/sysconfig/pcmcia  
...
```

⑰ 检查包含 /bin/vi 文件的包的正确性:

```
# rpm -Vf /bin/vi
```

⑱ 查询 kxicq 包的名称与版本:

```
# rpm -q kxicq  
kxicq-0.3.1-1
```



⑨ 显示 audit-0.5-1.i386.rpm 包的相关信息:

```
# rpm -qip audit-0.5-1.i386.rpm
warning: audit-0.5-1.i386.rpm: V3 DSA signature: NOKEY,
key ID 4f2a6fd2
Name       : audit           Relocations: (not relocatable)
Version    : 0.5             Vendor: Red Hat, Inc.
Release    : 1              Build Date: 公元2004年09月02日 (周四)
                                01时47分47秒
Install Date: (not installed) Build Host: tweety.build.redhat.com
Group      : system         Source RPM: audit-0.5-1.src.rpm
Size       : 29928          License: GPL
Signature  : DSA/SHA1, 公元2004年10月21日 (周四) 02时51分55秒,
                                Key ID b44269d04f2a6fd2
Packager   : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL        : http://people.redhat.com/faith/audit/
Summary    : Audit user space tools for 2.6 kernel auditing.
Description :

Author: Rik Faith
Maintainer: Charlie Bennett (ccb@redhat.com)
```

The audit package contains the user space utilities for storing and processing the audit records generate by the audit subsystem in the Linux 2.6 kernel.

⑩ 列出 kxicq-0.3.1-1.i386.rpm 包所包含的程序:

```
# rpm -qlp kxicq-0.3.1-1.i386.rpm
warning: kxicq-0.3.1-1.i386.rpm: V3 RSA/MD5 signature:
                                NOKEY, key ID 67ad71d5
/opt/kde/bin/kxicq
/opt/kde/share/applnk/Internet/KXicq.kdelnk
/opt/kde/share/apps/kxicq
...
```

⑪ 查询 kxicq-0.3.1-1.i386.rpm 包的相依性:

```
# rpm -qpR kxicq-0.3.1-1.i386.rpm
warning: kxicq-0.3.1-1.i386.rpm: V3 RSA/MD5 signature:
                                NOKEY, key ID 67ad71d5
```

R


```

kdelibs >= 1.1
ld-linux.so.2
libX11.so.6
libXext.so.6
...

```

与这些文件具有相依性

② 查询kxicq包当前文件的状态:

```

# rpm -qs kxicq
normal          /opt/kde/bin/kxicq
normal          /opt/kde/share/applnk/Internet/KXicq.kdelnk
normal          /opt/kde/share/apps/kxicq
...

```

③ 列出ppp包的所有设定文件:

```

# rpm -qc ppp
/etc/pam.d/ppp
/etc/ppp/chap-secrets
/etc/ppp/options
/etc/ppp/pap-secrets

```

④ 列出kxicq包的说明文件有哪些:

```

# rpm -qd kxicq
/usr/doc/kxicq-0.3.1/AUTHORS
/usr/doc/kxicq-0.3.1/COPYING
/usr/doc/kxicq-0.3.1/ChangeLog
...

```

⑤ 查询/usr/bin/ispell文件所属的说明文件有哪些:

```

# rpm -qdf /usr/bin/ispell
/usr/share/doc/aspell-0.33.7.1/README
/usr/share/doc/aspell-0.33.7.1/TODO
/usr/share/doc/aspell-0.33.7.1/man-html/1_
Introduction.html
...

```

R



⑳ 查询系统中所有已安装的RPM包:

```
# rpm -qa  
bzip2-libs-1.0.2-8  
glib-1.2.10-10  
losetup-2.11y-9  
...
```

㉑ 查询系统所有安装RPM包的文件列表:

```
# rpm -qal  
/usr/lib/libbz2.so.1  
/usr/lib/libbz2.so.1.0.2  
/usr/lib/libglib-1.2.so.0  
...
```

㉒ 查询/bin/chgrp文件所属的包:

```
# rpm -qf /bin/chgrp  
coreutils-4.5.3-19
```

㉓ 从FTP站直接安装rxvt-2.7.8-6.i386.rpm包:

```
# rpm -ivh ftp://rpmfind.net/linux/PLD/current/dists/ra/  
PLD/i386/PLD/RPMS/rxvt-2.7.8-6.i386.rpm
```

㉔ 从FTP站上查询 rxvt-2.7.8-6.i386.rpm包所包含的程序:

```
# rpm -qlp ftp://rpmfind.net/linux/PLD/current/dists/ra/  
PLD/i386/PLD/RPMS/rxvt-2.7.8-6.i386.rpm
```



rsh(remote shell)

常用指数: 

功能说明: 远程登录的shell。

语 法: rsh [-dn] [-l <用户名称>] [主机名称或 IP 地址] [执行命令]

包 名 称: rsh。

相关命令: rcp, rlogin。

补充说明: rsh提供用户环境,也就是shell,以便命令能够在指定的远程主机上执行。

参 数:

-d	使用socket层级的排错功能
-l <用户名称>	指定要登录远程主机的用户名称。假设没有使用这项参数,则rsh会使用现在的账号登录远程主机
-n	把输入的命令导向代号为/dev/null的特殊周边设备

范 例:

① 以当前账号登录到 linux.flag.com.tw 主机:

```
# rsh linux.flag.com.tw
```

```
Password: ← 输入账号的密码
```

```
Last login: Sun Jul 13 15:08:05 from 203.74.205.36
```

```
#
```

② 以irchen账号登录到linux.flag.com.tw主机：

```
# rsh -l irchen linux.flag.com.tw
```

③ 利用rsh登录到linux.flag.com.tw主机，并执行“cat text”命令：

```
# rsh linux.flag.com.tw cat text
```

在远程主机执行的命令

④ 利用rsh登录到linux.flag.com.tw主机，执行ls命令，并将结果存在本地端的local_file文件：

```
# rsh linux.flag.com.tw ls >> local_file
```

↑
存储在本地端

⑤ 利用rsh登录到linux.flag.com.tw主机，执行ls命令，并将结果存在该主机的remote_file文件：

```
# rsh linux.flag.com.tw ls ">>" remote_file
```

↑
加上双引号，则将文件存在远程主机

runlevel

常用指数：



功能说明：显示当前系统的执行等级。

语 法：runlevel [登录记录文件]

包 名 称：SysVinit。

相关命令: chkconfig, ntsysv, telinit。

补充说明: runlevel 命令会读取登录记录文件, 然后显示前一次跟当前的执行等级 (runlevel), 两者之间以一个空格隔开。假如没有读到前一次执行等级的记录, 则以“N”显示。Linux操作系统将执行等级分成以下7种:

执行等级	执行的状态
0	关机 (请不要切换到此等级)
1	单人用户模式的文字界面
2	多人用户模式的文字界面, 但不具有网络文件系统 (NFS) 功能
3	多人用户模式的文字界面, 具有全部的网络功能
4	某些发行版的Linux使用此等级进入 X Window
5	某些发行版的Linux则使用此等级进入 X Window
6	重新开机

参 数:

登录记录文件	指定要读取的登录记录文件, 默认会读取 /var/run/utmp 记录文件
--------	--

范 例:

显示当前的执行等级, 切换到 3 的执行等级后, 再显示一次当前的执行等级:

```
# runlevel
N 5
↑ ↑
| |
| | 当前的执行等级
| | 前一次执行等级
| |
| |
# telinit 3 ← 切换到执行等级 3
# runlevel
5 3
```

R

samba

常用指数:



功能说明: Samba服务器控制。

语法: samba [start] [stop] [status] [restart]

包名称: samba-server (Mandrake Linux)。

相关命令: smbclient, smbd, testparm。

补充说明: Red Hat Linux/Fedora Core已无此命令，以下各项操作使用Mandrake Linux为例。samba为script文件，可启动、停止Samba服务器或返回当前的状态。

Samba服务器是一组应用程序包，将SMB (Server Message Block) 协议移植到Linux上，使Linux系统能整合到微软的网络系统中。启动Linux的Samba服务器后，网络上其他Windows95/98/NT的系统即可存取Linux系统的文件与打印机，而Linux系统也能存取Windows所共享出来的资源。

samba命令实际上会调用smbd (Samba服务器程序) 与nmbd (NetBIOS名称解析服务器程序)，来执行相关的命令。

参数:

start	启动 Samba 服务器的服务
stop	停止 Samba 服务器的服务
status	显示 Samba 服务器当前的状态
restart	重新启动 Samba 服务器

范 例:

① 启动Samba服务器的服务:

```
# samba start
```

```
Starting SMB services: [ OK ]
```

```
Starting NMB services: [ OK ]
```

② 显示Samba服务器当前的状态:

```
# samba status
```

```
smbd (pid 792) is running...
```

```
nmbd (pid 802) is running...
```

← 正在执行中

③ 重新启动Samba服务器:

```
# samba restart
```

```
Shutting down SMB services: [ OK ]
```

```
Shutting down NMB services: [ OK ]
```

```
Starting SMB services: [ OK ]
```

```
Starting NMB services: [ OK ]
```

④ 停止Samba服务器的服务:

```
# samba stop
```

```
Shutting down SMB services: [ OK ]
```

```
Shutting down NMB services: [ OK ]
```



screen

常用指数:



功能说明: 多重窗口管理程序。

语 法: screen [-AmRvx] [-d <操作名称>] [-h <行数>] [-ls] [-r <操作名称>] [-s <shell>] [-S <操作名称>] [-wipe]

包 名 称: screen。

补充说明: screen为多重窗口管理程序。此处所谓的窗口，是指一个全屏幕的文本模式画面。通常只有在使用telnet登录主机，或是使用老式的终端时，才有可能用到screen程序。

执行screen命令时，会启动screen操作，screen默认的操作名称为“<pid>.<终端>.<主机名称>”。screen操作刚开始会与启动它的终端连接（attach）。用户可将screen操作自终端上离线（detach），然后再从其他终端上恢复离线的screen操作。

参 数:

-A	将所有的窗口都调整为当前终端的大小
-d <操作名称>	将指定的screen操作离线
-h <行数>	指定窗口的缓冲区行数
-m	即使当前已在操作中的screen操作，仍强制建立新的screen操作
-r <操作名称>	恢复离线的screen操作
-R	先试图恢复离线的操作。若找不到离线的操作，即建立新screen操作
-s <shell>	指定建立新窗口时，所要执行的shell

-S <操作名称>	指定新建立 screen 操作的名称, 此名称将会取默认操作名称中 <终端>.<主机名称> 的位置
-v	显示版本信息
-x	恢复离线的 screen 操作
-ls 或 --list	显示当前所有的 screen 操作
-wipe	检查当前所有的screen操作, 并删除已经无法使用的screen操作 (标示为 "dead")

范 例:

① 执行screen命令, 可利用稍后所介绍的操作命令来控制窗口:

```
# screen
```

② 显示当前所有的screen操作:

```
# screen -ls
```

```
There is a screen on:
```

```
929.pts-1.irchen (Attached)
```

```
1 Socket in /tmp/screen/S-root. ← 在Mandrake Linux中, 此行为“1 Socket in /root/tmp.”
```

③ 当前只有一个连接中的screen操作时, 将此screen操作离线:

```
# screen -d ← 将连接中的screen操作离线
```

```
[remote detached]
```

```
# screen -ls ← 显示当前所有的screen操作
```

There is a screen on:

929.pts-1.irchen

(Detached)

↑ “Detached” 表示已经
离线的screen操作

1 Socket in /tmp/screens/S-root.

④ 将离线的screen操作恢复连接:

screen -r ← 连接原本离线的screen操作

screen -ls ← 显示当前所有的screen操作

There is a screen on:

929.pts-1.irchen

(Attached)

↑ 此screen操作已经恢
复连接

1 Socket in /tmp/screens/S-root.

而执行以下命令也有同样的结果:

screen -x

⑤ 建立新的screen操作:

screen -m ← 建立新的screen操作

screen -ls ← 显示当前所有的screen操作

There are screens on:

929.pts-1.irchen

(Attached)

951.pts-2.irchen

(Attached)

← 新建立的screen
操作

2 Sockets in /tmp/screens/S-root.

⑥ 将指定的screen操作离线:

```
# screen -d 951.pts-2.irchen
```

将此screen操作离线

```
# screen -ls ← 显示当前所有的screen操作
```

```
There are screens on:
```

```
929.pts-1.irchen (Attached)
```

```
951.pts-2.irchen (Detached) ← 此screen操作  
已经离线
```

```
2 Sockets /tmp/screens/S-root.
```

而执行以下 3 种方式也有同样的结果:

```
# screen -d 951
```

↑ 指定执行程序的识别码

```
# screen -d pts-2
```

↑ 指定终端

```
# screen -d pts-2.irchen
```

指定终端和主机名称

⑦ 将指定的 screen 操作恢复连接:

```
# screen -r 951.pts-2.irchen
```

将恢复此screen操作

```
# screen -ls ← 显示当前所有的screen操作
```

```
There are screens on:
```

```
929.pts-1.irchen (Attached)
```

```
951.pts-2.irchen (Attached) ← 此screen操作已经  
恢复连接
```

```
2 Sockets /tmp/screens/S-root.
```

而执行以下3种方式也有同样的结果：

```
# screen -r 951
```

↑ 只指定执行程序的识别码

```
# screen -r pts-2
```

↑ 只指定终端

```
# screen -r pts-2.irchen
```

指定终端和主机名称

⑧ 建立新的screen操作,并特别指定操作名称:

```
# screen -S myscreen ← 指定操作名称为"myscreen"
```

```
# screen -ls ← 显示当前所有的screen操作
```

```
There is a screen on:
```

```
1038.myscreen (Attached)
```

└─ 新建立的作业名称

```
1 Socket /tmp/screens/S-root.
```

⑨ 尝试恢复离线的screen操作,若找不到离线的操作,则建立新的screen操作:

```
# screen -R
```

⑩ 建立新的screen操作,并指定使用tcsh为所执行的shell:

```
# screen -s tcsh
```

⑪ 删除标示为“dead”的screen操作:

```
# screen -wipe
```

操作命令：执行screen后，外观与一般shell下的画面相同，但按下特定的组合键可进行screen的操作。以下说明常用的操作命令（**Ctrl + A C**键表示先同时按**Ctrl**键与**A**键，然后再按**C**键，请注意第二个按键有大小写的区别）：

Ctrl + A	显示选择窗口的提示。出现提示后，请按数字键，选择要切换的窗口
Ctrl + A	进入指定的窗口
Ctrl + A Ctrl + P	切换至上一个窗口
Ctrl + A Ctrl + N	切换至下一个窗口
Ctrl + A Ctrl + A	切换至前一个窗口
Ctrl + A Ctrl + C	建立新窗口，并在新窗口中执行shell，然后切换至新建的窗口
Ctrl + A C	建立新窗口
Ctrl + A C	清除当前窗口的内容
Ctrl + A Ctrl + d	将当前的screen操作与所在的终端分离（detach）。用户可在其他终端重新登录，再执行“screen -x”命令，即可恢复离线的screen操作
Ctrl + A K	删除当前的窗口
Ctrl + A i	显示当前窗口的信息
Ctrl + A N	显示当前窗口的编号
Ctrl + A h	存储当前窗口的内容成“screenlog.n”，其中“n”为当前窗口的编号
Ctrl + A Ctrl + t	显示系统当前的时间、主机名称及负载情况
Ctrl + A Ctrl + W	显示当前所打开窗口的清单
Ctrl + A X +	锁住当前窗口，需要输入账号密码，才能再执行命令
Ctrl + A m +	显示最近一次显示过的信息
Ctrl + A L	重新显示画面
Ctrl + A V	显示版本信息
Ctrl + ?	显示按键说明

sed (stream editor)

常用指数:



功能说明: 利用 script来处理文本文件。

语法: sed [-hnV] [-e <script>] [-f <script 文件>] [文本文件]

包名称: sed。

相关命令: ed, jed, joe, vi, vim, tr。

补充说明: sed可依照script的命令来处理、编辑文本文件。script命令可直接在命令列中输入，或是写成 script 文件。

参数:

-e <script>或--expression=<script>	以选项中指定的script来处理输入的文本文件
-f <script 文件> 或--file=<script 文件>	以选项中指定的script文件来处理输入的文本文件
-h 或 --help	显示帮助
-n、--quiet 或--silent	仅显示script处理后的结果。若不使用此参数，则原来文本文件中未经script处理的部分也会一起显示
-V 或 --version	显示版本信息

在参数中使用的script语法如下:

[行号] [/查找字符串/命令 <参数>]

而其详细说明如下所示:

[行号]	[行号] 中可指定输入文本文件的行号。若是用[行号, 行号]的格式，则是指定第1个行号与第2个行号之间的每一行。若在 [行号] 后加上惊叹号，则是指定除了[行号]以外的每一行。 若不使用 [行号]，且没有指定 /查找字符串/，则是处理输入文件中的每一行
a<字符串>	在指定的行号后新增一行<字符串>

c<字符串>	以 <字符串> 取代指定的行号
d	删除指定的行号
i<字符串>	与 a 命令类似,但是在指定的行号前新增一行
p	显示指定的行号
r<文本文件>	先处理此处所指定的文本文件,然后再处理命令行中所指定的文本文件
s/<查找字符串> /<取代字符串> /<取代方式>	在指定的行号中寻找 <查找字符串>,再以 <取代字符串> 来取代<查找字符串>。而取代方式共有 3 种: n 取代第 n 个找到的 <查找字符串> g 取代所有找到的 <查找字符串> p 取代后,再显示一次该行
w<文本文件>	在此文本文件填入指定字符
y/<查找字符> /<取代字符>/	以 <取代字符> 来取代所有找到的 <查找字符>。<查找字符> 与 <取代字符> 中都可指定一个以上的字符,但两者长度必须相同

范 例:

① 查找含有“target”字符串的行号,然后在后面新增一行,其内容为“A New Line”:

```

# cat sed_script          ← 显示script文件内容
/target/a\
A New Line
# cat textfile           ← 显示textfile文件内容
This is the 1st line
This is the target line
This is the last line   ← 以sed_script中的命令来
# sed -f sed_script textfile ← 处理textfile
This is the 1st line
This is the target line ← 这一行包含查找的字符串
A New Line              ← 从后面新增一行
This is the last line

```


- ② 将textfile（文件内容与第1个范例相同）第1、2行的“is”字符串取代为12:

```
# sed -e 1,2 s/is/12/g textfile
```

将所有的“is”字符串取代为12
处理第1、2行

```
Th12 12 the 1st l1ne
```

```
Th12 12 the target l1ne
```

```
This is the last line ← 第3行不受影响
```

- ③ 将textfile（文件内容与第1个范例相同）第1、2行的“i”字符取代为1，“s”字符取代为2:

```
# sed -e 1,2y/is/12/ textfile
```

将“i”字符取代为1，“s”取代为2

```
Th12 12 the 12t l1ne
```

```
Th12 12 the target l1ne
```

```
This is the last line
```

- ④ 将textfile（文件内容与第1个范例相同）第2、3行写入newfile文件:

```
# cat sed_script ← 显示script文件内容
```

```
1!w newfile ← 1!表示第1行以外的每一行，在textfile这个文件中就是指第2、3行
```

```
# sed -f sed_script textfile
```

```
# cat newfile ← 显示newfile文件内容
```

```
This is the target line
```

```
This is the last line
```

set

常用指数: **功能说明:** 设定shell。**语 法:** set [+abCdefhHklmnpPtuvx]**包 名 称:** bash。**相关命令:** declare, export, setenv。**补充说明:** set为bash shell中查询或设定环境变量的命令。set命令能依照不同的需求设定所使用shell的执行方式。而set也能查询变量, shell可将变量导出成为环境变量(使用export命令), 让所有的命令都可存取环境变量。若不加任何参数, 则会显示当前所有变量。**参 数:**

-a	标示已修改的变量, 以供输出至环境变量
-b	使被中止的后台程序立刻返回执行情况
-C	转向所产生的文件无法覆写已存在的文件
-d	shell默认会用哈希表(hash table)记忆使用过的命令, 以加速指令的执行。使用“-d”参数可取消此功能
-e	若命令返回值不等于0, 则立即退出shell
-f	取消使用通配字符。shell有3种通配字符: “*”、“?”及 “[abc]”。例如“ab*”表示以ab为开头的字符串。“ab?”则是ab开头, 长度为3个字符的字符串。“ab[a-c]”意指aba、abb及abc等3个字符串
-h	自动记录函数的所在位置
-H	可利用“!”加上<命令编号>的方式来执行history中记录的命令。shell默认会启动此参数
-k	命令所给的参数都会被视为此命令的环境变量
-l	记录for循环的变量名称

-m	使用监视模式。在此模式下，后台执行的程序若执行结束，会显示执行结果
-n	只读取命令，而不实际执行。通常用来测试 shell script 是否正确
-p	启动优先顺序模式
-P	启动“-P”参数后，执行命令时，会以实际的文件或目录来取代符号链接。例如，dirlnk是dirsrc的符号链接，则执行“cd dirlnk”命令时，当前目录会变成dirsrc
-t	执行完随后的命令，即退出 shell
-u	当执行时使用到未定义过的变量，则显示错误信息
-v	显示shell所读取的输入值
-x	执行命令后，会先显示该命令及所下的参数
+<参数>	取消某个set曾启动的参数。此<参数>可为以上介绍过的参数。例如执行set -x后，可执行set +x取消

范 例：

① 查询当前所有的变量：

```

# set          ← 查询当前系统环境变量的情况
BASH=/bin/bash ← bash 文件所在位置
BASH_ENV=/root/.bashrc ← bash 设定文件所在位置
BASH_VERSINFO=([0]="2" [1]="05b" [2]="0" [3]="1" [4]="release"
[5]=" i386-redhat-linux-gnu" ) ← bash 版本信息
BASH_VERSION=' 2.05b.0(1)-release' ← bash 的版本
COLORS=/etc/DIR_COLORS.xterm
COLUMNS=80 ← 纯文本的画面宽度为80列
DIRSTACK=()
EUID=0
GROUPS=()
G_BROKEN_FILENAMES=1
HISTFILE=/root/.bash_history ← 命令历史记录文件所在的位置
HISTFILESIZE=1000 ← 命令历史记录文件最多记录1000行
HISTSIZ=1000 ← 命令历史最多保存 1000 条
HOME=/root ← 用户的根目录

```

```

HOSTNAME=free.flag.com.tw ← 主机名称
HOSTTYPE=i386 ← 电脑类型, i386 表示是 x86 平台
IFS='#' \t\n'
INPUTRC=/etc/inputrc
LAMHELPPFILE=/etc/lam/lam-helpfile
LANG=zh_CN.gb18030 ← 语言设定为中国区 GB18030 码
LESSOPEN='|/usr/bin/lesspipe.sh %s'
LINES=24 ← 纯文本模式画面的高度为 24 行
LOGNAME=root ← 用户账号的名称
LS_COLORS='no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00; 35:bd=40;33;01:cd=40;
33;01:or=01;05;37;41:mi=01;05;37;41:ex=00;32:*.cmd=00;32:*.exe=00;32:*.
com=00;32:*.bat=00;32:*.sh=00;32:*.csh=00;32:*.tar=00;31:*.
tgz=00;31:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z=00;
31:*.gz=00;31:*.bz2=00;31:*.bz=00;31:*.tz=00;31:*.rpm=00;31:*.cpio=00;31:*.
jpg=00;35:*.gif=00;35:*.bmp=00;35:*.xbm=00;35:*.xpm=00;35:*.png=00;35:*.
tif=00;35:'
MACHTYPE=i386-redhat-linux-gnu
MAIL=/var/spool/mail/root ← 用户邮件存放目录
MAILCHECK=60 ← 每60秒就检查一次是否有新的电子邮件
OPTERR=1
OPTIND=1
OSTYPE=linux-gnu ← 操作系统为 Linux
PATH=/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin
:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/
bin:/root/bin ← 执行命令时的查找路径
PIPESTATUS=( [0]=" 0" )
PPID=2349
PROMPT_COMMAND='echo -ne "\033]0;#{USER}@# {HOSTNAME%%.*}
:#{PWD/##HOME/~}\007"'
PS1=' [\u@\h \W]\# ← shell 的提示符号
PS2=' > '
PS4=' + '
PVM_ROOT=/usr/share/pvm3

```

```

PVM_RSH=/usr/bin/rsh
PWD=/root ← 当前用户所在的目录
QTDIR=/usr/lib/qt-3.1
SHELL=/bin/bash ← 当前使用的 shell 是 bash
SHELLOPTS=braceexpand:emacs:hashall:histexpand:history:in
teractive-comments:monitor
SHLVL=1
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
SSH_CLIENT=' 192.168.0.39 3358 22'
SSH_CONNECTION=' 192.168.0.39 3358 192.168.0.139 22'
SSH_TTY=/dev/pts/0
SUPPORTED=zh_TW.Big5:zh_TW:zh
TERM=linux ← 当前使用的终端类型为 linux (文本模式)
UID=0 ← 当前用户的 UID
USER=root ← 当前用户的账号
USERNAME=root ← 当前用户的名称
XPVM_ROOT=/usr/share/pvm3/xpvm
_=-1

```

② 使转向的文件无法覆写已经存在的文件：

```

# ls ← 显示当前目录的内容
file1 file2 ← 目录中有这两个文件
# cat file1 > file2 ← 转向时所产生的文件会覆写现有的文件
# set -C ← 便转向时所产生的文件无法覆写现有的档案
# cat file1 > file2
bash: file2: Cannot clobber existing file ← 无法覆写现有的文件
# set +C ← 恢复原来的设定
# cat file1 > file2 ← 可以覆写现有的文件

```

③ 使shell不接受通配字符:

```
# ls          ← 显示当前目录的内容
file1 file2 file3 ← 此目录当前有3个以file字符串开头的文件
# set -f      ← 取消使用通配字符
# rm file*
rm: cannot remove 'file*': No such file or directory ←
                只会删除文件名称恰为file*的文件, 而不会删除
                以file字符串开头的文件。
```

④ 当执行时使用到未曾定义过的参数, 则显示错误信息:

```
# echo $alpha ← 没有定义过的alpha变量
                ← 不显示任何信息

# set -u
# echo $alpha ← 使用到未定义的alpha变量
bash: alpha: unbound variable ← 出现错误信息
```

⑤ 取消执行history记录的功能:

```
# history 5 ← 列出最近5次所执行的命令
693 echo $alpha
694 ls
695 locate spell
696 history
697 history 5

# !695 ← 执行编号695号的命令
# set +H ← 取消执行history记录的功能
# !695 ← 再一次执行编号695号的功能
bash: !695: command not found ← 出现错误信息
```

⑥ 设定当执行命令时, 会先显示该命令及所下的参数:

```
# set -x ← 启动此功能
# cat shortfile ← 执行任意一个命令
+ cat shortfile ← 会先显示该命令及所执行的参数
this is a test file ← 命令实际执行的结果
```

⑦ 执行命令，以原始文件或目录来取代符号链接：

```
# ls -l
总计 4
drwxr-xr-x  2 root  root  4096  7月 1 17:31 dir
lrwxrwxrwx  1 root  root      3  7月 1 17:32 dir_lnk -> dir ←
```

dir_lnk是指向dir目录的符号链接

```
# cd dir_lnk      ← 切换到此符号链接目录
# pwd            ← 显示当前的目录
/root/dir/dir_lnk
# cd ..         ← 回到父目录
# set -P       ← 执行命令时以原始文件为准
# cd dir_lnk   ← 切换到符号链接的目录
# pwd
/root/dir/dir  ← 已经切换到原来指向的dir目录
```

setenv

常用指数：



(set environment variable)

功能说明： 查询或显示环境变量。

语 法： setenv [变量名称 变量值]

包 名 称： tcsh。

相关命令： set。

补充说明： setenv 为 tcsh shall 中查询或设定环境变量的命令。

若不加上 [变量名称 变量值]，则会显示当前所有的环境变量。

参 数： 无。

范 例:

将HISTSIZE变量设为500:

```
# setenv
HOSTNAME=free.flag.com.tw      ← 主机名称
...
HISTSIZE=1000                  ← HISTSIZE 当前设为 1000
...
USERNAME=root                  ← 用户名称
MAIL=/var/spool/mail/root      ← 用户的邮件目录
PATH=/usr/local/sbin:/usr/sbin:/sbin:/usr/kerberos/sbin:/usr
/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr
/sbin:/usr/bin:/usr/X11R6/bin:/root/bin:/root/bin ←
                                                    执行命令时的查找路径
...
# setenv HISTSIZE 500
# setenv
HOSTNAME=free.flag.com.tw
...
HISTSIZE=500                   ← HISTSIZE已改为 500
...
```

setserial

常用指数:



S

功能说明: 设置或显示串行端口的相关信息。

语 法: setserial [-abgGqvVz] [设备] [串行端口参数]

包 名 称: setserial。

相关命令: statserial。

补充说明: setserial可用来设定串行端口或显示当前的设置。串行端口可供设置的相当多,不过最主要的还是IRQ、I/O及传输速率等参数。

参 数:

-a	显示详细信息
-b	显示摘要信息
-g	显示串行端口的相关信息
-G	以命令列参数的格式来显示信息。即输出结果可直接作为 setserial 的输入参数
-q	执行时显示较少的信息
-v	执行时显示较多的信息
-V	显示版本信息
-z	设置前, 先将所有的标志归零
串行端口参数	
autoconfig	让kernel自动侦测并设置串行端口
auto_irq	让kernel自动侦测并设置 IRQ
^auto_irq	配合“autoconfig”参数一起使用时, 可关闭自动设置 IRQ 的功能
baud_base	设置基本速率。一般所能设的最大值为115 200
callout_nohup	当串行端口为对外传输设备时, 即使已侦测不到载体 (例如, 电话线的连接), 仍然不要挂断 tty
^callout_nohup	当串行端口为对外传输设备时, 若侦测不到载体, 即挂断 tty
close_delay<延迟时间>	当远程数据机关闭后, 本地端数据机维持在 DTR (Data Terminal Ready, 也就是已准备好传输数据) 状态的时间。单位为百分之一秒, 默认值为 50
closing_wait<等待时间>	在关闭接收器 (Receiver) 之前, kernel 等待串行端口传输数据的时间。单位为百分之一秒, 默认值为 3 000
divisor <divisor 值>	自定义传输速率, 必须与“spd_cust”参数一起使用
fourport	将串行端口设为 AST Fourport 卡
^fourport	关闭将串行端口设为 AST Fourport 卡的设置
irq <IRQ 编号>	设置 IRQ
low_latency	减少串行端口接收数据时的延迟时间。如此会提高 CPU 的负荷, 但可增加数据流的稳定性, 适用于需要即时数据的应用程序

pgrp_lockout	锁定传输端口, 不让其他程序组使用
^pgrp_lockout	解除 pgrp_lockout 的锁定状态
port <串行端口编号>	设置串行端口的编号
session_lockout	锁定传输端口, 不让其他程序使用
^session_lockout	解除 session_lockout 的锁定状态
skip_test	配合“autoconfig”参数一起使用时, 可关闭自动侦测 UART的功能
^skip_test	让 kernel 自动侦测并设置 UART
spd_cust	使用自定义的传输速率, 传输速率为 baud_base 除以 divisor
spd_hi	使用57.6 Kbps传输速率
spd_normal	使用38.4 Kbps传输速率
spd_vhi	使用115 Kbps传输速率
split_termios	传送与接收的连接端口使用不同的 termios (Terminal I/O 的设置)
^split_termios	传送与接收的连接端口使用相同的 termios (Terminal I/O 的设置)
uart <UART类型>	设置 UART (Universal Asynchronous Receiver/Transmitter)的类型

范 例:

① 显示第1个串行端口当前的状态:

```
# setserial /dev/ttyS0
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 5
```

② 显示第1个串行端口的摘要信息:

```
# setserial -b /dev/ttyS0
/dev/ttyS0 at 0x03f8 (irq = 5) is a 16550A
```


setup

常用指数:



功能说明: 设置公用程序。

语 法: setup

包 名 称: setuptool (Red Hat Linux/Fedora Core)。

补充说明: 此为Red Hat Linux/Fedora Core才有的命令。setup是一个设置公用程序，提供图形界面的操作方式，共有8个项目可供设置：

1. 登录认证方法
2. 防火墙态设置
3. 鼠标配置设置
4. 网络配置设置
5. 打印机配置设置
6. 开机时所要启动的系统服务
7. 声卡配置设置
8. 时区设置

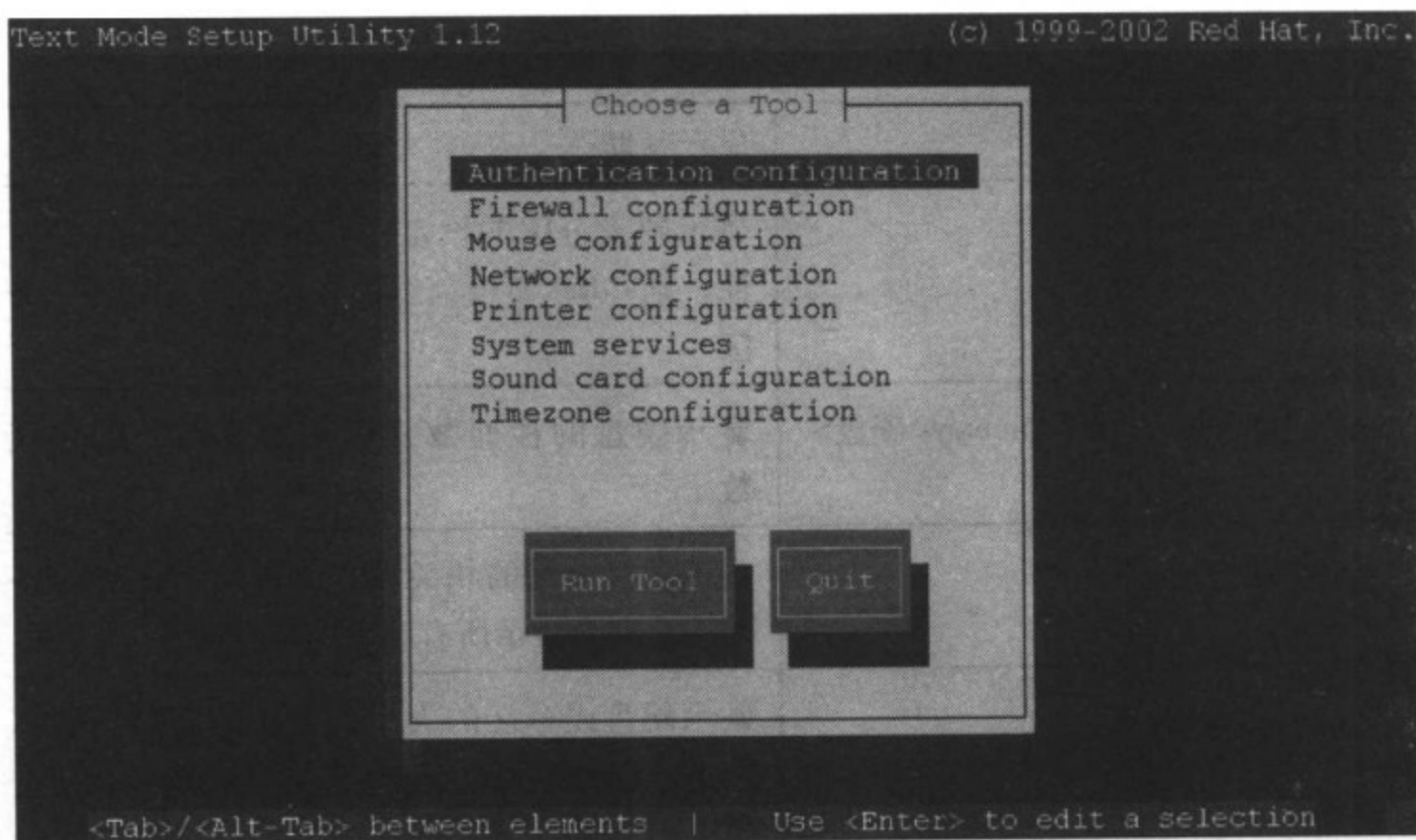
参 数: 无。

范 例:

执行设定程序，可利用 **Tab**、**Alt + Tab** 和 **Enter** 键设置：

