常用命令集

cal

使用权限

所有使用者

使用方式

cal [-mjy] [month [year]]

说明

显示日历。若只有一个参数，则代表年份(1-9999)，显示该年的年历。年份必须全部写出；``cal 89\ 将不会是显示 1989 年的年历。使用两个参数，则表示月份及年份。若没有参数则显示这个月的月历。
1752 年 9 月第 3 日起改用西洋新历，因这时大部份的国家都采用新历，有 10 天被去除，所以该月份的月历有些不同。在此之前为西洋旧历。

参数

-m : 以星期一为每周的第一天方式显示。
-j : 以凯撒历显示，即以一月一日起的天数显示。
-y : 显示今年年历。

范例

cal : 显示本月的月历。
[root@mylinux /root]# cal 2001
cal 5 2001 : 显示公元 2001 年 5 月月历。
cal -m : 以星期一为每周的第一天方式，显示本月的月历。
cal -jy : 以一月一日起的天数显示今年的年历。

crontab

使用权限

所有使用者

使用方式

crontab [-u user] filecrontab [-u user] { -l | -r | -e }

说明

crontab 是用来让使用者在固定时间或固定间隔执行程式之用，换句话说，也就是类似使用者的时程表。-u user 是指设定指定 user 的时程表，这个前提是你必须要有其权限(比如说是 root)才能够指定他人的时程表。如果不使用 -u user 的话，就是表示设定自己的时程表。

参数

-e : 执行文字编辑器来设定时程表，内定的文字编辑器是 VI，如果你想用别的文字编辑器，则请先设定 VISUAL 环境变数来指定使用那个文字编辑器(比如说 setenv VISUAL joe)
-r : 删除目前的时程表
-l : 列出目前的时程表
时程表的格式如下 :
f1 f2 f3 f4 f5 program
其中 f1 是表示分钟，f2 表示小时，f3 表示一个月份中的第几日，f4 表示月份，f5 表示一个星期中的第几天。program 表示要执行的程式。
当 f1 为 * 时表示每分钟都要执行 program，f2 为 * 时表示每小时都要执行程式，其余类推
当 f1 为 a-b 时表示从第 a 分钟到第 b 分钟这段时间内要执行，f2 为 a-b 时表示从第 a 到第 b 小时都要执行，其余类推
当 f1 为 */n 时表示每 n 分钟个时间间隔执行一次，f2 为 */n 表示每 n 小时个时间间隔执行一次，其余类推
当 f1 为 a, b, c,... 时表示第 a, b, c,... 分钟要执行，f2 为 a, b, c,... 时表示第 a, b, c...个小时要执行，其余类推
使用者也可以将所有的设定先存放在档案 file 中，用 crontab file 的方式来设定时程表。

例子

每月每天每小时的第 0 分钟执行一次 /bin/ls :

0 7 * * * /bin/ls
在 12 月内, 每天的早上 6 点到 12 点中，每隔 20 分钟执行一次 /usr/bin/backup:
0 6-12/3 * 12 * /usr/bin/backup
周一到周五每天下午 5:00 寄一封信给 alex@domain.name :
0 17 * * 1-5 mail -s "hi" alex@domain.name < /tmp/maildata
每月每天的午夜 0 点 20 分, 2 点 20 分, 4 点 20 分....执行 echo "haha"
20 0-23/2 * * * echo "haha"

注意

当程式在你所指定的时间执行后，系统会寄一封信给你，显示该程式执行的内容，若是你不希望收到这样的信，请在每一行空一格之后加上 > /dev/null 2>&1 即可。

date

使用权限

所有使用者

使用方式

date [-u] [-d datestr] [-s datestr] [--utc] [--universal]
[--date=datestr] [--set=datestr] [--help] [--version] [+FORMAT] [MMDDhhmm[[CC]YY][.ss]]

说明

date 可以用来显示或设定系统的日期与时间，在显示方面，使用者可以设定欲显示的格式, 格式设定为一个加号后接数个标记，其中可用的标记列表如下:
时间方面 :
% : 印出 %
%n : 下一行
%t : 跳格
%H : 小时(00..23)
%I : 小时(01..12)
%k : 小时(0..23)
%l : 小时(1..12)
%M : 分钟(00..59)
%p : 显示本地 AM 或 PM
%r : 直接显示时间 (12 小时制，格式为 hh:mm:ss [AP]M)
%s : 从 1970 年 1 月 1 日 00:00:00 UTC 到目前为止的秒数

%S : 秒(00..61)
%T : 直接显示时间 (24 小时制)
%X : 相当于 %H:%M:%S
%Z : 显示时区
日期方面 :
%a : 星期几 (Sun..Sat)
%A : 星期几 (Sunday..Saturday)
%b : 月份 (Jan..Dec)
%B : 月份 (January..December)
%c : 直接显示日期与时间
%d : 日 (01..31)
%D : 直接显示日期 (mm/dd/yy)
%h : 同 %b
%j : 一年中的第几天 (001..366)
%m : 月份 (01..12)
%U : 一年中的第几周 (00..53) (以 Sunday 为一周的第一天的情形)
%w : 一周中的第几天 (0..6)
%W : 一年中的第几周 (00..53) (以 Monday 为一周的第一天的情形)
%x : 直接显示日期 (mm/dd/yy)
%y : 年份的最后两位数字 (00.99)
%Y : 完整年份 (0000..9999)
若是不以加号作为开头，则表示要设定时间，而时间格式为 MMDDhhmm[[CC]YY][.ss]，其中 MM 为月份，DD 为日，hh 为小时，mm 为分钟，CC 为年份前两位数字，YY 为年份后两位数字，ss 为秒数

参数

-d datestr : 显示 datestr 中所设定的时间 (非系统时间)
--help : 显示辅助讯息
-s datestr : 将系统时间设为 datestr 中所设定的时间
-u : 显示目前的格林威治时间
--version : 显示版本编号

例子

显示时间后跳行，再显示目前日期 :
date +%T%n%D
显示月份与日数 :
date +%B %d
显示日期与设定时间(12:34:56) :
date --date 12:34:56

注意

当你不希望出现无意义的 0 时(比如说 1999/03/07)，则可以在标记中插入 - 符号，比如说 date +%-H:%-M:%-S 会把时分秒中无意义的 0 给去掉，像是原本的 08:09:04 会变为 8:9:4。另外，只有取得权限者(比如说 root)才能设定系统时间。
当你以 root 身分更改了系统时间之后，请记得以 clock -w 来将系统时间写入 CMOS 中，这样下次重新开机时系统时间才会持续抱持最新的正确值。

sleep

使用权限

所有使用者

使用方式

sleep [--help] [--version] number[smhd]

说明

sleep 可以用来将目前动作延迟一段时间

参数

--help : 显示辅助讯息
--version : 显示版本编号
number : 时间长度，后面可接 s、m、h 或 d
其中 s 为秒，m 为 分钟，h 为小时，d 为日数

例子

显示目前时间后延迟 1 分钟，之后再次显示时间 :
date;sleep 1m;date

time

使用权限

所有使用者

使用方式

time [options] COMMAND [arguments]

说明

time 指令的用途，在于量测特定指令执行时所需消耗的时间及系统资源等资讯。例如 CPU 时间、记忆体、输入输出等等。需要特别注意的是，部分资讯在 Linux 上显示不出来。这是因为在 Linux 上部分资源的分配函式与 time 指令所预设的方式并不相同，以致于 time 指令无法取得这些资料。

使用方式

-o or --output=FILE
设定结果输出档。这个选项会将 time 的输出写入 所指定的档案中。如果档案已经存在，系统将覆写其内容。
-a or --append
配合 -o 使用，会将结果写到档案的末端，而不会覆盖掉原来的内容。
-f FORMAT or --format=FORMAT
以 FORMAT 字串设定显示方式。当这个选项没有被设定的时候，会用系统预设的格式。不过你可以用环境变数 time 来设定这个格式，如此一来就不必每次登入系统都要设定一次。一般设定上，你可以用\t 表示跳栏，或者是用\n 表示换行。
每一项资料要用 % 做为前导。如果要在字串中使用百分比符号，就用 。（学过 C 语言的人大概会觉得很熟悉）

time 指令可以显示的资源有四大项，分别是：
Time resources
Memory resources
IO resources
Command info
详细的内容如下：
Time Resources
E 执行指令所花费的时间，格式是：[hour]:minute:second。请注意这个数字并不代表实际的 CPU 时间。

e 执行指令所花费的时间，单位是秒。请注意这个数字并不代表实际的 CPU 时间。
S 指令执行时在核心模式（kernel mode）所花费的时间，单位是秒。
U 指令执行时在使用者模式（user mode）所花费的时间，单位是秒。
P 执行指令时 CPU 的占用比例。其实这个数字就是核心模式加上使用者模式的 CPU 时间除以总时间。
Memory Resources
M 执行时所占用的实体记忆体的最大值。单位是 KB
t 执行时所占用的实体记忆体的平均值，单位是 KB
K 执行程序所占用的记忆体总量（stack+data+text）的平均大小，单位是 KB
D 执行程序的自有资料区（unshared data area）的平均大小，单位是 KB
p 执行程序的自有堆叠（unshared stack）的平均大小，单位是 KB
X 执行程序间共享内容（shared text）的平均值，单位是 KB
Z 系统记忆体页的大小，单位是 byte。对同一个系统来说这是个常数
IO Resources
F 此程序的主要记忆体页错误发生次数。所谓的主要记忆体页错误是指某一记忆体页已经置换到置换档（swap file)中，而且已经分配给其他程序。此时该页的内容必须从置换档里再读出来。
R 此程序的次要记忆体页错误发生次数。所谓的次要记忆体页错误是指某一记忆体页虽然已经置换到置换档中，但尚未分配给其他程序。此时该页的内容并未被破坏，不必从置换档里读出来
W 此程序被交换到置换档的次数
c 此程序被强迫中断（像是分配到的 CPU 时间耗尽）的次数
w 此程序自愿中断（像是在等待某一个 I/O 执行完毕，像是磁碟读取等等）的次数
I 此程序所输入的档案数
O 此程序所输出的档案数
r 此程序所收到的 Socket Message
s 此程序所送出的 Socket Message
k 此程序所收到的信号 (Signal)数量
Command Info
C 执行时的参数以及指令名称
x 指令的结束代码 (Exit Status)

-p or --portability
这个选项会自动把显示格式设定成为：
real %e
user %U
sys %S
这么做的目的是为了与 POSIX 规格相容。

-v or --verbose
这个选项会把所有程式中用到的资源通通列出来，不但如一般英文语句，还有说明。对不想花时间去熟习格式设定或是刚刚开始接触这个指令的人相当有用。

范例

利用下面的指令
time -v ps -aux
我们可以获得执行 ps -aux 的结果和所花费的系统资源。如下面所列的资料：
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.4 1096 472 ? S Apr19 0:04 init
root 2 0.0 0.0 0 0 ? SW Apr19 0:00 [kflushd]
root 3 0.0 0.0 0 0 ? SW Apr19 0:00 [kpiod]
.....
root 24269 0.0 1.0 2692 996 pts/3 R 12:16 0:00 ps -aux
Command being timed: "ps -aux"
User time (seconds): 0.05
System time (seconds): 0.06
Percent of CPU this job got: 68%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0:00.16
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 0
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 238
Minor (reclaiming a frame) page faults: 46
Voluntary context switches: 0
Involuntary context switches: 0
Swaps: 0
File system inputs: 0
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0

uptime

使用权限

所有使用者

使用方式

uptime [-V]

说明

uptime 提供使用者下面的资讯，不需其他参数：
现在的时间
系统开机运转到现在经过的时间
连线的使用者数量
最近一分钟，五分钟和十五分钟的系统负载

参数

-V 显示版本资讯。

范例

uptime
其结果为：
10:41am up 5 days, 10 min, 1 users, load average: 0.00, 0.00, 1.99

chfn

使用权限

所有使用者

用法

shell>> chfn

说明

提供使用者更改个人资讯，用于 finger and mail username

范例

shell>> chfn
Changing finger information for user
Password: [del]
Name[]:Johnney Huang ### 提供 finger 时的资料
Office[]:NCCU
Office Phone[]: [del]
Home Phone[]: [del]

chsh

使用权限

所有使用者

用法

shell>> chsh

说明

更改使用者 shell 设定

范例

shell>> chsh
Changing fihanging shell for user1
Password: [del]
New shell [/bin/tcsh]: ### [是目前使用的 shell]
[del]
shell>> chsh -l ### 展示 /etc/shells 档案内容
/bin/bash
/bin/sh
/bin/ash
/bin/bsh
/bin/tcsh
/bin/csh

finger

使用权限

所有使用者

使用方式

finger [options] user[@address]

说明

finger 可以让使用者查询一些其他使用者的资料。会列出来的资料有：
Login Name
User Name
Home directory
Shell
Login status
mail status
.plan
.project
.forward
其中 .plan ，.project 和 .forward 就是使用者在他的 Home Directory 里的 .plan ， .project 和 .forward 等档案里的资料。如果没有就没有。finger 指令并不限定于在同一伺服器上查询，也可以寻找某一个远端伺服器上的使用者。只要给一个像是 E-mail address 一般的地址即可。

参数

-l 多行显示。
-s 单行显示。这个选项只显示登入名称，真实姓名，终端机名称，闲置时间，登入时间，办公室号码及电话号码。如果所查询的使用者是远端伺服器的使用者，这个选项无效。

范例

下列指令可以查询本机管理员的资料：
finger root
其结果如下：
Login: root Name: root
Directory: /root Shell: /bin/bash
Never logged in.

No mail.
No Plan.

last

使用权限

所有使用者

使用方式

shell>> last [options]

说明

显示系统开机以来获是从每月初登入者的讯息

参数

-R 省略 hostname 的栏位
-num 展示前 num 个
username 展示 username 的登入讯息
tty 限制登入讯息包含终端机代号
范例：
shell>> last -R -2
johnney pts/1 Mon Aug 14 20:42 still logged in
johnney pts/0 Mon Aug 14 19:59 still logged in
wtmp begins Tue Aug 1 09:01:10 2000 ### /var/log/wtmp

shell>> last -2 minery
minery pts/0 140.119.217.115 Mon Aug 14 18:37 - 18:40 (00:03)
minery pts/0 140.119.217.115 Mon Aug 14 17:22 - 17:24 (00:02)
wtmp begins Tue Aug 1 09:01:10 2000

login

这个命令都不会就不要干算了！呵呵我也不在这里多费笔墨耽误大家美好青春了^_^

passwd

使用权限

所有使用者

使用方式

passwd [-k] [-l] [-u [-f]] [-d] [-S] [username]

说明

用来更改使用者的密码

参数

-k
-l
-u
-f
-d 关闭使用者的密码认证功能, 使用者在登入时将可以不用输入密码, 只有具备 root 权限的使用者方可使用.
-S 显示指定使用者的密码认证种类, 只有具备 root 权限的使用者方可使用.
[username] 指定帐号名称.

who

使用权线

所有使用者都可使用

使用方式

who - [husfV] [user]

说明

显示系统中有那些使用者正在上面，显示的资料包含了使用者 ID，使用的终端机，从那边连上来的，上线时间，呆滞时间，CPU 使用量，动作等等。

参数

-h : 不要显示标题列
-u : 不要显示使用者的动作/工作
-s : 使用简短的格式来显示
-f : 不要显示使用者的上线位置
-V : 显示程式版本

cat

使用权限

所有使用者

使用方式

cat [-AbeEnstTuv] [--help] [--version] fileName

说明

把档案串连接后传到基本输出（萤幕或加 > fileName 到另一个档案）

参数

-n 或 --number 由 1 开始对所有输出的行数编号
-b 或 --number-nonblank 和 -n 相似，只不过对于空白行不编号
-s 或 --squeeze-blank 当遇到有连续两行以上的空白行，就代换为一行的空白行
-v 或 --show-nonprinting

范例

cat -n textfile1 > textfile2 把 textfile1 的档案内容加上行号后输入 textfile2 这个档案里
cat -b textfile1 textfile2 >> textfile3 把 textfile1 和 textfile2 的档案内容加上行号（空白行不

加）之后将内容附加到 textfile3

cd

使用权限

所有使用者

使用方式

cd [dirName]

说明

变换工作目录至 dirName。 其中 dirName 表示法可为绝对路径或相对路径。若目录名称省略，则变换至使用者的 home directory (也就是刚 login 时所在的目录)。
另外，"~" 也表示为 home directory 的意思，"." 则是表示目前所在的目录，".."则表示目前目录位置的上一层目录。

范例

跳到 /usr/bin/ :
cd /usr/bin
跳到自己的 home directory :
cd ~
跳到目前目录的上上两层 :
cd ../..

chmod

使用权限

所有使用者

使用方式

chmod [-cfvR] [--help] [--version] mode file...