```
# setup
```

S



sfdisk

常用指数:



功能说明: 硬盘分区工具程序。

语法: `sfdisk [-?Tvx] [-d <硬盘>] [-g <硬盘>] [-l <硬盘>] [-s <硬盘或分区>] [-V <硬盘>]`

包名称: util-linux。

相关命令: cfdisk, fdisk。

补充说明: sfdisk为硬盘分区工具程序，可显示分区的设定信息，并检查分区是否正常。

sfdisk也可用来建立分区。由于建立硬盘分区兹事体大，建议还是使用为个别操作系统设计的硬盘分区工具程序比较妥当，如用fdisk建立DOS分区；用Disk Druid或sfdisk建立Linux的分区。

参 数:

-? 或 --help	显示帮助
-d <硬盘>	显示硬盘分区的设定。若将显示结果存为文件，则sfdisk便可利用该文件来修复损毁的分区
-g <硬盘>或-show-geometry<硬盘>	显示硬盘的柱面数、磁头数及每轨的扇区数
-l <硬盘>	显示硬盘分区的相关设定。若不指定<硬盘>选项，则会显示所有硬盘上的分区
-s <硬盘或分区>	显示硬盘或分区的大小，单位为区块。若不指定<硬盘或分区>选项，则会显示每个硬盘的大小
-T 或 --list-types	显示所有sfdisk能辨识的文件系统ID
-v 或 --version	显示版本信息
-V <硬盘> 或--verify <硬盘>	检查硬盘分区是否正常
-x 或 --show-extend	显示扩展分区中的逻辑分区

范 例:

① 显示系统中每个硬盘的大小:

```
# sfdisk -s
/dev/hda: 6297480 ← 第1个IDE硬盘的大小
/dev/hdb: 2063880 ← 第2个IDE硬盘的大小
total: 8361360 blocks ← 所有硬盘容量的总和，单位为扇区数，若一个扇区的大小为1024字节，则此系统中所有硬盘的总容量为8361360字节
```

② 显示/dev/hdb硬盘的大小:

```
# sfdisk -s /dev/hdb
2063880
```

③ 显示 /dev/had3 分区的大小:

```
# sfdisk -s /dev/hda3
2562367
```

④ 显示系统中所有硬盘的柱面数、磁头数及每轨的扇区数:

```
# sfdisk -g
/dev/hda: 784 cylinders, 255 heads, 63 sectors/track
/dev/hdb: 1023 cylinders, 64 heads, 63 sectors/track
```

⑤ 显示/dev/hdb硬盘的柱面数、磁头数及每轨的扇区数:

```
# sfdisk -g /dev/hdb
/dev/hdb: 1023 cylinders, 64 heads, 63 sectors/track
```

⑥ 显示系统中所有硬盘分区的相关信息:

```
# sfdisk -l
Disk /dev/hda: 784 cylinders, 255 heads, 63 sectors/track
```

↑
第1个IDE硬盘

Units = cylinders of 8225280 bytes, blocks of 1024 bytes,
counting from 0

Device	Boot	Start	End	#cyls	#blocks	Id	System
/dev/hda1	*	0+	254	255	-2048256	b	Win95 FAT32
/dev/hda2		255	350	96	771120	5	Extended
/dev/hda3		351	669	319	2562367	+ 83	Linux
/dev/hda4		670	682	13	104422	+ 82	Linux swap
/dev/hda5		255+	350	96-	771088	+ b	Win95 FAT32

```
Disk /dev/hdb: 1023 cylinders, 64 heads, 63 sectors/track
```

↑
第2个IDE硬盘

Units = cylinders of 2064384 bytes, blocks of 1024 bytes,
counting from 0

Device	Boot	Start	End	#cyls	#blocks	Id	System
/dev/hdb1		0+	639	640	- 1290208+83		Linux
/dev/hdb2		640	1022	383	772128	b	Win95 FAT32

/dev/hdb3	0	-	0	0	0	0	Empty
/dev/hdb4	0	-	0	0	0	0	Empty

分区名称 是否为可开机分区 此分区的起始柱面数 此分区的结束柱面数 此分区的总柱面数 此分区的大小 文件系统ID 文件系统的类型

⑦ 显示/dev/hdb硬盘分区的相关信息:

```
# sfdisk -l /dev/hdb
```

⑧ 检查系统中所有硬盘的分区是否正常:

```
# sfdisk -v
```

```
/dev/hda: OK
```

```
Warning: no primary partition is marked bootable (active)
```

```
This does not matter for LILO, but the DOS MBR will not boot this disk.
```

```
/dev/hdb: OK
```

由于dev/hdb并没有设定任何可开机分区，故发出警告信息

⑨ 检查 /dev/hdb 硬盘的分区是否正常:

```
# sfdisk -v /dev/hdb
```

⑩ 显示第2个IDE硬盘分区的设定，并将结果存入文件中，以用来修复分区:

```
# sfdisk -d /dev/hdb
```

```
# partition table of /dev/hdb
```

```
unit: sectors
```

```
/dev/hdb1 : start= 63, size= 2580417, Id=83
```



```

/dev/hdb2 : start= 2580480, size= 1544256, Id= b
/dev/hdb3 : start=      0, size=      0, Id= 0
/dev/hdb4 : start=      0, size=      0, Id= 0
# sfdisk -d /dev/hdb > hdb.partition ← 存入hdb.partition
                                         文件中
# sfdisk /dev/hdb < hdb.partition ← 用来修复分区

```

⑩ 显示所有sfdisk能辨识的文件系统ID:

```

# sfdisk -T
Id  Name
  0  Empty
  1  FAT12
...
  b  Win95 FAT32      ← FAT32文件系统的ID是b
  c  Win95 FAT32 (LBA)
  e  Win95 FAT16 (LBA)
...
82  Linux swap      ← Linux swap文件系统的ID是82
83  Linux           ← Linux默认文件系统的ID是83
...

```

shapcfg (shaper configure)

常用指数:



功能说明: 限制网络设备的流量。

语 法: shapcfg attach [流量限制器] [网络设备]或shapcfg speed
[流量限制器] [频宽]

包名称: shapcfg (Red Hat Linux)。

补充说明: 此为Red Hat Linux才有的命令。自Linux 2.1.15开始, 支持流量限制 (Traffic Shaping) 的功能。流量限制能限制系统中每个网络设备对外的传输频宽, 让管理员能进一步控制网络资源的分配。

流量限制在应用时，必须通过流量限制器（Traffic Shaper）来控制实际的网络设备。流量限制器可视为一种网络接口驱动程序，与实际的网络设备结合后，便可控制网络设备的对外传输频宽。

参 数：

attach	将流量限制器与实际的网络设备结合
speed	设定流量限制器的对外传输频宽

范 例：

① 将 shaper0 流量限制器与 eth1 网络卡结合：

```
# shapecfg attach shaper0 eth1
```

② 将 shaper0 流量限制器的对外传输频宽设为 64 000：

```
# shapecfg speed shaper0 64000
```

shutdown

常用指数：

功能说明：系统关机命令。

语 法：shutdown [-cfFhknr] [-t 秒数] [时间] [警告信息]

包 名 称：SysVinit。

相关命令：halt, reboot。

补充说明：shutdown命令可以关闭所有程序，并依用户的需要，进行重新开机或关机的操作。shutdown命令可以指定系统在特定时间关机，但仅限于当日。若需指定在特定日期、时间关机，则须搭配at命令执行（相关说明请参考at命令）。

参 数:

-c	当执行“shutdown -h 11:50”命令时，只要按 Ctrl + Alt 键就可以中断关机的命令。若是执行“shutdown -h 11:50 &”将 shutdown 丢到后台时，则需要使用 shutdown -c 将前一个 shutdown 命令取消
-f	重新启动时不执行 fsck
-F	重新启动时执行 fsck
-h	将系统关机
-k	只是送出信息给所有用户，但不会实际关机
-n	不调用 init 程序进行关机，而由 shutdown 自己进行（一般关机程序是由 shutdown 调用 init 来做关机操作）。不建议用户采取此种关机方式
-r	shutdown 之后重新启动
-t <秒数>	送出警告信息和删除信号之间要延迟多少秒。警告信息可以提醒执行 vi 或 mail 等程序的用户尽快存文件，删除信号会删除 vi 或 mail 等程序
[时间]	设定多久时间后执行 shutdown 命令。时间参数有 hh:mm 或 +m 等两种模式。hh:mm 表示在几点几分执行 shutdown。例如“shutdown 10:45”表示 10:45 执行 shutdown。+m 表示 m 分钟后执行 shutdown。比较特别的用法是以 now 表示立即执行 shutdown
[警告信息]	要传送给所有登录用户的信息

范 例:

① 立即关闭系统:

```
# shutdown -h now
```

```
Broadcast message from root (pts/1) Thu Jul 3 00:55:55
2003): The system is going down for system halt NOW !!
```

② 指定在 22:00 时，将系统重新开机:

```
# shutdown -r 22:00
```

③ 指定5分钟后关机，同时送出警告信息给登录的用户：

```
# shutdown +5 "System will shutdown after 5 minutes"
Broadcast message from root (pts/1) (Thu Jul 3 00:58:40 2003):
System will shutdown after 5 minutes
The system is going DOWN to maintenance mode in 5 minutes !!
```

④ 立即将系统重新开机，当重新启动时指定需执行fsck命令：

```
# shutdown -r -F now
```

⑤ 以后台执行的方式，指定系统在22:00关机，然后再取消此关机命令：

```
# shutdown -h 22:00 &
```

```
[1] 1318
```

```
# ps ← 察看当前执行的程序
```

```
  PID TTY          TIME CMD
```

```
 1259 pts/3        00:00:00 bash
```

```
 1315 pts/3        00:00:00 shutdown
```

← shutdown命令已经在后台执行

```
 1316 pts/3        00:00:00 ps
```

```
# shutdown -c
```

← 取消执行中的shutdown命令

```
Shutdown cancelled.
```

```
[1]+  Done
```

```
shutdown -h 22:00
```

sleep

常用指数：



功能说明：暂停执行命令。

语法：sleep [--help] [--version] [时间]

包名称：coreutils。

补充说明：sleep可暂停一段指定的时间，然后再继续执行命令或工作。

参 数:

<code>--help</code>	显示帮助
<code>--version</code>	显示版本信息
[时间]	[时间] 的格式为整数后再加上单位。单位共有 s (秒)、m (分)、h (时) 与 d (日) 等 4 种, 默认为 s

范 例:

先暂停10分钟再打印printdoc文件, 并指定此工作在后台执行:

```
# (sleep 10m; lpr printdoc) &
```

slocate

常用指数:



(security enhanced version of the GNU locate)

功能说明: 查找文件或目录。

语 法: `slocate [-quv] [-d <目录>] [-e <目录...>] [-f <文件系统...>] [-l <0 或 1>] [-n <数目>] [-U <目录>] [--help] [--version] [查找的文件]`

包 名 称: slocate。

相关命令: find, locate, updatedb, whereis, which。

补充说明: slocate本身具有一个数据库, 里面放了系统中文件与目录的相关信息。若要查找文件或目录时, 利用slocate命令直接在已建立好的数据库中查找, 可加快查找的速度。

参 数:

-d <目录>或--database <目录>	指定数据库所在的目录
-e <目录...>	使用“-u”或“-U”参数建立数据库时，不建立指定目录的数据库
-f <文件系统...>	使用“-u”或“-U”参数建立数据库时，不建立指定文件系统的数据库
-l <0 或 1>	设定执行时的安全核查等级。设为0时，关闭安全核查功能，查找速度较快；设为1时，则启动安全核查功能
-n <数目>	列出查找结果时，指定最多要列出几个结果
-q	执行时显示较少的信息
-u	更新slocate数据库。必须定期更新数据库，才能确保正确的查找结果。执行updatedb命令也可更新 slocate 数据库
-U <目录>	只建立指定目录的数据库
-v	显示命令执行过程
--help	显示帮助
--version	显示版本信息

范 例:

① 建立 slocate 数据库:

```
# slocate -u
```

② 在数据库中查找“slocate”:

```
# slocate slocate
/var/lib/slocate
/var/lib/slocate/slocate.db.tmp
/var/lib/slocate/slocate.db
/etc/cron.daily/slocate.cron
/usr/bin/slocate
/usr/bin/dislocate
...
```

③ 数据库中查找“fdisk”，列出查找结果时，只列出前面3个结果：

```
# locate fdisk
```

← 先以默认方式查找

```
/usr/lib/anaconda/iw/fdisk_gui.py
/usr/lib/anaconda/iw/fdisk_gui.pyc
/usr/lib/anaconda/iw/fdisk_gui.pyo
/usr/lib/anaconda/textw/fdisk_text.py
/usr/lib/anaconda/textw/fdisk_text.pyc
/usr/lib/anaconda/textw/fdisk_text.pyo
...
```

默认会列出全部的查找结果

```
# locate -n 3 fdisk
```

← 只列出前面3个结果

```
/usr/lib/anaconda/iw/fdisk_gui.py
/usr/lib/anaconda/iw/fdisk_gui.pyc
/usr/lib/anaconda/iw/fdisk_gui.pyo
```

④ 更新locate数据库，更新时只建立/home目录的库料库：

```
# locate -U /home
```

⑤ 更新locate数据库，但不建立/mnt目录的数据库：

```
# locate -e /mnt
```

⑥ 更新locate数据库，但不建立属于vfat文件系统的数据库：

```
# locate -u -f vfat
```

slrn

常用指数: **功能说明:** 新闻组阅读程序。**语 法:** slrn [-Cdn] [-f <新闻组文件>] [-h <服务器>] [--create] [--help] [--version]**包 名 称:** slrn。**相关命令:** getlist, nntpget, tin。**补充说明:** slrn是一个很容易使用的新闻组阅读程序, 提供全屏幕的图形操作界面, 让用户能轻松地阅读与管理新闻组。**参 数:**

-C	使用彩色模式 (不检查屏幕是否支持)
-d	下载组清单时, 也一并下载每个组的说明。使用此参数会延长下载的时间
-f <新闻组文件>	指定要存储组内容的文件, 默认的文件名为 .newsrc
-h <服务器>	指定新闻组服务器
-n	不要检查是否有新的新闻组。使用此参数会有较快的启动时间
-p	设定新闻组服务器的连接端口编号, 默认为 119
-create	第一次连上某台新闻组服务器时必须使用此参数, 以便下载服务器中的组清单, 并存储在指定的 newsrc 文件
-help	显示帮助
--version	显示版本信息

范 例:

- ① 第一次连上news.hinet.net新闻组服务器, 建立并指定用户根目录下的.newsrc为新闻组文件, 连上服务器后, 可利用稍后所介绍的操作命令来浏览新闻组:


```
# slrn -h news.hinet.net -f ~/.newsrc -create
slrn 0.9.7.4 (Jul 21 1999 14:38:46)
Reading startup file /etc/slrn.rc.
Using newsrc file /root/.newsrc for server news.hinet.
net.
Connecting to host news.hinet.net ...
Connected to host. Posting Ok.
--The next step may take a while if the NNTP connection
is slow.--
Creating /root/.newsrc.....
```

② 连上news.hinet.net新闻组服务器:

```
# slrn -h news.hinet.net
```

③ 连上news.hinet.net新闻组服务器, 连接时不要检查是否有新的新闻组:

```
# slrn -h news.hinet.net -n
```

④ 连上news.hinet.net新闻组服务器, 当下载新闻组清单时, 一并下载组说明:

```
# slrn -h news.hinet.net -d
```

操作命令: 进入slrn的全屏幕画面后, 因所在的画面不同, 可能会有不一样的操作命令。常用的操作命令说明如下:

通用命令	
?	显示帮助
Page Up 或 Ctrl + u	上一页
Page Down 或 Ctrl + d	下一页
Ctrl + z	暂停程序

组画面操作命令

Space 或 Enter	读取当前组的文章
↑	移至上一个组
↓	移至下一个组
q	结束程序
p	公布文章
L	切换“只显示尚有文章未读取的组/显示所有的组”
L	切换“只显示已订阅的组/显示所有的组”
u	取消订阅当前的组
s	订阅当前的组
c	将当前组中的全部文章标示为已读取
A	订阅指定的组(必须自行输入组名称)
/	查找指定已订阅的组
G	依据服务器上的现状,重新整理新闻组
X	存储 .newsrc
Esc >	移至最后一个组
Esc <	移至第一个组

文章画面操作命令

Space	选择文章,或在文章中往下一页
↑	移至上一篇文章
↓	移至下一篇文章
Del 或 b	在文章中往上一页
d	将当前的文章标示为已读取,然后移至下一篇尚未阅读的文章
u	将当前的文章标示为尚未读取,然后移至下一篇文章
h	只显示文章标题窗口,而不显示文章内容
Ctrl + ↓	使文章标题窗口显示区域减少一行
↑	使文章标题窗口显示区域增加一行
F	响应当前的文章
F	将当前的文章转寄给其他人

r	直接回信给文章作者
O	将当前的文章 (或是文章系列) 存储为文件
q	回到组画面
Esc >	移至最后一篇文章
Esc <	移至第一篇文章
N	移至下一篇尚未读取的文章
p	移至上一篇尚未读取的文章
A	往前查找作者
A	往后查找作者
S	往前查找主题
S	往后查找主题
h	隐藏文章内容窗口
N	读取下一个组

smbclient (samba client)

常用指数: 

功能说明: 可存取 SMB/CIFS 服务器的客户端程序。

语 法: smbclient [网络资源] [密码] [-EhN] [-b <缓冲区大小>] [-d <排错层级>] [-i <范围>] [-I <IP 地址>] [-l <记录文件>] [-L <服务器名称>] [-M <NetBIOS 名称>] [-n <服务器名称>] [-O <连接字选项>] [-p <TCP 连接端口>] [-R <名称解析顺序>] [-s <目录>] [-t <服务器字码>] [-T <tar 选项>] [-U <用户名称>] [-W <工作组>]

包 名 称: samba-client。

相关命令: samba, smbd。

补充说明: SMB与CIFS (衍生自SMB) 为服务器通信协议, 使得Linux及MS Windows操作系统的资源得以共享。smbclient可让Linux系统存取Windows系统所共享的资源。

参 数:

[网络资源]	[网络资源] 的格式为 //服务器名称/资源共享名称。例如, 服务器的名称为smbserver (NetBIOS名称), 共享的目录为share, 则[网络资源]为: //smbserver/share
[密码]	输入存取网络资源所需的密码
-b <缓冲区大小>	指定自服务器端下载文件或上传文件到服务器端时, 使用的缓冲大小, 默认值为65 520字节
-d <排错层级>	指定记录文件所记载事件的详细程度。<排错层级> 可从0到10。0表示只记录严重的错误与警告信息 (也就是最不详细), 此为默认值
-E	将信息送至标准错误输出设备, 而不是默认的标准输出设备
-h	显示帮助
-i <范围>	设定NetBIOS名称范围 (scope)
-I <IP 地址>	指定服务器的IP地址
-l <记录文件>	指定记录文件的名称
-L <服务器名称>	显示服务器端所共享出来的所有资源
-M <NetBIOS 名称>	可利用WinPopup协议, 将信息送给选项中所指定的主机
-n <NetBIOS 名称>	指定客户端所要使用的NetBIOS名称
-N	不用询问密码
-O <连接字选项>	设定客户端TCP连接字 (socket) 的选项
-p <TCP 连接端口>	指定服务器端TCP连接端口编号。SMB/CIFS默认会使用139
-R <名称解析顺序>	设定NetBIOS名称解析的顺序。总共有4种解析NetBIOS名称的方法, 默认的顺序为lmhosts、host、wins和bcast。 lmhosts利用lmhosts文件来解析名称。host则是用hosts文件、NIS及DNS查找来解析名称。wins是以WINS服务器来解析名称。bcast则是以广播的方式来解析名称
s <目录>	指定smb.conf所在的目录。smb.conf为Samba服务器与smbclient的设定文件
-t <服务器字符码>	设定用何种字符码来解释服务器端的文件名称

-T <tar 选项>	<p>备份服务器端共享的全部文件，并打包成 tar 格式的文件。以下为常用的 <tar 选项>：</p> <p>c 在本地端建立一个tar文件</p> <p>a 备份后重设文件的备份标志（Archive bit）。通常会与g及c选项一起使用</p> <p>g 仅备份已设定备份标志的文件，只有与C选项合用时才有效</p> <p>I <文件或目录> 指定要包含的文件或目录</p> <p>N <文件> 以 <文件> 中所指定文件的日期为基准，仅备份比该日期新的文件，只有与C选项合用时才有效</p> <p>q 执行时不显示详细的执行经过</p> <p>x <文件> 将放在本地端的备份文件回存到服务器端的共享目录</p> <p>X <文件或目录> 指定要排除的文件或目录</p>
-U <用户名称>	指定用户名称。若不使用此参数，则默认会以USER或LOGNAME环境变量作为用户名称
-W <工作组>	指定工作组（Workgroup）名称，而不使用smb.conf文件的设定

范 例：

① 连上SMB服务器共享的资源。SMB服务器的电脑名称为top，共享目录的名称为commdir：

```
# smbclient //top/commdir
added interface ip=203.74.205.36 bcast=203.74.205.255
nmask=255.255.255.0
Got a positive name query response from 203.74.205.39 ( 203.74.
205.39 )
Password: ← 输入密码
smb: \> ← 出现提示符，可利用稍后所介绍的操作命令来存取文件
```

② 连上SMB服务器共享的资源。SMB服务器的电脑名称为top，共享目录的名称为commdir，登录的密码为mysecret：

```
# smbclient //top/commdir mysecret
```


- ③ 连上SMB服务器共享的资源。SMB服务器的电脑名称为top，共享目录的名称为sharedir，此目录登录时不需检查密码：

```
# smbclient //top/sharedir -N
added interface ip=203.74.205.36 bcast=203.74.205.255
nmask=255.255.255.0
Got a positive name query response from 203.74.205.39 ( 203.
74.205.39 )
smb: \> ←不用提供密码即可使用此资源
```

- ④ 备份SMB服务器共享的资源。SMB服务器的电脑名称为top，共享目录的名称为sharedir，此目录登录时不需检查密码，并备份该目录下的所有文件，备份文件名为backup.tar：

```
# smbclient //top/sharedir -N -Tc backup.tar
added interface ip=203.74.205.36 bcast=203.74.205.255
nmask=255.255.255.0
Got a positive name query response from 203.74.205.39 (
203.74.205.39 )
    1007 ( 196.7 kb/s) \FRUNLOG.TXT
    3163 ( 308.9 kb/s) \NETLOG.TXT } 备份所有文件的过程
...
    61440 ( 540.5 kb/s) \ASD.EXE
tar: dumped 41 files and directories ← 总共备份的文件及目录数
Total bytes written: 460288 ← 压缩的文件大小
```

- ⑤ 备份SMB服务器共享的资源。SMB服务器的电脑名称为top，共享目录的名称为sharedir，此目录登录时不需检查密码，并备份该目录下的所有文件，备份文件名为backup.tar，备份时不显示详细的执行过程：

```
# smbclient //top/sharedir -N -Tcq backup.tar
added interface ip=203.74.205.36 bcast=203.74.205.255
nmask=255.255.255.0
Got a positive name query response from 203.74.205.39 ( 203.
74.205.39 )
tar: dumped 41 files and directories
Total bytes written: 460288
```

← 与上例比较，并没有显示备份文件的过程

- ⑥ 备份SMB服务器共享的资源。SMB服务器的电脑名称为top，共享目录的名称为sharedir，此目录登录时不需检查密码，并备份该目录下除了testfile.bmp文件之外的所有文件，备份文件名为backup.tar：

```
# smbclient //top/sharedir -N -TcX backup.tar test file.bmp
```

↑
指定排除的文件

- ⑦ 备份SMB服务器共享的资源。SMB服务器的电脑名称为top，共享目录的名称为sharedir，此目录登录时不需检查密码，并备份该目录下文件日期比file.txt还新的文件，备份文件名为backup.tar：

```
# smbclient //top/sharedir -N -TcN backup.tar file.txt
```

↑
只备份比此文件还新的文件

- ⑧ 将备份文件回存到SMB服务器。SMB服务器的电脑名称为top，共享目录的名称为sharedir，此目录登录时不需检查密码，现将backup.tar备份文件回存到该目录上：

```
# smbclient //top/sharedir -N -Tx backup.tar
added interface ip=203.74.205.36 bcast=203.74.205.255
nmask=255.255.255.0
Got a positive name query response from 203.74.205.39
( 203.74.205.39 )
restore tar file \FRUNLOG.TXT of size 1007 bytes
restore tar file \NETLOG.TXT of size 3163 bytes
...
restore tar file \ASD.EXE of size 61440 bytes
tar: restored 41 files and directories
```

回存所有文件的过程

← 总共回存的文件及目录数

⑨ 显示top主机共享的所有资源，登录时不需要检查密码：

```
# smbclient -L top -N
added interface ip=203.74.205.36 bcast=203.74.205.255
nmask=255.255.255.0
Got a positive name query response from 203.74.205.39
( 203.74.205.39 )
```

Sharename	Type	Comment
-----	----	-----
COMMDIR	Disk] 该主机分享出来的目录资源
SHAREDIR	Disk	
PRINTER#	Disk	
HP5L	Printer	← 该主机分享出来的打印机资源
...		

Server	Comment	
-----	-----	
EMILY	XEmily	
JOYCE	Joyce] 与该主机属于同一个工作组的主机
MICHELLE		
...		
Workgroup	Master	
-----	-----	
DOMAIN	EMSTPI	
FLAGPD1	MAIL_FLAGPD1] 该网络上其他的工作群组
LINUXOS	SHADOW	
...		

操作命令：

? 或 help <操作命令>	显示命令的说明
! <shell 命令>	执行shell命令
cd <目录>	切换服务器的工作目录
del <文件>	删除服务器上指定的文件
dir或ls	列出服务器端工作目录中的文件
exit 或 quit	切断与服务器的连接并结束程序
get <服务器端文件> <本地端文件名称>	将服务器端的文件下载到本地端。可指定要存储在本地端的文件名称
lcd <目录>	切换本地端的工作目录
lowercase	切换下载文件时, 是否要自动将文件名改成小写
mask <存取条件>	设定当执行mget或mput命令, 下载上传文件的条件。例如将存取条件设成 "*.txt", 则执行 "mget my*" 命令时, 会下载文件名称以 "my" 起始, 且副文件名为 ".txt" 的文件
mget <文件>	从服务器端下载多个文件到本地端
mput <文件>	将本地端的多个文件上传到服务器端
print <文件>	打印本地端的文件
printmod	切换打印文件时, 是以文本模式或二进制模式打印
prompt	切换执行mget或mput命令时, 是否要使用交谈模式来确认每个文件的传输
put <本地端文件>	将本地端的文件上传到服务器端, 并可指定存储在服务器端<服务器端文件名称>的文件名称
queue	显示打印机的队列
put <文件>	将本地端的文件上传到服务器端
rd<目录>或rmdir <目录>	移除服务器端的目录
recurse	切换当处理文件, 是否要连同所有的子目录一起处理
rm <文件>	移除服务器端的文件
tar <选项>	备份或回存服务器端的文件。可参照 "-t" 参数的说明
blocksize <区块大小>	指定备份文件时, 每个区块的大小
tarmod <备份方式>	指定采用何种方式备份文件, 有以下4种选项: full 备份所有的文件, 此为默认模式 inc 只备份已设定备份标志的文件 reset 备份后重新设定文件的备份标志 noreset 备份后不重新设定文件的备份标志

smbd (samba daemon)

常用指数: 

功能说明: Samba服务器程序。

语 法: `smbd [-aDhoV] [-d <排错层级>] [-l <记录文件>] [-O <连接字选项>] [-p <连接端口编号>] [-s <设定文件>]`

包 名 称: samba (Red Hat Linux/Fedora Core), samba-server (Mandrake Linux)。

相关命令: samba, smbclient, testparm。

补充说明: smbd为Samba服务器程序，可共享文件与打印机等网络资源，供Windows相关的客户端程序（包括MSCLIENT for DOS、Windows for Workgroups、Windows95/98/NT/2000等）存取。

若不指定任何参数，则smbd启动时会去读取smb.conf中的相关参数。在Mandrake Linux中，若只是要启动或关闭Samba服务器，建议使用samba命令（script文件）较为方便。

参 数:

-a	所有的连接记录都会加到记录文件中
-d <排错层级>	指定记录文件所记载事件的详细程度。<排错层级> 可从 0 到 10。0 表示只记录严重的错误与警告信息（也就是最不详细），此为默认值
-D	使用此参数时，smbd会以服务程序（daemon）的方式在后台执行。若客户端的存取不是很频繁，建议使用此参数
-h	显示帮助
-l <记录文件>	指定记录文件的名称
-o	每次启动时，会覆写原有的记录文件。若不使用此参数，默认会将新的记录附加在原有记录文件之后
-O <连接字选项>	设定连接字（socket）选项
-p <连接端口编号>	设定连接端口（port）编号，默认为139
-s <设定文件>	指定 smbd 的设定文件，默认为 smb.conf
-V	显示版本信息

范 例:

① 以后台执行的方式启动 Samba 服务器:

```
# smbd -D
```

② 以后台执行的方式启动Samba服务器，并指定 /var/log/samba/mylog.smb为记录文件:

```
# smbd -l /var/log/samba/mylog.smb
```

③ 以后台执行的方式启动Samba服务器，并指定/etc/mysmb.conf为设定文件:

```
# smbd -s /etc/mysmb.conf
```

sndconfig

常用指数: 

(sound card configure)

功能说明: 设置声卡。**语 法:** sndconfig [--help] [--noautoconfig] [--noprobe] [--quiet]**包 名 称:** sndconfig。**相关命令:** aumix。**补充说明:** sndconfig为声卡设置程序，支持PnP功能，可自动侦测并设置PnP声卡。**参 数:**

--help	显示帮助
--noautoconfig	不自动设置 PnP 的声卡
--noprobe	不自动侦测 PnP 声卡
--quiet	自动设置声卡

范 例：

- ① 设定系统中的声卡设备，会出现图形化界面，可用 **Tab**、**Alt** + **Tab** 和 **Enter** 键进行设定：

```
# sndconfig
```


- ② 设定系统中的声卡设备，执行时不自动侦测PnP声卡：

```
# sndconfig --noprobe
```

- ③ 设定系统中的声卡设备，执行时不自动设置PnP声卡：

```
# sndconfig -noautoconfig
```

sort

常用指数：

功能说明：将文本文件内容加以排序。

语 法：sort [-bcdfimnru] [-o <输出文件>] [-t <分隔字符>] [+<起始栏> -<结束栏>] [--help] [--version] [文件]

包 名 称：coreutils。

相关命令：comm。

补充说明：sort可针对文本文件的内容以行为单位来排序。默认的排序方法是从每行的第一个字符开始，以ASCII码的顺序来排序。也可以指定排序的栏，依该栏的内容来排序。

参 数:

-b	忽略每行前面开始处的空白字符
-c	检查文件是否已经按照顺序排序
-d	排序时,除了英文字母、数字及空白字符外,忽略其他的字符
-f	排序时,将小写字母视为大写字母
-i	排序时,除了040至176(八进制)之间的ASCII字符外,忽略其他的字符
-m	将排序好的数个文件合并
-n	依照数值的大小排序
-o <输出文件>	将排序后的结果存入指定的文件。若不使用此参数,则默认会将排序后的结果显示在屏幕上
-r	以相反的顺序(ASCII由大到小)来排序
-t <分隔字符>	指定排序时所用的栏分隔字符,默认的分隔字符为空白字符
-u	与“-c”参数一起使用,可检查并列有相同栏的行
+<起始栏> -<结束栏> -<结束栏>	以指定的栏来排序,范围由起始栏到结束位的前一栏。例如,以3、4、5等3个栏来排序,则参数应设为+3 -6。若只以一个栏排序,则不用指定
--help	显示帮助
--version	显示版本信息

范 例:

① 显示unsortfile文件的内容,并以默认的方式排序:

cat unsortfile ← 显示unsortfile文件的内容

```

jacket      May      5
sphenoid    September 12
alkane      June     7
irchen      October  14
alkane      June     7
  ↑         ↑         ↑
  栏0      栏1      栏2

```

sort unsortfile ← 以默认方式对文件进行排序

```
alkane      June      7
alkane      June      7
irchen     October  14
jacket     May       5
sphenoid   September 12
```

② 检查同第1个范例的unsortfile文件，输出有相同栏的行：

sort -cu unsortfile

```
sort: unsortfile: 3: 次序不正确: alkane June 7
```

③ 同第1个范例的unsortfile文件，以相反顺序排序此文件：

sort -r unsortfile

```
sphenoid   September  12
jacket     May       5
irchen     October  14
alkane     June      7
alkane     June      7
```

④ 同第1个范例的unsortfile文件，先以默认的方式排序栏2，再以数值大小排序此栏：

sort +2 unsortfile ← 以默认方式排序栏2

```
sphenoid   September  12
irchen     October  14
jacket     May       5
alkane     June      7
alkane     June      7
```

sort +2 -n unsortfile ← 以数值大小排序栏2

```
jacket     May       5
alkane     June      7
alkane     June      7
sphenoid   September  12
irchen     October  14
```

↑ 按照数目大小排列

⑤ 将file1、file2与file3混合排序后，存储成sortfile：

```
# sort -o sortfile file1 file2 file3
```

↑
所要存储的文件名称

⑥ 检查unsortfile及sortfile是否已经按照顺序排序：

```
# sort -c unsortfile
```

```
sort: unsortfile:3:错乱: alkane    June    7 ← 此文件未按顺序排序
```

```
# sort -c sortfile
```

```
# ← 不显示任何信息，则表示此文件已经排序过
```

SOX (sound exchange)

常用指数：

功能说明：音效文件转换程序。

语 法：sox [-hpV] [-v <音量>] [输入文件] [-ex] [-c <频道>] [-r <频率>] [-t <文件类型>] [输出文件] [stat]

包名称：sox。

补充说明：sox可转换各种音效文件的格式，同时也具有简单的音效处理功能。

sox当前支持 .8svx、.aiff、.au、.avr、.cdr、.cvs、.dat、.gsm、.hcom、.maud、.nul、.ogg、ossdsp、.sf、.sph、.smp、sunau、.txw、.vms、.voc、vorbis、.wav、.wve、.raw 等音效文件的格式。

参 数:

-c <频道>	指定音效文件的频道数。<频道> 总共有1、2、4等3种选项
-e	不用指定输出文件。通常与“stat”参数一起使用
-h	显示帮助
-p	加快转换的速度，但可能会牺牲输出文件的品质
-t <文件类型>	指定音效文件的文件类型。若不使用“-t”参数，则必须有适当的扩展文件名，以便sox能正确地辨识文件类型
-r <频率>	指定取样频率，单位为赫兹 (Hertz)
-v <音量>	调整输出文件的音量。<音量> 若大于1.0则输出文件的音量会变大；<音量>若小于1.0则输出文件的音量会变小
-V	执行时显示详细的说明
-x	音效文件为XINU格式
[stat]	分析输入文件，并显示统计结果

范 例:

① 将music.wav文件转换为.voc文件的格式，执行时显示执行过程：

```
# sox -V music.wav music.voc
sox: Detected file format type: wav

sox: Chunk fmt
sox: Chunk data
sox: Reading Wave file: Microsoft PCM format, 1 channel,
6000 samp/sec
sox: 6000 byte/sec, 1 block align, 8 bits/samp, 391211 data
bytes
sox: Input file music.wav: using sample rate 6000
      size bytes, encoding unsigned, 1 channel
sox: Input file music.wav: comment "music.wav"

sox: Output file music.voc: using sample rate 6000
      size bytes, encoding unsigned, 1 channel
sox: Output file: comment "music.wav"
# cat music.voc > /dev/dsp ←播放.voc文件
```


② 将music.wav文件转换为.voc文件的格式，转换时将音量放大4倍：

```
# sox -v 4 music.wav music.voc
```

③ 将music.wav文件转换为.voc文件的格式，并指定取样频率为44 100Hz：

```
# sox -r 44100 music.wav music.voc
```

④ 以较快速的方式将music.wav文件转换为.voc文件的格式：

```
# sox -p music.wav music.voc
```

⑤ 分析并显示music.wav音效文件的统计信息：

```
# sox music.wav -e stat
```

```
Samples read:          391211
Length (seconds):      65.201833
Scaled by:             2147483647.0
Maximum amplitude:     0.992188
Minimum amplitude:     -1.000000
Midline amplitude:     -0.003906
Mean norm:             0.250267
Mean amplitude:        0.009868
RMS amplitude:         0.332884
Maximum delta:         1.859375
Minimum delta:         0.000000
Mean delta:            0.205336
RMS delta:             0.297532
Rough frequency:       853
Volume adjustment:     1.000s
```

spell

常用指数:



功能说明: 拼字检查程序。

语 法: spell

包名称: aspell (Red Hat Linux/Fedora Core)。

相关命令: ispell, look。

补充说明: 此为Red Hat Linux/Fedora Core才有的命令。spell可从标准输入设备读取字符串,并显示拼错的词汇(按 **Ctrl + D** 键)。spell实际上为script文件,执行**ispell -l**命令。

参 数: 无。

范 例:

检查输入的文本是否有误:

```
# spell
```

```
All those moments would be lost in time
lost in
```

← 输入所要检查的句子后,按 **Enter** 键换行显示拼错的词汇

split

常用指数:



功能说明: 切割文件。

语 法: split [-<行数>] [-b <字节>] [-C <字节数>] [-l <行数>] [--help] [--version][要切割的文件] [输出文件名]

包名称: coreutils。

相关命令：csplit。

补充说明：split可将文件切成较小的文件，默认每1000行会切成一个小文件。split会按照参数或自动命名切割后的小文件。

参 数：

-<行数>或-l<行数>	指定每多少行就要切成一个小文件
-b <字节>	指定每多少字节就要切成一个小文件。<字节>后可再加上单位, 总共有 3 种选项:
	b 512 bytes
	k 1 024 bytes (1 KB)
	m 1 048 576 bytes (1 MB)
	如 -b 3b表示以1 536 (3*512) 字节为单位来切割
-C <字节数>	与“-b”参数类似, 但切割时尽量保持每行的完整性。<字节数> 后可在加上b、k、m等单位
--help	显示帮助
--version	显示版本信息
[输出文件名]	设定切割后文件的前置文件名, split会自动在前置文件名后再加上编号。若不设定此参数, 则默认会以x作为输出文件的前置文件名

范 例：

① 以200行为单位, 将largefile切割成小文件, 切割后文件的前置文件名为small:

```
# split -l 200 largefile small
# ls
largefile  smallaa  smallab  smallac  smallad  smallae
```

② 将largefile以10 KB为单位切割成小文件：

```
# split -b 10k largefile
# ls -l
总计 72
-rw-r--r--  1 root    root      31430  6月    29  15:58
largefile
-rw-r--r--  1 root    root      10240  7月    1  11:44 xaa
-rw-r--r--  1 root    root      10240  7月    1  11:44 xab
-rw-r--r--  1 root    root      10240  7月    1  11:44 xac
-rw-r--r--  1 root    root        710  7月    1  11:44 xad
```

不指定前置文件名，则使用默认的前置文件名"x" ↑

③ 将largefile以10 KB为单位切割成小文件，切割时尽量保持每行的完整性：

```
# split -C 10k largefile
# ls -l
总计 72
-rw-r--r--  1 root    root      31430  6月  29    15:58
largefile
-rw-r--r--  1 root    root      10225  7月    1  11:48 xaa
-rw-r--r--  1 root    root      10204  7月    1  11:48 xab
-rw-r--r--  1 root    root      10174  7月    1  11:48 xac
-rw-r--r--  1 root    root        827  7月    1  11:48 xad
```

与上例比较，由于为了要保持每行的完整性，所以xaa、xab及xac文件的大小并不会刚好等于10KB ↑

startx (start X Window)

常用指数: 

功能说明: 启动 X Window。

语 法: startx [程序] [--服务器设定]

包 名 称: XFree86。

补充说明: startx为启动X Window的script文件，实际上启动X Window的程序为xinit。

一般在使用startx时，通常不必加上任何参数。若想启动第2个X Window，可加上“-- :1”参数，若要启动第3个X Window，则加上“-- :2”，依此类推。用户也可以指定要在哪一个虚拟终端启动X Window，如“vt8”表示要在第8号终端启动X Window。

参 数:

[程序]	此处所指的[程序]为X Window所要执行程序的设定文件。startx默认会去读取.xinitrc中的设定，以决定在X Window中要执行哪些程序。[程序]中的设定较.xinitrc为优先
[--服务器设定]	startx默认会去读取.xserverrc中的设定，决定启动X Window的方式。[服务器设定]中的设定较.xserverrc 为优先

范 例:

① 启动第 2 个 X Window:

```
# startx -- :1
```

② 在第 9 号虚拟终端启动 X Window:

```
# startx -- vt9
```

stat (file status)

常用指数: 

功能说明: 显示 inode 内容。

语 法: stat [文件或目录...]

包 名 称: coreutils。

补充说明: stat以文本的格式来显示inode的内容。

参 数: 无。

范 例:

① 显示 sample 文件 inode 的信息:

```
# stat sample.txt
  File: `sample.txt!'
  Size: 5049             Blocks: 16             IO Block: 4096
Regular File
Device: 301h/769d      Inode: 282755        Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 0/root)   Gid: ( 0/root)
Access: 2003-07-03 17:59:24.000000000 +0800
Modify: 2003-07-03 17:59:24.000000000 +0800
Change: 2003-07-03 19:36:44.000000000 +0800
```

② 显示 dir1 及 dir2 两个目录 inode 的信息:

```
# stat dir1 dir2
```

statserial

常用指数: 

(status of serial port)

功能说明: 显示串行端口状态。

语 法: statserial [-dnx] [串行端口设备名称]

包 名 称: statserial。

相关命令: setserial。

补充说明: statserial可显示串行端口各个接脚的状态, 常用来判断串行端口是否正常。执行时, statserial默认会不断地每秒钟更新串行端口的状态, 若要停止执行, 请按 **Ctrl+C** 键。

参 数:

-d	以10进制数字来表示串行端口的状态。各个数字所表示的意义如下: 0 未使用 1 DTR (Data Terminal Ready) 2 RTS (Request To Send) 3 未使用 4 未使用 5 CTS (Clear To Send) 6 DCD (Data Carrier Detect) 7 RI (Ring Indicator) 8 DSR (Data Set Ready)
-n	仅显示一次串行端口的状态后即结束程序
-x	与“-d”参数类似, 但是以十六进制来表示

范 例:

① 显示/dev/ttyS0的状态:

```
# statserial /dev/ttyS0
```

```
Device: /dev/ttyS0
```

Signal Name	Pin (25)	Pin (9)	Direction (computer)	Status	Full Name
FG	1	-	-	-	Frame Ground
TxD	2	3	out	-	Transmit Data
RxD	3	2	in	-	Receive Data
RTS	4	7	out	1	Request To Send
CTS	5	8	in	0	Clear To Send
DSR	6	6	in	0	Data Set Ready
GND	7	5	-	-	Signal Ground
DCD	8	1	in	0	Data Carrier Detect
DTR	20	4	out	1	Data Terminal Ready
RI	22	9	in	0	Ring Indicator

```
# ←按 [Ctrl]+[C] 键结束程序
```

② 只显示一次 /dev/ttyS0 的状态后,即结束程序:

```
# statserial -n /dev/ttyS0
```

SU (super user)

常用指数: 

功能说明: 变更用户身份。

语 法: su [-] [-flmp] [-c <命令>] [-s <shell>] [--help] [--version]
[用户账号]

包名称: coreutils。

相关命令: sudo。

补充说明：su可让用户暂时变更登录的身份。变更时需输入所要变更的用户账号与密码（若原来的身份为root，则不需要输入密码）。

利用su改变身份后，默认并不会改变工作目录，但会变更HOME、SHELL、USER、LOGNAME等环境变量（若是新的身份为root，则不会变更USER与LOGNAME变量）。

参 数：

-c <命令> 或--command=<命令>	执行完指定的命令后，即恢复原来的身份
-f 或 --fast	适用于csh与tsch，使shell不用去读取启动文件
-, -l 或--login	改变身份时，也同时变更工作目录，以及HOME、SHELL、USER、LOGNAME等环境变量（即使新的身份为root，仍然会变更USER与LOGNAME变量）。此外，也会变更PATH变量
-m、-p或--preserve-environment	变更身份时，不要改变环境变量
-s <shell> 或--shell=<shell>	指定要执行的shell。若不使用此参数，则默认会执行/etc/passwd 中指定给用户的shell
--help	显示帮助
--vesion	显示版本信息
[用户账号]	指定要变更的用户。若不指定此参数，则默认变更为root

范 例:

① 原登录的用户账号为alkane，现以默认方式变更用户身份为sphenoid:

```

$ whoami          ← 显示当前身份
alkane
$ pwd            ← 显示工作目录
/home/alkane
$ echo $HOME     ← 显示HOME环境变量
/home/alkane
$ su sphenoid   ← 变更身份成sphenoid用户
Password:       ← 输入sphenoid账号的密码
$ whoami
sphenoid        ← 身份已改为sphenoid
$ pwd
/home/alkane    ← 工作目录没有变更
$ echo $HOME
/home/sphenoid ← HOME环境变量已经变更

```

② 原登录的用户账号为alkane，现变更用户身份为sphenoid，变更时同时变更工作目录及环境变量:

```

$ whoami          ← 显示当前身份
alkane
$ pwd            ← 显示工作目录
/home/alkane
$ echo $HOME     ← 显示HOME环境变量
/home/alkane
$ su - sphenoid ← 变更账号成sphenoid用户，同时变更工作
                  目录及环境变量
Password:       ← 输入sphenoid账号的密码
$ whoami
sphenoid        ← 身份已改为sphenoid
$ pwd
/home/sphenoid  ← 工作目录已经变更
$ echo $HOME
/home/sphenoid ← HOME环境变量已经变更

```

- ③ 原登录的用户账号为alkane，现变更用户身份为sphenoid，变更时
工作目录及环境变量均不更动：

```

$ whoami          ← 显示当前身份
alkane
$ pwd            ← 显示工作目录
/home/alkane
$ echo $HOME     ← 显示HOME环境变量
/home/alkane
$ su -m sphenoid ← 变更账号成sphenoid用户，但
                  不变更工作目录及环境变量
Password:       ← 输入sphenoid账号的密码
$ whoami
sphenoid        ← 身份已改为sphenoid
$ pwd
/home/alkane    ← 工作目录没有变更
$ echo $HOME
/home/alkane    ← HOME环境变量没有变更

```

- ④ 变更用户身份成sphenoid，变更后以tcsh当成shell：

```
$ su -s /bin/tcsh sphenoid
```

- ⑤ 以sphenoid身份检视/home/sphenoid目录的内容：

```
$ su -c "ls /home/sphenoid" - sphenoid
```

所要执行的命令

- ⑥ 以不用任何参数的方式来执行su命令：

```

$ su          ← 默认会变更身份为root
Password:    ← 输入root账号的密码
# whoami
root         ← 身份已改为root

```

- ⑦ 变更用户身份为root，变更时同时变更工作目录及环境变量：

```
$ su -
```

sudo

常用指数: 

功能说明: 以其他身份来执行命令。

语 法: `sudo [-bhHV] [-p <提示符号>] [-s <shell>] [-u <用户>]`
`[命令] 或 sudo [-klv]`

包 名 称: sudo。

相关命令: su。

补充说明: sudo 可让用户以其他的身份来执行指定的命令，默认的身份为root。在/etc/sudoers中设定了可执行sudo命令的用户。若未经授权的用户企图使用sudo，则会发出警告的邮件给管理员。

用户使用sudo时，必须先输入密码，之后有5分钟的有效期限，超过期限则必须重新输入密码。

参 数:

-b	在后台执行命令
-h	显示帮助
-H	将HOME环境变量设为新身份的HOME环境变量
-k	结束密码的有效期限，也就是下次再执行sudo时便需要输入密码
-l	列出当前用户可执行与无法执行的命令
-p <提示符号>	改变询问密码的提示符号（可自行指定）
-s <shell>	执行指定的 shell
-u <用户>	以指定的用户作为新的身份。若不加上此参数，则默认以root 作为新的身份
-v	延长密码有效期限5分钟
-V	显示版本信息

范 例:

① leon用户没有执行sudo的权利，却执行sudo命令：

```
$ reboot
```

```
reboot: must be superuser. ← leon没有执行reboot的权限
```

```
$ sudo shutdown now
```

```
leon is not in the sudoers file. This incident will be
reported. ← sudo会以leon的账号发出一封
           邮件给管理员，报告此事件
```

② leon用户有执行sudo的权力，而执行sudo命令：

```
$ sudo shutdown now
```

```
Password: ← 输入leon的密码即可以root身份执行命令
```

③ 列出用户可执行及无法执行的命令：

```
$ sudo -l
```

```
User may run the following commands on this host:
  (ALL) ALL
```

④ 将密码的有效期限延长5分钟：

```
$ sudo -v
```

⑤ 以sphenoid用户的身份检视/home/sphenoid目录下的文件：

```
$ sudo -u sphenoid ls /home/sphenoid
```

⑥ 中止密码的有效期限，下次再执行sudo时即需输入密码：

```
$ sudo -k
```



sum

常用指数:

**功能说明:** 计算文件的校验和与区块数。**语 法:** sum [-rs] [--help] [--version] [文件]**包 名 称:** coreutils。**相关命令:** cksum, md5sum。**补充说明:** sum可计算文件的校验和 (checksum)。校验和的长度为16位, 并提供System V与BSD等两种算法。sum同时也会显示文件的区块数 (block), 默认一个区块的大小为1 024字节。**参 数:**

-r	使用BSD兼容的算法来计算文件的校验和, 此为默认的算法
-s 或 --sysv	使用System V兼容的算法来计算文件的校验和。使用此参数时, 区块大小会设为512字节
--help	显示帮助
--version	显示版本信息

范 例:

① 以BSD兼容的算法计算file文件的校验和:

```
# sum file      ← 若不加任何参数则是使用BSD兼容的算法
20000 1541
  ↑      ↑
  校验和 区块数
```



② 以System V兼容的算法计算file文件的校验和:

```
# sum -s file    ← 使用System V兼容的算法
15767 3082 file
```

suspend

常用指数:



功能说明: 暂停执行 shell。

语 法: suspend [-f]

包 名 称: bash。

补充说明: suspend为shell内建命令,可暂停当前正在执行的shell。
若要恢复,则必须使用SIGCONT信息。

参 数:

-f	若当前执行的shell为登录的shell,则suspend默认无法暂停此shell。若要强迫暂停登录的shell,则必须使用“-f”参数
----	--

范 例:

暂停当前的shell:


```
# suspend
```

```
-bash:suspend: Can't suspend a login shell ← suspend无法
```

```
# suspend -f ← 强制暂停当前的shell
```

暂停登录的
shell

SVGATextMode

常用指数: 

功能说明: 增强文本模式的显示画面。

语 法: SVGATextMode [-acdfhmnrsv] [-t <设定文件>] [模式]

包 名 称: SVGATextMode (Mandrake Linux)。

补充说明: 此为Mandrake Linux才有的命令。SVGATextMode可用来设定文本模式下的显示画面, 包括分辨率、字体与更新频率等等。SVGATextMode当前所支持的显示晶片, 除了标准VGA以外, 还支持ET系列、S3系列、Matrox (Mystique及Millennium)、ATI (MACH32以前) 等常见的显示晶片。

参 数:

-a	如果新显示模式的屏幕大小与原先不同时, SVGATextMode会执行必要的系统设定; 若屏幕大小不变, 则 SVGATextMode不会执行相关的设定。使用“-a”参数时, 不管屏幕大小是否改变, SVGATextMode 均会重新执行相关的系统设定
-c	维持原有的 VGA 时脉
-d	执行时显示详细的信息, 供排错时参考
-f	不要执行设定文件中有关载入字体的 FontProg 项目
-h	显示帮助
-m	当调整屏幕大小时, 若出现内存不足的信息, 可使用此参数减少其他终端所占用的内存, 以取得较大的可用内存空间
-n	仅测试指定的模式, 并报告测试结果, 而不会实际变更显示模式
-r	不要执行设定文件中有关通知或重设屏幕大小的 RestProg 项目
-s	显示设定文件中所有可用的模式
-t<设定文件>	指定设定文件。若不使用此选项, 则使用默认的/etc/TextConfig为设定文件

-v	SVGATextMode 在设定新的显示模式时, 默认会先检查垂直与水平的更新频率是否在设定文件所指定的范围内, 如果不在范围内, 则不设定新的显示模式。若不要执行此项检查, 请使用“-v”参数
[模式]	[模式] 参数必须是设定文件中模式的名称

范 例:

① 显示设定文件中所有可用的模式:

```
# SVGATextMode -s
```


② 变更文本模式的显示画面:

```
# SVGATextMode Super116x48 ← 设定文件中必须有Super116×48模式
Chipset = 'S3', Textmode clock = 75.00 MHz, 116x48 chars,
CharCell = 9x16. Refresh = 55.93kHz/69.9Hz.
SVGATextMode: Syncing disks...
```

③ 变更文本模式的显示画面, 但仅做测试而不实际变更显示模式:

```
# SVGATextMode -n Super116x48
```

swapoff

常用指数: 

功能说明: 关闭系统交换区 (swap area)。

语 法: swapoff [-ahV] [设备]

包 名 称: mount。

相关命令: mkswap, swapon。

补充说明：swapoff实际上为swapon的符号链接，可用来关闭系统的交换区。

参 数：

-a	关闭 /etc/fstab 文件中所有设定为 swap 的设备
-h	显示帮助
-V	显示版本信息

范 例：


① 关闭 /dev/hda3交换区：

```
# swapoff /dev/hda3
```

② 关闭/etc/fstab文件中所有设定为交换区的设备：

```
# swapoff -a
```

swapon

常用指数：

功能说明：启动系统交换区 (swap area)。

语 法：swapon [-ahsV] [-p <优先顺序>] [设备]

包 名 称：mount。

相关命令：mkswap, swapoff。

补充说明：Linux系统的内存管理必须使用交换区来建立虚拟内存。交换区为独立的磁盘分区。建立好磁盘分区后，还必须使用swapon将分区启动成为系统可用的交换区。若要关闭启动的交换区，请使用swapoff命令。

参 数:

-a	将/etc/fstab文件中所有设定为swap的设备, 启动为交换区
-h	显示帮助
-p <优先顺序>	指定交换区的优先顺序, <优先顺序>可从0到32 767
-s	显示交换区的使用情况
-V	显示版本信息

范 例:

① 启动/dev/hda3交换区:

```
# swapon /dev/hda3 ← 启动交换区
```

② 显示 /dev/hda3 交换区的使用情况:

```
# swapon -s /dev/hda3
```

Filename	Type	Size	Used	Priority
/dev/hda3	partition	72288	4520	-1

③ 启动/etc/fstab文件中所有设定为交换区的设备:

```
# swapon -a
```

④ 启动/dev/hda3交换区, 并指定优先权为2:

```
# swapon -p 2 /dev/hda3
```

swatch (simple watcher)

常用指数: 

功能说明: 系统监控程序。

语 法: swatch [-c <设定文件>] [-f <记录文件>] [-r <时间>] [-t <记录文件>]

包 名 称: swatch (Mandrake Linux)。

补充说明: 此为Mandrake Linux才有的命令。swatch可用来监控系统记录文件,并在发现特定的事件时,执行指定的操作(例如,发送邮件通知管理员,或是执行指定的script)。swatch所监控的事件以及对应事件的操作都保存在swatch的设定文件中。默认的设定文件为用户专属目录下的.swatchrc。

然而在Mandrake Linux的默认用户根目录下并没有.swatchrc设定文件,可以将/usr/share/doc/swatch-3.0.4/examples/swatchrc.personal文件复制到用户专属目录下的.swatchrc,然后修改.swatchrc所要监控的事件及执行的操作。

参 数:

-c <设定文件>	指定设定文件,而不使用默认的设定文件
-f <记录文件>	检查指定的记录文件,检查完毕后不会继续监控该记录文件
-r <时间>	在指定的时间重新启动。<时间>的格式如下:
	+hh:mm hh为小时,mm为分钟。若是设为+3:30,表示经过3小时30分钟后会重新启动
	hh:mm[am或pm] 在指定的时间重新启动。若是设为8:50pm,表示在晚上的8点50分会重新启动
-t <记录文件>	检查指定的记录文件,并且会监控加入记录文件中的后续记录。默认检查的记录文件为/var/log/syslog

范 例:

- ① 执行swatch后, 若有不正常的登录操作, 则会发出哔声, 并以反白的方式显示警告信息:

```
# swatch
```

```
*** swatch-3.0.4 (pid:2321) started at 四 7月 10 19:50:45 CST 2003
Jul 10 19:51:03 free sshd[2363]: Illegal user genie from 192.168.0.39
Jul 10 19:51:03 free sshd[2363]: Failed none for illegal user genie
from 192.168.0.39 port 2156
Jul 10 19:54:01 free login(pam_unix)[2372]: authentication failure;
logname=LOGIN uid=0 euid=0 tty=tty3 ruser= rhost= user=alkane
Jul 10 19:54:04 free login[2372]: FAILED LOGIN 1 FROM (null) FOR alkane,
Authentication failure
```

swatch警告有不正常的登录操作


- ② 执行swatch, 并指定在3小时后重新启动swatch:

```
# swatch -r +3:00
```

- ③ 以用户专属目录下的myswatchrc设定文件来执行swatch命令:

```
# swatch -c ~/myswatchrc
```

symlinks (symbolic links)

常用指数: 

功能说明: 维护符号链接的工具程序。

语 法: symlinks [-cdrstv] [目录]

包 名 称: symlinks。

相关命令: cp, ln。

补充说明： `symlinks`可检查目录中的符号链接，并显示符号链接类型。

以下为`symlinks`可判断的符号链接类型：

- `absolute`：符号链接使用了绝对路径。这类符号链接可能会因为载入点改变而无法使用。
- `dangling`：原始文件已经不存在。
- `lengthy`：符号链接的路径中包含了多余的“`../`”。
- `messy`：符号链接的路径中包含了多余的“`/`”。
- `other_fs`：原始文件位于其他文件系统中。
- `relative`：符号链接使用了相对路径。

参 数：

-c	将使用绝对路径的符号链接转换为相对路径
-d	移除dangling类型的符号链接
-r	检查目录下所有子目录中的符号链接
-s	检查lengthy类型的符号链接
-t	与“-c”一起使用时，会显示如何将绝对路径的符号链接转换为相对路径，但不会实际转换
-v	显示所有类型的符号链接。若不使用“-v”参数，则默认不会显示relative类型的符号链接

S

范 例：

① 检查并列出生成/root/dir目录中的符号链接：

```
# symlinks /root/dir
```

absolute: /root/dir/file4_lnk -> /tmp/file4 ← 此符号链接使用了绝对路径

other_fs: /root/dir/file2_lnk -> /mnt/c/file2 ← 此符号链接的原始文件位于其他的文件系统

dangling: /root/dir/file1_lnk -> file1 ← 此符号链接的原始文件已经不存在

符号链接的类型 符号链接的位置 原始文件的位置

- ② 检查并列出的/root/dir目录中的符号链接，显示时也列出采用相对路径的符号链接：

```
# symlinks -v /root/dir
absolute: /root/dir/file4_lnk -> /tmp/file4
other_fs: /root/dir/file2_lnk -> /mnt/c/file2
dangling: /root/dir/file1_lnk -> file1
relative: /root/dir/file3_lnk -> file3 ←多列出了relative
                                         类型的符号链接
```

- ③ 检查并列出的/root/dir目录包含所有子目录的符号链接：

```
# symlinks -r /root/dir
absolute: /root/dir/file4_lnk -> /tmp/file4
other_fs: /root/dir/file2_lnk -> /mnt/c/file2
dangling: /root/dir/file1_lnk -> file1
absolute: /root/dir/subdir/file5 -> /root/file5 ←
                                         也列出子目录下的符号链接
```

- ④ 删除/root/dir目录中dangling类型的符号链接：

```
# symlinks -d /root/dir ← 删除dangling类型的符号链接
absolute: /root/dir/file4_lnk -> /tmp/file4
other_fs: /root/dir/file2_lnk -> /mnt/c/file2
dangling: /root/dir/file1_lnk -> file1
deleted:  /root/dir/file1_lnk -> file1
```

↑ 执行删除的操作

⑤ 将/root/dir目录中，absolute类型的符号链接转换成relative类型：

```
# symlinks -c /root/dir
absolute: /root/dir/file4_lnk -> /tmp/file4
changed:  /root/dir/file4_lnk -> ../../tmp/file4
      ↑
      |
      | 绝对路径已经改为相对路径
      |
      |
other_fs: /root/dir/file2_lnk -> /mnt/c/file2
```

sync (synchronize)

常用指数：

功能说明：将内存缓冲区内的数据写入磁盘。

语 法：sync [--help] [--version]

包 名 称：coreutils。

相关命令：halt。

补充说明：在Linux系统中，当数据需要存入磁盘时，通常会先放到缓冲区内，等到适当的时刻再写入磁盘，如此可提高系统的执行效率。sync命令则用来强制将缓冲区内的数据立即写入磁盘中。用户通常不需执行sync命令，因为系统会自动执行update或bdfush，将缓冲区的数据写入磁盘。只有在update或bdfush因故无法执行，或等不及执行shutdown便要关机时，才需手动执行sync命令。

参 数：

--help	显示帮助
--version	显示版本信息

范 例：

立即将缓冲区中的数据写入磁盘：

```
# sync
```


tac (reverse concatenate)

常用指数:



功能说明: 连接多个文件, 并将其内容反序输出到标准输出设备。

语 法: tac [-br] [-s <间隔字符>] [--help] [--version] [文件...]

包 名 称: coreutils。

相关命令: cat, cut, head, tail。

补充说明: tac命令和cat命令类似, 但输出相反顺序的数据。它会读取所给予的文件的内容, 并将其内容反序输出在标准的输出设备上面。若不指定任何文件名称, 或是所给予的文件名为“-”, 则tac命令会从标准输入设备读取数据, 然后再把所得到的数据输出至输出设备。反序输出以行为单位, 也就是文件的第1行会显示在后面, 最后一行则显示在第1行。

参 数:

-b 或 --before	把间隔字符的字符串放在文件头记录的前面
-r 或 --regex	把间隔字符的字符串视同一般表达方式
-s <间隔字符> 或 --separator=<间隔字符>	用指定的间隔字符取代新增行控制字符
--help	显示帮助
--version	显示版本信息

范 例:

① 先列出file1文件的内容, 再反序列出此文件的内容:

```
# cat file1 ← 列出 file1 的内容
line1 in file1
line2 in file1
line3 in file1
# tac file1 ← 反序列出 file1 的内容
```

```
line2 in file1
line1 in file1
```

② 将键盘所输入的数据反序输出：

```
# tac          ← 执行时不加任何参数
1st line      |
2nd line      | 输入要反序输出的内容，按 Enter 键
2nd line      | 完成输入后，请按 Ctrl + d 键
1st line      | 反序输出的内容
```

③ 将 textfile 文件反序输出至打印机：

```
# tac textfile | lpr
```

tail

常用指数：



功能说明：输出文件内容的最后部分。

语法：tail [-fqv] [-c <显示数目>] [-n <显示行数>] [--help] [--version] [文件...]

包名称：coreutils。

相关命令：cat, cut, head, tail。

补充说明：tail命令会读取指定文件的内容，并将其内容最后的部分输出到标准输出设备。若不指定任何文件名称，或是指定的文件名为“-”，则tail命令会从标准输入设备读取数据。默认读取的行数是最后10行，假设所给予的文件超过一个以上，tail会在列出的文件内容前面，加上符号“==>”和“<==”，并在两个符号之间标出文件名称。

参 数:

-c<显示数目>或--bytes= <显示数目>	设定要显示多少数据, 单位以Byte计算。可在指定的数字后面加上“b”、“k”和“m”, 各表示512Bytes、1 024Bytes和1 048 576Bytes。例如设成20, 表示列出20个字符的数据, 若是2m, 则是列出2MB大小的数据
-f 或 --follow	读取到文件最后面时, 持续反复尝试读取更多的数据。本参数可应用于一些内容持续增加的文件, 比如记录文件等, 让用户得以随时观看到最新加入的内容
-n <显示行数>或--lines= <显示行数>	设定要显示多少行的内容。如显示行数设为5, tail命令会显示文件最后5行的内容。我们可在显示行数前面加上“+”号, 表示要从该指定行数开始列出文件内容, 例如显示行数设成+10, 表示从该文件内容的第10行开始列出, 直到文件结束为止
-q、--quiet 或 --silent	不显示文件名称
-v 或 --verbose	显示文件名称
--help	显示帮助
--version	显示版本信息

范 例:

① 列出 textfile 文本文件最后面的内容:

```
# tail textfile
```

← 默认会列出最后 10 行的内容

② 列出textfile文本文件最后5行的内容:

```
# tail -n 5 textfile
```

③ 列出textfile文本文件最后200Bytes的内容:

```
# tail -c 200 textfile
```

④ 列出textfile文本文件最后512Bytes的内容：

```
# tail -c 1b textfile
```

↑
"b"表示512Bytes


⑤ 列出当前目录下所有扩展名为.txt文件最后面的内容：

```
# tail *.txt
```

⑥ 列出当前目录下所有文件名称以“file”起始的文件最后面的内容，列出时不显示每个文件的名称：

```
# tail -q file*
```

talk

常用指数：

功能说明：与其他用户交谈。

语 法：talk [用户名称] [终端编号]

包 名 称：talk。

相关命令：mesg, ytalk。

补充说明：借由talk命令，用户可以和另一个用户线上交谈，假如想和其他主机的用户交谈，则用户名称必须加上其主机名称或IP地址，例如用户名称为thief，主机名称是lady.heart.org，完整的用户名称就是“thief@lady.heart.org”。

如果关闭他人要求跟你交谈的功能，可使用mesg命令。

参 数: 无。

范 例:

① terry用户要求跟同一主机的alkane用户交谈:

```
$ talk alkane
```

此时alkane用户会收到以下的信息:

```
Message from Talk_Daemon@linux.flag.com.tw at 9:21 ...
talk: connection requested by terry@linux.flag.com.tw.
talk: respond with: talk terry@linux.flag.com.tw
```

如果alkane愿意跟terry交谈, 则执行以下命令:

```
$ talk terry
```

若alkane与terry不在同一台主机上, 则执行"talk terry@linux.flag.com.tw"

② alkane用户当前在本地主机登录超过一次, 而terry用户要求跟alkane交谈, 此时需特别指定终端编号:

```
$ talk alkane tty4 ← 指定跟第4号终端的alkane用户交谈
```

③ 要求跟superlinux.flag.com.tw主机的jacket用户交谈:

```
$ talk jacket@superlinux.flag.com.tw
```

tar (tape archive)

常用指数: 

功能说明: 备份文件。

语 法: tar [-ABcdgGhiklmMoOpPrRsStuUvwWxzZ] [-b <区块数目>] [-C <目的目录>] [-f <备份文件>] [-F <script 文件>] [-K <文件>] [-L <媒体容量>] [-N <日期时间>] [-T <范本文件>] [-V <卷名称>] [-X <范本文件>] [-<设备编号><存储密度>] [--after-date=<日期时间>] [--atime-preserve] [--backup=<备份方式>] [--checkpoint] [--concatenate] [--confirmation] [--delete] [--exclude=<范本样式>] [--force-local] [--group=<组名称>] [--help] [--ignore-failed-read] [--new-volume-script=<script 文件>] [--newer-mtime=<日期时间>] [--no-recursion] [--null] [--numeric-owner] [--owner=<用户名称>] [--posix] [--preserve] [--preserve-order] [--preserve-permissions] [--record-size=<区块数目>] [--recursive-unlink] [--remove-files] [--rsh-command=<执行命令>] [--same-owner] [--suffix=<备份字尾字符串>] [--totals] [--use-compress-program = <执行命令>] [--version] [--volno-file=<编号文件>] [文件或目录...]

包 名 称: tar。

相关命令: compress, cpio, gzip。

补充说明: tar是用来建立、还原备份文件的工具程序，它可以加入、解开备份文件内的文件。备份的文件可存放在硬盘里、软盘上或是磁带机。利用tar命令可将很多文件或目录打包成一个文件，便于保存。

参 数:

-A 或 --catenate	新增备份文件至已存在备份文件之后
-b<区块数目>或--blocking-factor=<区块数目>	设定每条记录的区块数目, 每个区块大小为512 Bytes。例如设定区块数目为4, 则记录大小就是2 048 Bytes
-B 或 --read-full-records	读取数据时重设区块大小。本参数仅对 BSD (Berkeley Software Distribution) 4.2 的管道有效
-c 或 --create	建立新的备份文件
-C<目的目录>或--directory=<目的目录>	切换到指定的目录
-d、--diff或--compare	对比备份文件内容和系统中文件内容的差异
-f <备份文件> 或--file <主机名称><文件>	指定备份文件。该文件亦能设定所存放文件用的外围设备
-F<script文件>或--info-script=<script文件>	每次更换磁带时, 就执行指定的script文件。这项参数配合“-M”参数使用
-g 或 --listed-incremental	处理新GNU格式的大量备份
-G 或 --incremental	处理旧GNU格式的大量备份
-h 或 --dereference	不建立符号链接, 直接复制该连接所指向的原始文件
-i 或 --ignore-zeros	忽略备份文件中的0 Byte区块, 也就是EOF
-k 或 --keep-old-files	解开备份文件时, 不覆盖已有的文件
-K<文件>或--starting-file<文件>	从指定的文件开始还原
-l 或 --one-file-system	复制的文件或目录所在的文件系统, 必须与 tar命令执行时所处的文件系统相同, 否则不予以复制
-L<媒体容量> 或--tape-length<媒体容量>	设定存储媒体的容量, 单位以1 024 Bytes计算。例如容量设成1 000, 则能够存放的空间为1 024 000 Bytes
-m 或 --modification-time	还原文件时, 不变更文件的异动时间
-M 或 --multi-volume	在建立、还原备份文件或列出其中内容时, 采用多卷模式

-N <日期时间> 或 --newer <日期时间>	只将较指定日期更新的文件存储到备份文件里
-o、--old-archive 或 --portability	将数据写入备份文件时使用 V7 格式
-O 或 --to-stdout	把从备份文件里还原的文件输出至标准输出设备
-p 或 --same-permissions	用原来的文件权限还原文件
-P 或 --absolute-paths	文件名使用绝对名称，不删除文件名称前的 “/” 号
-r 或 --append	新增文件至已存在的备份文件的结尾部分
-R 或 --record-number	列出每个文件在备份文件中的区块编号
-s 或 --same-order	还原文件的顺序，和备份文件内的存放顺序相同
-S 或 --sparse	假如一个文件内含大量的连续0字节，则将此文件存成稀疏文件。稀疏文件是一种内含大量连续0字节的文件，这种现象称之为坑洞，许多的二进制文件都具有这种特性。假使文件系统有支持这种特性，这些坑洞将不会占用大量的存储区块，则对节省存放空间和增进系统效能皆有助益
-t 或 --list	列出备份文件的内容
-T <范本文件> 或 --files-from =<范本文件>	指定范本文件，其内含有一个或多个范本样式，让tar解开或建立符合设定条件的文件
-u 或 --update	仅置换较备份文件内的文件更新的文件
-U 或 --unlink-first	解开压缩文件还原文件之前先解除文件的连接
-v 或 --verbose	显示命令执行过程
-V <卷名称> 或 --label <卷名称>	建立使用指定的卷名称的备份文件
-w 或 --interactive	处理每个文件时，均要求用户确认
-W 或 --verify	写入备份文件后，确认文件正确无误
-x、--extract 或 --get	从备份文件中还原文件
-X <范本文件> 或 --exclude-from <范本文件>	指定范本文件，其内含有一个或多个范本样式，让tar排除符合设定条件的文件
-z、--gzip 或 --ungzip	通过gzip命令处理备份文件

-Z、-compress 或-uncompress	通过compress命令处理备份文件
-<设备编号><存储密度>	设定备份用的外围设备编号及存放数据的密度，设备编号的范围是0~7，密度则分为“l”、“m”、“h”，表示低、中、高密度
--after-date <日期时间>	此参数的效果和指定“-N”参数相同
--atime-preserve	不变更文件的存取时间
--backup=<备份方式>	移除文件前先行备份。备份方式指定为“numbered”（或“t”）将使用备份编号，在字尾会加上“~1~”字符串，且数字编号依次递增。备份方式指定为“simple”（或“never”）将使用简单备份，默认的备份字尾字符串是“~”，也能通过“--suffix”参数来设定。备份方式指定为“existing”（或“nil”）将使用当前方式，程序会先检查是否存在备份编号，若有则采用备份编号，若无则采用简单备份
--checkpoint	读取备份文件时列出目录名称
--concatenate	此参数的效果和指定“-A”参数相同
--confirmation	此参数的效果和指定“-w”参数相同
--delete	从备份文件中删除指定的文件。若备份用的媒体是磁带，则无法单独删除某个文件
--exclude=<范本样式>	排除符合范本样式的文件
--force-local	强制于本地主机进行备份操作
--group=<组名称>	把加入备份文件中的文件的所属组设成指定的组
--help	显示帮助
--ignore-failed-read	忽略数据读取错误，不中断程序的执行
--new-volume-script=<script 文件>	此参数的效果和指定“-F”参数相同
--newer-mtime=<日期时间>	只存储异动过的文件
--no-recursion	不做递归处理，也就是指定目录下的所有文件及子目录不予处理
--null	从null设备读取文件名称
--numeric-owner	以用户识别码及组织识别码，取代用户名称和组名称

<code>--owner=<用户名称></code>	把加入备份文件中的文件的拥有者设成指定的用户
<code>--posix</code>	将数据写入备份文件时使用POSIX格式
<code>--preserve</code>	此参数的效果和同时指定“-ps”参数相同
<code>--preserve-order</code>	此参数的效果和指定“-A”参数相同
<code>--preserve-permissions</code>	此参数的效果和指定“-p”参数相同
<code>--record-size=<区块数目></code>	此参数的效果和指定“-b”参数相同
<code>--recursive-unlink</code>	解开压缩文件还原目录之前，先解除整个目录下所有文件的连接
<code>--remove-files</code>	文件加入备份文件后，就将其移除
<code>--rsh-command=<执行命令></code>	设定要在远程主机上面执行的命令，以取代rsh命令
<code>--same-owner</code>	尝试以相同的文件拥有者还原文件
<code>--suffix=<备份字尾字符串></code>	移除文件前先行备份。备份文件的字尾会被加上一个备份字符串，默认的备份字尾字符串是符号“~”。通过这个参数可以改变备份字尾字符串
<code>--totals</code>	备份文件建立后，列出文件大小
<code>--use-compress-program = <执行命令></code>	通过指定的命令处理备份文件
<code>--version</code>	显示版本信息
<code>--volno-file=<编号文件></code>	使用指定文件内的编号取代默认的卷编号

范 例：

① 备份当前目录下所有文件和子目录，并指定备份文件名为 backup.tar:

```
# tar -c -f backup.tar .
```

↑ 指定备份文件文件名
↑ 建立新的备份文件

② 备份/home/alkane目录下所有文件和子目录，并指定备份文件名为 backup.tar，执行时显示命令执行过程：

```
# tar -cv -f backup.tar /home/alkane
tar: Removing leading `/' from archive names
home/alkane/
home/alkane/.Xdefaults
home/alkane/.bash_logout
home/alkane/.bash_profile
...
```

} 显示备份文件的过程

③ 备份backupdir目录下所有文件和子目录，备份时一并检查备份文件是否正确，并指定备份文件文件名为backup.tar，执行时显示命令执行过程：

```
# tar -cvWf backup.tar backupdir
backupdir/
backupdir/dir/
backupdir/dir/file1
backupdir/file1
backupdir/.file1
Verify backupdir/
Verify backupdir/dir/
Verify backupdir/dir/file1
Verify backupdir/file1
Verify backupdir/.file1
```

} 显示备份文件的过程

} 显示检查文件的过程

④ 备份backupdir目录下所有文件和子目录，备份后显示备份文件的文件大小，并指定备份文件文件名为backup.tar：

```
# tar --totals -cf backup.tar backupdir
Total bytes written: 10240 ← 备份文件大小为10240Bytes
```



- ⑤ 备份backupdir目录下所有文件和子目录，备份时询问用户是否要备份每个文件，并指定备份文件文件名为backup.tar:

```
# tar -cwf backup.tar backupdir
```

```
add backupdir?y ← 按 y 键表示同意
```

```
add backupdir/dir?
```

```
...
```

- ⑥ 备份backupdir目录下所有文件和子目录，并指定备份文件中的文件的拥有者为alkane，所属组为users，备份文件文件名为backup.tar:

```
# tar --owner=alkane --group=users -cf backup.tar backupdir
```

- ⑦ 备份backupdir目录下所有文件和子目录，但不备份文件名称以“file”起始的文件，并指定备份文件文件名为backup.tar:

```
# tar --exclude=file* -cf backup.tar backupdir
```

- ⑧ 备份backupdir目录下所有文件和子目录，若该目录有符号链接，则直接备份该连接所指向的原始文件，并指定备份文件文件名为backup.tar:

```
# tar -chf backup.tar backupdir
```

- ⑨ 备份backupdir目录下的所有文件及目录，但不包括子目录下的文件和目录，也不备份文件名称以“.”开头的文件，并指定备份文件文件名为backup.tar，执行时显示命令执行过程:

```
# ls -aR backupdir/ ← 显示backupdir目录下的所有文件及目录
```

```
backupdir/:
```

```
. .. .file1 dir file1
```

```
backupdir/dir:
```

```
. .. file1
```

```
# tar -cvf backup.tar --no-recursion backupdir/*
```

```
backupdir/dir/    } 此例并没有备份/backupdir/.file及
backupdir/file1  } /backupdir/file1文件
```

⑩ 备份backupdir目录下所有文件和子目录，备份完后再以gzip命令压缩，并指定备份文件文件名为backup.tar.gz：

```
# tar -czf backup.tar.gz backupdir
```

⑪ 备份backupdir目录下所有文件和子目录，备份完后再以compress命令压缩，并指定备份文件文件名为backup.tar.Z：

```
# tar -czf backup.tar.Z backupdir
```

⑫ 备份backupdir目录下所有文件和子目录，备份完后再以zip命令压缩，并指定备份文件文件名为backup.tar.zip：

```
# tar --use-compress-program zip -cf backup.tar.zip backupdir
```

⑬ 列出 backup.tar备份文件的内容：

```
# tar -tf backup.tar
```

```
backupdir/
backupdir/dir/
backupdir/dir/file1
backupdir/file1
backupdir/.file1
```

⑭ 以详细的方式列出backup.tar备份文件的内容：

```
# tar -tvf backup.tar
```

```
drwxr-xr-x root/root      0 2003-07-05 13:52:15 backupdir/
```

```
drwxr-xr-x root/root 0 2003-07-05 11:21:50 backupdir/dir/
-rw-r--r-- root/root 77 2003-07-05 10:26:05 backupdir/dir/file1
-rw-r--r-- root/root 77 2003-07-05 10:25:55 backupdir/file1
-rw-r--r-- root/root 77 2003-07-05 10:34:14 backupdir/.file1
```

- ⑮ 列出backup.tar备份文件的内容，列出时同时显示每个文件的区块编号：

```
# tar -tRf backup.tar
block 0: backupdir/
block 1: backupdir/dir/
block 2: backupdir/dir/file1
block 4: backupdir/file1
block 6: backupdir/.file1
block 8: ** Block of NULs **
```

↑
区块编号

- ⑯ 列出曾以gzip命令压缩过的backup.tar.gz备份文件的内容：

```
# tar -tzf backup.tar.gz
```

- ⑰ 列出曾以compress命令压缩过的backup.tar.Z备份文件的内容：

```
# tar -tzf backup.tar.Z
```

- ⑱ 检查bakcup.tar备份文件的内容是否与文件系统上的文件一致：

```
# tar -df backup.tar
```

```
backupdir/file1: Mod time differs ← 此文件的修改时间改变了
backupdir/file1: Size differs ← 此文件的文件大小改变了
```



⑲ 将file文件加入到backup.tar备份文件中：

```
# ls ← 列出当前目录下的所有文件
backup.tar file
# tar -rf backup.tar file ← 将file文件加入备份文件
# ls ← 再列出当前目录下的所有文件
backup.tar file ← 所加入的file文件仍然保留
```

⑳ 将file文件加入到backup.tar备份文件中，加入后删除 file 文件：

```
# ls ← 列出当前目录下的所有文件
backup.tar file
# tar --remove-files -rf backup.tar file ← 将file文件加入
                                         备份文件后再删除
                                         此文件

# ls ← 再列出当前目录下的所有文件
backup.tar ← file文件已经不存在
```

㉑ 更新 backup.tar 备份文件中的 file 文件：

```
# tar -uf backup.tar file
```

㉒ 先查看backup.tar备份文件的内容，再删除此备份文件中指定的 backupdir/dir/file1文件：

```
# tar -tf backup.tar ← 列出backup.tar备份文件的内容
backupdir/
backupdir/dir/
backupdir/dir/file1
backupdir/file1
backupdir/.file1 删除备份文件中的backupdir/dir/file1文件
# tar --delete -f backup.tar backupdir/dir/file1 ←
# tar -tf backup.tar ← 列出backup.tar备份文件的内容
backupdir/
backupdir/dir/
backupdir/file1
backupdir/.file1
```

backupdir/dir/file1文件已经被删除

⑳ 解开 backup.tar 备份文件, 执行时显示命令执行过程:

```
# tar -xvf backup.tar
```

㉑ 解开曾以 gzip 命令压缩过的 backup.tar.gz 备份文件:

```
# tar -xzf backup.tar.gz
```

㉒ 解开曾以 compress 命令压缩过的 backup.tar.Z 备份文件:

```
# tar -xzf backup.tar.Z
```

㉓ 解开 backup.tar 备份文件, 解开时不覆盖现有的文件:

```
# tar -xkf backup.tar
```

㉔ 将 backup1.tar 备份文件中的文件加到 backup.tar 备份文件内:

```
# tar -A backup1.tar -f backup.tar
```

tcpdump

常用指数:



功能说明: 转储网络传输数据。

语 法: tcpdump [-adeflnNOpqStvx] [-c <包数目>] [-dd] [-ddd] [-F
<表达文件>] [-i <网络接口>] [-r <包文件>] [-s <包大小>]
[-tt] [-T <包类型>] [-vv] [-w <包文件>] [输出数据栏]

包 名 称: tcpdump。



补充说明：执行tcpdump命令可列出经过指定的网络接口的包文件头，在Linux操作系统中，用户必须是系统管理者才能执行这项命令，除非tcpdump具备SUID的权限设定，否则一般用户无法使用它。可指定欲输出数据的栏位，若不给予栏位名称，则tcpdump会转出所有可供输出的数据。

参 数：

-a	尝试将网络和广播地址转换成名称
-c <包数目>	收到指定的包数目之后,就停止进行转储操作
-d	把编译过的包编码转换成可阅读的格式,并转储至标准输出
-dd	把编译过的包编码转换成 C 语言的格式,并转储至标准输出
-ddd	把编译过的包编码转换成十进制数字的格式,并转储至标准输出
-e	在每行转储数据上显示连接层级的文件头
-f	用数字显示网络地址
-F <表达文件>	指定内含表达方式的文件,而不在命令列设定表达方式
-i <网络接口>	使用指定的网络接口送出包。一般而言,第 1 张网卡的接口代号通常是 eth0,可执行 ifconfig 命令查询网络接口代号
-l	使用标准输出行的缓冲区
-n	不把主机的网络地址转换成名称
-N	不列出域名,例如某主机的名称为 zone.tomato.com,则使用本参数后,仅会列出 tomato.com
-O	不将包编码最优化。这个参数只在最优化出现错误时才有作用
-p	不让网络接口进入混杂模式
-q	快速输出,仅列出少数的传输协议信息
-r <包文件>	从指定的文件读取包数据,该文件可由参数“-w”产生。假设将包文件设定为“-”,则 tcpdump 会从标准输入设备读取数据
-s <包大小>	设定每个包的大小,单位以 Byte 计算。默认值为68,可通过此参数另行设定
-S	用绝对而非相对数值列出 TCP 关联数
-t	不在每行转储数据上显示时间戳记

-tt	在每行转出数据上显示未经格式化的时间戳记
-T <包类型>	强制将 [输出数据栏]指定的包转译成设定之包类型。包类型可分为“cnpf” (Cisco Netflw protocol)、“rpc” (Remote Procedure Call)、“rtp” (Real Time Applications Protocol)、“rtcp” (Real Time Applications Control Protocol)、“vat” (Visual Audio Tool) 和 “wb” (Distributed White Board)
-v	详细显示命令执行过程
-v v	更详细显示命令执行过程
-x	用十六进制列出包数据
-w <包文件>	把包数据写入指定的文件, 该文件可由参数“-r”读取。假设将包文件设定为“-”, 则 tcpdump会把数据写入标准输出设备

范 例:

① 列出通过默认网络接口的包文件头:

```
# tcpdump
```

```
tcpdump: listening on eth0
```

```
14:33:34.255603 linux.flag.com.tw.ssh > 192.168.0.39.3213: P
1276272621:
```

```
1276272665(44) ack 1811674177 win 7504 (DF) [tos 0x10]
```

```
14:33:34.257542 linux.flag.com.tw.32782 > dns.is.net.tw.domain:
33959+ PTR? 39.0.168.192.in-addr.arpa. (43) (DF)
```

```
14:33:34.302771 dns.is.net.tw.domain > linux.flag.com.tw.32782:
33959 NXDomain 0/1/0 (120)
```

```
14:33:34.303914 linux.flag.com.tw.32782 > dns.is.net.tw.domain:
33960+ PTR? 1.128.62.210.in-addr.arpa. (43) (DF)
```

```
...
```

② 列出通过默认网络接口的包文件头, 但只列出 10 个包:

```
# tcpdump -c 10
```

③ 列出通过默认网络接口的包文件头, 执行时显示命令执行过程:

```
# tcpdump -v
```

tee

常用指数:



功能说明: 读取标准输入的数据, 并将其内容输出成文件。

语法: tee [-ai] [--help] [--version] [文件...]

包名称: coreutils。

补充说明: tee命令会从标准输入设备读取数据, 将其内容输出到标准输出设备, 并同时存储成文件。我们可利用tee把管道导入的数据存成文件, 甚至一次存储数份文件。

参数:

-a 或 --append	附加到已有文件的后面, 而非覆盖它。如果给予tee 命令的文件名称已经存在, 默认会覆盖该文件的内容。加上此参数后, 数据会新增在该文件内容的最后面, 而不会删除原先内容
-i 或 --ignore-interrupts	忽略中断信号
--help	显示帮助
--version	显示版本信息

范例:

① 从键盘输入一段文本后, 再将此段文本存成 output 文件:

```
# tee output
```

```
The mount command invokes the mount system call after
doing some... ← 输入要存成文件的文字
```

```
The mount command invokes the mount system call after doing
some ... ← 屏幕上会出现一模一样的字符串, 完成输入后按 [Ctrl]+[C] 键结束
```

```
# cat output ← 列出output文件的内容
```

```
The mount command invokes the mount system call after
doing some ...
```

T

② 从键盘输入一段文本，并将此段文本附加在已有的output文件之后：

```
# tee -a output
```

③ 列出文本文件slayers.story的内容，同时复制3份副本，文件名称分别为ss-copy1、ss-copy2和ss-copy3：

```
# cat slayers.story | tee ss-copy1 ss-copy2 ss-copy3
```

④ 寻找所有文件中包含“xcin”字符串的文件，除了打印出结果外，并将结果存储成xcin_result：

```
# locate xcin | tee xcin_result | lpr
```

telinit

常用指数：



功能说明：切换系统的执行等级。

语 法：telinit [-t <秒数>] [0123456abcqQsS]

包 名 称：SysVinit。

相关命令：chkconfig, ntsysv, runlevel。

补充说明：为了要符合于各种执行场合的需求，Linux操作系统将操作环境分成了7个执行等级（runlevel），可设定在不同的执行等级所要启动的服务。/etc/inittab设定文件用来指定系统开机时所要进入的执行等级及启动的服务。系统完成开机后，若想变更执行等级，则可使用telinit命令来立即切换系统的执行等级。

不过需注意的是，telinit所切换的执行等级只对此次开机有效。若想更改以后每次启动Linux时所进入的执行等级，还是请修改/etc/inittab设定文件。

Linux 操作系统将执行等级分成以下 7 种：

执行等级	执行的状态
0	关机 (请不要切换到此等级)
1	单人用户模式的文本界面
2	多人用户模式的文本界面, 但不具有网络文件系统 (NFS) 功能
3	多人用户模式的文本界面, 具有全部的网络功能
4	某些发行版的 Linux 使用此等级进入 X Window
5	某些发行版的 Linux 使用此等级进入 X Window
6	重新开机

参 数：

0 ~ 6	切换到指定的执行等级
a、b 或 c	切换到指定的 a、b 或 c 执行等级 (当前已很少使用此 3 种执行等级)
q 或 Q	重新检查 /etc/inittab 设定文件
s 或 S	进入单人用户模式
-t <秒数>	在等待指定的秒数之后, 即切换执行等级。默认值为 5 秒

范 例：

① 先显示当前的执行等级后, 然后再切换到执行等级 3:

```
# runlevel ← 显示当前执行等级
```

```
N 5
```



当前执行等级

```
# telinit 3
```

② 切换到单人用户模式的执行等级：

```
# telinit s
```

③ 等待10秒钟之后，切换到执行等级 5：

```
# telinit -t 10 5
```

telnet

常用指数：

功能说明：远程登录。

语 法：telnet [-8acdEfFKLrx] [-e <脱离字符>] [-k <领域名称>] [-l <用户名称>] [-n <记录文件>] [-S <服务类型>] [-X <认证类型>] [主机名称或 IP 地址 <通信端口>]

包名称：telnet-client (Red Hat Linux), telnet (Mandrake Linux/ Fedora Core)。

补充说明：执行telnet命令开启终端阶段操作，并登录远程主机。用户可指定通信端口或使用默认的通信端口。单独执行telnet命令，它会进入命令模式，执行 ? 或help命令就可查询所有可用命令，执行 ? [命令名称] 或 help [命令名称] 命令则能显示该命令的帮助。

参 数：

-8	允许使用8位字符数据, 包括输入与输出
-a	尝试自动登录远程系统。本参数会读取环境变量USER所设定的用户名称, 作为登录账号
-c	不读取用户根目录里的 .telnetrc 文件
-d	启动排错模式
-e <脱离字符>	设定脱离字符。如果使用这项参数却没指定脱离字符, 则telnet命令将不会使用脱离字符

-E	滤除脱离字符
-f	此参数的效果和指定“-F”参数相同
-F	使用Kerberos V5认证时，加上此参数可把本地主机的认证数据，上传至远程主机
-k <领域名称>	使用Kerberos认证时，加上此参数让远程主机采用指定的域名，而非该主机的域名
-K	不自动登录远程主机
-l <用户名称>	指定要登录远程主机的用户名称。假设没有使用这项参数，则telnet会询问欲使用的登录账号
-L	允许输出8位字符数据
-n <记录文件>	指定文件记录相关信息
-r	使用类似rlogin命令的用户界面。在这个模式中，默认的脱离字符是“~”，除非用参数“-e”去改变它
-S <服务类型>	设定telnet连线所需的IP TOS (Type of Service) 信息
-x	假设主机有支持数据加密的功能，就使用它
-X <认证类型>	关闭指定的认证类型。认证类型分为：“kerberos”、“kerberos_v5”以及“password”

范 例：

① 登录到linux.flag.com.tw主机：

```
# telnet linux.flag.com.tw
Trying 203.74.205.30...
Connected to linux.flag.com.tw.
Escape character is '^]'.
Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8 on an i686
login: sphenoid ← 输入账号名称
Password: ← 输入密码
Last login: Mon Jul 7 12:17:34 on tty5
$
```

T

② 以sphenoid账号登录到linux.flag.com.tw主机：

```
# telnet -l sphenoid linux.flag.com.tw
Trying 203.74.205.30...
Connected to linux.flag.com.tw.
Escape character is '^]'.
Password:          ← 直接输入该账号的密码即可
Last login: Mon Jul  7 17:02:25 from 203.74.205.36
$
```

testparm (test parameter)

常用指数：



功能说明：测试 Samba 的设置是否正确无误。


语 法：testparm [-hs] [设定文件] [<主机名称> <IP 地址>]

包 名 称：Samba-common。

相关命令：samba, smbd。

补充说明：执行testparm命令可以简单测试Samba的设定文件，假如测试结果无误，Samba常驻服务就能正确载入该设定值，但并不保证其后的操作如预期一样一切正常。默认的设定文件为 /etc/samba/smb.conf，当然也可以另行指定。主机名称和IP地址必须成对指定，用来测试设定文件内的“hosts allow”和“hosts deny”能否正常操作。

参 数：

-h	显示帮助
-s	不显示提示符号等待用户按下  键，就直接列出Samba服务定义信息

范 例:

① 编辑smb.conf设定文件之后, 测试其内容是否正确无误:

```
# testparm
```

```
Load smb config files from /etc/smb.conf
```

```
Processing section "[homes]"
```

```
Processing section "[printers]"
```

```
Processing section "[tmp]"
```

```
Processing section "[public]"
```

```
Loaded services file OK.
```

```
Press enter to see a dump of your service definitions
```

按下  键显示Samba服务定义信息

测试过程与结果

```
# Global parameters
```

```
[global]
```

```
workgroup = MYGROUP
```

```
netbios name =
```


```
netbios aliases =
```

```
...
```

Samba服务定义信息

② 测试 /etc/mysmb.conf 设定文件是否正确无误:

```
# testparm /etc/mysmb.conf
```

③ 测试Samba设定文件是否正确无误, 显示时不等待用户按下  键, 即列出Samba服务定义信息:

```
# testparm -s
```

tftp (trivial file transfer protocol)

常用指数:



功能说明: 传输文件。

语 法: tftp [主机名称或 IP 地址]

包 名 称: tftp (Mandrake Linux/Fedora Core)。

补充说明：此为Mandrake Linux/Fedora Core才有的命令。在互联网上要传输文件，除了一般常用的FTP传输协议之外，还可以使用另一种传输协议—TFTP (Trivial File Transfer Protocol)。这种传输协议并不提供身份认证的功能，同时在服务器端与客户端也没有复杂的文件传输机制，使得传输文件时的负荷降低。不过当前只有少数的服务器提供这一传输协议。而TFTP命令就是TFTP客户端程序。

参 数：无。

范 例：

连线具有TFTP服务的 ftp.flag.com.tw 主机：

```
# tftp ftp.flag.com.tw
```

操作命令：

? <操作命令>	列出操作命令的简要说明。若不指定 <操作命令>，则列出所有可用的操作命令
ascii	将传输文件的模式设成 ASCII 模式
binary	将传输文件的模式设成二进制模式
connect<主机地址>	设定要传输文件的主机地址。由于TFTP传输协议并不像FTP在客户端与服务器端间建立永久性的连线，此操作命令的目的在于告知tftp程序，当执行get或put操作命令时，所要传输文件的主机地址
get <文件...>	下载远程的文件
mode <传输模式>	指定传输文件时要使用的模式，可指定的模式有“ascii”和“binary”
put <文件...>	上传本地端的文件
quit	结束TFTP程序
rexmt <秒数>	设定每个包重新传输的超时秒数
status	显示当前的传输状态
timeout <秒数>	设定整体传输的超时秒数
trace	设定是否要使用追踪包模式
verbose	切换简繁模式

tgatoppm

常用指数:



(targa bitmap to portable pixmap)

功能说明: 转换图像文件。**语 法:** tgatoppm [-headerdump] [TGA 图像文件]**包名称:** netpbm-progs (Red Hat Linux/Fedora Core), netpbm (Mandrake Linux)。**补充说明:** 执行tgatoppm命令, 读取指定的TGA格式图像文件(默认最后的扩展名为.tga), 并把该文件转成Portable Pixmap(默认最后的扩展名为.ppm)格式后, 输出到标准输出设备。**参 数:**

-headerdump	将文件信息转出至标准错误输出, 以便于排错
-------------	-----------------------

范 例:

把mypicture.tga图像文件转成Portable Pixmap格式:

```
# tgatoppm mypicture.tga > mypicture.ppm
```

tifftopnm

常用指数:



(tag image file format to portable anymap)

功能说明: 转换图像文件。**语 法:** tifftopnm [-headerdump] [TIFF 图像文件]

T

包名称: netpbm-progs (Red Hat Linux/Fedora Core), netpbm (Mandrake Linux)。

补充说明: 执行tifftopnm命令, 读取指定的TIFF格式图像文件 (默认最后的扩展名为.tif), 并把该文件转成Portable Anymap (默认最后的扩展名为.pnm) 格式后, 输出到标准输出设备。Anymap的格式可以是Bitmap、Graymap或Pixmap, 视输入的文件格式而定。

参 数:

-headerdump	把TIFF文件的信息转出至标准错误输出, 便于排错
-------------	---------------------------

范 例:

把mypicture.tif图像文件转成Portable Anymap格式:

```
# tifftopnm mypicture.tif > mypicture.pnm
```

timeconfig

常用指数: 

功能说明: 设置时区。

语 法: timeconfig [--arc] [--test] [时区名称]

包名称: redhat-config-date (Red Hat Linux/Fedora Core), system-config-date (Fedora Core 2&3)。

相关命令: clock, date, hwclock, rdate。

补充说明: 这是Red Hat公司遵循GPL规则所开发的程序 (Red Hat Linux/Fedora Core才有此命令), 具有交互式操作界面, 可以轻易地利用方向键和空白键等, 设置系统时间所属的时区。也可指定参数和时区名称, 直接设置时区而不通过交互式界面。

参 数:

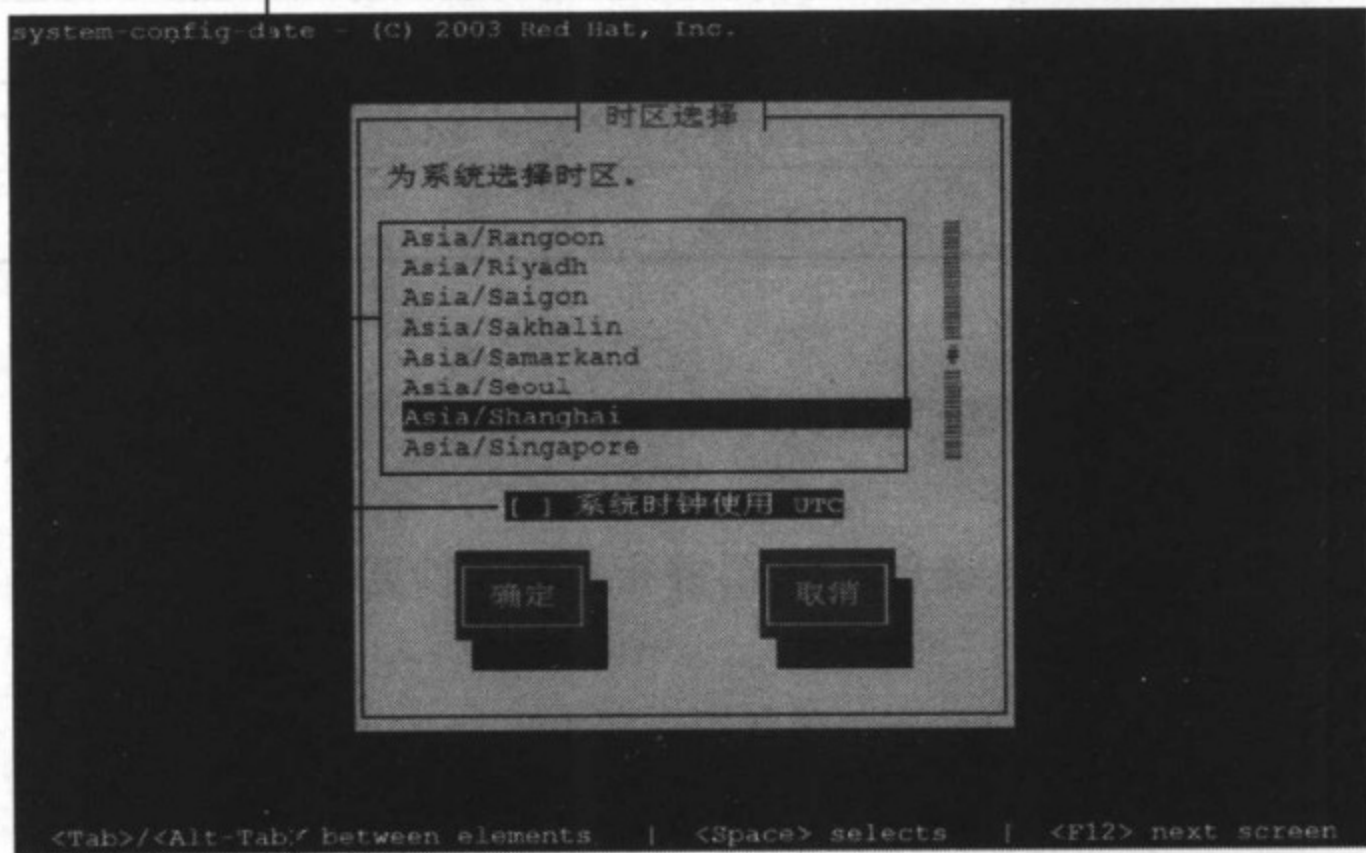
--arc	使用Alpha硬件架构的格式存储系统时间。如果不是使用Digital的Alpha 硬件平台,此参数将没有任何作用
--test	仅作测试,并不真正改变系统的时区

范 例:

- ① 使用timeconfig程序设定时区,可使用 **Tab**、**Alt + Tab** 及空格键来设置时区:

```
# timeconfig
```

选择时区,若在中国,则选择Asia/Shanghai



选择此项,则将硬件时钟设成格林威治标准时间

- ② 使用timeconfig程序设定时区,但仅作测试,并不真正改变系统的时区:

```
# timeconfig --test
```

tload (tty load)

常用指数:



功能说明: 显示系统负载状况。

语 法: tload [-V] [-d <间隔秒数>] [-s <刻度大小>] [终端编号]

包 名 称: procps。

相关命令: top。

补充说明: tload命令使用ASCII字符简单地以文本模式显示系统负载状态。假设不给予终端编号,则会在执行tload命令的终端显示负载情形。

参 数:

-d <间隔秒数>	设定tload侦测系统负载的间隔时间,单位以秒计算
-s <刻度大小>	设定图表的垂直刻度大小,单位以行计算
-V	显示版本信息

范 例:

① 显示当前终端的系统负载状况:

```
# tload
```

② 显示当前终端的系统负载状况,并指定每3秒更新一次信息:

```
# tload -d 3
```

③ 显示第3号终端的系统负载状况:

```
# tload /dev/tty3
```

tmpwatch (temporary watch)

常用指数: 

功能说明: 删除临时文件。

语 法: tmpwatch [-afqv] [--test] [过期时间] [目录...]

包 名 称: tmpwatch。

补充说明: 执行tmpwatch命令可删除不必要的临时文件，可以设定文件过期的时间，单位以小时计算。tmpwatch会将指定目录里的过期文件统统清除，通常会指定/tmp、/var/tmp等临时目录。它采用文件的最后存取时间来判断是否过期，而非使用最后异动时间。

参 数:

-a 或 --all	删除任何类型的文件。tmpwatch默认只会删除一般文件或空目录, 但不会删除符号链接等特殊类型的文件
-f 或 --force	强制删除文件或目录, 其效果类似“rm”命令的“-f”参数
-q 或 --quiet	不显示命令执行过程
-v 或 --verbose	详细显示命令执行过程。如重复指定这项参数多次, 则会显示更为详细的相关信息
--test	仅作测试, 并不真正删除文件或目录。本参数需配合“-v”参数使用

范 例:

① 删除/tmp和/var/tmp目录中超过10日不曾使用过的文件:

```
# tmpwatch 240 /tmp /var/tmp
```

↑
10日共计240小时

② 删除/tmp目录中超过10日不曾使用过的文件，执行时显示命令执行过程：

```
# tmpwatch -v 240 /tmp
```

③ 删除/tmp目录中超过10日不曾使用过的文件，包括符号链接及特殊文件：

```
# tmpwatch -a 240 /tmp
```


④ 强制删除/tmp目录中超过10日不曾使用过的文件：

```
# tmpwatch -f 240 /tmp
```

⑤ 仅作测试，不真正删除/tmp目录中超过10日不曾使用过的文件，执行时显示命令执行过程：

```
# tmpwatch -v --test 240 /tmp
```

top

常用指数：

功能说明：显示、管理执行中的程序。

语 法：top [bciqsS] [d <间隔秒数>] [n <执行次数>]

包 名 称：procps。

相关命令：free, kill, ps, renice, tload。

补充说明：执行top命令可显示当前正在系统中执行的程序，并通过它所提供的互动式界面，用功能键加以管理。



参 数:

b	使用批次模式。本模式中top将不接受任何热键输入命令, 除非达到参数“n”所设定的执行次数, 或被强制中断才会停止
c	列出程序时, 显示每个程序的完整命令, 包括命令名称、路径和参数等相关信息
d <间隔秒数>	设定 top 监控程序执行状况的间隔时间, 单位以秒计算
i	执行top命令时, 忽略闲置或是已成为僵尸 (Zombie) 的程序。若子程序已经当掉, 而父程序却仍不知道, 该子程序便称为僵尸
n <执行次数>	设定监控信息的更新次数。当达到指定的次数之后, top命令就会自动结束并回到命令列
q	持续监控程序执行的状况。若是系统管理者使用这项参数, 则top的优先权将是最高等级, 可不断更新程序执行的信息

范 例:

① 显示系统程序的执行状态, 可采用稍后所介绍的操作命令来改变要显示的信息:

```
# top
```

② 显示系统程序的执行状态, 并指定每3秒钟更新一次信息:

```
# top d 3
```

③ 不断地更新、显示系统程序的执行状态:

```
# top q
```

④ 显示系统程序的执行状态, 当更新达 10 次之后, 即结束程序:

```
# top n 10
```

- ⑤ 显示系统程序的执行状态，显示时列出每个程序的完整命令，包括命令名称、路径和参数等信息：

top c

- ⑥ 显示系统程序的执行状态，但不显示闲置中及已成为僵尸的程序：

top i

- ⑦ 采用批次模式显示系统的执行状态，此时所有的功能键均没有任何作用：

top b

- ⑧ 使用累计模式来显示系统程序的执行状态：

top S

操作命令：

k	中止执行中的程序
S	设定更新数据的时间间隔
r	调整执行程序的优先权
S	切换是否以累计模式显示
i	切换是否显示闲置中及成为僵尸的程序
L	切换是否显示平均负载及时间相关的信息
M	切换是否显示内存相关的信息
T	切换是否显示 CPU 及程序的相关信息
c	切换是否显示完整的命令列栏位

n	依程序 ID 排序
A	依程序执行的先后顺序排序
P	依 CPU 的使用率排序, 此为默认值
M	依内存的使用量排序
T	依时间栏位排序
F 或 F	新增或删除要显示的栏位
O 或 O	改变显示栏位的顺序
n 或 ,	设定最多要显示几个程序。若设为 0, 表示不做任何限制
Space	立即更新显示中的数据
h 或 ?	显示帮助
Ctrl + L	重新显示画面
W	将当前的设定值写入 ~/.toprc 设定文件中
q	结束程序

touch

常用指数:



T

功能说明: 改变文件或目录时间。

语法: touch [-acfm] [-d <日期时间>] [-r <参考文件>] [-t <日期时间>] [--help] [--version] [文件或目录...]

包名称: coreutils。

补充说明: 使用touch命令可更改文件或目录的日期时间, 包括存取时间和异动时间。假如不指定日期和时间, 则会以现在的时间为依据。

参 数:

-a、--time=atime、--time=access或--time=use	只更改存取时间
-c 或 --no-create	不建立任何文件
-d <日期时间>	使用指定的日期时间,而非现在的时间。touch 命令会自动尝试解读日期与时间的格式,它可以包含月份名称 (Jan、Feb...)、时区名称、上下午 (am/pm)...等
-f	此参数将忽略不予处理,仅负责解决 BSD 版本 touch命令的兼容性问题
-m、--time=mtime 或--time=modify	只更改异动时间
-r <参考文件>或--reference=<参考文件>	把指定文件的日期时间,都设成和参考文件的日期时间相同
-t <日期时间>	使用指定的日期时间,而非现在的时间。设定的格式依次是年份、月份、日期、小时、分钟、秒数,如“200302231300.00”即表示公元2003年2月23日13时0分0秒。年份可以指定2位数,若数字范围是00~68,将被视为公元2000~2068年,若数字是69~99,则会被当成公元1969~1999年。年份、秒数皆可省略不予指定
--help	显示帮助
--version	显示版本信息

范 例:

① 将 file文件的存取时间设为当前时间:

```
# touch -a file
```

② 将file文件的异动时间设为当前时间:

```
# touch -m file
```

③ 将file1文件的异动时间与存取时间设成与file文件一致：

```
# touch -r file file1
```

④ 把/home/alkane目录下的所有文件，其存取和异动时间统统设成公元2003年7月12日17时30分50秒：

```
# touch -t 200307121730.50 /home/alkane/*
```

tr (translate character)

常用指数：



功能说明：转换字符。

语 法：tr [-cdst] [--help] [--version] [第 1 字符集] [第 2 字符集]

包 名 称：coreutils。

相关命令：sed。

补充说明：tr命令从标准输入设备读取数据，经过字符串转译后，输出到标准输出设备。字符集除了指定字符串之外（如“xyz”），还能采用下列的格式：

- \<字符字码>：使用 1~3 位数的八进制字码指定字符。
- \\：反斜线字符“\”。
- \a：声响控制字符。
- \b：倒退字符。
- \f：FF 控制字符。

- \n: 新增行控制字符。
- \r: CR 控制字符。
- \t: 水平制表字符。
- \v: 垂直制表字符。
- <第1字符>-<第2字符>: 包含第1字符到第2字符间的所有字符, 以递增顺序排列。例如“a-z”表示所有小写的英文本母。
- [<字符>*]: 运用在第2字符集, 重复指定的字符, 直到等于第1字符集的字数为止。
- [<字符>*<重复次数>]: 重复指定的字符达到设定的次数为止, 默认之重复次数以十进制数字表示, 若重复次数的开头为0, 则采用八进制数字表示。
- [:alnum:]: 所有的字母和数字。
- [:alpha:]: 所有的字母。
- [:blank:]: 所有的水平空格字符。
- [:cntrl:]: 所有的控制字符。
- [:digit:]: 所有的数字。
- [:graph:]: 所有可打印的字符, 但不包括空格字符。
- [:lower:]: 所有小写的字母。
- [:print:]: 所有可打印的字符, 包括空格字符。
- [:punct:]: 所有的标点符号。
- [:space:]: 所有的水平与垂直空格字符。

- [:upper:]: 所有大写的字母。
- [:xdigit:]: 十六进制数字。
- [=<字符>=]: 所有等于指定字符的字符。

参 数:

-c 或 --complement	取代所有不属于第 1 字符集的字符
-d 或 --delete	删除所有属于第 1 字符集中的字符
-s 或 --squeeze-repeats	把连续重复的字符以单独一个字符表示
-t 或 --truncate-set1	先删除第 1 字符集较第 2 字符集多出的字符
--help	显示帮助
--version	显示版本信息

范 例:

- ① 把文本文件nevergiveup里的小写字母统统换成大写字母，然后存储成文件NEVERGIVEUP:

```
# tr a-z A-Z < nevergiveup > NEVERGIVEUP
```

或

```
# tr [:lower:] [:upper:] < nevergiveup > NEVERGIVEUP
```

- ② 由键盘输入一段文本后，将“a”至“d”的小写字母转换成大写字母:

```
# tr a-d A-D
```

```
abcdefgh ← 输入任一字符串
```

```
ABCDefgh
```

└─ 只有 "a" 至 "d" 的小写字母被转成大写

- ③ 将dollar.txt文本文件里连续出现的“a”字符替换为单一字符，并以名称dollar.neo存文件：

```
# tr -s [a] < dollar.txt > dollar.neo
```

- ④ 将text文本文件里非英文字母的字符替换成空白字符，并将结果存成text_output文件：

```
# tr -c "[:alpha:]" "[ *]" < text1 > text_output
```

注意：此处为空白字符

- ⑤ 将letter文本文件中所有的数字字符删除，并将结果存成letter_output文件：

```
# tr -d [:digit:] < letter > letter_output
```

traceroute

常用指数：



(trace route packets)

功能说明：显示包到主机间的路径。

语 法：traceroute [-dFIrvx] [-f <存活数值>] [-g <网关>...] [-i <网络接口>] [-m <存活数值>] [-p <通信端口>] [-s <来源地址>] [-t <服务类型>] [-w <超时秒数>] [主机名称或 IP 地址] [包大小]

包 名 称：traceroute。



相关命令： ping。

补充说明： traceroute命令追踪网络包的路由途径，默认包大小是40 Bytes，用户可另行设定。

参 数：

-d	使用 Socket 层级的排错功能
-f <存活数值>	设定第 1 个侦测包的存活数值 TTL 的大小，范围从 1~255。每个 IP 包都会拥有这项数值，以便决定该包是否应放弃。当包经过路由器，TTL 数值就会被扣减，直到数值为 0 时，路由器就会放弃这个包，且发出 ICMP 的信息通知送出该包的主机
-F	设定勿离断 (Don' t Fragment) 位
-g <网关>	设定来源路由网关，最多可设定 8 个
-i <网络接口>	使用指定的网络接口送出包。一般而言，第 1 张网卡的接口代号通常是 eth0，可执行 ifconfig 命令查询网络接口代号
-I	使用 ICMP 响应取代 UDP 数据信息
-m <存活数值>	设定侦测包的最大存活数值 TTL 的大小，也就是最大跃程数目，范围从 1~255，默认值为 30
-n	直接使用 IP 地址而非主机名称，节省数据往来名称服务器的时间
-p <通信端口>	设定 UDP 传输协议的通信端口，默认是 33 434
-r	忽略普通的 Routing Table，直接将包送到远程主机上。如果该主机并非本区域网络内的一份子，则会传回错误信息
-s <来源地址>	设定本地主机送出包的 IP 地址
-t <服务数值>	设定侦测包的 TOS 数值，该数值必须是十进制整数，范围在 0~255 之间
-v	详细显示命令的执行过程
-w <超时秒数>	设定等待远程主机回报的时间，单位以秒计算，默认值为 5 秒
-x	开启或关闭包的正确性检验

范 例:

① 显示本地主机到www.flag.com.tw的包传递路径:

```
# traceroute www.flag.com.tw
traceroute to www.flag.com.tw (210.62.132.10), 30 hops max,
38 byte packets
 1  192.168.0.3 (192.168.0.3)  0.346 ms  0.227 ms  0.174 ms
 2  c29.h061013225.is.net.tw (61.13.225.29)  1.432 ms
1. 387 ms  1.970 ms
 3  10.1.1.70 (10.1.1.70)  42.500 ms  41.140 ms  43.756 ms
 4  TP-R3661-28.rt.is.net.tw (202.52.70.158)  44.031 ms
66.894 ms  43.965 ms
 5  c10.h210062132.is.net.tw (210.62.132.10)  44.004 ms
47.794 ms  42.228 ms
```

② 显示本地主机到www.flag.com.tw的包传递路径, 追踪包时直接使用IP地址而非主机名称:

```
# traceroute -n www.flag.com.tw
traceroute to www.flag.com.tw (210.62.132.10), 30 hops
max, 38 byte packets
 1  192.168.0.3  0.279 ms  0.221 ms  0.215 ms
 2  61.13.225.29  1.427 ms  1.376 ms  1.378 ms
 3  10.1.1.70  42.781 ms  109.999 ms  41.502 ms
 4  202.52.70.158  123.293 ms  48.348 ms  109.305 ms
 5  210.62.132.10  50.680 ms  55.032 ms  44.096 ms
```

③ 显示本地主机到 www.flag.com.tw 的包传递路径, 显示时详细列出命令的执行过程:

```
# tarceroute -v www.flag.com.tw
```

tree

常用指数: 

功能说明: 以树状图列出目录的内容。

语 法: tree [-aACdDfFgilmNpqstux] [-I <范本样式>] [-P <范本样式>] [目录...]

包 名 称: tree。

相关命令: ls。

补充说明: 执行tree命令，它会列出指定目录下的所有文件，包括子目录里的文件。若不指定目录名称，则会列出当前目录(含子目录)中的全部内容。

参 数:

-a	显示所有文件和目录, 包括以“.”为名称开头字符的文件、当前目录“.”与上层目录“..”
-A	使用ANSI绘图字符显示树状图, 而非以ASCII字符组合
-C	在文件和目录清单加上色彩, 便于区分各种类型
-d	只显示目录, 而不列出文件
-D	列出文件或目录的异动时间
-f	在每个文件或目录之前, 显示完整的相对路径名称
-F	在执行文件、目录、Socket、符号链接、管道名称后面, 各自加上“*”、“/”、“=”、“@”、“ ”号。其效果类似ls命令的“-F”参数
-g	列出文件或目录的所属组名称, 没有对应的名称时, 则显示组织别码
-i	不以阶梯状列出文件或目录名称
-I<范本样式>	不显示符合范本样式的文件
-l	如遇到性质为符号链接的目录, 直接列出该连接所指向的原始目录



-n	不在文件和目录清单加上色彩。若搭配参数“-C”使用,则本参数无效
-N	直接列出文件和目录名称,包括控制字符。使用此参数ls命令会将控制字符视同普通字符一并列出
-p	列出权限标示。其效果类似ls命令的“-l”参数
-P <范本样式>	只显示符合范本样式的文件。当欲寻找的文件以“.”为名称开头字符时,必须同时指定参数“-a”方能列出该文件
-q	用“?”号取代控制字符,列出文件和目录名称
-s	列出文件或目录大小
-t	用文件和目录的异动时间排序
-u	列出文件或目录的拥有者名称,没有对应的名称时,则显示用户识别码
-x	将范围局限在当前的文件系统中,若指定目录下的某些子目录存放于另一个文件系统中,则将该子目录予以排除在寻找范围外。其效果类似find命令的“-xdev”参数

范 例:

① 以树状图列出当前目录的内容:

```
# tree
.
...
|-- boot
|   |-- System.map -> System.map-2.4.20-8
|   |-- System.map-2.4.20-8
|   |-- boot.b
|   |-- chain.b
|   |-- config-2.4.20-8
|   |-- grub
|   |   |-- device.map
|   |   |-- e2fs_stage1_5
...

```

② 以树状图列出当前目录的内容, 同时显示其异动时间:

```
# tree -D
.
...
|-- [Jul  2  0:30] boot
|   |-- [Jul  2  0:30] System.map -> System.map-2.4.20-8
|   |-- [Mar 14  7:01] System.map-2.4.20-8
|   |-- [Jan 25 13:05] boot.b
|   |-- [Jan 25 13:05] chain.b
|   |-- [Mar 14  7:01] config-2.4.20-8
|   |-- [Jul  2  2:32] grub
|       |-- [Jul  2  2:32] device.map
|       |-- [Jul  2  2:32] e2fs_stage1_5
...

```

③ 以树状图列出当前目录的内容, 同时列出完整的路径:

```
# tree -f
.
...
|-- ./boot
|   |-- ./boot/System.map -> System.map-2.4.20-8
|   |-- ./boot/System.map-2.4.20-8
|   |-- ./boot/boot.b
|   |-- ./boot/chain.b
|   |-- ./boot/config-2.4.20-8
|   |-- ./boot/grub
|       |-- ./boot/grub/device.map
|       |-- ./boot/grub/e2fs_stage1_5
...

```

④ 以树状图列出当前目录的内容，同时列出其拥有者：

```
# tree -u
.
|-- [alkane  ]  alkane
|-- [genie   ]  genie
|   |-- [root   ]  test
|   `-- [genie  ]  test1
`-- [tony    ]  tony
```

5 directories, 0 files

⑤ 以树状图列出当前目录的内容，同时列出所属组：

```
# tree -g
```

⑥ 以树状图列出当前目录的内容，同时显示其文件或目录的大小：

```
# tree -s
.
|-- [      4096]  alkane
|-- [      4096]  genie
|   |-- [      4096]  test
|   `-- [      4096]  test1
`-- [      4096]  tony
```

5 directories, 0 files

⑦ 以树状图列出当前目录的内容，同时显示其文件或目录的权限：

```
# tree -p
.
|-- [drwx-----]  alkane
|-- [drwx-----]  genie
```



```
| |-- [drwxr-xr-x] test  
| |-- [drwxrwxr-x] test1  
|-- [drwx-----] tony
```

5 directories, 0 files

⑧ 列出当前目录的内容，但不以阶梯方式显示：

```
# tree -i
```

```
.  
alkane  
genie  
test  
test1  
tony
```

5 directories, 0 files

⑨ 以树状图列出当前目录中，名称以file起始的文件：

```
# tree -P file*
```

⑩ 以树状图列出当前目录中，名称不以file起始的文件：

```
# tree -I file*
```

⑪ 依异动时间的顺序，以树状图列出当前目录的内容：

```
# tree -t
```

⑫ 以ANSI绘图字符列出当前目录的内容：

```
# tree -A
```



- ⑬ 以树状图列出当前目录的内容，对于执行文件、目录、Socket、符号链接及管道，分别在名称后面加上特殊符号：

```
# tree -F
```

- ⑭ 以树状图列出 /home/alkane 目录下所有文件和目录，包括以 “.” 起始字符的文件：

```
# tree -a /home/alkane
```

- ⑮ 以树状图列出 /home/alkane 及 /home/sphenoid 目录下的内容，只列出目录并不显示文件：

```
# tree -d /home/alkane /home/sphenoid
```

ttmkfdir

常用指数：

功能说明： 建立TTF字体的索引文件。

语 法： ttmkfdir [-cfp] [-d <目录>] [-m <数目>] [-o <文件>]

包 名 称： ttmkfdir (Red Hat Linux/Fedora Core), freetype-tools (Mandrake Linux)。

相关命令： xfs, xset。

补充说明： ttmkfdir命令会分析指定目录所有的TTF (True Type Font) 字体，然后建立字体的索引文件。建立好索引文件后，即可修改 /etc/X11/XF86Config设定文件（有的Linux发行版的设定文件是 /etc/X11/fs/config），将此目录加入字体的搜寻路径，即可在X Window中使用该目录下的字体。

参 数:

-c	以较宽松的标准分析 TTF 字体
-d <目录>	指定要建立字体索引文件的目录。默认值为当前的工作目录
-f	对于不知名字体厂商所造的字体也建立索引文件
-m <数目>	建立索引文件时, 指定最多能容忍几个字符发生错误。默认值为 5 个字符
-o <文件>	指定要输出文件的名称。默认值是输出到标准输出设备
-p	建立索引文件时, 考虑 panose 信息

范 例:

① 建立现在工作目录的字体索引文件, 并将结果存成 fonts.dir;

```
# ttmkfdir > fonts.dir
```

② 而执行以下命令也有同样结果, 并将结果存成 fonts.dir;

```
# ttmkfdir -o fonts.dir
```

③ 建立/mnt/c/windows/font目录的的字体索引文件, 并将结果存成 fonts.dir;

```
# ttmkfdir -d /mnt/c/windows/font -o /mnt/c/windows/font/
fonts.dir
```



tty (teletypewriter)

常用指数: 

功能说明: 显示终端连接标准输入设备的文件名称。

语 法: `tty [-s] [--help] [--version]`

包 名 称: `coreutils`。

补充说明: 在Linux操作系统中, 所有外围设备都有其名称与代号, 这些名称代号以特殊文件的类型存放于/dev目录下。可执行tty命令查询连接当前使用的终端的文件名称。

参 数:

<code>-s</code> 、 <code>--silent</code> 或 <code>--quiet</code>	不显示任何信息, 只回传状态代码。状态代码为0、1、2或3, 依次表示标准输入设备TTY、标准输入设备不是TTY、命令语法错误, 以及写入错误
<code>--help</code>	显示帮助
<code>--version</code>	显示版本信息

范 例:

① 查询当前使用终端的外围设备文件名称:

```
# tty
/dev/tty3
```

② 查询当前使用终端的外围设备文件名称, 并以状态代码的方式将结果传回:

```
# tty -s ← 表示当前正在使用第3号终端
```

tunelp (tune line printer)

常用指数: 

功能说明: 改变打印设备的参数。

语 法: tunelp [外围设备代号] [-rs] [-a <on/off>] [-c <缓冲循环>] [-C <on/off>] [-i <中断要求>] [-o <on/off>] [-q <on/off>] [-t <缓冲时间>] [-w <等待循环>]

包 名 称: util-linux。

相关命令: lpc, lpd, lpq, lpr, lprm。

补充说明: 使用tunelp命令可设定连接打印设备的特殊文件, 例如位于/dev目录下的lp0、lp1或lp2, 配合适当的参数调整其设定值能有效改善打印机的效能, 或是排除无法正常运作的原因。仅指定外围设备代号而不给予任何参数, tunelp命令会显示该设备所使用的中断要求 (IRQ, Interrupt Request)。

参 数:

-a <on/off>	若使用“on”开启这个选项, 则打印操作发生错误时, 系统会中断并放弃该操作。默认值为“off”, 系统会反覆尝试将数据送至打印机
-c <缓冲循环>	设定将数据送至打印机缓冲区的等待循环数目, 默认值为250, 对大多数的打印机而言, 120 是个不错的选择。假设打印机有足够的缓冲区 (如激光打印机), 则可设成10
-C <on/off>	若使用“on”开启这个选项, 系统会特别侦测、确认打印机的状况
-i <中断要求>	设定与指定打印机相连的并行端口的中断要求。如果中断要求的设定值不是0, 则参数“-t”和“-c”都将无效
-o <on/off>	此参数的效果和指定“-a”参数类似, 但它会检查指定的打印机是否仍在线操作
-q <on/off>	若使用“on”开启这个选项, 系统会显示指定的打印机所使用的中断要求

-r	重设与打印机相连的并行端口
-s	回报打印机的状态, 该状态以 0~255 的十进制数字表示
-t <缓冲时间>	设定将数据送至打印机缓冲区的等待时间, 单位以百分之一秒计算。默认值为10, 如果不在乎系统负载问题, 可把缓冲时间设成0。假如不在乎打印速度, 或是打印机老旧缓慢, 则500是个不错的设定值
-w <等待循环>	设定打印机忙碌时的等待循环数目, 默认值为0

范 例:

① 查询 /dev/lp0 连接的打印机所使用的中断要求:

```
# tunelp /dev/lp0
/dev/lp0 using IRQ 5
```

② 回报 /dev/lp0 连接打印机的状态:

```
# tunelp /dev/lp0 -s
/dev/lp0 status is 223, on-line
```

③ 将送至与 /dev/lp0 连接之打印机缓冲区的等待时间设为 5 秒钟:

```
# tunelp /dev/lp0 -t 500
```



时间单位为百分之一秒


④ 将送至与 /dev/lp0 连接的打印机缓冲区的等待循环数目设为 10:

```
# tunelp /dev/lp0 -c 10
```

⑤ 重设 /dev/lp0 连接的打印机并行端口:

```
# tunelp /dev/lp0 -r
```

ulimit

常用指数: 

功能说明: 控制shell程序的资源。

语 法: ulimit [-aHS] [-c <core 文件上限>] [-d <数据段大小>] [-f <文件大小>] [-m <内存大小>] [-n <文件数目>] [-p <缓冲区大小>] [-s <堆栈大小>] [-t <CPU 时间>] [-u <程序数目>] [-v <虚拟内存大小>]

包 名 称: bash。

补充说明: ulimit为shell内建命令，可用来控制shell执行程序的资源。

参 数:

-a	显示当前资源限制的设定
-c <core 文件上限>	设定 core 文件的最大值, 单位为区块
-d <数据段大小>	程序数据段 (Data segment) 的最大值, 单位为 KB
-f <文件大小>	shell 所能建立的最大文件, 单位为区块
-H	设定资源的硬性限制 (hard limit), 也就是管理员所设下的限制
-m <内存大小>	指定可使用内存的上限, 单位为 KB
-n <文件数目>	指定同一时间最多可开启的文件数
-p <缓冲区大小>	指定管道缓冲区 (Pipe buffer) 的大小, 单位为 512字节
-s <堆栈大小>	指定堆栈的上限, 单位为 KB
-S	设定资源的弹性限制 (soft limit)。这是由用户在管理员所设的硬性资源限制之下, 自行再加上的限制
-t <CPU 时间>	指定 CPU 使用时间的上限, 单位为秒
-u <程序数目>	用户最多可启动的程序数目
-v <虚拟内存大小>	指定可使用的虚拟内存上限, 单位为 KB

范 例:

① 显示当前资源限制的设定值:

```
# ulimit -a
core file size (blocks)          1000000
data seg size (kbytes)           unlimited
file size (blocks)              unlimited
max locked memory (kbytes)       unlimited
max memory size (kbytes)         unlimited
open files                       1024
pipe size (512 bytes)            8
stack size (kbytes)              8192
cpu time (seconds)               unlimited
max user processes               2048
virtual memory (kbytes)          unlimited
```

② 将core文件大小的上限设为0, 也就是不让系统产生core文件:

```
# ulimit -c 0
```

③ 将数据段的最大值设为5 120 KB:

```
# ulimit -d 5120
```

④ 将shell所能建立的最大文件设为10 240个区块:

```
# ulimit -f 10240
```

⑤ 将可使用内存的上限设为20 480 KB:

```
# ulimit -m 20480
```



⑥ 设定同一时间最多可以开启的文件数为1 024个：

```
# ulimit -n 1024
```


⑦ 设定堆栈的上限为4 096 KB：

```
# ulimit -s 4096
```

⑧ 设定用户最多可启动1 024个程序：

```
# ulimit -u 1024
```

umask

常用指数：

功能说明：指定在建立文件时默认的权限掩码。

语 法：umask [-S] [权限掩码]

包 名 称：bash。

相关命令：chmod。

补充说明：umask可用来设定 [权限掩码]。[权限掩码]是由3个八进制的数字所组成，将现有的存取权限减掉权限掩码后即可产生建立文件时默认的权限。例如，root的权限为777，若[权限掩码]设为022，两者相减后可得755。

参 数：

-S	以文本的方式 (rwx) 来表示权限掩码
----	----------------------

范 例:

① 显示当前用户（此例为root）的权限掩码:

```
# umask
```

```
0022
```

```
# umask -S
```

```
u=rwx,g=rx,o=rx ← 表示同组与其他用户拥有所有的权限
```

② 将权限掩码设为 000:

```
# umask 000
```

```
# umask -S
```

```
u=rwx,g=rwx,o=rwx ← 表示同组与其他用户也有写入的权限
```

umount

常用指数: 

功能说明: 卸除文件系统。

语 法: umount [-ahnrV] [-t <文件系统类型>] [文件系统]

包 名 称: mount。

相关命令: mmount, mount。

补充说明: umount可卸除当前挂在Linux目录中的文件系统。若文件系统当前正处于使用状态（例如，有开启的文件），则必须先等工作处理完后才能卸除文件系统。



参 数:

-a	卸除 /etc/mtab 记录文件中的所有文件系统
-h	显示帮助
-n	卸除时不要将信息存入 /etc/mtab 文件中
-r	若无法成功卸除, 则尝试以只读的方式重新载入文件系统
-t <文件系统类型>	仅卸除选项中所指定的文件系统。文件系统类型有以下几项: minix Linux 最早使用的文件系统 ext2 Linux 当前最常用的文件系统 msdos MS-DOS 的 FAT vfat Windows 95 / 98 的 VFAT nfs 网络文件系统 iso9660 CD-ROM 光盘的标准文件系统 ntfs Windows NT 4.0/2000 的文件系统 hpfs OS/2 文件系统 Windows NT 3.51 之前的版本也使用 HPFS auto 自动侦测文件系统
-v	执行时显示详细的信息
-V	显示版本信息
[文件系统]	除了直接指定文件系统外, 也可以用设备名称 (例如/dev/hda3) 或载入点 (例如 /mnt/dos622) 来表示文件系统

范 例:

① 卸除当前挂在 /mnt/dos622 中的文件系统, 执行时显示详细信息:

```
# umount -v /mnt/dos622
/dev/hda4 umounted
```

② 卸除当前载入的 /dev/fd0 设备的文件系统 (即软驱):

```
# umount /dev/fd0
```

- ③ 卸除/mnt/c文件系统，如果无法卸除，则尝试以只读方式重新载入文件系统：

```
# umount -r /mnt/c
```

```
umount: /dev/hda1 busy - remounted read-only
```

以只读方式重新载入

- ④ 卸除 /etc/mtab 记录文件中的所有文件系统：

```
# umount -a
```

- ⑤ 卸除 /etc/mtab 记录文件中 iso9660 类型的文件系统 (即光驱)：

```
# umount -a -t iso9660
```

指定卸除的 iso9660 文件系统

unalias

常用指数：



功能说明：删除别名。

语 法：unalias [-a] [别名]

包 名 称：bash。

相关命令：alias。

补充说明：unalias 为 shell 内建命令，可删除别名设定。

参 数：


-a	删除全部的别名
----	---------

范 例:**① 删除copy别名:**

```
# alias copy
alias copy='cp -iv'          ← 当前 copy 别名设定
# unalias copy                ← 删除 copy 别名
# alias copy
alias: 'copy' not found      ← 找不到 copy 别名了
```

② 删除所有的别名:

```
# unalias -a
```

uname常用指数: **功能说明:** 显示系统信息。**语 法:** `uname [-amnrsv] [--help] [--version]`**包 名 称:** coreutils。**相关命令:** `lsb_release`, `make`, `patch`。**补充说明:** `uname`可显示电脑以及操作系统的相关信息。

参 数:

-a 或 --all	显示全部的信息。此时会依照 <code>snrvm</code> 参数的顺序将全部信息列出
-m 或 --machine	显示电脑类型
-n 或 --nodename	显示在网络上的主机名称
-r 或 --release	显示操作系统的发行版本(release)
-s 或 --sysname	显示操作系统名称。若不加任何参数,则默认会使用“ <code>uname -s</code> ”
-v 或 --kernel-version	显示操作系统的版本
--help	显示帮助
--version	显示版本信息

范 例:

① 显示操作系统名称:

```
# uname
Linux
```

② 执行以下命令,也有相同结果:

```
# uname -s
```

③ 显示系统在网络上的主机名称:

```
# uname -n
mylinux.flag.com.tw
```

④ 显示操作系统的发行编号:

```
# uname -r
2.4.20-8
```

⑤ 显示操作系统的版本:

```
# uname -v
#1 Thu Mar 13 17:54:28 EST 2003
```


⑥ 显示系统的电脑类型:

```
# uname -m
i686
```

⑦ 显示系统所有的信息:

```
# uname -a
Linux mylinux.flag.com.tw 2.4.20-8 #1 Thu Mar 13 17:54:28
EST 2003 i686 i686 i386 GNU/Linux
```

unrarj

常用指数: 

功能说明: 解压缩 .arj 文件。

语 法: unrarj [eltx] [.arj 压缩文件]

包 名 称: mc (Red Hat Linux)。

补充说明: 此为Red Hat Linux才有的命令。arj的压缩格式是当前常用的压缩格式之一，unrarj则为.arj压缩文件的解压缩程序。

参 数:

e	解压缩 .arj 文件
l	显示压缩文件内所包含的文件
t	检查压缩文件是否正确
x	解压缩时保留原有的路径

范 例:

① 列出replicant.arj压缩文件中所包含的文件:

```
# unarj l replicant.arj
```

```
UNARJ (Demo version) 2.63 Copyright (c) 1991-2000 ARJ Software, Inc.
```

```
Processing archive: replicant.arj
```

```
Archive created: 2003-07-05 14:25:48, modified: 2003-07-05 14:25:48
```

```
Filename      Original Compressed Ratio DateTime modified CRC-32
```

```
-----
```

Filename	Original	Compressed	Ratio	DateTime	modified	CRC-32
DECKARD	38260	12739	0.333	99-11-17	14:40:18	5D6551DF
PRIS	788	515	0.654	99-11-18	13:40:48	EE64FB50
LEON	9675	3691	0.381	99-11-18	14:13:44	D9F9DF0F
ROY	262420	67183	0.256	99-11-11	18:07:30	4AF968D0
RACHAEL	231428	68324	0.295	99-11-18	11:00:46	016DC1AF

```
-----
```

```
5 files 542571 152452 0.281 99-11-19 14:25:48
```

② 测试replicant.arj压缩文件是否正确:

```
# unarj t replicant.arj
```

```
UNARJ (Demo version) 2.63 Copyright (c) 1991-2000 ARJ Software, Inc.
```

```
Processing archive: replicant.arj
```

```
Archive created: 2003-07-05 14:25:48, modified: 2003-07-05 14:25:48
```

```
48
```

Testing	DECKARD	CRC OK
Testing	PRIS	CRC OK
Testing	LEON	CRC OK
Testing	ROY	CRC OK
Testing	RACHAEL	CRC OK

```
5 file(s)
```

③ 解开 replicant.arj 压缩文件:

```
# unarj e replicant.arj
UNARJ (Demo version) 2.63 Copyright (c) 1991-2000 ARJ
Software, Inc.
Processing archive: replicant.arj
Archive created: 2003-07-05 14:25:48, modified: 2003-07-
05 14:25:48
Extracting deckard           Binary file!  CRC OK
Extracting pris              Binary file!  CRC OK
Extracting leon              Binary file!  CRC OK
Extracting roy               Binary file!  CRC OK
Extracting rachael          Binary file!  CRC OK
    5 file(s)
```

uncompress

常用指数:



功能说明: 解压缩 .Z 文件。

语法: uncompress [-cvV] [.Z 压缩文件]

包名称: ncompress (Red Hat Linux/Fedora Core)。

相关命令: compress。

补充说明: 此为Red Hat Linux /Fedora Core才有的命令。uncompress 为.Z 压缩文件的解压缩程序。.Z压缩文件为compress程序所压缩的文件。uncompress将压缩文件解开后,即删除原来的.Z文件。

参数:

-c	将解压缩的结果显示在屏幕上,但不删除.Z文件
-v	执行时显示详细的信息
-V	显示版本信息
[.Z 压缩文件]	只要输入.Z前面的主文件名即可

范 例:

① 解压缩 file.Z 文件:

```
# ls
file.Z          ← .Z 压缩文件
# uncompress -v file ← 解压缩时, 显示命令执行过程
file.Z:  -- replaced with file
# ls
file           ← 解压缩后的文件 (file.Z 文件被删除了)
```

② 解压缩file.Z文件将结果显示在屏幕上, 执行结束后并不删除原始文件:

```
# uncompress -c file.Z
line 1
line 2
line 3
line 4
...
      ] ← 解压缩文件的内容
```

U

uniq

常用指数:



功能说明: 检查及删除文本文件中重复出现的行列。

语 法: `uniq [-cdu] [-f <栏位>] [-s <字符位置>] [-w <字符>] [--help][--version] [输入文件] [输出文件]`

包 名 称: coreutils。

相关命令: sort。

补充说明: uniq可检查文本文件中重复出现的行列。由于uniq使用循序式的比较法, 因此文本文件必须先经过排序 (可利用sort命令), 才能找出重复出现的行列。

若不加任何参数，则会删除重复出现的行列。

参 数：

-c 或 --count	在每行旁边显示该行重复出现的次数
-d 或 --repeated	仅显示重复出现的行列
-f <栏> 或 --skip-fields=<栏>	忽略比较指定的栏
-s <字符位置> 或 --skip-chars=<字符位置>	忽略比较指定的字符
-u 或 --unique	仅显示出现一次的行列
-w <字符> 或 --check-chars=<字符>	指定每行最多要比较前面几个字符
--help	显示帮助
--version	显示版本信息
[输入文件]	指定已排序好的文本文件
[输出文件]	指定输出的文件。若不使用此参数，则会将结果显示在屏幕上

范 例：

① 删除 textfile 中重复出现的行列：

```
# cat textfile          ← 显示 textfile 文件的内容
This is the 1st line
This is the 2nd line
This is the 2nd line
This is the last line
This ix the last line
# uniq textfile         ← 删除重复出现的行列
This is the 1st line
This is the 2nd line
This is the last line
This ix the last line
```

② 删除textfile文件重复出现的行列，比较时不比较每行的第7个字符：

```
# uniq -s 7 textfile
This is the 1st line
This is the 2nd line
This is the last line
```

③ 删除textfile文件重复出现的行列，比较时不比较每行的第2个栏：

```
# uniq -f 2 textfile
This is the 1st line
This is the 2nd line
This is the last line
```

④ 删除textfile文件重复的行列，比较时最多只比较每行前7个字符：

```
# uniq -w 7 textfile
This is the 1st line
This ix the last line
```

← 由于每行只比较7个字符，所以前4行会被视为重复的行列

⑤ 计算textfile文件每行出现的次数：

```
# uniq -c textfile
 1 This is the 1st line
 2 This is the 2nd line
 1 This is the last line
 1 This ix the last line
```

↑
计算的结果

⑥ 显示textfile文件重复出现的行列：

```
# uniq -d textfile
This is the 2nd line
```

⑦ 显示textfile文件只出现一次的行列:

```
# uniq -u textfile
This is the 1st line
This is the last line
This ix the last line
```

unset

常用指数: 

功能说明: 删除变量或函数。

语法: unset [-fv] [变量或函数名称]

包名称: bash。

相关命令: set。

补充说明: unset为shell内建命令, 可删除变量或函数。

参数:

-f	仅删除函数
-v	仅删除变量

范 例:

删除HISTSIZ变量:

```
# set
BASH=/bin/bash
HISTFILESIZE=1000
HISTSIZ=1000
HOME=/root
...
```

← 显示现有的变量



U

```
# unset HISTSIZE          ← 删除 HISTSIZE 变量
# set
BASH=/bin/bash
HISTFILESIZE=1000
HOME=/root                ← HISTSIZE 变量已经删除
...
```

unzip

常用指数: 

功能说明: 解压缩 zip 文件。

语 法: unzip [-cflptuvz] [-abCjLMnoqsVX] [-Z] [-P <密码>] [.zip
文件] [文件] [-d <目录>] [-x <文件>]

包 名 称: unzip。

相关命令: zip, zipinfo。

补充说明: unzip为.zip压缩文件的解压缩程序。zip的压缩格式是当前常用的压缩格式。

第1个参数([-cflptuvz])用来指定unzip所要执行的工作。

第2个参数 ([-abCjLMnoPqsVX] [-P <密码>]) 则指定工作的选项。

参 数:

-c	将解压缩的结果显示到屏幕上,并对字符作适当的转换
-f	更新现有的文件。unzip 会检查压缩文件中有哪些文件与当前目录中的文件重复,若压缩文件中的文件较新,则解压缩这些较新的文件到目录中,覆盖目录中较旧的文件
-l	显示压缩文件内所包含的文件

-p	与“-c”参数类似, 会将解压缩的结果显示到屏幕上, 但不会执行任何的转换
-t	检查压缩文件是否正确
-u	与“-f”参数类似, 但是除了更新现有的文件外, 也会将压缩文件中的其他文件解压缩至目录中
-v	执行时显示详细的信息
-z	仅显示压缩文件的备注文本
-a	对文本文件进行必要的字符转换
-b	不要对文本文件进行字符转换
-C	压缩文件中的文件名称区分大小写
-j	不处理压缩文件中原有的目录路径, 即将所有文件(包含子目录下的文件)解压缩到当前目录
-L	将压缩文件中的全部文件名改为小写, 此功能只适用于压缩此文件的操作系统文件名没有大小写的区别(如 DOS)
-M	将输出结果送至 more 程序处理
-n	解压缩时不要覆盖原有的文件
-o	不必先询问用户, unzip 直接覆盖原有的文件
-P <密码>	使用 zip 的密码选项
-q	执行时不显示任何信息
-s	将文件名中的空白字符转换为底线字符
-V	保留 VMS 的文件版本信息
-X	解压缩时同时回存文件原来的 UID / GID
[.zip 文件]	指定 .zip 压缩文件
[文件]	指定要处理 .zip 压缩文件中的哪些文件。若不加上此参数, 则默认会处理 .zip 压缩文件中的全部文件
-d <目录>	指定文件解压缩后所要存储的目录。默认目录为当前的目录
-x <文件>	指定不要处理 .zip 压缩文件中的哪些文件

范 例:

① 压缩test.zip压缩文件到当前目录下:

```
# unzip test.zip
Archive:  test.zip
  inflating: file
  inflating: file2
  inflating: file3
    creating: subdir/
  inflating: subdir/file1
```

② 将test.zip解压缩到/temp目录下:

```
# unzip test.zip -d /temp
Archive:  test.zip
  inflating: /temp/file
  inflating: /temp/file2
  inflating: /temp/file3
    creating: /temp/subdir/
  inflating: /temp/subdir/file1
```

③ 将test.zip压缩文件除了file2及file3文件外,解压缩到当前目录下:

```
# unzip test.zip -x file2 file3
Archive:  test.zip
  inflating: file
    creating: subdir/
  inflating: subdir/file1
```

④ 列出test.zip压缩文件所包含的文件:

```
# unzip -l test.zip
Archive:  test.zip
  Length      Date    Time    Name
-----
-----
```

```

10240    05-07-03    15:49    file    ← 原始文件的大小、修
                                     改日期及名称

      255    12-07-03    15:24    file2

49844    12-07-03    15:25    file3

      0     12-07-03    15:26    subdir/

      75    05-07-03    15:13    subdir/file1
-----
60414    5 files ← 所有压缩文件大小的
                                     总和、压缩文件及目
                                     录个数的总和

```

⑤ 列出test.zip压缩文件的详细信息:

```

# unzip -z test.zip
Archive:  test.zip 23602 bytes          5 files
-rw-r--r--  2.2 ntf  10240  t-  defN   5-Jul-00  15:49  file
-rw-----  2.2 ntf   255    t-  defN  12-Jul-00  15:24  file2
-rwxr-xr-x  2.2 ntf  49844  t-  defN  12-Jul-00  15:25  file3
drwxr-xr-x  2.2 ntf   0      t-  stor  12-Jul-00  15:26  subdir/
-rw-r--r--  2.2 ntf   75    t-  defN   5-Jul-00  15:13  subdir/
file1
5 files, 60414 bytes uncompressed, 22891 bytes compressed:  62.1%

```

⑥ 测试test.zip压缩文件是否正确:

```

# unzip -t test.zip
Archive:  test.zip
testing:  file           OK
testing:  file2          OK
testing:  file3          OK
testing:  subdir/        OK
testing:  subdir/file1   OK
No errors detected in compressed data of test.zip. ←
                                                    此压缩文件正确无误

```

- ⑦ 解压缩test.zip压缩文件，并指定将压缩文件内子目录下的所有文件全部解压到当前目录：

```
# unzip -j test.zip
```

```
Archive: test.zip
```

```
  inflating: file
```

```
  inflating: file2
```

```
  inflating: file3
```

```
  inflating: file1
```

← 子目录下的文件也解压缩到当前目录

- ⑧ 解压缩test.zip压缩文件，解压时不要覆盖原有的文件：