说明

Linux/Unix 的档案存取权限分为三级 : 档案拥有者、群组、其他。利用 chmod 可以藉以控制档案如何被他人所存取。

参数

mode : 权限设定字串，格式如下 : [ugoa...][[+-=][rwxX]...][,...]，其中 u 表示该档案的拥有者，g 表示与该档案的拥有者属于同一个群体(group)者，o 表示其他以外的人，a 表示这三者皆是。
+ 表示增加权限、- 表示取消权限、= 表示唯一设定权限。
r 表示可读取，w 表示可写入，x 表示可执行，X 表示只有当该档案是个子目录或者该档案已经被设定过为可执行。
-c : 若该档案权限确实已经更改，才显示其更改动作
-f : 若该档案权限无法被更改也不要显示错误讯息
-v : 显示权限变更的详细资料
-R : 对目前目录下的所有档案与子目录进行相同的权限变更(即以递回的方式逐个变更)
--help : 显示辅助说明
--version : 显示版本

范例

将档案 file1.txt 设为所有人皆可读取 :
chmod ugo+r file1.txt
将档案 file1.txt 设为所有人皆可读取 :
chmod a+r file1.txt
将档案 file1.txt 与 file2.txt 设为该档案拥有者，与其所属同一个群体者可写入，但其他以外的人则不可写入:
chmod ug+w,o-w file1.txt file2.txt
将 ex1.py 设定为只有该档案拥有者可以执行 :
chmod u+x ex1.py
将目前目录下的所有档案与子目录皆设为任何人可读取 :
chmod -R a+r *
此外 chmod 也可以用数字来表示权限如 chmod 777 file
语法为：chmod abc file
其中 a,b,c 各为一个数字，分别表示 User、Group、及 Other 的权限。
r=4，w=2，x=1
若要 rwx 属性则 4+2+1=7；
若要 rw-属性则 4+2=6；
若要 r-x 属性则 4+1=5。
范例：

chmod a=rwx file 和 chmod 777 file 效果相同
chmod ug=rwx,o=x file 和 chmod 771 file 效果相同
若用 chmod 4755 filename 可使此程式具有 root 的权限

chown

使用权限

root

使用方式

chmod [-cfhvR] [--help] [--version] user[:group] file...

说明

Linux/Unix 是多人多工作业系统，所有的档案皆有拥有者。利用 chown 可以将档案的拥有者加以改变。一般来说，这个指令只有是由系统管理者(root)所使用，一般使用者没有权限可以改变别人的档案拥有者，也没有权限可以自己的档案拥有者改设为别人。只有系统管理者(root)才有这样的权限。

参数

user : 新的档案拥有者的使用者 ID
group : 新的档案拥有者的使用者群体(group)
-c: 若该档案拥有者确实已经更改，才显示其更改动作
-f : 若该档案拥有者无法被更改也不要显示错误讯息
-h : 只对于连结(link)进行变更，而非该 link 真正指向的档案
-v : 显示拥有者变更的详细资料
-R : 对目前目录下的所有档案与子目录进行相同的拥有者变更(即以递回的方式逐个变更)
--help : 显示辅助说明--version : 显示版本

范例

将档案 file1.txt 的拥有者设为 users 群体的使用者 jessie :
chown jessie:users file1.txt
将目前目录下的所有档案与子目录的拥有者皆设为 users 群体的使用者 lamport :
chmod -R lamport:users *

cp

使用权限

所有使用者

使用方式

cp [options] source dest
cp [options] source... directory

说明

将一个档案拷贝至另一档案，或将数个档案拷贝至另一目录。

参数

-a 尽可能将档案状态、权限等资料都照原状予以复制。
-r 若 source 中含有目录名，则将目录下之档案亦皆依序拷贝至目的地。
-f 若目的地已经有相同档名的档案存在，则在复制前先予以删除再行复制。

范例

将档案 aaa 复制(已存在)，并命名为 bbb :
cp aaa bbb
将所有的 C 语言程式拷贝至 Finished 子目录中 :
cp *.c Finished

cut

使用权限

所有使用者

用法

cut -cnum1-num2 filename

说明

显示每行从开头算起 num1 到 num2 的文字。

范例

shell>> cat example
test2
this is test1
shell>> cut -c0-6 example ## print 开头算起前 6 个字元
test2
this i

find

用法

find

使用说明

将档案系统内符合 expression 的档案列出来。你可以指要档案的名称、类别、时间、大小、权限等不同资讯的组合，只有完全相符的才会被列出来。
find 根据下列规则判断 path 和 expression，在命令列上第一个 - ( ) , ! 之前的部份为 path，之后的是 expression。如果 path 是空字串则使用目前路径，如果 expression 是空字串则使用 -print 为预设 expression
expression 中可使用的选项有二三十个之多，在此只介绍最常用的部份。
-mount, -xdev : 只检查和指定目录在同一个档案系统下的档案，避免列出其它档案系统中的档案
-amin n : 在过去 n 分钟内被读取过
-anewer file : 比档案 file 更晚被读取过的档案
-atime n : 在过去 n 天过读取过的档案
-cmin n : 在过去 n 分钟内被修改过
-cnewer file :比档案 file 更新的档案
-ctime n : 在过去 n 天过修改过的档案
-empty : 空的档案-gid n or -group name : gid 是 n 或是 group 名称是 name
-ipath p, -path p : 路径名称符合 p 的档案，ipath 会忽略大小写
-name name, -iname name : 档案名称符合 name 的档案。iname 会忽略大小写
-size n : 档案大小 是 n 单位，b 代表 512 位元组的区块，c 表示字元数，k 表示 kilo bytes，

w 是二个位元组。-type
c : 档案类型是 c 的档案。
d: 目录
c: 字型装置档案
b: 区块装置档案
p: 具名贮列
f: 一般档案
l: 符号连结
s: socket
-pid n : process id 是 n 的档案
你可以使用 ( ) 将运算式分隔，并使用下列运算。
exp1 -and exp2
! expr
-not expr
exp1 -or exp2
exp1, exp2

范例

将目前目录及其子目录下所有延伸档名是 c 的档案列出来。
find . -name "*.c"
将目前目录其其下子目录中所有一般档案列出
find . -ftype f
将目前目录及其子目录下所有最近 20 分钟内更新过的档案列出
find . -ctime -20

less

使用权限

所有使用者

使用方式

less [Option] filename

说明

less 的作用与 more 十分相似，都可以用来浏览文字档案的内容，不同的是 less 允许使用者往回卷动以浏览已经看过的部份，同时因为 less 并未在一开始就读入整个档案，因此在

遇上大型档案的开启时，会比一般的文书编辑器(如 vi)来的快速。

ln

使用权限

所有使用者

使用方式

ln [options] source dist
其中 option 的格式为 :
[-bdfinsvF] [-S backup-suffix] [-V {numbered,existing,simple}][--help] [--version] [--]
说明 : Linux/Unix 档案系统中，有所谓的连结(link)，我们可以将其视为档案的别名，而连结又可分为两种 : 硬连结(hard link)与软连结(symbolic link)，硬连结的意思是一个档案可以有多个名称，而软连结的方式则是产生一个特殊的档案，该档案的内容是指向另一个档案的位置。硬连结是存在同一个档案系统中，而软连结却可以跨越不同的档案系统。
ln source dist 是产生一个连结(dist)到 source，至于使用硬连结或软链结则由参数决定。
不论是硬连结或软链结都不会将原本的档案复制一份，只会占用非常少量的磁碟空间。
参数
-f : 链结时先将与 dist 同档名的档案删除
-d : 允许系统管理者硬链结自己的目录
-i : 在删除与 dist 同档名的档案时先进行询问
-n: 在进行软连结时，将 dist 视为一般的档案
-s : 进行软链结(symbolic link)
-v : 在连结之前显示其档名
-b : 将在链结时会被覆写或删除的档案进行备份
-S SUFFIX : 将备份的档案都加上 SUFFIX 的字尾
-V METHOD : 指定备份的方式
--help : 显示辅助说明
--version: 显示版本

范例

将档案 yy 产生一个 symbolic link : zz
ln -s yy zz
将档案 yy 产生一个 hard link : zz
ln yy xx

locate

使用权限

所有使用者

使用方式

locate [-q] [-d] [--database=]
locate [-r] [--regexp=]
locate [-qv] [-o] [--output=]
locate [-e] [-f] <[-l] [-c]<[-U] [-u]>
locate [-Vh] [--version] [--help]

说明

locate 让使用者可以很快速的搜寻档案系统内是否有指定的档案。其方法是先建立一个包括系统内所有档案名称及路径的资料库，之后当寻找时就只需查询这个资料库，而不必实际深入档案系统之中了。在一般的 distribution 之中，资料库的建立都被放在 contab 中自动执行。一般使用者在使用时只要用# locate your_file_name 的形式就可以了。
参数：
-u
-U 建立资料库，-u 会由根目录开始，-U 则可以指定开始的位置。
-e 将排除在寻找的范围之外。
-l 如果 是 1．则启动安全模式。在安全模式下，使用者不会看到权限无法看到的档案。这会始速度减慢，因为 locate 必须至实际的档案系统中取得档案的权限资料。
-f 将特定的档案系统排除在外，例如我们没有到理要把 proc 档案系统中的档案放在资料库中。
-q 安静模式，不会显示任何错误讯息。
-n 至多显示 个输出。
-r 使用正规运算式 做寻找的条件。
-o 指定资料库存的名称。
-d 指定资料库的路径
-h 显示辅助讯息
-v 显示更多的讯息
-V 显示程式的版本讯息

范例

locate chdrv : 寻找所有叫 chdrv 的档案

locate -n 100 a.out : 寻找所有叫 a.out 的档案，但最多只显示 100 个
locate -u : 建立资料库

ls

使用权限

所有使用者

使用方式

ls [-alrtAFR] [name...]

说明

显示指定工作目录下之内容（列出目前工作目录所含之档案及子目录)。

参数

-a 显示所有档案及目录 (ls 内定将档案名或目录名称开头为"."的视为隐藏档，不会列出)
-l 除档案名称外，亦将档案型态、权限、拥有者、档案大小等资讯详细列出
-r 将档案以相反次序显示(原定依英文字母次序)
-t 将档案依建立时间之先后次序列出
-A 同 -a ，但不列出 "." (目前目录) 及 ".." (父目录)
-F 在列出的档案名称后加一符号；例如可执行档则加 "*", 目录则加 "/"
-R 若目录下有档案，则以下之档案亦皆依序列出

范例

列出目前工作目录下所有名称是 s 开头的档案，愈新的排愈后面 :
ls -ltr s*
将 /bin 目录以下所有目录及档案详细资料列出 :
ls -lR /bin
列出目前工作目录下所有档案及目录；目录于名称后加 "/", 可执行档于名称后加 "*":
ls -AF

more

使用权限

所有使用者

使用方式

more [-dlfpcsu] [-num] [+/pattern] [+linenum] [fileNames..]

说明

类似 cat，不过会以一页一页的显示方便使用者逐页阅读，而最基本的指令就是按空白键（space）就往下一页显示，按 b 键就会往回（back）一页显示，而且还有搜寻字串的功能（与 vi 相似），使用中的说明文件，请按 h 。
参数
-num 一次显示的行数
-d 提示使用者，在画面下方显示 [Press space to continue, q to quit.] ，如果使用者按错键，则会显示[Press h for instructions.] 而不是哔声
-l 取消遇见特殊字元 ^L（送纸字元）时会暂停的功能
-f 计算行数时，以实际上的行数，而非自动换行过后的行数（有些单行字数太长的会被扩展为两行或两行以上）
-p 不以卷动的方式显示每一页，而是先清除萤幕后再显示内容
-c 跟 -p 相似，不同的是先显示内容再清除其他旧资料
-s 当遇到有连续两行以上的空白行，就代换为一行的空白行
-u 不显示下引号 （根据环境变数 TERM 指定的 terminal 而有所不同）
+/ 在每个档案显示前搜寻该字串（pattern），然后从该字串之后开始显示
+num 从第 num 行开始显示 fileNames 欲显示内容的档案，可为复数个数

范例

more -s testfile 逐页显示 testfile 之档案内容，如有连续两行以上空白行则以一行空白行显示。
more +20 testfile 从第 20 行开始显示 testfile 之档案内容。

mv

使用权限

所有使用者

使用方式

mv [options] source dest
mv [options] source... directory

说明

将一个档案移至另一档案，或将数个档案移至另一目录。

参数

-i 若目的地已有同名档案，则先询问是否覆盖旧档。

范例

将档案 aaa 更名为 bbb :
mv aaa bbb
将所有的 C 语言程式移至 Finished 子目录中 :
mv -i *.c

rm

使用权限

所有使用者

使用方式

rm [options] name...

说明

删除档案及目录。

参数

-i 删除前逐一询问确认。
-f 即使原档案属性设为唯读，亦直接删除，无需逐一确认。
-r 将目录及以下之档案亦逐一删除。

范例

删除所有 C 语言程式档；删除前逐一询问确认 :
rm -i *.c
将 Finished 子目录及子目录中所有档案删除 :
rm -r Finished

rmdir

使用权限

于目前目录有适当权限的所有使用者

使用方式

rmdir [-p] dirName

说明

删除空的目录。

参数

-p 是当子目录被删除后使它也成为空目录的话，则顺便一并删除。
范例
将工作目录下，名为 AAA 的子目录删除 :
rmdir AAA
在工作目录下的 BBB 目录中，删除名为 Test 的子目录。若 Test 删除后，BBB 目录成为

空目录，则
BBB 亦予删除。
rmdir -p BBB/Test

split

使用权限

所有使用者

使用方式

split [OPTION] [INPUT [PREFIX]]

说明

将一个档案分割成数个。而从 INPUT 分割输出成固定大小的档案，其档名依序为 PREFIXaa, PREFIXab...；PREFIX 预设值为`x。若没有 INPUT 档或为 `-，则从标准输入读进资料。

参数

-b, --bytes=SIZE SIZE 值为每一输出档案的大小，单位为 byte。
-C, --line-bytes=SIZE 每一输出档中，单行的最大 byte 数。
-l, --lines=NUMBER NUMBER 值为每一输出档的列数大小。
-NUMBER 与 -l NUMBER 相同。
--verbose 于每个输出档被开启前，列印出侦错资讯到标准错误输出。
--help 显示辅助资讯然后离开。
--version 列出版本资讯然后离开。
SIZE 可加入单位: b 代表 512， k 代表 1K， m 代表 1 Meg。

范例

PostgresSQL 大型资料库备份与回存：
因 Postgres 允许表格大过你系统档案的最大容量，所以要将表格 dump 到单一的档案可能会有问题，使用 split 进行档案分割。
% pg_dump dbname | split -b 1m - filename.dump.
重新载入
% createdb dbname
% cat filename.dump.* | pgsql dbname

touch

使用权限

所有使用者

使用方式

touch [-acfm][-r reference-file] [--file=reference-file][-t MMDDhhmm[[CC]YY][.ss]]
[-d time] [--date=time] [--time={atime,access,use,mtime,modify}]
[--no-create] [--help] [--version] file1 [file2 ...]

说明

touch 指令改变档案的时间记录。 ls -l 可以显示档案的时间记录。

参数

a 改变档案的读取时间记录。
m 改变档案的修改时间记录。
c 假如目的档案不存在，不会建立新的档案。与 --no-create 的效果一样。
f 不使用，是为了与其他 unix 系统的相容性而保留。
r 使用参考档的时间记录，与 --file 的效果一样。
d 设定时间与日期，可以使用各种不同的格式。
t 设定档案的时间记录，格式与 date 指令相同。
--no-create 不会建立新档案。
--help 列出指令格式。
--version 列出版本讯息。

范例

最简单的使用方式，将档案的时候记录改为现在的时间。若档案不存在，系统会建立一个新的档案。
touch file
touch file1 file2
将 file 的时间记录改为 5 月 6 日 18 点 3 分，公元两千年。时间的格式可以参考 date 指令，至少需输入 MMDDHHmm ，就是月日时与分。
touch -c -t 05061803 file
touch -c -t 050618032000 file