```
# unzip -n test.zip
```

- ⑨ 解压缩test.zip压缩文件，解压时不需询问用户，直接覆盖原有的文件：

```
# unzip -o test.zip
```

- ⑩ 解压缩test.zip压缩文件，解压后保留原有的UID/GID：

```
# unzip -x test.zip
```

u

updatedb

常用指数：



(update slocate database)

功能说明：更新文件数据库。

语法：updatedb [-quv] [-e <目录...>] [-f <文件系统...>] [-l <0 或 1>] [-U <目录>] [--help] [--version]

包名称：slocate。

相关命令：locate, slocate。

补充说明： updatedb命令可用来更新搜寻文件的数据库，以确保执行 locate及slocate命令时，数据库是最新的。

参 数：

-e <目录...>	建立数据库时，不建立指定目录的数据库
-f <文件系统...>	建立数据库时，不建立指定文件系统的数据库
-l <0 或 1>	设定执行时的安全查核等级。设为0时，关闭安全查核功能，有较快的搜寻速度；若设为1，则启动安全查核功能
-q	执行时显示较少的信息
-u	更新 slocate 数据库。必须定期更新数据库，才能确保正确的搜寻结果。此参数为默认值
-U <目录>	只建立指定目录的数据库
-v	显示命令执行过程
--help	显示帮助
--version	显示版本信息

范 例：

① 更新slocate数据库，但只建立/home目录的数据库：

```
# updatedb -U /home
```


② 更新slocate数据库，但不建立/mnt目录的数据库：

```
# updatedb -e /mnt
```

③ 更新slocate数据库，但不建立属于vfat文件系统的数据库：

```
# updatedb -f vfat
```

useradd

常用指数: 

功能说明: 建立用户账号。

语 法: useradd [-mMnr] [-c <备注>] [-d <登录目录>] [-e <有效期限>] [-f <缓冲天数>] [-g <组>] [-G <组>] [-s <shell>] [-u <uid>] [用户账号] 或 useradd -D [-b] [-e <有效期限>] [-f <缓冲天数>] [-g <组>] [-s <shell>]

包名称: shadow-utils。

相关命令: groupadd, groupdel, groupmod, id, passwd, userdel, usermod。

补充说明: useradd 可用来建立用户账号。账号建好之后, 再用 passwd 命令设定账号的密码。而可用 userdel 删除账号。

使用 useradd 命令所建立的账号, 实际上是存储在 /etc/passwd 文本文件中。

参 数:

-c <备注>	加上备注文本。备注文本会存储在 passwd 的备注栏中
-d <登录目录>	指定用户登录时的启始目录
-D	变更默认值。默认值的选项说明如下: -b <用户目录> 在指定的 <用户目录> 下, 建立所有用户的 <登录目录>。默认值为 /home -e <有效期限> 指定默认账号的有效期限 -f <缓冲天数> 指定默认在密码过期多少天后即关闭该账号 -g <组> 指定用户默认所属的组 -s <shell> 指定使用默认的 shell
-e <有效期限>	指定账号的有效期限。<有效期限> 的格式为月份/日期/公元年后两位
-f <缓冲天数>	指定在密码过期后多少天即关闭该账号

-g <组>	指定用户所属的组
-G <组>	指定用户所属的附加组
-m	自动建立用户的登录目录
-M	不要自动建立用户的登录目录
-n	不建立以用户名称为名的组
-r	建立系统账号
-s <shell>	指定用户登录后所使用的 shell
-u <uid>	指定用户 ID

范 例：

① 建立名称为alkane的用户账号：

```
# useradd alkane
```

② 建立名称为alkane的用户账号，并指定登录目录为 /home/alkane1：

```
# useradd -d /home/alkane1 alkane
```

③ 建立名称为alkane的用户账号，并指定账号的有效期限为2003年12月31日：

```
# useradd -e 12/31/03 alkane
```

④ 建立名称为alkane的用户账号，并指定密码过期10天后即关闭该账号：

```
# useradd -f 10 alkane
```

U

⑤ 建立名称为alkane的用户账号，并指定所属组为users：

```
# useradd -g users alkane
```

⑥ 建立名称为alkane的用户账号，但不建立以alkane为名称的组：

```
# useradd -n alkane ←此时会将alkane所属组设为默认的users组
```

⑦ 建立名称为alkane的用户账号，并指定其登录后所使用的shell为C shell：

```
# useradd -s /bin/csh alkane
```

⑧ 建立名称为alkane的用户账号，并指定其用户识别码为507：

```
# useradd -u 507 alkane
```

⑨ 显示建立账号时的系统默认值：

```
# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
```

⑩ 更改建立账号时，默认使用的shell成C shell：

```
# useradd -D -s /bin/csh
```


⑪ 更改建立账号时，默认建立的用户目录是建在/myhome目录下：

```
# useradd -D -b /myhome
```

U



userconf

常用指数：

功能说明：用户账号设置程序。

语 法：userconf [--addgroup <组>] [--delgroup <组>] [--deluser <用户 ID>] [--help]

包 名 称：linuxconf (Mandrake Linux)。

相关命令：groupadd, groupdel, groupmod, id, useradd, userdel, usermod。

补充说明：此为Mandrake Linux才有的命令。userconf实际上为linuxconf的符号链接，提供图像界面的操作方式，供管理员建立与管理各类账号（用户账号、邮件账号或组等等）。若不加任何参数，即进入图像界面。

参 数：

--addgroup <组>	新增组
--delgroup <组>	删除组
--deluser <用户 ID>	删除用户账号
--help	显示帮助

范 例：

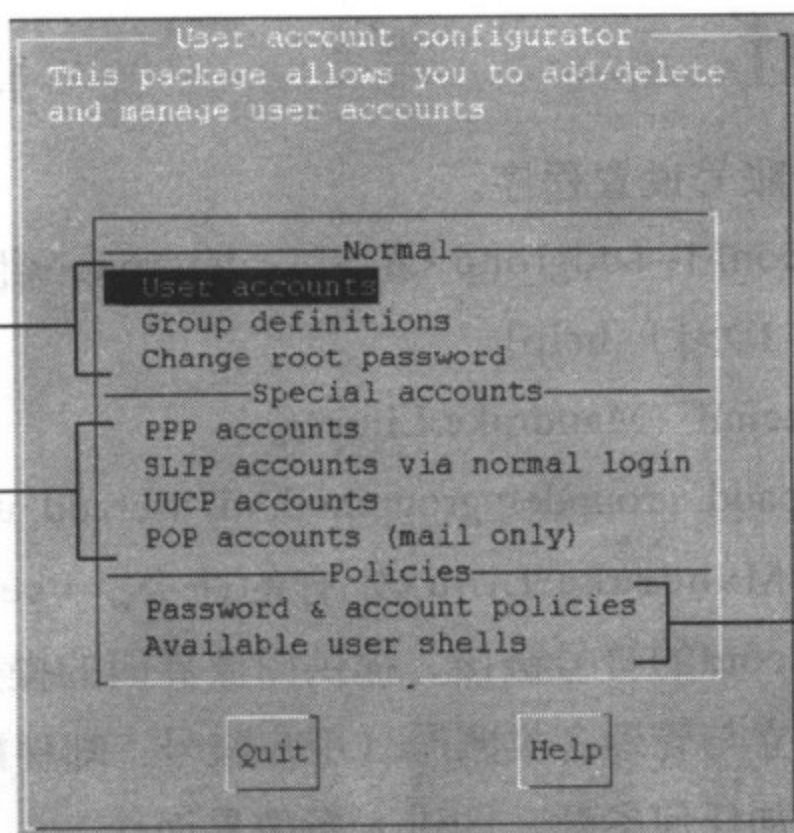
① 在文本模式下执行userconf，会出现以下画面，可管理各类账号：

```
# userconf
```



管理PPP、SLIP、UUCP及POP账号

管理用户、群组及系统管理者的账号



设定账号及密码的默认值，及各种可使用的shell

② 建立名称为newgroup的组：

```
# userconf --addgroup newgroup
```

③ 删除newgroup组：

```
# userconf --delgroup newgroup
```

U

userdel

常用指数：



功能说明：删除用户账号。

语法：userdel [-r] [用户账号]

包名称：shadow-utils。

相关命令：groupadd, groupdel, groupmod, useradd, usermod。

补充说明： userdel可删除用户账号与相关的文件。若不加参数，则仅删除用户账号，而不删除相关文件。

参 数：

-r	删除用户登录目录, 以及目录中所有文件.
----	----------------------

范 例：


① 删除名称为jacket的用户账号：

```
# userdel jacket
```

② 删除名称为jacket的用户账号，同时删除其登录目录：

```
# userdel -r jacket
```

usermod

常用指数：

功能说明： 修改用户账号。

语 法： usermod [-LmU] [-c <备注>] [-d <登录目录>] [-e <有效期限>] [-f <缓冲天数>] [-g <组>] [-G <组>] [-l <账号名称>] [-s <shell>] [-u <uid>] [用户账号]

包 名 称： shadow-utils。

相关命令： groupadd, groupdel, groupmod, id, useradd, userdel。

补充说明： usermod 可用来修改用户账号的各项设定。

参 数:

-c <备注>	修改用户账号的备注文本
-d <登录目录>	修改用户登录时的目录
-e <有效期限>	修改账号的有效期限。<有效期限> 的格式为公元年-月-日 (YYYY-MM-DD)
-f <缓冲天数>	修改在密码过期后多少天即关闭该账号
-g <组>	修改用户所属的组
-G <组>	修改用户所属的附加组
-l <账号名称>	修改用户账号名称
-L	锁定用户密码, 使密码无效
-m	本参数需搭配-d <登录目录> 参数使用, 可将指定目录设为新的用户根目录
-s <shell>	修改用户登录后所使用的 shell
-u <uid>	修改用户 ID
-U	解除密码锁定

范 例:

① 将用户账号jacket改名为 terry:

```
# usermod -l terry jacket
```

② 将terry用户的登录目录改为/home/alkane:

```
# usermod -d /home/alkane terry
```

③ 将terry用户账号的用户识别码改成 505:

```
# usermod -u 505 terry
```


④ 将terry用户登录后所使用的shell设为tcsh:

```
# usermod -s /bin/tcsh terry
```

⑤ 设定terry用户的账号, 当密码过期10天后即关闭该账号:

```
# usermod -f 10 terry
```

⑥ 更改terry用户的所属组成为users组:

```
# usermod -g users terry
```

⑦ 更改terry用户账号的有效期限为2003年9月12日:

```
# usermod -e 09/12/03 terry
```

⑧ 使terry用户的密码失效:

```
# usermod -L terry
```

uucico

常用指数:



功能说明: UUCP 文件传输服务程序。

语 法: uucico [-cCDeflqvwz] [-i <类型>] [-I <文件>] [-p <连接端口号码>] [-r0] [-r1] [-s <主机>] [-S <主机>] [-u <用户>] [-x <类型>] [--help]

包名称: uucp。

相关命令: uucp, uux, uustat, uulog, uuname。

U



补充说明： uucico是用来处理uucp或uux送到队列的文件传输工作。uucico有两种工作模式：主动模式（master）和从属模式（slave）。在主动模式下，uucico会调用远程主机；在从属模式下，uucico则接受远程主机的调用。

参 数：


-c 或 --quiet	当不执行任何工作时，不要更改记录文件的内容，及更新当前的状态（uustat命令能显示当前的状态）
-C 或 --ifwork	当有工作要执行时，才调用“-s”或“-S”参数所指定主机
-D 或 --nodetach	不要与控制终端离线
-e 或 --loop	在从属模式下执行，并且出现要求登录的提示画面。当工作执行完毕时，仍会出现登录提示画面，必须自行执行 kill 命令才可结束程序
-f 或 --force	当执行错误时，不等待任何时间即重新调用主机
-i <类型> 或 --stdin <类型>	当使用到标准输入设备时，指定连接端口的类型。当前支持的类型只有 TLI 界面
-I <文件> 或 --config <文件>	指定使用的设定文件
-l 或 --prompt	出现要求登录的提示画面
-p <连接端口号码> 或 -port <连接端口号码>	指定连接端口号码
-q 或 --nouu	不要启动 uuxqt 服务程序（uuxqt为 uux 服务程序，通常uucico所启动）
-r0 或 --slave	以从属模式启动
-r1 或 --master	以主动模式启动
-s <主机> 或 --system <主机>	调用指定的主机
-S <主机>	立即调用指定的主机。即使执行错误时，亦忽略任何等待的要求，立即重新调用指定的主机。此参数如同“-s <主机> -f”

-u <用户> 或--login <用户>	指定登录的用户账号,而不允许输入任意的登录账号。此参数只有管理者能使用。若与“-l”参数一起使用,则只会要求输入密码,而不会要求输入账号名称
-v 或 --version	显示版本信息,并且结束程序
-w 或 --wait	在主动模式下,执行调用动作后,进入执行“-e”参数时的无限循环,在工作执行完毕时,仍会出现登录提示画面,必须自行执行 kill 命令才可结束程序
-x <类型>、-X <类型> 或--debug <类型>	启动指定的排错模式,排错模式有: abnormal、chat、handshake、uucp-proto、proto、port、config、spooldir、execute、incoming、outgoing等。若要指定两种以上的排错模式,可在模式间用逗号分隔
-z 或 --try-next	当执行不成功时,尝试下一个选择而不结束程序
--help	显示帮助,并且结束程序

范 例:

在主动模式下启动 uucico:

```
# uucico -r1
```

uucp (UNIX to UNIX copy)常用指数: 

功能说明: 在UNIX系统之间传送文件。

语 法: uucp [-cCdfjmrRtvW] [-g <等级>] [-I <设定文件>] [-n <用户>] [-x <类型>] [--help] [来源...] [目的...]

包 名 称: uucp。

相关命令: uucico, uux, uustat, uulog, uupick。

补充说明：UUCP为UNIX系统之间，通过串行线来连线的协议。UUCP衍生出许多种规格，Linux上所移植的UUCP称为Taylor UUCP，可在TCP/IP上执行。UUCP的设定步骤相当繁复，使用前管理员必须先自行编写相关的设定文件。
uucp使用UUCP协议，主要的功能为传送文件。

参 数：

-c 或 --nocopy	不用将文件复制至缓冲区
-C 或 --copy	将文件复制至缓冲区, 此为默认值
-d 或 --directories	在复制文件时, 自动在 [目的...] 下建立必要的目录, 此为默认值
-f 或 --nodirectories	在复制文件时, 若需要在 [目的...] 下建立新目录, 则放弃执行该操作
-g <等级> 或 --grade <等级>	指定文件传送操作的优先顺序。<等级> 可从0至9、A至Z, 最后是a至z。0 的优先顺序最高, z 的优先顺序最低
-I <设定文件> 或 --config <设定文件>	指定 uucp 设定文件
-j 或 --jobid	显示操作编号 (Job ID)
-m 或 --mail	操作结束后, 以电子邮件报告操作是否顺利完成
-n <用户> 或 --notify <用户>	操作结束后, 以电子邮件向指定的用户报告操作是否顺利完成
-r 或 --nouucico	不要立即启动 uucico 服务程序, 仅将操作送至队列中, 待稍后再执行
-R 或 --recursive	若 [来源...] 为目录, 则将整个目录含子目录复制到 [目的...] 中
-t 或 --uuto	将最后一个参数视为“主机名称!用户”
-v 或 --version	显示版本信息
-W 或 --noexpand	不要将当前所在的目录加入路径
-x <类型> 或 --debug <类型>	启动指定的排错模式。uucp 支持 abnormal、config、spooldir 和 execute 等排错模式
--help	显示帮助

[来源...]	指定来源文件或路径。若指定的文件或目录不在本机上, 则格式为“主机名称!路径”
[目的...]	指定目的文件或路径。若指定的文件或目录不在本机上, 则格式为“主机名称!路径”

范 例:

将srdir目录的全部文件传送至远程主机rhost的UUCP公用目录下的tgdir目录下:

```
# uucp -d -R /srdir rhost!~/tgdir
```

↑
“~”表示远程主机中的 UUCP 公用目录

uudecode

常用指数:



功能说明: 将 uencode 所产生的编码文件转换回原来的格式。

语 法: uudecode [-o <文件>] [uencode 文件]

包 名 称: sharutils。

相关命令: uencode。

补充说明: uudecode 可将 uencode 所产生的 ASCII 文件转换回原来的格式。一般通常会将编码的文件扩展名取名为 .uue 以进行区别。

参 数:

-o <文件>	将转换的结果存为指定的文件。若不使用“-o”参数, 则会将结果显示到屏幕上
---------	---------------------------------------

范 例:

将 uuencode 所产生的asciifile转换回原来的格式,并存储为 binfile:

```
# uudecode -o binfile asciifile
```

uuencode

常用指数: 

功能说明: 将文件转换为 ASCII 编码的文件。

语 法: uuencode [-m] [要转换的文件] [输出设备]

包 名 称: sharutils。

相关命令: uudecode。

补充说明: uuencode可将非ASCII编码的文件(例如,二进制编码的文件,或是Big-5码的文件等)转换为ASCII编码的文件,以便在某些只能传送ASCII的媒介上(例如,某些电子邮件通信网关)传输非ASCII的数据。

若要将uuencode所产生的ASCII文件转换回原来的文件,请使用uudecode命令。

参 数:

-m	使用 BASE64 (由 MIME 所定义的一种格式) 的编码方式
----	-----------------------------------



范 例:

① 将binfile转换为ASCII格式并输出到屏幕:

```
# uuencode binfile /dev/stdout
begin 755 /dev/stdout
M5&AI<R!T86<@:7,@=&AE(#%S="!L:6YE+@HR;F0@=&%G(&QI;F4N"C)N9"!T
M86<@;&EN92X*5&AI<R!I<R!T86<@=&AE(#-R9"!L:6YE+@I7:&]L82`T(&QI
M;F4N"C4U-34U-34U-34U-2!T=`HU-34U-34U-34U-34@='0*86QL('1H;W-E
<(&UO;65N=',*86QL('1H97-E(&UO;65N=',*"@"`
`
end
```

② 将binfile转换为ASCII格式并存储成asciifile文件:

```
# uuencode binfile /dev/stdout > asciifile
```

uulog

常用指数:



功能说明: 显示UUCP记录文件。

语 法: uulog [-DFISv] [-<行数>] [-f <主机>] [-I <设定文件>] [-n
<行数>] [-s<主机>] [-u <用户>] [-X <层级>] [--help]

包 名 称: uucp。

相关命令: uucp。

补充说明: uulog可用来显示UUCP记录文件中的记录。由于记录文件通常相当庞大,因此uulog可加上参数,仅显示指定的记录(例如,与某台主机有关的记录)。

参 数:

-D 或 --debuglog	显示排错记录
-f <主机>或--follow=<主机>	与 -F 参数类似,但是仅显示与指定主机相关的记录
-F 或 --follow	持续显示记录文件,若随后有新增的记录,则会显示在最后面
-I <设定文件> 或--config <设定文件>	指定程序的设定文件
-<行数>、-n <行数>或 --lines <行数>	显示记录文件中,从最后面算起指定行数的数据
-s <主机>	仅显示记录文件中与指定主机相关的记录
-S 或 --statslog	显示统计记录
-u <用户> 或--suer <用户>	仅显示记录文件中与指定用户相关的记录
-v 或 --version	显示版本信息
-X <层级> 或--debug <层级>	设定排错层级
--help	显示帮助

范 例:

显示与rhost相关的记录:

```
# uulog -s rhost
```

uuname

常用指数: 

功能说明: 显示全部的 UUCP 远程主机。

语 法: uuname [-alv] [-I <设定文件>] [--help]

包 名 称: uucp。

相关命令： uucp, uucico。

补充说明： uuname 可显示 UUCP 远程主机。

参 数：

-a 或 --aliases	显示别名
-I <设定文件> 或 --config <设定文件>	指定程序的设定文件
-l 或 --local	显示本机名称
-v 或 --version	显示版本信息
--help	显示帮助

范 例：

① 显示所有可经由UUCP来连接的主机名称：

```
# uuname
```

② 显示本地主机的名称：

```
# uuname -l
```

uupick

常用指数：



U

功能说明： 处理传送进来的文件。

语 法： uupick [-v] [-I <设定文件>] [-s <主机>] [-x <层级>] [--help]

包 名 称： uucp。



相关命令： uucp。

补充说明： 当其他主机通过UUCP将文件传送进来时，可利用uupick命令取出这些文件。

参 数：

-I <设定文件> 或--config <设定文件>	指定设定文件
-s <主机> 或--system <主机>	处理由指定主机传送过来的文件
-x <层级> 或--debug <层级>	指定排错层级
-v 或 --version	显示版本信息
--help	显示帮助

范 例：

仅处理由rhost主机所传送进来的文件：

```
# uupick -s rhost
```

uustat

常用指数：



功能说明： 显示 UUCP 当前的状态。

语 法： uustat [-aeiKmMNpqQRv] [-B <行数>] [-c <命令>] [-C <命令>] [-I <设定文件>] [-k <工作>] [-o <小时>] [-r <工作>] [-s <主机>] [-S <主机>] [-u <用户>] [-U <用户>] [-W <附注>] [-y <小时>] [-x <层级>] [--help]

包 名 称： uucp。

相关命令： uucp, uucico。

补充说明： 执行uucp与uux命令后，会先将工作送到队列，再由 uucico 来执行工作。uustat可显示、删除或启动队列中等待执行的工作。

在队列中的工作可分为两类：一类是由uucp所启动，工作内容为传送文件；另一类则是由uux所启动，用来在远程电脑上执行命令。

参 数：

-a 或 --all	显示全部的 UUCP 工作
-B <行数> 或--mail-lines <行数>	与“-M”或“-N”参数一起使用，用来指定邮件中要包含多少行的信息
-c <命令> 或--command <命令>	显示与<命令>有关的工作
-C <命令> 或--not-command <命令>	显示与<命令>无关的工作
-e 或 --executions	仅显示待执行的工作（也就是由 uux 所启动的工作）
-i 或 --prompt	针对队列中的每项工作，询问使用是否要删除工作
-I <设定文件> 或--config <设定文件>	指定设定文件
-k <工作> 或--kill <工作>	删除指定的工作
-K 或 --kill-all	删除全部的工作
-m 或 --status	显示所有远程主机的状态
-M 或 --mail	将状态信息邮寄给 UUCP 管理员
-N 或 --notify	将状态信息分别邮寄给提出该项工作的用户
-o <小时> 或--older-than <小时>	显示队列中在指定时间以后的工作
-p 或 --ps	显示负责 UUCP 锁定程序的状态
-q 或 --list	显示每台远程主机上所要执行工作的状态

-Q 或 --no-list	不显示工作, 仅执行其他参数所指定的动作
-r <工作> 或 --rejuvenate <工作>	重新启动指定的工作
-R 或 --rejuvenate-all	重新启动全部的工作, 不能与“-k”参数合用
-s <主机> 或 --system <主机>	显示与指定主机有关的所有工作
-S <主机> 或 --not-system <主机>	显示与指定主机无关的所有工作
-v 或 --version	显示版本信息
-u <用户> 或 --user <用户>	显示与指定用户有关的所有工作
-U <用户> 或 --not-user <用户>	显示与指定用户无关的所有工作
-W <附注> 或 --comment <附注>	要放在邮件信息中的附注
-y <小时> 或 --younger-than <小时>	显示队列中在指定时间以前的工作
-x <层级> 或 --debug <层级>	指定排错层级
--help	显示帮助

范 例:

① 显示队列中全部的工作:

```
# uustat -a
```

② 删除编号17的工作:

```
# uustat -k 17
```



uuto

常用指数: 

功能说明: 将文件传送至远程的 UUCP 主机。

语 法: uuto [文件] [目的]

包 名 称: uucp。

相关命令: uucp。

补充说明: uuto为script文件, 它实际上会执行uucp, 用来将文件传送至远程UUCP主机, 并在完成工作后, 以邮件通知远程主机上的用户。


参 数: 相关参数请参考uucp命令。

范 例:

将文件复制至远程主机rhost的tgdir目录中, 执行后寄出邮件通知用户是否顺利完成工作:

```
# uuto -m /usr/bin/file2 rhost!tgdir
```

uux

常用指数: 

功能说明: 在远程的 UUCP 主机上执行命令。

语 法: uux [-bcCjlnrvz] [-a <地址>] [-g <等级>] [-I <文件>] [-x <层级>] [--help] [命令]

包 名 称: uucp。



相关命令： uucp, uucico, uustat。

补充说明： uux可在远程的UUCP主机上执行命令，或是执行本机上的命令，但在执行时会使用远程电脑的文件。

参 数：

-、-p 或 --stdin	直接从键盘读取要执行的命令
-a <地址> 或 --requestor <地址>	指定邮件地址,以便寄送状态信息
-b 或 --return-stdin	在屏幕上显示状态信息
-c 或 --nocopy	不用将文件复制至缓冲区,此为默认值
-C 或 --copy	将本机上的文件复制至缓冲区
-g <等级> 或 --grade <等级>	指定文件传送操作的优先顺序。<等级> 可从 0 至 9、A至Z,最后是a至z。0 的优先顺序高,z 的优先顺序最低
-I <文件> 或 --config <文件>	指定 uux 设定文件
-j 或 --jobid	显示操作编号
-l 或 --link	将本机上的文件连接至缓冲区
-n 或 --notification=no	无论发生任何状况,都不寄邮件通知用户
-r 或 --nouucico	不要立即启动 uucico 服务程序,仅将操作送至队列中,待稍后再执行
-v 或 --version	显示版本信息
-x <层级> 或 --debug <层级>	指定排错层级
-z 或 --notification=error	若发生错误,以邮件通知用户
--help	显示帮助

范 例:

- ① 在远程主机rhost上执行“rmail user1”命令，若发生错误，则以邮件通知用户：

```
# uux -z rhost!rmail user1
```

- ② 比较远程主机rhost1上的file1与rhost2上的file2，并将结果存储至本机当前目录的file.diff文件：

```
# uux'diff -c rhost1!~user1/file1 rhost2!~user2/file2  
>!file.diff'
```



vi

常用指数: **功能说明:** 编辑文本文件。**语 法:** vi [-befFghHlLnrRsVZ] [-c <执行命令>] [-d <外围设备代号>] [-o<窗口数目>] [-r <交换文件>] [-s <script 文件>] [-T <终端类型>] [-w <script 文件>] [-W <script 文件>] [--version] [+] [+<行数编号>] [+/<范本样式>] [--] [文件...]**包 名 称:** vim-minimal。**相关命令:** ed, jed, joe, sed, vim。**补充说明:** vi是Linux默认的文本编辑程序,它采用传统的命令界面,而非类似pico的互动式操作界面,因此在使用上较为不便,但功能却相当完善齐备。vi可编辑任何ASCII格式的文本文件,也具备完整的帮助手册,用户只需输入“:help”命令即能查询vi所有编辑命令。若所给予的文件名为“-”,则vi命令会从标准输入设备读取数据。**参 数:**

-b	启动二进制模式
-c <执行命令> 或 +<执行命令>	读取第 1 个文件的内容后,就执行该命令。命令之间若有空格字符,则以双引号标示起来,例如“goto bridge”
-d <外围设备代号>	使用指定的外围设备作为终端。本参数仅在Amiga电脑中方能使用
-e	使用 Ex 单列编辑模式,也可直接执行 ex 命令启动 Ex 模式
-f	使用前台模式,本参数仅在 Amiga 电脑中才有作用
-F	使用波斯 (Farsi/Persian) 模式。中东地区的文本书写方向为从右至左,可使用这项参数采用从右到左的编辑模式

-g	启动图形使用者界面 (Graphical User Interface)。vi 文本编辑程序在编译时必须开启 GUI 的支持, 否则无法使用这个模式
-h 或 --help	显示帮助
-H	使用希伯来 (Hebrew) 模式
-l	使用 Lisp 主机模式
-L	此参数的效果和单独指定“-r”参数 (不加文件名) 相同
-n	不使用交换文件。当在慢速存储设备 (如软盘) 上编辑文件时, 可关闭交换文件的功能, 提升少许效率。但使用这个参数之后, 将无法恢复毁损的文件
-o<窗口数目>	开启指定数目的窗口。举例来说, 窗口数目设成3, 在进入vi时就会开启 3个编辑窗口。假设没有指定窗口数目, 则会根据编辑的文件数目, 开启相等数目的窗口, 比如同时编辑5个文件, 就会开启 5个窗口
-r	列出交换文件, 可利用交换文件恢复某些毁损的文件
-r <交换文件>	使用指定的交换文件, 进入恢复模式。当编辑文件的程序当掉时, 正在编辑的文件可能遭到损坏, 利用交换文件可恢复正在编辑的文件, 交换文件的名称和原来的文件名相同, 差别只在最后的附文件名为“.swp”字符串
-R	使用只读模式, 仍然可以编辑已载入缓冲区的文件内容, 但无法将它保存
-s	寂静模式, 只有在 Ex 单行编辑模式方可使用本参数
-s <script 文件>	读取指定的script 文件的内容
-T <终端类型>	指定终端的类型, 只有在无法正常使用 vi 时, 方加上这项参数
-V	显示命令执行过程
-w <script 文件>	把输入的内容附加到指定的 script 文件
-W <script 文件>	此参数的效果和指定“-w”参数类似, 但会覆盖指定的script 文件
-Z	使用限制模式
--	结束参数设定, 在此参数之后的语句都将被视为文件名称。可借由本参数编辑以“-”号为文件名开头字符的文件
--version	显示版本信息
+	执行 vi 命令进入编辑模式时, 从最后一行开始编辑

+<列数编号>	执行 vi 命令进入编辑模式时, 从指定的行数开始编辑。例如起始行数设定为5, 则一进入编辑画面, 光标便会位于欲编辑的文件的第 5 行。默认值为 1, 即从第 1 行开始编辑
+/<范本样式>	指定字符串作为搜寻依据, 当找到符合条件的字符串时, 光标会停留在含有该字符串的行的开头

范 例:

① 编辑textfile文本文件:

```
# vi textfile
```

② 开启两个窗口, 编辑textfile文本文件, 并在进入vi后从含有“Internet”字符串之行开始编辑:

```
# vi -o2 +/Internet textfile
```

操作命令: vi 命令的操作命令可参考 vim 命令。

vim (vi improved)

常用指数: 

功能说明: 编辑文本文件。

语 法: vim [-bCefFghHlLnNrRsvVZ] [-c <执行命令>] [-d <外围设备代号>] [-i <信息文件>] [-o<窗口数目>] [-r <交换文件>] [-s <script 文件>] [-T <终端类型>] [-u <环境文件>] [-U <环境文件>] [-w <script 文件>] [-W <script 文件>] [--version] [+] [+<行数编号>] [+/<范本样式>] [--] [文件...]

包名称: vim-enhanced。

相关命令: ed, jed, joe, sed, vi。

补充说明：vim是Linux默认文本编辑程序vi的强化版，提供较多的功能，但依旧采用传统命令界面。vim可编辑任何ASCII格式的文本文件，也具备完整的帮助手册，只需输入“:help”命令即能查询vim所有编辑命令。若所给予的文件名为“-”，则vim命令会从标准输入设备读取数据。

参 数：

-b	启动二进制模式
-c <执行命令> 或+ <执行命令>	读取第 1 个文件的内容后,就执行该命令。命令之间若有空格字符,则以双引号标示起来,例如“goto bridge”
-C	开启兼容模式,使用旧版 vi 的编辑命令和环境
-d <外围设备代号>	使用指定的外围设备作为终端。本参数仅在 Amiga 电脑中方能使用
-e	使用 Ex 单行编辑模式,也可直接执行 ex 命令启动 Ex 模式
-f	使用前台模式,本参数仅在 Amiga 电脑中才有作用
-F	使用波斯 (Farsi/Persian) 模式。中东地区的文本书写方向为从右至左,用户可使用这项参数采用从右到左的编辑模式
-g	启动图形用户界面 (Graphical User Interface)。vim 文本编辑程序在编译时必须开启GUI的支持,否则无法使用这个模式
-h 或 --help	显示帮助
-H	使用希伯来 (Hebrew) 模式
-i <信息文件>	另外指定 vim 的信息文件。默认的文件名称为.viminfo,存放在每位使用者的专属目录下,可利用此参数改变它。若不想使用信息文件,可将文件名设为“NONE”
-l	使用 Lisp 主机模式
-L	此参数的效果和单独指定“-r”参数(不加文件名)相同
-n	不使用交换文件。当在慢速存储设备(如软盘)上编辑文件时,可关闭交换文件的功能,提升少许效率。但使用这个参数之后,将无法恢复毁损的文件
-N	关闭兼容模式,不使用旧版 vi 的编辑命令和环境

-o<窗口数目>	开启指定数目的窗口。举例来说, 窗口数目设成3, 在进入 vim 时就会开启 3个编辑窗口。假设没有指定窗口数目, 则会根据编辑的文件数目, 开启相等数目的窗口, 比如同时编辑 5 个文件, 就会开启 5 个窗口
-r	列出交换文件, 可利用交换文件恢复某些毁损的文件
-r <交换文件>	使用指定的交换文件, 进入恢复模式。当编辑文件的程序当掉时, 用户正在编辑的文件可能遭到损坏, 利用交换文件可恢复正在编辑的文件, 交换文件的名称和原来的文件名相同, 差别只在最后的附文件名为“.swp”字符串
-R	使用只读模式, 仍然可以编辑已载入缓冲区的文件内容, 但无法将其存入文件
-s	寂静模式, 只有在 Ex 单列编辑模式方可使用本参数
-s <script 文件>	读取指定script 文件的内容
-T <终端类型>	指定终端的类型, 只有在无法正常使用 vim 时, 方加上这项参数
-u <环境文件>	另外指定vim的环境文件。环境文件提供每位 vim 的使用者设定相关信息, 默认的文件名称为 .vimrc, 存放在使用者的专属目录下, 可利用此参数改变它。若不想使用环境文件, 可将文件名设为“NONE”
-U <环境文件>	此参数的效果和指定“-u”参数类似, 但用于 GUI 模式
-v	使用 vi 模式, 本参数仅在执行 ex 命令时有效
-V	显示命令执行过程
-w <script 文件>	把输入的内容附加到指定的 script 文件
-W <script 文件>	此参数的效果和指定“-w”参数类似, 但会覆盖指定的script 文件
-Z	使用限制模式
--	结束参数设定, 在此参数之后的叙述都将被视为文件名称。可借由本参数, 编辑以“-”号为文件名开头字符的文件
--version	显示版本信息
+	执行vim命令进入编辑模式时, 从最后一行开始编辑
+<列数编号>	执行 vim 命令进入编辑模式时, 从指定的行数开始编辑。例如起始列数设定为5, 则一进入编辑画面, 光标便会位于欲编辑的文件的第 5 行。默认值为 1, 即从第 1 行开始编辑
+/<范本样式>	指定字符串作为查找依据, 当找到符合条件的字符串时, 光标会停留在含有该字符串的行的开头

范 例:

① 编辑textfile文本文件，可使用稍后所介绍的操作命令来编辑文件：

```
# vim textfile
```

编辑区

```
Stop reading here. Go there, then return.

Send bug reports to yura@namesys.botik.ru.

mkreiserfs and other utilities are in reiserfs/utils, or wherever your
Linux provider put them. There is some disagreement about how useful
it is for users to get their fsck and mkreiserfs out of sync with the
version of reiserfs that is in their kernel, with many important
distributors wanting them out of sync.:-) Please try to remember to
recompile and reinstall fsck and mkreiserfs with every update of
reiserfs, this is a common source of confusion. Note that some of the
utilities cannot be compiled without accessing the balancing code
which is in the kernel code, and relocating the utilities may require
you to specify where that code can be found.

Yes, if you update your reiserfs kernel module you do have to
recompile your kernel, most of the time. The errors you get will be
quite cryptic if you forget to do so.

Real users, as opposed to folks who want to hack and then understand
what went wrong, will want REISERFS_CHECK off.

Hideous Commercial Fitch: Spread your development costs across other OS
```

光标所在的列数 光标所在的行数

当前光标位置在
整个文件中所占
位置的百分比

② 以只读模式开启textfile文本文件：

```
# vim -R textfile
```

③ 编辑textfile文本文件，并指定从第100行开始编辑：

```
# vim +100 textfile
```



④ 编辑textfile文本文件，并指定从最后一行开始编辑：

```
# vim + textfile
```

⑤ 编辑textfile文本文件，并将光标停留在含有“information”字符串之列的开头：

```
# vim +/information textfile
```

⑥ 开启两个窗口分别载入textfile及textfile1文本文件：

```
# vim -o2 textfile textfile1
```

⑦ 编辑textfile文本文件，并指定使用根目录下的.myvimrc设定文件：

```
# vim -u ~/.myvimrc textfile
```

操作命令：

切换不同模式

a	进入input mode。从当前光标所在位置的下一个字符开始输入
i	进入input mode。从光标所在位置开始插入新输入的字符
O	进入input mode。新增加一行，并将光标移到下一行的开头
Esc	由input mode切换到command mode
:	切换到last line mode

command mode 的操作命令

h	将光标向左移动一格
i	将光标向右移动一格

J	将光标向下移动一行
K	将光标向上移动一行
O	将光标移到该行的最前面
S	将光标移到该行的最后面
G	将光标移到最后一行的开头
w 或 W	将光标移到下一个单字
e	将光标移到当前所在单字的最后一个字符。如果光标所在的位置为该单字的最后一个字符,则跳到下一个单字的最后一个字符
b	将光标移到当前所在单字的第一个字符。如果光标所在的位置为该单字的第一个字符,则跳到上一个单字的第一个字符
Page Up 或 Ctrl + b	将光标向前移动一页
Page Down 或 Ctrl + f	将光标向后移动一页
Ctrl + u	将光标向前移动半页
Ctrl + d	将光标向后移动半页
Ctrl + e	将屏幕向下移动一行
Ctrl + y	将屏幕向上移动一行
y y	复制光标所在的行
y w	复制光标所在位置到整个单字结束的字符
p	将复制的字贴在当前光标所在的位置
d ← 或 X	删除光标所在位置的前一个字符
d → 或 X	删除光标所在位置的字符
d ↑	删除光标所在的行与前一行
d ↓	删除光标所在的行与下一行
d d	删除光标所在的行
D	删除一行中光标所在位置之后的所有字符

r	取代光标所在的字符
R	连续取代字符, 直到按 Esc 为止
Ctrl + g 或 Ctrl + G	显示总行数, 及光标当前位置在整个文件中所占位置的百分比
g G	“#”为数字, 将光标移到指定的行数。如按下 g G 表示将光标移到第 9 行
u	取消前一个动作
last line mode 的操作命令	
e <文件名称>	建立新文件
n <文件名称>	载入另外一个文件
w <文件名称>	存储文件
q	结束程序
q !	不管文件是否曾被修改过, 均强制结束程序
w	存储文件并结束程序

vlock (virtual console lock)

常用指数:



功能说明: 锁住虚拟控制台 (Console)。

语 法: vlock [-achv]

包 名 称: vlock。

补充说明: 执行vlock命令可锁住虚拟控制台, 避免他人使用。控制台多半是指全屏幕的终端或虚拟终端, 也就是所谓的TTY。单独执行vlock命令, 会锁住当前正在使用的控制台阶段操作, 待使用者输入密码后方能再度使用。

参 数:

-a 或 --all	锁住所有的主控台阶段操作, 如果在全屏幕的主控台中使用本参数, 则会用键盘切换终端的功能一起关闭
-c 或 --current	锁住当前的主控台阶段操作, 此为默认值
-h 或 --help	显示帮助
-v 或 --version	显示版本信息

范 例:

① 锁住当前执行的主控台阶段操作:

```
# vlock
```

```
This TTY is now locked.
```

```
Use Alt-function keys to switch to other consoles.
```

```
Please enter the password to unlock.
```

```
root's Password: ← 需要输入账号密码, 才能解除锁定状态
```

② 在全屏幕的主控台中, 锁住所有的主控台阶段操作:

```
# vlock -a
```

```
The entire console display is now completely locked.
```

```
You will not be able to switch to another virtual console.
```

```
Please enter the password to unlock.
```

```
root's Password:
```

W

常用指数: **功能说明:** 显示当前登录系统的用户信息。**语 法:** w [-fhlsuV] [用户名称]**包 名 称:** procps。**相关命令:** who, finger, free, ps, procinfo, top, tload。**补充说明:** 执行这项命令可得知当前登录系统的用户有哪些人, 以及他们正在执行的程序。单独执行w命令会显示所有的用户, 也可指定用户名称仅显示某位用户的相关信息。除登录的用户信息外, 还可得知现在的系统时间、系统已启动多久、当前共有多少用户登录系统, 还有在过去1、5、10分钟内系统的平均负载程度。**参 数:**

-f	开启或关闭显示用户从何处登录系统。本项功能在 Red Hat Linux/Fedora Core 默认为开启状态, 因此用户可通过这个参数关闭它, 不显示从何处登录系统的信息
-h	不显示各栏的标题行
-l	使用详细格式列表, 此为默认值
-s	使用简洁格式列表, 不显示用户登录的时间、终端阶段操作和程序所耗费的 CPU 时间
-u	忽略执行程序的名称, 以及该程序耗费 CPU 时间的信息
-V	显示版本信息

范 例:

① 查询现在有哪些用户登录系统:

```
# w
10:25:31 up 24 min, 5 users, load average: 0.15, 0.35, 0.50
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU   WHAT
sphenoid  tty3     -             10:11am    44.00s 0.22s  0.04s  joe
irchen    tty5     -             10:18am    36.00s 0.15s  0.08s  pine
alkane    pts/3    shadow.flag. 10:13am    16.00s 0.09s  0.03s  -bash
root      pts/4    -             10:13am    0.00s  0.07s  0.02s  w
```

↑
若是Mandrake Linux, 则默认不会显示此栏

② 查询现在有哪些用户登录系统, 但不显示用户从何处登录系统的信息:

```
# w -f
10:27:47 up 26 min, 5 users, load average: 0.07,
0.26, 0.44
USER      TTY      LOGIN@      IDLE   JCPU   PCPU   WHAT
sphenoid  tty3     10:11am    2:42   0.22s  0.04s  joe
irchen    tty5     10:18am    2:34   0.15s  0.08s  pine
alkane    pts/3    10:13am    2:14   0.09s  0.03s  -bash
root      pts/4    10:13am    0.00s  0.09s  0.03s  w -f
```

↑
与上例比较, 少了 "FROM" 栏

③ 查询现在有哪些用户登录系统, 显示时不列出各栏的标题行:

```
# w -h
← 没有列出标题行
sphenoid  tty3     -             10:11am    1:18   0.22s  0.04s  joe
irchen    tty5     -             10:18am    1:10   0.15s  0.08s  pine
alkane    pts/3    shadow.flag. 10:13am    50.00s 0.09s  0.03s  -bash
root      pts/4    -             10:13am    0.00s  0.07s  0.02s  w-h
```

④ 以较精简的格式列出现在有哪些用户登录系统:

```
# w -s
 10:26:35 up 25 min,  5 users,  load average: 0.08, 0.29,
0.46
USER      TTY      FROM          IDLE      WHAT
sphenoid  tty3    -             1:54     joe
irchen    tty5    -             1:46     pine
alkane    pts/3   shadow.flag.  1:26     -bash
root      pts/4   -             0.00s    w -s
```

少了 "LOGIN@"、"JCPU" 及 "PCPU" 栏

⑤ 查询当前登录的alkane用户:

```
# w alkane
 10:35:29 up 34 min,  5 users,  load average: 0.00, 0.05, 0.25
USER      TTY      FROM          LOGIN@    IDLE      JCPU      PCPU      WHAT
alkane    pts/3   shadow.flag.  10:13am  10:22    0.09s    0.03s    -bash
```

wait

常用指数:



功能说明: 等待程序回报其状态。

语法: wait [程序识别码或命令名称]

包名称: bash。

补充说明: wait命令用于等待某个程序的返回值, 假如所给予的是命令名称, 则会等待该命令下所有子程序的返回值。若不指定任何识别码或名称, 则wait命令会等待正在活动的所有子程序并返回状态代码“0”, 假设指定不存在的识别码或名称, wait命令会返回状态代码“127”。


参 数: 无。

范 例:

等待识别码为1263的程序的返回值:

```
# wait 1263
```

wall (write all)

常用指数: 

功能说明: 传送信息。

语 法: wall [公告信息]

包 名 称: SysVinit。

相关命令: mesg, write。

补充说明: 通过wall命令可将信息发送给每位同意接收公众信息的终端用户, 若不给予其信息内容, 则wall命令会从标准输入设备读取数据, 然后再把所得到的数据传送给所有终端用户。

参 数: 无。

范 例:

① 发送信息“Give Me a Reason to Live!”的信息给所有终端用户:

```
# wall Give Me a Reason to Live!
```

```
Broadcast message from root (tty1)(Thu Jul 3 16:48:53 2003)
```

```
Give Me a Reason to Live!
```



- ② 将message.txt文本文件内的文本以信息的方式传送给所有终端用户:

```
# wall < message.txt
```

WC (word count)

常用指数:



功能说明: 计算字数。

语 法: wc [-clLmw] [--help] [--version] [文件...]

包名称: coreutils。

补充说明: 利用wc命令, 可以计算文件的byte数、字数、字符数或是行数, 若不指定任何文件名称, 或是所给予的文件名为“-”, 则wc命令会从标准输入设备读取数据。假设不给予参数, wc命令会一并显示行数、字数和byte数。

参 数:

-c 或 --bytes	只显示 byte 数
-l 或 --lines	只显示行数
-L 或 --max-line-length	显示文件中最长一行的字符数
-m 或 --chars	只显示字符数
-w 或 --words	只显示字数
--help	显示帮助
version	显示版本信息

范 例:

- ① 计算textfile文本文件的行数:

```
# wc -l textfile
```

```
66 textfile
```

← 此文件有 66 行

② 计算textfile文本文件的字数：

```
# wc -w textfile
  174 textfile
```

← 此文件包含了 174 个字

③ 计算textfile文本文件的byte数：

```
# wc -c textfile
 1291 textfile
```

← 此文件大小为 1291 bytes

④ 计算textfile文本文件的列数、字数及byte数：

```
# wc textfile
   66   174  1291 textfile
```

⑤ 计算textfile文本文件中最长一行的字符数：

```
# wc -L textfile
   77 textfile
```

← 此文件最长一行的长度为 77个字符

W

wget

常用指数：



功能说明：从互联网 (WWW) 下载文件。

语 法：wget [-bcdFhHkLmNqrsSvVx] [-a <记录文件>] [-A <扩展名>,...] [-B <基础连接>] [-C <on/off>] [-D <领域名称>,...] [-e <执行命令>] [-g <on/off>] [-i <URL 文件>] [-I <目录>,...] [-l <目录层级>] [-nc] [-nd] [-nh] [-nH] [-np] [-nr] [-nv] [-o <记录文件>] [-O <输出文件>] [-P <目录字首字符串>] [-Q <空间限制>] [-R <扩展名>,...] [-t <尝试次数>] [-T <逾时秒数>] [-U <识别标签>] [-w <间隔秒数>] [-X <目录>,...] [-Y <on/off>] [--delete-after] [--dot-style=<显示样式>]

[--exclude-domains=<领域名称>,...] [--follow-ftp] [--header=<文件头字符串>] [--http-passwd=<密码>] [--http-user=<用户名称>] [--ignore-length] [--passive-ftp] [--proxy-passwd=<密码>] [--proxy-user=<用户名称>] [--retr-symlinks] [--spider] [URL...]

包名称: wget。

相关命令: lynx。

补充说明: wget是用来大量抓取网络数据的程序，它十分类似MS Windows的Teleport软件，但不具备交互式操作界面。wget支持 HTTP和FTP传输协议，并可在用户未登录系统的状况下于后台中执行，也就是可利用计时器让它在某些时间自行启动下载网络数据。

参 数:

-a<记录文件>或--append-output=<记录文件>	把信息输出至记录文件内，而非标准输出设备。默认会记录所有命令执行过程，可用参数“-nv”让它只输出更新及错误信息
-A <扩展名> 或--accept=<扩展名>	设定接受哪些文件的扩展名，两两之间以“,”号间隔
-b 或 --background	在后台进行下载操作
-B <基础连接> 或--base-herf=<基础连接>	设定基本参考的连接地址
-c 或 --continue	从上次离开 wget 程序的地方继续下载文件
-C <on/off> 或--cache=<on/off>	开启或关闭服务器数据缓存，默认值为“on”
-d 或 --debug	详细显示命令执行过程，以便于排错
-D <领域名称> 或--domains=<领域名称>	设定接受哪些领域名称，两两之间以“,”号间隔

-e <执行命令> 或 --execute=<执行命令>	启动 wget 后就执行该命令, 其效果和在与 .wgetrc 环境文件中设定执行命令相同
-F 或 --force-html	将输入的文件视为 HTML 格式
-g <on/off> 或 --glob=<on/off>	开启或关闭 FTP 通过通配字符下载文件。例如设定欲下载的文件名称为 "*.jpeg", 则会下载所有最后扩展名为 ".jpeg" 的文件
-h 或 --help	显示帮助
-H 或 --span-hosts	递归处理时, 允许跟随连接指向下载数据
-i <URL 文件> 或 --input-file=<URL 文件>	从指定的文件里读取 URL
-I <目录> 或 --include-directories=<目录>	设定接受哪些目录, 两两之间以 "," 号间隔
-k 或 --convert-links	把不相关的连接转换成相关连接
-l <目录层级> 或 --level=<目录层级>	设定递归处理深入目录的最大层级, 默认值为 5, 即从指定的地址开始向下深入 5 层目录, 如设成 0 表示没有限制
-L 或 --relative	只下载有关联的连接
-m 或 --mirror	此参数的效果和同时指定 "-rN" 参数相同
-nc 或 --no-clobber	不覆盖已有的文件
-nd 或 --no-directories	不建立目录, 所有下载的文件都会存放在当前目录中。如果名称重复, 则会在文件后面依次加上扩展名 ".1"、".2"、".3" ...
-nh 或 --no-host-lookup	不花费时间在名称服务器上寻找主机
-nH 或 --no-host-directories	不建立主机名称的目录
-np 或 --no-parent	不下载指定地址的上层目录的数据
-nr 或 --dont-remove-listing	不删除 ".listing" 文件
-nv 或 --non-verbose	不详细显示命令执行过程, 只输出更新及错误信息
-N 或 --timestamping	不下载比本地主机的文件更旧的文件。假设远程主机上的文件的异动时间与本地主机的文件相同或更旧, 且文件大小相同, 则 wget 将不会下载该文件

-o <记录文件> 或--output-file=<记录文件>	此参数的效果和指定“-a”参数类似,但指定的记录文件若已存在,则wget会覆盖该文件,而非仅将信息附加在文件的后面
-O <输出文件> 或--output-document=<输出文件>	设定存放输出数据的文件名称,若所给予的文件名为“-”,则 wget 命令会把所得到的数据输出至输出设备
-P <目录字首字符串> 或--directory-prefix=<目录字首字符串>	设定目录名称的字首字符串,默认为“.”号,就是当前目录
-q 或 --quiet	不显示命令执行过程
-Q <空间限制> 或--quota=<空间限制>	设定下载数据的磁盘空间限制,单位以Byte计算,可在空间限制的后面加上“K”或“M”字符,表示用KB或MB计算
-r 或 --recursive	递归处理,将指定目录下的所有文件及子目录一起处理
-R <扩展名> 或--reject=<扩展名>	设定排除哪些文件的扩展名,两两之间以“,”号间隔
-s 或 --save-headers	把 HTTP 的文件头存成文件
-S 或 --server-response	列出 HTTP 或 FTP 服务器的数据文件头或响应数据
-t <尝试次数> 或--tries=<尝试次数>	设定反复尝试的次数,当尝试次数设为0时,则没有次数限制
-T <超时秒数> 或--timeout=<超时秒数>	设定读取数据的时间限制,单位以秒计算,默认为 900 秒
-U <识别标签> 或--user-agent=<识别标签>	使用自行设定的识别标签,而非默认的“Wget”字符串加上其版本编号
-v 或 --verbose	显示命令执行过程,此为默认值
-V 或 --version	显示版本信息
-w <间隔秒数> 或--wait=<间隔秒数>	设定两次下载操作间的间隔时间,单位以秒计算
-x 或 --force-directories	强制建立目录
-X <目录> 或--exclude-directories=<目录>	设定排除哪些目录,两两之间以“,”号间隔

-Y<on/off>或 --proxy=<on/off>	开启或关闭代理服务器, 默认为 “on”
--delete-after	删除已下载的文件
--dot-style=<显示样式>	设定下载操作进行时的显示样式
--exclude-domains=<领域名称>	设定排除哪些领域名称, 两两之间以 “,” 号间隔
--follow-ftp	从 HTML 文件中下载 FTP 连接的文件
--header=<文件头字符串>	在文件头加入指定的字符串
--http-passwd=<密码>	设定登录 HTTP 服务器的密码
--http-user=<用户名称>	设定登录 HTTP 服务器的账号
--ignore-length	忽略文件头 “Content-Length” 字符串
--passive-ftp	使用 PASV 模式连接数据, 只有在搭配安全系统而遭遇问题时, 才需附加这项参数
--proxy-passwd=<密码>	设定登录代理服务器的密码
--proxy-user=<用户名称>	设定登录代理服务器的账号
--retr-symlinks	下载 FTP 的符号链接
--spider	不下载任何数据

范 例:

① 下载 www.flag.com.tw 网站首页的数据:

```
# wget www.flag.com.tw
```

```
--18:08:25-- http://www.flag.com.tw/
```

```
=> `index.html`
```

```
正在查找主机 www.flag.com.tw... 完成。
```

```
正在连接 www.flag.com.tw[210.62.132.10]:80... 连上了。
```

```
已送出 HTTP 要求, 正在等候响应... 200 OK
```

```
长度: 23,327 [text/html]
```

```
100%[======>] 23,327 605.33B/s ETA 00:00
```

```
18:09:04 (605.33 B/s) — 已存储 `index.html` [23327/23327]
```

② 下载 `www.flag.com.tw` 首页往下5层的网页数据:

```
# wget -r -l 5 www.flag.com.tw
```

③ 下载 `www.flag.com.tw` 网页数据, 并将其存储在 `/home/alkane/downloadWEB` 目录下:

```
# wget -P /home/alkane/downloadWEB www.flag.com.tw
```

④ 不断尝试地从 `www.flag.com.tw` 下载网页数据:

```
# wget -t 0 www.flag.com.tw
```

⑤ 在后台下载 `www.flag.com.tw` 网页数据:

```
# wget -b www.flag.com.tw
```

⑥ 下载 `www.flag.com.tw` 网页数据, 但不通过代理服务器下载:

```
# wget --proxy=off www.flag.com.tw
```

⑦ 下载 `www.flag.com.tw` 网页数据, 但不下载指定的图形文件类型:

```
# wget -R .jpg,.gif,.jpeg www.flag.com.tw
```

⑧ 下载 `www.flag.com.tw` 网页数据, 并限定最多只能使用20MB的磁盘空间:

```
# wget -Q 20M www.flag.com.tw
```

⑨ 下载 `www.flag.com.tw` 网页数据, 并指定不要产生记录文件:

```
# wget -o /dev/null www.flag.com.tw
```

⑩ 下载 `www.flag.com.tw` 网页数据, 执行时不显示命令执行过程:

```
# wget -nv www.flag.com.tw
```

whereis (where is file)

常用指数: 

功能说明: 查找文件。

语法: whereis [-bfmsu] [-B <目录>...] [-M <目录>...] [-S <目录>...] [文件...]

包名称: util-linux。

相关命令: find, locate, slocate, which。

补充说明: whereis命令会在特定目录中查找符合条件的文件，这些文件的类型会属于原始码、二进制文件或是说明文件。若不刻意指定目录，whereis默认会去寻找的目录如下：

- /bin
- /usr/etc
- /usr/local
- /usr/local/bin
- /usr/local/man
- /etc
- /usr/games
- /usr/man
- /usr/local/etc
- /sbin
- /usr/include
- /usr/sbin
- /usr/local/games
- /usr/bin
- /usr/lib
- /usr/src
- /usr/local/lib

找到合乎条件的文件之后，whereis会把它们全部列示出来。

参 数:

-b	只查找二进制文件
-B <目录>	只在设定的目录下查找二进制文件
-f	不显示文件名前的路径名称。本参数需配合“-B”、“-M”或“-S”使用
-m	只查找帮助文件
-M <目录>	只在设定的目录下寻找帮助文件
-s	只查找源代码文件
-S <目录>	只在设定的目录下寻找原始码文件
-u	寻找不包含指定类型的文件

范 例:

① 找出lilo命令与其说明文件的存放位置:

```
# whereis lilo
lilo: /sbin/lilo /usr/share/man/man8/lilo.8.gz
```

② 找出lilo命令的二进制文件:

```
# whereis -b lilo
lilo: /sbin/lilo
```

③ 找出lilo命令的说明文件:

```
# whereis -m lilo
lilo: /usr/share/man/man8/lilo.8.gz
```

④ 找出/bin目录下所有文件的说明文件:

```
# whereis -m /bin/*
arch: /usr/share/man/man1/arch.1.gz
ash: /usr/share/man/man1/ash.1.gz
ash: /usr/share/man/man1/ash.1.gz
...
```

which

常用指数:



功能说明: 查找文件。

语 法: which [文件...]

包 名 称: which。

相关命令: find, locate, slocate, whereis。

补充说明: which命令会在环境变数\$PATH设定的目录里查找符合条件的文件。

参 数: 无。

范 例:


① 查找lilo文件:

```
# which lilo
/sbin/lilo      ← 列出文件所在的位置
```

② 查找bash2与pico等两个文件:

```
# which bash2 pico
/bin/bash2
/usr/bin/pico
```

who

常用指数: 

功能说明: 显示当前登录系统的用户信息。

语 法: who [am i] [-bdHimqrsuw] [--help] [--version] [记录文件]

包 名 称: coreutils。

相关命令: finger, mesg, w, rwho。

补充说明: 执行这项命令可得知当前有哪些用户登录系统, 单独执行who命令会列出登录账号、使用的终端、登录时间, 以及从何处登录或正在使用哪个X显示器。可在who命令之后加上记录文件的名称, 取代存放于/var/run目录里的utmp文件 (默认值), 通常是指定/var/log目录下的wtmp文件, 用来查询以前的用户登录记录。假设加在who命令后面的字符串是“am i”, 则会显示本地主机的名称和本身账号的相关信息。

参 数:

am i	只列出与自己相关的登录信息
-b 或 --boot	显示系统最近一次启动的时间
-d 或 --dead	显示已死掉的程序 (dead processes)
-H 或 --heading	显示各栏位的标题行
-i、-u或--idle	显示闲置时间, 若该用户在前一分钟之内有进行任何动作, 将标示成“.”号, 如果该用户已超过24小时没有任何动作, 则标示出“old”字符串
-m	此参数的效果和指定“am i”字符串相同
-q 或 --count	只显示登录系统的账号名称和总人数。使用本参数将忽略其他任何参数
-r 或 --runlevel	显示当前的执行等级 (runlevel)
-s	此参数将忽略不予处理, 仅负责解决who命令其他版本的兼容性问题
-w、-T、--mesg、--message 或--writable	显示用户的信息状态栏。信息的状态标示共有3种: “+”、“-”和“?”, 分别表示接受 write 命令传来的信息、不接受 write 命令的信息, 以及找不到终端外围设备
--help	显示帮助
--version	显示版本信息

范 例:

① 列出当前登录系统的用户:

```
# who
tony      tty1      Jul      3 11:30
alkane    tty4      Jul      3 11:32
root      pts/0     Jul      3 10:53 (192.168.0.33)
root      pts/1     Jul      3 11:09 (192.168.0.39)
```


② 列出当前登录系统的用户，并显示各栏的标题行：

```
# who -H
```

NAME	LINE	TIME	COMMENT
tony	tty1	Jul 3 11:30	
alkane	tty4	Jul 3 11:32	
root	pts/0	Jul 3 10:53	(192.168.0.33)
root	pts/1	Jul 3 11:09	(192.168.0.39)

③ 列出当前登录系统的用户，并显示标题列及闲置时间：

```
# who -Hu
```

NAME	LINE	TIME	IDLE	PID	COMMENT
tony	tty1	Jul 3 11:30	00:08	1916	
alkane	tty4	Jul 3 11:32	00:02	1919	
root	pts/0	Jul 3 10:53	.	2024	(192.168.0.33)
root	pts/1	Jul 3 11:09	00:09	2220	(192.168.0.39)

④ 列出当前登录系统的用户，并显示标题行及信息状态栏：

```
# who -Hw
```

NAME	LINE	TIME	COMMENT
tony	+ tty1	Jul 3 11:30	
alkane	+ tty4	Jul 3 11:32	
root	- pts/0	Jul 3 10:53	(192.168.0.33)
root	- pts/1	Jul 3 11:09	(192.168.0.39)

⑤ 列出当前登录系统用户的名称以及总人数：

```
# who -q
```

```
tony alkane root root
```

```
# users=4
```

⑥ 显示系统最近一次启动的时间：

```
# who -b
      system boot  Jul  3 10:05
```


⑦ 显示系统当前的执行等级 (runlevel)：

```
# who -r
      run-level 5  Jul  3 10:05                last=S
```

↑
当前的执行等级为 5

⑧ 列出自己登录的信息：

```
# who am i
      root          pts/0          Jul  3 10:53          (192.168.0.33)
      |             |             |             |
      账号名称      登录的终端      登录的时间      登录的地址
```

常用指数：

whoami

功能说明：显示用户名称。

语 法：whoami [--help] [--version]

包 名 称：coreutils。

相关命令：id。

补充说明：显示自己的用户名称，本命令相当于执行“id -un”命令。

参 数:

--help	显示帮助
--version	显示版本信息

范 例:

列出自己的用户名称:

```
$ whoami
kinglore
```

whois

常用指数: 

功能说明: 查找并显示指定网址的相关信息。

语 法: whois [--help][--verbose][网址]

包 名 称: jwhois (Red Hat Linux/Fedora Core), whois (Mandrake Linux)。

相关命令: jwhois。

补充说明: whois命令会向whois服务器查询并显示指定网址的相关信息。当不知道某个网址是属于哪一个单位时,即可通过此命令查询。

参 数:

--help	显示帮助
--verbose	显示较详细的信息

范 例:

向whois服务器的数据库查询66.218.71.88网址的相关信息:

```
# whois 66.218.71.88
[查询 whois.arin.net]
[whois.arin.net]
```

```
OrgName:      Yahoo!           ← 此网址的注册单位为 Yahoo!
OrgID:        YAOO
Address:      701 First Avenue
City:         Sunnyvale      StateProv: CA
PostalCode:   94089
Country:     US
NetRange:    66.218.64.0 - 66.218.95.255
CIDR:        66.218.64.0/19
...
```

其他相关信息

write

常用指数:



功能说明: 传送信息。

语 法: write [用户名称] [终端编号]

包 名 称: util-linux。

相关命令: mesg, wall。

补充说明: 通过write命令可传递信息给另一位登录系统的用户，当输入完毕后，键入EOF表示信息结束，write命令就会将信息传给对方。假如接受信息的用户不只登录本地主机一次，用户可以指定接收信息的终端编号。

参 数: 无。

范 例:

① 发送信息给另一位登录系统的用户alkane:

write alkane

I love you! ← 输入完毕后, 用 **Ctrl + c** 或 **Ctrl + d** 键送出信息

② 发送信息给第 5 号终端的 alkane 用户:

write alkane tty5

本手册用于描述... 关于用户... 关于系统... 关于网络... 关于安全... 关于性能... 关于兼容性... 关于扩展性... 关于可维护性... 关于易用性... 关于可靠性... 关于安全性... 关于性能... 关于兼容性... 关于扩展性... 关于可维护性... 关于易用性... 关于可靠性... 关于安全性...



...	...
...	...
...	...
...	...
...	...
...	...



xauth (X authority)

常用指数: 

功能说明: 编辑修改X server授权信息。

语法: xauth [-biqv] [-f <授权文件>] [命令...]

包名称: XFree86-xauth (Red Hat Linux/Fedora Core 1), xorg-x11-xauth(Fedora Core 2&3), XFree86 (Mandrake Linux 1~10), xorg-x11 (Mandrake Linux 10.1)。

相关命令: xhost。

补充说明: 本程序用于编辑、显示连接X server的授权信息, 它可解开某台主机上的授权信息记录, 并将该记录与另一个记录连接。xauth能使用命令进行交互式操作, 也可应用在script文件中。除了在xauth互动模式下使用generate命令外, xauth并不会和X server进行沟通, 一般而言它并不负责产生记载授权信息的文件, xdm才负责此类工作。

参数:

-b	解开授权文件的防护限制
-f <授权文件>	指定授权文件, 默认的文件名称为 .Xauthority
-i	忽略授权文件的防护限制
-q	关闭额外信息
-v	启动额外信息

范例:

① 执行xauth命令进入命令行互动模式, 可利用稍后介绍的操作命令存取授权信息:

```
# xauth                                     Xauthority授权
Using authority file /root/.Xauthority. ← 文件所在路径
xauth>                                     ← xauth命令行互动模式提示符号
```

- ② 在xauth的命令提示符号下，列出当前用户所拥有的Magic Cookie (X server和X client的沟通密码)：

```
xauth> list
```

```
wki.samu.org/unix:0 MIT-MAGIC-COOKIE-1 284ea0a21192d793c22a62be7508242d
```

```
localhost:0 MIT-MAGIC-COOKIE-1 fd2a9e618ab9744969e75458b5ff770d
```

```
wki.samu.org:0 MIT-MAGIC-COOKIE-1 284ea0a21192d793c22a62be7508242d
```

显示设备名称

产生Cookie文件的协议

沟通密码值，每个字符为十六进制数值 (0-F)，占用4位(bit)，32个数值共128位

```
xauth> exit
```

← 离开xauth命令行互动模式，若有任何异动将会自动存储

- ③ 把显示设备local:1的沟通密码值df6a7e6kkab9711999e75458b5fe63e0加入授权文件中：

```
xauth> add local:1 MIT-MAGIC-COOKIE-1 df6a7e6kkab9711999e75458b5fe63e0
```

增加设备

显示设备名称

协议类别

沟通密码值

```
xauth> list
```

```
wki.samu.org/unix:0 MIT-MAGIC-COOKIE-1 284ea0a21192d793c22a62be7508242d
```

```
localhost:0 MIT-MAGIC-COOKIE-1 fd2a9e618ab9744969e75458b5ff770d
```

```
wki.samu.org:0 MIT-MAGIC-COOKIE-1 284ea0a21192d793c22a62be7508242d
```

```
local:1 MIT-MAGIC-COOKIE-1 df6a7e6kkab9711999e75458b5fe63e0
```

新增项目已经加入

```
xauth> exit
```

← 离开 xauth 命令行互动模式，自动存储变更的设定

- ④ 不使用默认的授权文件.Xauthotity，另行指定/tmp目录下的.anotherXauth文件为授权文件：

```
# xauth -f /tmp/.anotherXauth
```

操作命令:

?	列出可用的命令
add <显示设备><协议类别><密码值>	新增 Magic Cookie
exit	离开命令行模式, 内容若有异动会自动存储
extract <密码文件><显示设备...>	将指定的设备内容存放到指定的密码文件
help <命令名称>	列出指定命令的说明, 若没有给予命令名称, 则会列出所有命令的说明
info	显示密码文件相关信息
list <显示设备...>	列出指定显示设备的内容, 若没有给予设备名称, 则会列出所有显示设备的内容
merge <密码文件...>	合并多个密码文件的内容
nextract <密码文件> <显示设备...>	效果类似extract命令, 但输出格式为十六进制数值, 而非ASCII码
nmerge<密码文件...>	效果类似merge命令, 但输入格式为十六进制数值, 而非ASCII码
quit	效果类似 exit 命令, 但不会存储修改过的内容
remove<显示设备...>	移除指定的设备的Magic Cookie
source <命令文件>	从指定的文件内读取 xauth 的命令

xf86config

常用指数: **功能说明:** 设置 X Window。**语 法:** xf86config**包 名 称:** XFree86 (Mandrake Linux 1~10), xorg-x11(Mandrake Linux 10.1)。**补充说明:** 这是Mandrake Linux用来设定X Window的程序, 它具有互动式的问答界面, 让用户轻松完成X Window环境的设定。可通过xf86config程序, 设定鼠标与键盘、显卡和显示器, 并选择欲使用的画面分辨率与色彩深度。

参 数: 无。

范 例:

使用xf86config程序设置X Window:

xf86config

```
This program will create a basic XF86Config file, based on menu selections you
make.

The XF86Config file usually resides in /usr/X11R6/etc/X11 or /etc/X11. A sample
XF86Config file is supplied with XFree86; it is configured for a standard
VGA card and monitor with 640x480 resolution. This program will ask for a
pathname when it is ready to write the file.

You can either take the sample XF86Config as a base and edit it for your
configuration, or let this program produce a base XF86Config file for your
configuration and fine-tune it.

Before continuing with this program, make sure you know what video card
you have, and preferably also the chipset it uses and the amount of video
memory on your video card. SuperProbe may be able to help with this.