将 file 的时间记录改变成与 referencefile 一样。
touch -r referencefile file
将 file 的时间记录改成 5 月 6 日 18 点 3 分，公元两千年。时间可以使用 am, pm 或是 24 小时的格式，日期可以使用其他格式如 6 May 2000 。
touch -d "6:03pm" file
touch -d "05/06/2000" file
touch -d "6:03pm 05/06/2000" file

at

使用权限

所有使用者

使用方式

at -V [-q queue] [-f file] [-mldbv] TIME

说明

at 可以让使用者指定在 TIME 这个特定时刻执行某个程式或指令，TIME 的格式是 HH:MM 其中的 HH 为小时，MM 为分钟，甚至你也可以指定 am, pm, midnight, noon, teatime(就是下午 4 点锺)等口语词。
如果想要指定超过一天内的时间，则可以用 MMDDYY 或者 MM/DD/YY 的格式，其中 MM 是分钟，DD 是第几日，YY 是指年份。另外，使用者甚至也可以使用像是 now + 时间间隔来弹性指定时间，其中的时间间隔可以是 minutes,hours, days, weeks
另外，使用者也可指定 today 或 tomorrow 来表示今天或明天。当指定了时间并按下 enter 之后，at 会进入交谈模式并要求输入指令或程式，当你输入完后按下 ctrl+D 即可完成所有动作，至于执行的结果将会寄回你的帐号中。

参数

-V : 印出版本编号
-q : 使用指定的伫列(Queue)来储存，at 的资料是存放在所谓的 queue 中，使用者可以同时使用多个 queue，而 queue 的编号为 a, b, c... z 以及 A, B, ... Z 共 52 个
-m : 即使程式/指令执行完成后没有输出结果, 也要寄封信给使用者
-f file : 读入预先写好的命令档。使用者不一定要使用交谈模式来输入，可以先将所有的指定先写入档案后再一次读入
-l : 列出所有的指定 (使用者也可以直接使用 atq 而不用 at -l)
-d : 删除指定 (使用者也可以直接使用 atrm 而不用 at -d)

-v : 列出所有已经完成但尚未删除的指定

例子

三天后的下午 5 点锺执行 /bin/ls :
at 5pm + 3 days /bin/ls
三个星期后的下午 5 点锺执行 /bin/ls :
at 5pm + 2 weeks /bin/ls
明天的 17:20 执行 /bin/date :
at 17:20 tomorrow /bin/date
1999 年的最后一天的最后一分钟印出 the end of world !
at 23:59 12/31/1999 echo the end of world !

/etc/aliases

使用权限

系统管理者

使用方式

请用 newaliases 更新资料库

说明

sendmail 会使用一个在/etc/aliases 中的档案做使用者名称转换的动作。当 sendmail 收到一个要送给 xxx 的信时，它会依据 aliases 档的内容送给另一个使用者。这个功能可以创造一个只有在信件系统内才有效的使用者。例如 mailing list 就会用到这个功能，在 mailinglist 中，我们可能会创造一个叫 redlinux@link.ece.uci.edu 的 mailinglist，但实际上并没有一个叫 redlinux 的使用者。实际 aliases 档的内容是将送给这个使用者的信都收给 mailing list 处理程式负责分送的工作。
/etc/aliases 是一个文字模式的档案，sendmail 需要一个二进位格式的 /etc/aliases.db。
newaliases 的功能传是将 /etc/aliases 转换成一个 sendmail 所能了解的资料库。

范例

newaliases
下面命令会做相同的事，
sendmail -bi

相关命令:

mail, mailq, newaliases, sendmail

mail

使用权限

所有使用者

使用方式

mail [-iInv] [-s subject] [-c cc-addr] [-b bcc-addr] user1 [user2 ...]

说明

mail 不仅只是一个指令， mail 还是一个电子邮件程式，不过利用 mail 来读信的人应该很少吧！对于系统管理者来说 mail 就很有用，因为管理者可以用 mail 写成 script ，定期寄一些备忘录提醒系统的使用者。

参数

i 忽略 tty 的中断讯号。 (interrupt)
I 强迫设成互动模式。 (Interactive)
v 列印出讯息，例如送信的地点、状态等等。 (verbose)
n 不读入 mail.rc 设定档。
s 邮件标题。
c cc 邮件地址。
b bcc 邮件地址。

范例

将信件送给一个或以上的电子邮件地址，由于没有加入其他的选项，使用者必须输入标题与信件的内容等。而 user2 没有主机位置，就会送给邮件伺服器的 user2 使用者。
mail user1@email.address
mail user1@email.address user2
将 mail.txt 的内容寄给 user2 同时 cc 给 user1 。如果将这一行指令设成 cronjob 就可以定时将备忘录寄给系统使用者。

mail -s 标题 -c user1 user2 < mail.txt

mesg

使用权限

所有使用者

使用方式

mesg [y|n]

说明

决定是否允许其他人传讯息到自己的终端机介面

参数

y : 允许讯息传到终端机介面上。
n : 不允许讯息传到终端机介面上 。
如果没有设定，则讯息传递与否则由终端机界面目前状态而定。

例子

改变目前讯息设定，改成不允许讯息传到终端机介面上 :
mesg n
与 mesg 相关的指令有： talk，write，wall。

talk

使用权限

所有使用者

使用方式

talk person [ttyname]

说明

与其他使用者对谈

参数

person: 预备对谈的使用者帐号，如果该使用者在其他机器上，则可输入 person@machine.name
ttyname : 如果使用者同时有两个以上的 tty 连线，可以自行选择合适的 tty 传讯息

例子

1 :与现在机器上的使用者 Rollaend 对谈，此时 Rollaend 只有一个连线 :
talk Rollaend
接下来就是等 Rollaend 回应，若 Rollaend 接受，则 Rollaend 输入 `talk jzlee`即可开始对谈，结束请按 ctrl+c
2 :与 linuxfab.cx 上的使用者 Rollaend 对谈，使用 pts/2 来对谈 :
talk Rollaend@linuxfab.cx pts/2
接下来就是等 Rollaend 回应，若 Rollaend 接受，则 Rollaend 输入 `talk jzlee@jzlee.home`即可开始对谈，结束请按 ctrl+c
注意 ：若萤幕的字会出现不正常的字元，试着按 ctrl+l 更新萤幕画面。

wall

使用权限

所有使用者

使用方式

wall [message]

使用说明

wall 会将讯息传给每一个 mesg 设定为 yes 的上线使用者。当使用终端机介面做为标准传入时, 讯息结束时需加上 EOF (通常用 Ctrl+D)

例子

传讯息"hi" 给每一个使用者 :
wall hi

write

使用权限

所有使用者

使用方式

write user [ttyname]

说明

传讯息给其他使用者

参数

user : 预备传讯息的使用者帐号
ttyname : 如果使用者同时有两个以上的 tty 连线，可以自行选择合适的 tty 传讯息

例子

1 :传讯息给 Rollaend，此时 Rollaend 只有一个连线 :
write Rollaend
接下来就是将讯息打上去，结束请按 ctrl+c
2 :传讯息给 Rollaend，Rollaend 的连线有 pts/2，pts/3 :
write Rollaend pts/2
接下来就是将讯息打上去，结束请按 ctrl+c
注意 ：若对方设定 mesg n，则此时讯席将无法传给对方

kill

使用权限

所有使用者

使用方式

kill [-s signal | -p] [-a] pid ...
kill -l [signal]

说明

kill 送出一个特定的信号 (signal) 给行程 id 为 pid 的行程根据该信号而做特定的动作, 若没有指定, 预设是送出终止(TERM) 的信号

参数

-s (signal) : 其中可用的讯号有 HUP (1), KILL (9), TERM (15),分别代表着重跑, 砍掉, 结束; 详细的信号可以用 kill -l
-p : 印出 pid , 并不送出信号
-l (signal) : 列出所有可用的信号名称

范例

将 pid 为 323 的行程砍掉 (kill) :
kill -9 323
将 pid 为 456 的行程重跑 (restart) :
kill -HUP 456

nice

使用权限

所有使用者

使用方式

nice [-n adjustment] [-adjustment] [--adjustment=adjustment]
[--help] [--version] [command [arg...]]

说明

以更改过的优先序来执行程式，如果未指定程式，则会印出目前的排程优先序，内定的 adjustment 为 10，范围为 -20 (最高优先序) 到 19 (最低优先序)

参数

-n adjustment, -adjustment, --adjustment=adjustment
皆为将该原有优先序的增加 adjustment
--help 显示求助讯息
--version 显示版本资讯

范例

将 ls 的优先序加 1 并执行 :
nice -n 1 ls
将 ls 的优先序加 10 并执行 :
nice ls 将 ls 的优先序加 10 并执行
注意 : 优先序 (priority) 为作业系统用来决定 CPU 分配的参数，Linux 使用『回合制 (round-robin)』的演算法来做 CPU 排程，优先序越高，所可能获得的 CPU 时间就越多。

ps

使用权限

所有使用者

使用方式

ps [options] [--help]

说明

显示瞬间行程 (process) 的动态

参数

ps 的参数非常多, 在此仅列出几个常用的参数并大略介绍含义
-A 列出所有的行程
-w 显示加宽可以显示较多的资讯
-au 显示较详细的资讯
-aux 显示所有包含其他使用者的行程
au(x) 输出格式 :
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
USER: 行程拥有者
PID: pid
%CPU: 占用的 CPU 使用率
%MEM: 占用的记忆体使用率
VSZ: 占用的虚拟记忆体大小
RSS: 占用的记忆体大小
TTY: 终端的次要装置号码 (minor device number of tty)
STAT: 该行程的状态:
D: 不可中断的静止 (通悸□□缜 b 进行 I/O 动作)
R: 正在执行中
S: 静止状态
T: 暂停执行
Z: 不存在但暂时无法消除
W: 没有足够的记忆体分页可分配
<: 高优先序的行程
N: 低优先序的行程
L: 有记忆体分页分配并锁在记忆体内 (即时系统或捱 A I/O)
START: 行程开始时间
TIME: 执行的时间
COMMAND:所执行的指令
范例
ps
PID TTY TIME CMD
2791 ttyp0 00:00:00 tcsh
3092 ttyp0 00:00:00 ps
% ps -A
PID TTY TIME CMD
1 ? 00:00:03 init
2 ? 00:00:00 kflushd

3 ? 00:00:00 kpiod
4 ? 00:00:00 kswapd
5 ? 00:00:00 mdrecoveryd
.....
% ps -aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.7 1096 472 ? S Sep10 0:03 init [3]
root 2 0.0 0.0 0 0 ? SW Sep10 0:00 [kflushd]
root 3 0.0 0.0 0 0 ? SW Sep10 0:00 [kpiod]
root 4 0.0 0.0 0 0 ? SW Sep10 0:00 [kswapd]
.....

pstree

使用权限

所有使用者

使用方式

pstree [-a] [-c] [-h|-Hpid] [-l] [-n] [-p] [-u] [-G|-U] [pid|user]
pstree -V

说明

将所有行程以树状图显示, 树状图将会以 pid (如果有指定) 或是以 init 这个基本行程为根 (root) ,如果有指定使用者 id , 则树状图会只显示该使用者所拥有的行程

参数

-a 显示该行程的完整指令及参数, 如果是被记忆体置换出去的行程则会加上括号
-c 如果有重覆的行程名, 则分开列出 (预设值是会在前面加上 *

renice

使用权限

所有使用者

使用方式

renice priority [[-p] pid ...] [[-g] pgrp ...][[-u] user ...]

说明

重新指定一个或多个行程(Process)的优先序(一个或多个将根据所下的参数而定)

参数

-p pid 重新指定行程的 id 为 pid 的行程的优先序
-g pgrp 重新指定行程群组(process group)的 id 为 pgrp 的行程 (一个或多个) 的优先序
-u user 重新指定行程拥有者为 user 的行程的优先序

范例

将行程 id 为 987 及 32 的行程与行程拥有者为 daemon 及 root 的优先序号码加 1 :
renice +1 987 -u daemon root -p 32
注意 : 每一个行程(Process)都有一个唯一的 (unique) id

top

使用权限

所有使用者

使用方式

top [-] [d delay] [q] [c] [S] [s] [i] [n] [b]

说明

即时显示 process 的动态

参数

d : 改变显示的更新速度，或是在交谈式指令列(interactive command)按 s
q : 没有任何延迟的显示速度，如果使用者是有 superuser 的权限，则 top 将会以最高的优先序执行
c : 切换显示模式，共有两种模式，一是只显示执行档的名称，另一种是显示完整的路径与名称 S : 累积模式，会将已完成或消失的子行程 (dead child process) 的 CPU time 累积起来
s : 安全模式，将交谈式指令取消, 避免潜在的危机
i : 不显示任何闲置 (idle) 或无用 (zombie) 的行程
n : 更新的次数，完成后将会退出 top
b : 批次档模式，搭配 "n" 参数一起使用，可以用来将 top 的结果输出到档案内

范例

显示更新十次后退出 ;
top -n 10
使用者将不能利用交谈式指令来对行程下命令 :
top -s
将更新显示二次的结果输入到名称为 top.log 的档案里 :
top -n 2 -b < top.log

skill

使用权限

所有使用者

使用方式

skill [signal to send] [options] 选择程序的规则

说明

送个讯号给正在执行的程序,预设的讯息为 TERM (中断) , 较常使用的讯息为 HUP , INT, KILL , STOP , CONT ,和 0
讯息有三种写法:分别为 -9 , -SIGKILL , -KILL , 可以使用 -l 或 -L 已列出可使用的讯息。

参数

-f 快速模式/尚未完成
-i 互动模式/ 每个动作将要被确认
-v 详细输出/ 列出所选择程序的资讯
-w 智能警告讯息/ 尚未完成
-n 没有动作/ 显示程序代号
选择程序的规则可以是, 终端机代号,使用者名称,程序代号,命令名称。
-t 终端机代号 (tty 或 pty)
-u 使用者名称
-p 程序代号 (pid)
-c 命令名称 可使用的讯号:
以下列出已知的讯号名称,讯号代号,功能。
名称 (代号) 功能/ 描述
ALRM 14 离开
HUP 1 离开
INT 2 离开
KILL 9 离开/ 强迫关闭
PIPE 13 离开
POLL 离开
PROF 离开
TERM 15 离开
USR1 离开
USR2 离开
VTALRM 离开
STKFLT 离开/ 只适用于 i386, m68k, arm 和 ppc 硬体
UNUSED 离开/ 只适用于 i386, m68k, arm 和 ppc 硬体
TSTP 停止 /产生与内容相关的行为
TTIN 停止 /产生与内容相关的行为
TTOU 停止 /产生与内容相关的行为
STOP 停止 /强迫关闭
CONT 从新启动 /如果在停止状态则从新启动,否则忽略
PWR 忽略 /在某些系统中会离开
WINCH 忽略
CHLD 忽略
ABRT 6 核心
FPE 8 核心
ILL 4 核心
QUIT 3 核心
SEGV 11 核心
TRAP 5 核心
SYS 核心 /或许尚未实作

EMT 核心 /或许尚未实作
BUS 核心 /核心失败
XCPU 核心 /核心失败
XFSZ 核心 /核心失败

范例

停止所有在 PTY 装置上的程序
skill -KILL -v pts/*
停止三个使用者 user1 , user2 , user3
skill -STOP user1 user2 user3
其他相关的命令: kill

expr

使用权限

所有使用者

说明

字串长度
shell>> expr length "this is a test"
14
数字商数
shell>> expr 14 % 9
5
从位置处抓取字串
shell>> expr substr "this is a test" 3
5
is is
数字串 only the first character
shell>> expr index "testforthegame" e
2
字串真实重现
shell>> expr quote thisisatestformela
thisisatestformela

tr

1.比方说要把目录下所有的大写档名换为小写档名?
似乎有很多方式，"tr"是其中一种:

```
#!/bin/sh
dir="/tmp/testdir";
files=`find $dir -type f`;
for i in $files
do
dir_name=`dirname $i`;
ori_filename=`basename $i`
new_filename=`echo $ori_filename | tr [:upper:] [:lower:]` > /dev/null;

#echo $new_filename;
mv $dir_name/$ori_filename $dir_name/$new_filename
done
```

2.自己试验中...lowercase to uppercase

```
tr abcdef...[del] ABCDE...[del]
tr a-z A-Z
tr [:lower:] [:upper:]
shell>> echo "this is a test" | tr a-z
A-Z > www
shell>> cat www
THIS IS A TEST
```

3.去掉不想要的字串

```
shell>> tr -d this ###  去掉有关  t.e.s.t
this
man
man
test
e
```

4.取代字串

```
shell>> tr -s "this" "TEST"

this
TEST
th
TE
```

clear

用途

清除屏幕用。

使用方法

在 console 上输入 clear。

Reset/tset

使用方法

tset [-IQqrs] [-] [-e ch] [-i ch] [-k ch] [-m mapping] [terminal]