Press enter to continue, or ctrl-c to abort.
```

欲进行设定, 请
按 **Enter** 键继续

若欲取消设定, 请
按 **Ctrl + c** 键离开

xfs (X Font Server)

常用指数:



功能说明: 提供 X Window 字体服务器。

语 法: xfs [-config <配置文件>] [-daemon] [-port <通信端口>]

包 名 称: XFree86-xfs, xorg-x11-xfs (Fedora Core 2&3/Mandrake Linux 10.1)。

相关命令： ttmkfdir, xset。

补充说明： xfs是一个常驻服务，它提供XWindow字体方面的需求，可修改/etc/rc.d/rc.local，让xfs在系统开机时就自动执行。

参 数：

-config <配置文件>	指定配置设定文件，默认的配置文件的位于 /usr/X11R6/lib/X11/fs 目录下，文件名称为 config
-daemon	以常驻服务的类型启动 X Font Server
-port <通信端口>	指定通信端口编号，默认 xfs 采用第 7100 通信端口

范 例：

① 启动X Font Server提供X Window字体服务：

```
# xfs -daemon
```

② 另行指定配置文件xfsconfig供xfs使用：

```
# xfs -config /home/tony/xfsconfig
```

文件存放路径 配置设定文件名称

③ 使用第7000通信端口提供X Font Server的服务：

```
# xfs -port 7000
```

xhost

常用指数: 

功能说明: 控制存取 X server 的主机。

语 法: xhost [+/-] [+/-<主机名称或 IP 地址>...]

包 名 称: XFree86, xorg-x11-xfs (Fedora Core 2&3/Mandrake Linux 10.1)。

补充说明: 利用本命令可新增、删除允许存取 X server 的主机, 以提供基本的保密控管与系统安全。执行 xhost 命令时, 如不赋予任何参数, 则会显示当前存取控制的相关信息。

参 数:

+	允许任何主机存取本地端的 X server, 即关闭存取控制
-	只允许列于授权清单内的主机得以存取本地端的 X server, 即启动存取控制
+<主机名称或IP地址>	允许某台主机存取本地端的 X server, 也就是把该主机加入授权清单里。前面的“+”号可以省略
-<主机名称或IP地址>	不允许某台主机存取本地端的 X server, 也就是把该主机从授权清单里移除

范 例:

① 列出当前有哪些主机可存取本地端的 X server:

```
# xhost
```

```
access control enabled, only authorized clients can connect
```

```
INET:205.47.203.53 ←允许存取 X server 的主机, 此处列出其 IP 地址
```

② 允许主机 shougo.linker.edu 存取本地端 X server:

```
# xhost +shougo.linker.edu
```

此处必须使用有效的主机名称,即在 DNS (Domain Name System) 有登录的名称,倘若没有合法名称,可用该主机的 IP 地址替代
shougo.linker.edu being added to access control list ←
加入授权清单

③ 移除主机shougo.linker.edu对本地端X server的存取权限:

```
# xhost -shougo.linker.edu
```

```
shougo.linker.edu being removed from access control list
```

④ 开放任何主机皆可存取本地端 X server:

```
# xhost +
```

```
access control disabled, clients can connect from any host ←
```

存取控管已经关闭, 可从任何主机进行存取动作

⑤ 重新启动存取控管, 只接受在授权清单里主机的存取要求:

```
# xhost -
```

```
access control enabled, only authorized clients can connect
```

xlsatoms

常用指数:



功能说明: 列出X Server定义的成分。

语法: `xlsatoms [-display <显示器编号>] [-format <输出格式>] [-name <成分名称>] [-range <列表范围>]`

包名称: XFree86, xorg-x11(Fedora Core 2&3/Mandrake Linux 10.1)。

补充说明: 执行xlsatoms命令会列出X Server内部所有定义的成分，每个成分都有其编号，可利用参数设定列表范围，或直接指定欲查询的成分名称。

参数:

<code>-display <显示器编号></code>	指定X Server连接的显示器编号, 该编号由0开始计算, 依次递增
<code>-format <输出格式></code>	设定成分清单的列表格式, 可使用控制字符改变显示样式, 默认值为“%ld\t%s”
<code>-name <成分名称></code>	列出指定的成分, 若不包含该成分, 则会出现错误信息
<code>-range <列表范围></code>	设定成分清单的列表范围。例如指定为“10-300”, 就只会列出编号10~300的成分名称, 设成“-300”会从第1个成分列至第300个, “10-”则会由第10个成分列至最后一个

范例:

① 列出X Server第30~250号的定义成分:

```
# xlsatoms -range 30-250
30      RGB_RED_MAP
31      STRING
32      VISUALID
...
```

② 列出名称为“ISO8859”的成分：

```
# xlsatoms -name ISO8859
211 ISO8859
```

xlsclients

常用指数：



功能说明：列出显示器中的客户端应用程序。

语法：xlsclients [-al] [-display <显示器编号>] [-m <最大命令长度>]

包名称：XFree86, xorg-x11(Fedora Core 2&3/ Mandrake Linux 10.1)。

补充说明：执行xlsclients命令可列出某个显示器中正在执行的客户端应用程序信息。

参数：

-a	列出所有显示器的客户端应用程序信息
-display <显示器编号>	指定X Server连接的显示器编号, 该编号由0开始计算, 依次递增
-l	使用详细格式列表。加上这项参数之后, xlsclients命令会将窗口、图示、主机名称, 以及Class提示一并列出
-m <最大命令长度>	设定显示命令信息的最大长度, 单位以字符计算。默认值为10 000字符

范例：

① 利用xlsclients命令, 查询有哪些X Window的客户端应用程序正在执行：



```
# xlsclients
free.flag.com.tw  xcin
free.flag.com.tw  /usr/bin/gnome-session
free.flag.com.tw  gnome-settings-daemon
...
free.flag.com.tw  /usr/libexec/notification-area-applet
free.flag.com.tw  gnome-terminal
free.flag.com.tw  xcin
```

② 执行xlsclients命令，详细列出正在执行的客户端应用程序：

```
# xlsclients -l
Window 0x600006:
  Machine: free.flag.com.tw
  Name: xcin 2.5.3-pre2
  Icon Name: xcin
  Command: xcin
  Instance/Class: xcin/xcin
Window 0x400001:
  Machine: free.flag.com.tw
  Name: gnome-session
  Command: /usr/bin/gnome-session
  Instance/Class: gnome-session/Gnome-session
...
```



③ 列出所有显示器的客户端应用程序信息：

```
# xlsclients -a
free.flag.com.tw  xcin
free.flag.com.tw  /usr/bin/gnome-session
...
```



- ④ 查询有哪些X Window的客户端应用程序正在执行，并限制显示信息的最大长度为10个字符：

```
# xlsclients -m 10
free.flag.com.tw  xcin
free.flag.com.tw  /usr/bin/g
free.flag.com.tw  gnome-sett
...
free.flag.com.tw  /usr/libex
free.flag.com.tw  gnome-term
free.flag.com.tw  xcin
```

超过 10 个字符的部分都会被截除

X

xlsfonts

常用指数：



功能说明：列出 X Server 使用的字体。

语 法：xlsfonts [-lCImou] [-n <显示栏数>] [-w <每行字符数>]
[-display <主机名称或 IP 地址>:<显示器编号>] [-fn <范
本样式>] [-ll] [-lll]

包 名 称：XFree86, xorg-x11(Fedora Core 2&3/Mandrake Linux 10.1)。

补充说明：执行xlsfonts命令会显示当前X Server可使用的字体，也能使用范本样式仅列出符合条件的字体。



参 数:

-l	仅用单一栏列出字体清单。此参数的效果和指定“-n 1”参数相同
-C	使用多个栏列出字体清单。此参数的效果和指定“-n 0”参数相同
-display<主机名称或 IP 地址>:<显示器编号>	指定显示画面的主机名称,以及欲在该主机上的哪个显示器显示画面。显示器编号由0开始依次递增
-fn <范本样式>	只列出符合范本样式指定条件的字体,可使用通配字符“*”或“?”号设定范本样式
-l	除字体名称外,同时列出字体的属性(此参数为小写的“L”)
-ll	此参数的效果和指定“-l”参数类似,但显示更详细的信息
-lll	此参数的效果和指定“-ll”参数类似,但显示最详细的信息
-m	配合参数“-l”使用时,一并列出字体大小的上下限
-n <显示栏位数>	设定每列显示的栏位数,默认 xlsfonts 会自动根据参数“-w”所指定的字符数目,决定要显示几个栏
-o	以OpenFont的形式列出字体清单
-u	列出字体清单时不依照其名称排序
-w <每行字符数>	设定每行的最大字符数。默认值为每行79个字符,可用这项参数去改变它

范 例:

① 列出所有字体的信息:

```
# xlsfonts
-2rebels-adventure-semi bold-r-normal--0-0-0-0-p-0-ascii-0
-2rebels-beast wars-medium-r-normal--0-0-0-0-p-0-ascii-0
-2rebels-embargo-medium-r-normal--0-0-0-0-p-0-ascii-0
...
variable
vga12x24
vga8x16
```

② 详细列出所有字体的信息：

```
# xlsfonts -l
DIR  MIN  MAX  EXIST  DFLT  PROP  ASC  DESC  NAME
-->  0  255  some  0   34  8    4 -2rebels-embargo-medium-r-normal-
-0-0-0-0-p-0-ascii-0
-->  0  255  some  0   27  11   4 -adobe-courier-bold-i-normal-
-0-0-0-0-m-0-iso8859-1
-->  0  255  some  0   27  11   4 -adobe-courier-bold-i-normal-
-0-0-0-0-m-0-iso8859-2
...
-->  0  255  some  0   27  11   3 variable
-->  0  255  all   32  21  22   2 vga12x24
-->  1  255  some  32  21  14   2 vga8x16
```

③ 列出名称里包含“vga”字符串的字体详细信息：

```
# xlsfonts -l -fn vga*
DIR  MIN  MAX  EXIST  DFLT  PROP  ASC  DESC  NAME
-->  0  255  all   32  21  22   2 vga12x24
-->  1  255  some  32  21  14   2 vga8x16
```

④ 使用多栏位简单列出名称包含“vga”字符串的字体信息：

```
# xlsfonts -C -fn vga*
vga8x16          vga12x24
```

X



xplaymidi

常用指数: 

功能说明: 播放音乐文件。

语 法: xplaymidi [-48adefglkMrvwz] [-A <频道掩码>] [-c <频道掩码>] [-C <混音层级>] [-D <周边设备代号>] [-E <频道掩码>] [-F <频道掩码>] [-G <频道掩码>] [-i <频道掩码>] [-p <频道编号,><乐器种类>...] [-P <频道编号>] [-R <初始化层级>] [-t <节拍速度>] [-V <频道编号,> <播放速度>...] [-x <频道编号>] [MIDI 音乐文件...]

包 名 称: playmidi-X11 (Mandrake Linux)。

相关命令: playmidi, sndconfig。

补充说明: xplaymidi是Mandrake Linux才有的命令。MIDI是乐器和电脑之间的标准沟通界面，它记录着每个播放频道所对应的乐器种类与节拍速度等信息。利用xplaymidi命令可以播放MIDI音乐文件。若不给予任何文件名称或参数，则会显示帮助。

参 数:

-4	输出至OPL-3音源晶片
-8	在UltraSound声卡上, 强制使用8位取样
-a	输出至AWE32声卡
-A <频道掩码>	把指定的频道以AWE32声卡播放
-c <频道掩码>	仅播放指定的频道, 不播放其他频道的音乐。频道掩码需用十六进制指定

-C <混音层级>	设定混音初始化的层级, 范围从 0~127
-d	不播放任何打击乐器, 包括鼓声
-D <周边设备代号>	使用指定的 MIDI 设备播放音乐
-e	输出至外部 MIDI 音源器
-E <频道掩码>	用指定的频道以外部的 MIDI 音源器播放
-f	输出至 FM 音源
-F <频道掩码>	把指定的频道以 FM 音源播放
-g	输出至 UltraSound 声卡
-G <频道掩码>	把指定的频道以 UltraSound 声卡播放
-i <频道掩码>	忽略指定的频道, 不播放该频道的音乐。频道掩码需用十六进制字码设定
-l	列出 GM 乐器种类的编号对照表
-k	删除所有正在使用 /dev/sequencer 的程序
-M	把 MT-32 的频道对照表转换成 GM 的频道对照表
-p <频道编号,>	用指定的乐器播放设定的频道。若没有设定频道, 则<乐器种类>会用该乐器播放所有的频道。乐器种类的范围从 1~128
-P <频道编号>	在指定的频道播放所有打击乐器
-r	播放音乐时, 以 ANSI 绘图字符即时显示相关信息
-R<初始化层级>	设定 reverb 初始化的层级, 范围从 0~127
-t <节拍速度>	设定播放时的节拍速度
-v	显示命令执行过程
-V <频道编号> <播放速度>	设定频道的播放速度。若没有指定频道, 则所有的频道都会以设定的速度为准
-w	等待所有正在使用 /dev/sequencer 的程序
-x <频道编号>	在可播放的频道掩码中排除指定的频道
-z	忽略所有的频道。本参数仅供特别的应用程序使用

范 例:

- ① 播放MIDI音乐文件xgmusic.midi, 把第4、9频道分别指定成第55、82号乐器, 并将第7、10频道的速度设定为128:

```
# xplaymidi -p 4,55,9,82 -v 7,128,10,128 xgmusic.midi
```

连续设定频道的播放速度, 组别之间用“,”号区隔
连续设定频道使用的乐器, 组别之间用“,”号区隔

- ② 在AWE32声卡上播放music.midi音乐文件:

```
# xplaymidi -a music.midi
```

```
Playmidi 2.4 Copyright (C) 1994-1997 Nathan I. Laredo, AWE32  
by ...
```

```
This is free software with ABSOLUTELY NO WARRANTY.
```

```
For details please see the file COPYING.
```

- ③ 使用外部的音源器播放valkyrie.midi音乐文件:

```
# xplaymidi -e valkyrie.midi
```

- ④ 列出GM乐器所有种类的编号对照表:

```
# xplaymidi -l
```

```
Playmidi 2.4 Copyright (C) 1994-1997 Nathan I. Laredo, AWE32 by ...
```

```
This is free software with ABSOLUTELY NO WARRANTY.
```

```
For details please see the file COPYING.
```

```
Gravis Ultrasound Program Info:
```

```
1 acpiano 2 britepno 3 synpiano 4 honky 5 epiano1 6 epiano2
```

```
7 hrpschrd 8 clavinet 9 celeste 10 glocken 11 musicbox 12 vibes
```

```
.....
```



- ⑤ 在AWE32声卡上播放music.midi音乐文件，但不播放任何打击乐器的声音：

```
# xplaymidi -a -d music.midi
```

- ⑥ 在AWE32声卡上，指定以第10频道播放music.midi音乐文件：

```
# xplaymidi -a -P10 music.midi
```

- ⑦ 在UltraSound声卡播放music.midi音乐文件，并指定不使用第5频道播放：

```
# xplaymidi -g -x5 music.midi
```

- ⑧ 在UltraSound声卡上，以82的混音层级播放music.midi音乐文件：

```
# xplaymidi -g C82 music.midi
```

- ⑨ 使用FM音源播放whatsgoingon.midi音乐文件：

```
# xplaymidi -f whatsgoingon.midi
```

- ⑩ 强制UltraSound声卡使用8位取样，播放trackboy.midi音乐文件：

```
# xplaymidi -g -8 trackboy.midi
```

x



xset

常用指数: **功能说明:** 设置 X Window 的使用偏好。**语 法:** xset [b <on/off>] [c <on/off>] [dpms <省电状态>] [fp default] [led <on/off>] [r <on/off>] [s <on/off>] [q] [-display <显示设备>] [<+/->dpms] [<+/->fp <字体路径>]**包 名 称:** xset , xorg-x11(Fedora Core 2&3/Mandrake Linux 10.1)。**补充说明:** 本程序可调整与 X Window 相关部分的设定。**参 数:**

b <on/off>	开、关电脑蜂鸣器的声音
c <on/off>	开、关键盘按键的声响
dpms <省电状态>	决定DPMS的省电状态, 其值可以是“standby”、“suspend”或“off”
fp default	恢复默认的字体查找路径
led <on/off>	开、关键盘上的 LED 指示灯
r <on/off>	开、关自动反复发送按键信号
s <on/off>	开、关屏幕保护功能
q	显示当前与 xset 相关的设定
-display <显示设备>	指定欲使用的显示设备
<+/->dpms	开、关 DPMS 省电功能
<+/->fp <字体路径>	增、删字体查找路径

X

范 例:

① 检视当前与xset相关的系统状态:

```
# xset q
```

Keyboard Control:

```
auto repeat: on    key click percent: 0    LED mask: 00000000
```

```
auto repeat delay: 500    repeat rate: 30
```

```
auto repeating keys: 00ffffffdffffbbf
```

```
fadffffffffffdfe5ff
```

```
fffffffffffffffffff
```

```
fffffffffffffffffff
```

```
bell percent: 50    bell pitch: 400    bell duration: 100
```

Pointer Control:

```
acceleration: 2/1    threshold: 4
```

Screen Saver:

```
prefer blanking: yes    allow exposures: yes
```

```
timeout: 600    cycle: 600
```

Colors:

```
default colormap: 0x20    BlackPixel: 0    WhitePixel: 16777215
```

Font Path:

```
/root/.gnome2/share/cursor-fonts,unix/:7100,/root/.gnome2/share/fonts
```

Bug Mode: compatibility mode is disabled

DPMS (Energy Star):

```
prefer blanking: yes    allow exposures: yes
```

```
timeout: 600    cycle: 600
```

...

② 利用xset命令, 加上字体查找路径/usr/X11R6/lib/X11/fonts/TTF/:

```
# xset +fp /usr/X11R6/lib/X11/fonts/TTF/
```


③ 利用xset命令，移除字体查找路径/usr/X11R6/lib/X11/fonts/TTF/:

```
# xset -fp /usr/X11R6/lib/X11/fonts/TTF/
```

④ 恢复字体查找路径的默认值:

```
# xset fp default
```

⑤ 开启电脑蜂鸣器与键盘按键的声音:

```
# xset b on c on
```

⑥ 关闭屏幕保护与DPMS省电功能:

```
# xset s off dpms off
```

x



yes

常用指数:

**功能说明:** 响应字符串。**语法:** `yes [--help] [--version] [响应字符串...]`**包名称:** coreutils。**补充说明:** `yes`命令会不断响应所给予的字符串，直到该程序被删除为止。**参数:**

<code>--help</code>	显示帮助
<code>--version</code>	显示版本信息

范 例:① 利用`yes`命令重复响应 “You can change the future... Beyond the time...”

字符串:

```
# yes You can change the future... Beyond the time...
You can change the future... Beyond the time...
You can change the future... Beyond the time...
You can change the future... Beyond the time...
.....
```

② 执行 “`rm -i`” 命令时，利用`yes`命令自动响应 “y” 字符，删除文件:

建立管道，把 `yes` 命令的输出结果转为 `rm` 命令的输入数据

```
# yes y | rm -i *
rm: 是否移除普通文件 'README'?
rm: 是否移除普通文件 'lock.txt'?
rm: 是否移除普通文件 'outfile.txt'?
.....
```

自动回答 “y” 删除文件

ytalk

常用指数:



功能说明: 与其他用户对话。

语法: ytalk [-isxY] [-h <主机名称或 IP 地址>] [用户名称...]

包名称: ytalk。

相关命令: mesg, talk。

补充说明: 借由ytalk命令可以和其他用户线上对话，假如想和其他主机的用户对话，在用户名称后加上其主机名称或IP地址即可。例如用户名称为thief，主机名称是lady.heart.org，完整的用户名称就是“thief@lady.heart.org”，假设还要指定终端编号（如第5台终端），完整名称则为“thief@lady.heart.org#tty5”或“thief#tty5@lady.heart.org”。假如该用户所登录的是本地主机，在其名称之后直接加上终端编号，就能指定特定的终端，比如“thief#tty5”。基本上ytalk和talk非常类似，但它可同时与多个用户对话，而非一对一对话。

参数:

-h <主机名称或 IP 地址>	指定对话对象所在的远程主机
-i	用提醒声响代替显示信息
-s	在命令提示符号下开启 ytalk 对话框
-x	关闭图像界面。当在X Window的文本模式窗口里执行 ytalk 命令时，默认会自动显示图像界面对话框，可用此参数关闭该功能
-Y	所有必须响应“yes”或“no”的问题，都必须用大写英文字母“Y”或“N”回答

范 例:

① 和第3个终端的用户 karashim 对话:

```
$ ytalk karashim#tty3
```

接收端用户karashim所看到的画面如下:

```
$
```

```
Message from Talk_Daemon@shadowside.org.ru at 11:53 ...
talk: connection requested by red-oct@shadowside.org.ru.
talk: respond with: talk red-oct@shadowside.org.ru
```

```
talk red-oct@shadowside.org.ru
```

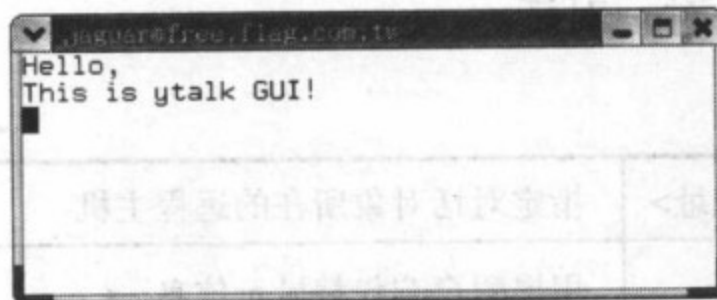
← 欲接受对话的要求, 请输入响应字符串

② 和mission.to.mars.org主机上的用户front3对话:

```
# ytalk front3@mission.to.mars.org
```

③ 在X Window里使用虚拟终端执行ytalk命令, 与本地主机用户 jaguar对话, 即出现图像界面的对话框:

```
# ytalk jaguar
```



如不需要以图像界面显示, 请使用参数“-x”:

```
# ytalk -x jaguar
```

yuvsplittoppm

常用指数:



(yuv to portable pixmap)

功能说明: 转换视频文件。**语 法:** yuvsplittoppm [Y 视频文件] [U 视频文件] [V 视频文件]
[宽度] [高度] [-ccir601]**包 名 称:** netpbm-progs (Red Hat Linux/Fedora Core), netpbm
(Mandrake Linux)。**相关命令:** yuvtoppm。

补充说明: 执行yuvsplittoppm命令时, 它将读取指定的YUV格式视频文件 (默认最后的扩展名为“.Y”、“.U”和“.V”), 并把该文件转成Portable Pixmap (默认最后的扩展名为“.ppm”) 格式后, 输出到标准输出设备。由于YUV视频文件是一种RAW格式, 因此必须指定其宽度与高度, 单位以像素计算。可在PAL (Phase Alternation Line) 类比电视标准规格或数码相机视频规格里见到YUV, 它和RGB的基本转换公式如下所示:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ 0.1684 & -0.3316 & 0.5000 \\ 0.5000 & -0.4178 & -0.0813 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0000 & 0.0000 & 1.4020 \\ 1.0000 & -0.3441 & -0.7139 \\ 1.0000 & 1.7718 & 0.0000 \end{bmatrix} \cdot \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

参 数:

-ccir601	转换时采用CCIR (ITU-R) 601的标准, 它的平行扩展标准是CCIR 656, 当前已被转换成众所周知的MPEG (Moving Picture Experts Group)标准
----------	--

范 例:

① 把视频文件digicam.Y、digicam.U与digicam.V转成Portable Pixmap格式:


```
# yuvsplittoppm digicam.Y digicam.U digicam.V 640 480 > digicam.ppm
```

② 采用CCIR 601标准, 把视频文件digivideo.Y、digivideo.U与digivideo.V转成Portable Pixmap格式:

```
# yuvsplittoppm digivideo.Y digivideo.U digivideo.V 360 288 -ccir601 > digivideo.ppm
```

Y

yuvtoppm

常用指数: 


(yuv to portable pixmap)

功能说明: 转换视频文件。**语 法:** yuvtoppm [宽度] [高度] [Abekas YUV 视频文件]**包名称:** netpbm-progs (Red Hat Linux/Fedora Core), netpbm (Mandrake Linux)。**相关命令:** yuvsplittoppm。**补充说明:** 执行yuvtoppm命令时, 它将读取指定的Abekas YUV格式视频文件, 并把该文件转成Portable Pixmap (默认最后的扩展名为.ppm) 格式后, 输出到标准输出设备。由于YUV视频文件是一种RAW格式, 因此必须指定其宽度与高度, 单位以像素计算。**参 数:** 无。**范 例:**

把Abekas视频系统产生的YUV视频文件videosystem转成Portable Pixmap格式:

```
# yuvtoppm videosystem 800 600 > videosystem.ppm
```

zcat (gzip concatenate)

常用指数: 

功能说明: 连接多个压缩文件, 并将它们的内容输出到标准输出设备。

语 法: zcat [-fhLV] [文件...]

包 名 称: gzip。

相关命令: cat, gzip, gunzip。

补充说明: zcat命令会读取所给予的压缩文件的内容, 并将其内容输出在标准的输出设备上面。若不指定任何文件名称, 或是所给予的文件名为“-”, 则zcat命令会从标准输入设备读取数据, 然后再把所得到的数据输出至输出设备。也可运用shell的特殊字符“>”和“>>”, 把多个文件的内容合并成一个文件。事实上, zcat就是gunzip的硬链接, 其效果相当于“gunzip -c”命令。

参 数:

-f 或 --force	强行解开压缩文件
-h 或 --help	显示帮助
-L 或 --license	显示版本与版权信息
-V 或 --version	显示版本信息

范 例:

① 文件README.gz是一个压缩文件, 用zcat命令直接观看其内容:

```
# zcat README.gz ← 原先的内容必须是纯文本文件, 否则会看见一堆乱码
```

② 解开压缩文件textfile1.gz与textfile2.gz, 并利用特殊字符“>”将它们合成一个文件textfile3.gz:


```
# zcat textfile1.gz textfile2.gz > textfile3.gz
```

zip

常用指数: **功能说明:** 压缩文件。**语 法:** zip [-AcdDfFghjJklLmoqrSTuvVwXy\$] [-b <工作目录>] [-n <字尾字符串>] [-t <日期时间>] [-<压缩率>] [-ll] [压缩文件] [文件...] [-i <范本样式>] [-x <范本样式>]

Z

包 名 称: zip。**相关命令:** unzip, zipinfo。**补充说明:** zip是一个使用广泛的压缩程序,其版本横跨十多种操作系统与硬件架构平台,文件经它压缩过后,会另外产生具有“.zip”扩展名的压缩文件,解压缩时则使用unzip命令。zip和另一个知名压缩程序pkzip (Phil Katz's ZIP)兼容,不论压缩或解压缩。若所给予的文件名为“-”或“-@”,则zip命令会从标准输入设备读取数据;若欲将压缩结果输出到输出设备,则请指定压缩文件的名称为“-”号。假设欲压缩、解压的文件是硬链接或符号链接,zip命令会读取该连接所指向、连接的原始文件的内容,将它压缩、解压后存文件。注意连接的文件经过压缩再解压之后,其连接将不复存在,而成为与原始文件内容完全相同的另一个文件。**参 数:**

-A	调整可执行的自动解压缩文件
-b <工作目录>	指定暂时存放文件的目录
-c	替每个被压缩的文件加上注释,zip 会显示提示符号一一询问用户,不加注释的文件可按  键略过
-d	从压缩文件内删除指定的文件
-D	压缩文件内不建立目录名称。默认 zip 命令会在压缩文件内单独建立目的名称

-f	此参数的效果和指定“-u”参数类似,但不仅更新已有的文件,如果某些文件原本不存在于压缩文件内,使用本参数会一并将其加入压缩文件中
-F	尝试修复已损坏的压缩文件。使用本参数 zip 会尝试去修复毁损的压缩文件,注意修复压缩文件的失败率很高,请先备份原来的压缩文件。如重复指定这项参数“-FF”,则 zip 会深入压缩文件检验特殊签名,验证文件的正确性并尽可能去修复它
-g	将文件压缩后附加在已有的压缩文件之后,而非另行建立新的压缩文件。若附加操作失败,zip 会尝试恢复原先的压缩文件,此项动作一旦失败,则该压缩文件将毁损无法使用
-h	显示帮助
-i <范本样式>	只压缩符合条件的文件。用户可设定范本样式,让 zip 只压缩符合指定范本的文件
-j	只存储文件名称及其内容,而不存放任何目录名称。默认 zip 命令会存储文件与当前目录之相对路径的名称
-J	删除压缩文件前面不必要的数,例如自动解压缩文件的文件头
-k	使用 MS-DOS 兼容格式的文件名称。在 DOS 环境下某些文件名称无法使用,举例而言,拥有两个或两个以上“.”号的文件名,就无法被 DOS 的 FAT-12/FAT-16 文件系统所接受
-l	压缩文件时,把“LF”(Line Feed)字符置换成“LF+CR”字符。借由本参数可在 DOS 环境中使用 UNIX 格式的纯文本文件。注意,欲压缩的文件若是二进制文件,则不该使用这个参数
-ll	压缩文件时,把“LF+CR”字符置换成“LF”字符。借由本参数可在 UNIX 环境中使用 DOS 格式的纯文本文件,注意欲压缩的文件若是二进制文件,则不该使用这个参数
-L	显示版权信息
-m	将文件压缩并加入压缩文件后,删除原始文件,即把文件移到压缩文件中。假设某目录下的文件在移至压缩文件内之后变成空目录,则该目录将一并被删除。利用这个参数可节省少许磁盘空间,但请注意,压缩文件的失败几率虽然极低,却不表示完全没有任何风险,一旦压缩失败而原始文件又已遭删除,将会造成相当程度的困扰

-n <字尾字符串>	不压缩具有特定字尾字符串的文件。某些文件已经过特殊编码大幅缩减文件大小(如 GIF 和 JPEG 图像文件),再使用 zip 对其压缩除耗费大量时间之外,几乎没有任何效果,使用本参数能指定这类文件的字尾字符串,两两之间用“:”号区隔,让zip直接存储这些文件而不予压缩,节省压缩时间。举例来说,字尾字符串设为“.gif.jpg”,当zip遇到最后扩展名是“.gif”或“.jpg”字符串的文件时,就会不经压缩而直接存入压缩文件内
-o	以压缩文件内拥有最新异动时间的文件为准,将压缩文件的异动时间设成和该文件相同
-q	不显示命令执行过程
-r	递归处理,将指定目录下的所有文件及子目录一并处理
-S	包含系统和隐藏文件,此参数仅在某些操作系统(如 DOS)中有效,对于 UNIX 之类的操作系统没有作用
-t <日期时间>	把压缩文件的日期设成指定的日期。例如日期时间设为“120599”,表示公元1999年12月5日
-T	检查备份文件内的每个文件是否正确无误
-u	更换较新的文件至压缩文件内。利用此参数可把较新的文件(以异动时间为准)压缩后,加入压缩文件中,置换原本压缩文件里较旧的文件
-v	显示命令执行过程或显示版本信息
-V	存储 VMS (Virtual Memory System) 操作系统的文件属性。本参数仅在 VMS 操作系统下有效,且产生的压缩文件也只能在VMS 中使用
-w	在文件名称里加入版本编号,本参数仅在 VMS 操作系统下有效
-x <范本样式>	压缩时排除符合条件的文件。可设定范本样式,让 zip 排除符合指定范本的文件
-X	不存储额外的文件属性,例如 OS/2 系统的扩展属性 (Extended Attribute),以及 UNIX 的用户和组织别码
-y	直接存储符号链接,而非该连接所指向的文件,本参数仅在UNIX 之类的系统下有效
-z	替压缩文件加上注释,zip 会显示提示符号询问用户,在注释最后输入 EOF 控制字符即可 (UNIX 请使用 █+█, VAX (Virtual Address eXtension)/VMS、OS/2 和 MS-DOS 请使用 █+█ 输入 EOF 控制字符)
-\$	存储第 1 个被压缩文件所在磁盘的卷册名称,本参数仅在某些系统(如 OS/2 或 MS-DOS)中有效,对 UNIX 之类的操作系统无效

-<压缩率>	压缩率是一个介于1~9的数值, 默认值为6, 指定越大的数值, 压缩率就会越高。压缩率越高, 产生的压缩文件就更小, 但耗费在压缩过程的时间也就越多
--------	--

范 例:

① 将文件名为monk的文件, 压缩后存储成名称为ninja.zip的压缩文件:

```
# zip ninja monk
```

② 把/home/bishop目录下的所有文件及子目录一并压缩, 压缩文件文件名为lord.zip:

```
# zip -r lord /home/bishop
```

③ 移除压缩文件summoner.zip内的文件psychic:

```
# zip -d summoner psychic
```

④ 把/home/cleric目录下的所有文件及子目录一并压缩, 但不含最后扩展名为.log的文件, 压缩文件文件名为 samurai.zip:

```
# zip -r samurai /home/cleric -x *.log
```

⑤ 将文件名为wakizashi的文件, 压缩成名称为niginata.zip的压缩文件, 并指定使用/var/tmp为工作目录:

```
# zip -b /var/tmp niginata wakizashi
```

└─ 指定暂时存放文件的目录

Z



- ⑥ 把/home/zeta目录里的所有文件压缩成kickstart.zip压缩文件，且一一替每个文件加上注释：

```
# zip -c kickstart /home/zeta/*
  adding: key.sah (deflated 34%)
  adding: lock.sah (stored 0%)
...
  adding: xsetiathome (deflated 57%)
```

Enter comment for key.sah: ← 询问第 1 个文件的注释
Key File for SETI ← 输入第 1 个文件的注释
 Enter comment for lock.sah: ← 询问第 2 个文件的注释，依此
 模式至最后一个文件为止

...

- ⑦ 压缩当前目录下所有最后扩展名为.sas的文件，压缩文件文件名为UK-SAS.zip:

```
# zip UK-SAS * -i *.sas
```

- ⑧ 把当前目录里的所有文件压缩成jumpstart.zip压缩文件，最后替压缩文件加上注释：

```
# zip -c jumpstart *
  adding: key.sah (deflated 34%)
  adding: lock.sah (stored 0%)
...
  adding: xsetiathome (deflated 57%)
```

enter new zip file comment (end with .): ← 询问压缩文件注释
SETI at Home ← 输入第 1 行文件注释
for Linux glibc2.1 ← 输入第 2 行文件注释
 ← 注释输入完成后，请按 **Ctrl+D** 键，
 输入 EOF 控制字符，
 结束输入注释的操作

- ⑨ 采用高压缩率的方式将/var/tmp/animeshow目录里的文件打包成压缩文件JP-Anime.zip:

```
# zip -9 JP-Anime /var/tmp/animeshow/*
```



指定压缩率为 "9", 产生更小的压缩文件

zipinfo (zip information)

常用指数: 

功能说明: 列出压缩文件信息。

语 法: zipinfo [-12hlmMstTvz] [压缩文件] [文件...] [-x <范本样式>]

包 名 称: unzip。

相关命令: unzip, zip。

补充说明: 执行zipinfo命令可得知zip压缩文件的详细信息, 包括存取权限、加密状态、压缩方式、操作系统版本, 以及压缩程序的文件系统等等。不附加任何参数直接给予其压缩文件名称, 显示的效果类似执行“ls-l”与“unzip-v”命令的综合体, 并加上一些压缩文件信息。

参 数:

-1	只列出文件名称。本参数会取消其他参数的功能
-2	此参数的效果和指定“-1”参数类似, 但可搭配“-h”、“-t”和“-z”参数使用
-h	只列出压缩文件的文件名称、大小(单位以Byte计算), 以及压缩文件内所包含的文件数目
-l	此参数的效果和指定“-m”参数类似, 但会列出原始文件的大小, 而非每个文件的压缩率
-m	此参数的效果和指定“-s”参数类似, 但多列出每个文件的压缩率
-M	若信息内容超过一个画面, 则采用类似 more 命令的方式列出信息。zipinfo 会尝试侦测画面大小, 假设无法侦测, 则会以24行为准
-s	用类似执行“ls -l”命令的效果列出压缩文件内容, 此为默认值
-t	只列出压缩文件的内所包含的文件数目、压缩前后的文件大小及压缩率

-T	将压缩文件内每个文件的日期时间,用年、月、日、时、分、秒的顺序列出,例如“19991215.150030”表示公元1999年12月15日15时0分30秒
-v	详细显示压缩文件内每一个文件的信息
-x <范本样式>	不列出符合条件的文件信息
-z	如果压缩文件内含有注释,就将该注释显示出来

范 例:

① 列出压缩文件SETIpack.zip的内容信息:

```
# zipinfo SETIpack
```

```
Archive: SETIpack.zip 330593 bytes 12 files
drwxr-xr-x 2.3 unx 0 bx stor 11-Jul-03 15:30 seti/
-r-xr-xr-x 2.3 unx 131000 bx defN 11-Jul-03 15:30 seti/setiathome
-rw-r--r-- 2.3 unx 0 bx stor 11-Jul-03 15:30 seti/lock.sah
-rw-r--r-- 2.3 unx 132 tx defN 11-Jul-03 15:30 seti/key.sah
-rw-r--r-- 2.3 unx 32 tx defN 11-Jul-03 15:30 seti/version.sah
-rw-r--r-- 2.3 unx 4 tx stor 11-Jul-03 15:30 seti/pid.sah
-rw-r--r-- 2.3 unx 371 tx defN 11-Jul-03 15:30 seti/temp.sah
-rw-r--r-- 2.3 unx 347 tx defN 11-Jul-03 15:30 seti/user_info.sah
-rw-r--r-- 2.3 unx 991 tx defN 11-Jul-03 15:30 seti/outfile.sah
-rw-r--r-- 2.3 unx 4082 tx defN 11-Jul-03 15:30 seti/state.sah
-rw-r--r-- 2.3 unx 356283 tx defN 11-Jul-03 15:30 seti/work_unit.sah
-rw-r--r-- 2.3 unx 161 tx defN 11-Jul-03 15:30 seti/result_header.sah
12 files, 493403 bytes uncompressed, 328687 bytes compressed: 33.4%
```

② 详细列出压缩文件SETIpack.zip内每一个文件的信息:

```
# zipinfo -v SETIpack
```

```
Archive: SETIpack.zip 330593 bytes 12 files
```

```
End-of-central-directory record:
```

```
-----
```

```
Actual offset of end-of-central-dir record: 330571 (00050B4Bh)
Expected offset of end-of-central-dir record: 330571 (00050B4Bh)
(based on the length of the central directory and its expected
offset)
```

This zipfile constitutes the sole disk of a single-part archive; its central directory contains 12 entries. The central directory is 990 (000003DEh) bytes long, and its (expected) offset in bytes from the beginning of the zipfile is 329581 (0005076Dh).

There is no zipfile comment.

Central directory entry #1: ← 第 1 个文件

seti/

offset of local header from start of archive: 0 (00000000h) bytes
 file system or operating system of origin: Unix
 version of encoding software: 2.3
 ...

The central-directory extra field contains:

- A subfield with ID 0x5455 (universal time) and 5 data bytes.
 The local extra field has UTC/GMT modification/access times.
- A subfield with ID 0x7855 (Unix UID/GID) and 0 data bytes.

There is no file comment.

Central directory entry #2: ← 第 2 个文件

seti/setiathome

offset of local header from start of archive: 65 (00000041h) bytes
 file system or operating system of origin: Unix
 ...
 ...
 - A subfield with ID 0x7855 (Unix UID/GID) and 0 data bytes.

There is no file comment.

Central directory entry #12: ← 第 12 个文件

```
seti/result_header.sah
```

```
offset of local header from start of archive: 329374 (0005069Eh)
bytes
```

```
file system or operating system of origin: Unix
```

```
...
```

```
...
```

```
- A subfield with ID 0x7855 (Unix UID/GID) and 0 data bytes.
```

```
There is no file comment.
```

③ 只列出压缩文件SETIpack.zip内的文件名称:

```
# zipinfo -l SETIpack
seti/
seti/setiathome
seti/lock.sah
seti/key.sah
seti/version.sah
seti/pid.sah
seti/temp.sah
seti/user_info.sah
seti/outfile.sah
seti/state.sah
seti/work_unit.sah
seti/result_header.sah
```

④ 列出压缩文件SETIpack.zip的内容信息,同时显示压缩文件的注释:

```
# zipinfo -z SETIpack
Archive: SETIpack.zip 330593 bytes 12 files
SETI@home client.
Platform: i386-pc-linux-gnu-gnulibc2.1
Version: 3.03
drwxr-xr-x 2.3 unx 0 bx stor 11-Jul-03 15:30 seti/
-r-xr-xr-x 2.3 unx 131000 bx defN 11-Jul-03 15:30 seti/setiathome
-rw-r--r-- 2.3 unx 0 bx stor 11-Jul-03 15:30 seti/lock.sah
...
-rw-r--r-- 2.3 unx 356283 tx defN 11-Jul-03 15:30 seti/work_unit.sah
-rw-r--r-- 2.3 unx 161 tx defN 11-Jul-03 15:30 seti/result_header.sah
12 files, 493403 bytes uncompressed, 328687 bytes compressed: 33.4%
```

压缩文件注释内容