使用说明

reset 其实和 tset 是一同个命令，它的用途是设定终端机的状态。一般而言，这个命令会自动的从环境变数、命令列或是其它的组态档决定目前终端机的型态。如果指定型态是? 的话，这个程式会要求使用者输入终端机的型别。
软驱片 由于这个程式会将终端机设回原始的状态，除了在 login 时使用外，当系统终端机因为程式不正常执行而进入一些奇怪的状态时，你也可以用它来重设终端机 o
例如不小心把二进位档用 cat 指令进到终端机，常会有终端机不再回应键盘输入，或是回应一些奇怪字元的问题。此时就可以用 reset 将终端机回复至原始状态。

参数

-p 将终端机类别显示在萤幕上，但不做设定的动作。这个命令可以用来取得目前终端机的类别。
-e ch 将 erase 字元设成 ch
-i ch 将中断字元设成 ch
-k ch 将删除一行的字元设成 ch
-I 不要做设定的动作，如果没有使用选项 -Q 的话，erase、中断及删除字元的目前值依然会送到萤幕上。
-Q 不要显示 erase、中断及删除字元的值到萤幕上。
-r 将终端机类别印在萤幕上。
-s 将设定 TERM 用的命令用字串的型式送到终端机中，通常在 .login 或 .profile 中用

范例

让使用者输入一个终端机型别并将终端机设到该型别的预设状态。
reset ?
将 erase 字元设定 control-h
reset -e ^B
将设定用的字串显示在萤幕上
reset -s
Erase is control-B (^B).
Kill is control-U (^U).
Interrupt is control-C (^C).
TERM=xterm;

compress

使用权限

所有使用者

使用方式

compress [-dfvcV] [-b maxbits] [file ...]

说明

compress 是一个相当古老的 unix 档案压缩指令，压缩后的档案会加上一个 .Z 延伸档名以区别未压缩的档案，压缩后的档案可以以 uncompress 解压。若要将数个档案压成一个压缩档，必须先将档案 tar 起来再压缩。由于 gzip 可以产生更理想的压缩比例，一般人多已改用 gzip 为档案压缩工具。

参数

c 输出结果至标准输出设备（一般指荧幕）
f 强迫写入档案，若目的档已经存在，则会被覆盖 (force)
v 将程式执行的讯息印在荧幕上 (verbose)
b 设定共同字串数的上限，以位元计算，可以设定的值为 9 至 16 bits 。由于值越大，能使用的共同字串就 越多，压缩比例就越大，所以一般使用预设值 16 bits (bits)
d 将压缩档解压缩
V 列出版本讯息

范例

将 source.dat 压缩成 source.dat.Z ，若 source.dat.Z 已经存在，内容则会被压缩档覆盖。
compress -f source.dat
将 source.dat 压缩成 source.dat.Z ，并列印出压缩比例。
-v 与 -f 可以一起使用
compress -vf source.dat
将压缩后的资料输出后再导入 target.dat.Z 可以改变压缩档名。
compress -c source.dat > target.dat.Z
-b 的值越大，压缩比例就越大，范围是 9-16 ，预设值是 16 。
compress -b 12 source.dat
将 source.dat.Z 解压成 source.dat ，若档案已经存在，使用者按 y 以确定覆盖档案，若使用-df 程式则会自动覆盖档案。由于系统会自动加入 .Z 为延伸档名，所以 source.dat 会自动当作 source.dat.Z 处理。
compress -d source.dat
compress -d source.dat.Z

lpd

使用权限

所有使用者

使用方式

lpd [-l] [#port]
lpd 是一个常驻的印表机管理程式，它会根据 /etc/printcap 的内容来管理本地或远端的印表机。/etc/printcap 中定义的每一个印表机必须在/var/lpd 中有一个相对应的目录，目录中以 cf 开头的档案表示一个等待送到适当装置的印表工作。这个档案通常是由 lpr 所产生。
lpr 和 lpd 组成了一个可以离线工作的系统，当你使用 lpr 时，印表机不需要能立即可用，甚至不用存在。lpd 会自动监视印表机的状况，当印表机上线后，便立即将档案送交处理。这个得所有的应用程式不必等待印表机完成前一工作。

参数

-l: 将一些除错讯息显示在标准输出上。
#port: 一般而言，lpd 会使用 getservbyname 取得适当的 TCP/IP port，你可以使用这个参数强迫 lpd 使用指定的 port。

范例

这个程式通常是由 /etc/rc.d 中的程式在系统启始阶段执行。

lpq

作用

显示列表机贮列中未完成的工作

用法

lpq [l] [P] [user]

说明

lpq 会显示由 lpd 所管理的列表机贮列中未完成的项目。

范例

1. 显示所有在 lp 列表机贮列中的工作
lpq –P lp Rank Owner Job Files Total Size1st root 238
(standard input) 1428646 bytes
相关函数
lpr,lpc,lpd

lpr

使用权限

所有使用者

使用方式

lpr [-P printer]
将档案或是由标准输入送进来的资料送到印表机贮列之中，印表机管理程式 lpd 会在稍后将这个档案送给适当的程式或装置处理。lpr 可以用来将料资送给本地或是远端的主机来处

理。

参数

-p Printer: 将资料送至指定的印表机 Printer，预设值为 lp。

范例

将 http://www.c 和 kkk.c 送到印表机 lp。
lpr -Plp http://www.ckkk.c

lprm

作用

将一个工作由印表机贮列中移除

用法

/usr/bin/lprm [P] [file...]

说明

尚未完成的印表机工作会被放在印表机贮列之中，这个命令可用来将常未送到印表机的工作取消。由于每一个印表机都有一个独立的贮列，你可以用 -P 这个命令设定想要作用的印列机。如果没有设定的话，会使用系统预设的印表机。
这个命令会检查使用者是否有足够的权限删除指定的档案，一般而言，只有档案的拥有者或是系统管理员才有这个权限。

范例

将印表机 hpprinter 中的第 1123 号工作移除
prm -Phpprinter 1123
将第 1011 号工作由预设印表机中移除
lprm 1011

fdformat

使用权限

所有使用者

使用方式

fdformat [-n] device

使用说明

对指定的软碟机装置进行低阶格式化。使用这个指令对软碟格式化的时候，最好指定像是下面的装置：
/dev/fd0d360 软驱机 A: ，磁片为 360KB 软驱
/dev/fd0h1440 软驱机 A: ，磁片为 1.4MB 软驱
/dev/fd1h1200 软驱机 B: ，磁片为 1.2MB 软驱
如果使用像是 /dev/fd0 之类的装置，如果里面的软驱不是标准容量，格式化可能会失败。在这种情况之下，使用者可以用 setfdprm 指令先行指定必要参数。

参数

-n 关闭确认功能。这个选项会关闭格式化之后的确认步骤。

范例

fdformat -n /dev/fd0h1440
软驱机 A 的磁片格式化成 1.4MB 的磁片。并且省略确认的步骤。

mformat

使用权限

所有使用者

使用方式：

mformat [-t cylinders] [-h heads] [-s sectors]
[-l volume_label] [-F] [-I fsVer-sion] [-S sizecode] [-2 sectors_on_track_0]
[-M software_sector_size] [-a] [-X] [-C] [-H hidden_sectors] [-r root_sectors]
[-B boot_sector] [-0 rate_on_track_0] [-A rate_on_other_tracks] [-1] [-k] drive
在已经做过低阶格式化的磁片上建立 DOS 档案系统。如果在编译 mtools 的时候把 USE_2M 的参数打开，部分与 2M 格式相关的参数就会发生作用。否则这些参数（像是 S,2,1,M）不会发生作用。

参数

-t 磁柱（synlider）数
-h 磁头（head）数
-s 每一磁轨的磁区数
-l 标签
-F 将软驱格式化为 FAT32 格式，不过这个参数还在实验中。
-I 设定 FAT32 中的版本号。这当然也还在实验中。
-S 磁区大小代码，计算方式为 sector = 2^(大小代码+7)
-c 磁丛（cluster）的磁区数。如果所给定的数字会导致磁丛数超过 FAT 表的限制，mformat 会自动放大磁区数。
-s
-M 软体磁区大小。这个数字就是系统回报的磁区大小。通常是和实际的大小相同。
-a 如果加上这个参数，mformat 会产生一组 Atari 系统的序号给这块软碟。
-X 将软碟格式化成 XDF 格式。使用前必须先用 xdfcopy 指令对软碟作低阶格式化的动作。
-C 产生一个可以安装 MS-DOS 档案系统的软驱影像档（disk image）。当然对一个实体软驱机下这个参数是没有意义的。
-H 隐藏磁区的数目。这通常适用在格式化硬碟的分割区时，因为通常一个分割区的前面还有分割表。这个参数未经测试，能不用就不用。
-n 软驱序号
-r 根目录的大小，单位是磁区数。这个参数只对 FAT12 和 FAT16 有效。
-B 使用所指定的档案或是设备的开机磁区做为这片磁片或分割区的开机磁区。当然当中的硬体参数会随之更动。
-k 尽量保持原有的开机磁区。
-0 第 0 轨的资料传输率
-A 第 0 轨以外的资料传输率
-2 使用 2m 格式
-1 不使用 2m 格式

范例

mformat a:
这样会用预设值把 a: （就是 /dev/fd0）里的软盘格式化。

mkdosfs

使用权限

所有使用者

使用方式

mkdosfs [-c | -l filename]
[-f number_of_FATs]
[-F FAT_size]
[-i volume_id]
[-m message_file]
[-n volume_name]
[-r root_dir_entry]
[-s sector_per_cluster]
[-v]
device
[block_count]

说明

建立 DOS 档案系统。 device 指你想要建立 DOS 档案系统的装置代号。像是 /dev/hda1 等等。 block_count 则是你希望配置的区块数。如果 block_count 没有指定则系统会自动替你计算符合该装置大小的区块数。

参数

-c 建立档案系统之前先检查是否有坏轨。
-l 从得定的档案中读取坏轨记录。
-f 指定档案配置表(FAT , File Allocation Table)的数量。预设值为 2 。目前 Linux 的 FAT 档案系统不支援超过
2 个 FAT 表。通常这个不需要改。
-F 指定 FAT 表的大小，通常是 12 或是 16 个位元组。12 位元组通常用于软盘，16 位

元组用于一般硬碟的分割区，也就是所谓的
FAT16 格式。这个值通常系统会自己选定适当的值。在软盘上用 FAT16 通常不会发生作用，反之在硬碟上用 FAT12 亦然。
-i 指定 Volume ID。一般是一个 4 个位元组的数字，像是 2e203a47 。如果不给系统会自己产生。
-m 当使用者试图用这片磁片或是分割区开机，而上面没有作业系统时，系统会给使用者一段警告讯息。这个参数就是用来变更这个讯息的。你可以先用档案编辑好，然后用这个参数指定，或是用
-m -
这样系统会要求你直接输入这段文字。要特别注意的是，档案里的字串长度不要超过 418 个字，包括展开的跳栏符号（TAB）和换行符号（换行符号在 DOS 底下算两个字元！）
-n 指定 Volume Name，就是软驱标签。如同在 DOS 底下的 format 指令一样，给不给都可以。没有预设值。
-r 指定根目录底下的最大档案数。这里所谓的档案数包括目录。预设值是在软碟上是 112 或是 224 ，在硬碟上是 512。没事不要改这个数字。
-s 每一个磁丛（cluster）的磁区数。必须是 2 的次方数。不过除非你知道你在作什么，这个值不要乱给。
-v 提供额外的讯息

范例

mkdosfs -n Tester /dev/fd0
将 A 槽里的软盘格式化为 DOS 格式，并将标签设为 Tester

Linux 备份与压缩命令

用户经常需要备份计算机系统中的数据，为了节省存储空间，常常将备份文件进行压缩。下面分别介绍备份与压缩的命令。

tar

作用
tar 可以为文件和目录创建档案。利用 tar，用户可以为某一特定文件创建档案（备份文件），也可以在档案中改变文件，或者向档案中加入新的文件。tar 最初被用来在磁带上创建档案，现在，用户可以在任何设备上创建档案，如软盘。利用 tar 命令，可以把一大堆的文件和目录全部打包成一个文件，这对于备份文件或将几个文件组合成为一个文件以便于网络传输是非常有用的。Linux 上的 tar 是 GNU 版本的。
语法
tar [主选项+辅选项] 文件或者目录
使用该命令时，主选项是必须要有的，它告诉 tar 要做什么事情，辅选项是辅助使用的，可

以选用。

参数

c 创建新的档案文件。如果用户想备份一个目录或是一些文件，就要选择这个选项。

r 把要存档的文件追加到档案文件的末尾。例如用户已经作好备份文件，又发现还有一个目录或是一些文件忘记备份了，这时可以使用该选项，将忘记的目录或文件追加到备份文件中。

t 列出档案文件的内容，查看已经备份了哪些文件。

u 更新文件。就是说，用新增的文件取代原备份文件，如果在备份文件中找不到要更新的文件，则把它追加到备份文件的最后。

x 从档案文件中释放文件。

辅助选项

b 该选项是为磁带机设定的。其后跟一数字，用来说明区块的大小，系统预设值为 20（20*512 bytes）。

f 使用档案文件或设备，这个选项通常是必选的。

k 保存已经存在的文件。例如我们把某个文件还原，在还原的过程中，遇到相同的文件，不会进行覆盖。

m 在还原文件时，把所有文件的修改时间设定为现在。

M 创建多卷的档案文件，以便在几个磁盘中存放。

v 详细报告 tar 处理的文件信息。如无此选项，tar 不报告文件信息。

w 每一步都要求确认。

z 用 gzip 来压缩/解压缩文件，加上该选项后可以将档案文件进行压缩，但还原时也一定要使用该选项进行解压缩。

例子

1：把/home 目录下包括它的子目录全部做备份文件，备份文件名为 usr.tar。

$ tar cvf usr.tar /home

2：把/home 目录下包括它的子目录全部做备份文件，并进行压缩，备份文件名为 usr.tar.gz 。

$ tar czvf usr.tar.gz /home

3：把 usr.tar.gz 这个备份文件还原并解压缩。

$ tar xzvf usr.tar.gz

4：查看 usr.tar 备份文件的内容，并以分屏方式显示在显示器上。

$ tar tvf usr.tar | more

要将文件备份到一个特定的设备，只需把设备名作为备份文件名。

5：用户在/dev/fd0 设备的软盘中创建一个备份文件，并将/home 目录中所有的文件都拷贝到备份文件中。

$ tar cf /dev/fd0 /home

要恢复设备磁盘中的文件，可使用 xf 选项：

$ tar xf /dev/fd0

如果用户备份的文件大小超过设备可用的存贮空间，如软盘，您可以创建一个多卷的 tar 备份文件。M 选项指示 tar 命令提示您使用一个新的存贮设备，当使用 M 选项向一个软驱进行存档时，tar 命令在一张软盘已满的时候会提醒您再放入一张新的软盘。这样您就可以把 tar 档案存入几张磁盘中。

$ tar cMf /dev/fd0 /home

要恢复几张盘中的档案，只要将第一张放入软驱，然后输入有 x 和 M 选项的 tar 命令。在必

要时您会被提醒放入另外一张软盘。
$ tar xMf /dev/fd0

gzip

作用

减少文件大小有两个明显的好处，一是可以减少存储空间，二是通过网络传输文件时，可以减少传输的时间。gzip 是在 Linux 系统中经常使用的一个对文件进行压缩和解压缩的命令，既方便又好用。

语法

gzip [选项] 压缩（解压缩）的文件名

参数

-c 将输出写到标准输出上，并保留原有文件。
-d 将压缩文件解压。
-l 对每个压缩文件，显示下列字段：
压缩文件的大小
未压缩文件的大小
压缩比
未压缩文件的名字
-r 递归式地查找指定目录并压缩其中的所有文件或者是解压缩。
-t 测试，检查压缩文件是否完整。
-v 对每一个压缩和解压的文件，显示文件名和压缩比。
-num 用指定的数字 num 调整压缩的速度，-1 或--fast 表示最快压缩方法（低压缩比），-9 或--best 表示最慢压缩方法（高压缩比）。系统缺省值为 6。

假设一个目录/home 下有文件 mm.txt、sort.txt、xx.com。

例子

1：把/home 目录下的每个文件压缩成.gz 文件。
$ cd /home
$ gzip *
$ ls
m.txt.gz sort.txt.gz xx.com.gz
2：把例 1 中每个压缩的文件解压，并列出详细的信息。
$ gzip -dv *
mm.txt.gz 43.1%-----replaced with mm.txt
sort.txt.gz 43.1%-----replaced with sort.txt
xx.com.gz 43.1%-----replaced with xx.com
$ ls
mm.txt sort.txt xx.com
3：详细显示例 1 中每个压缩的文件的信息，并不解压。
$ gzip -l *
compressed uncompr. ratio uncompressed_name
277 445 43.1% mm.txt

278 445 43.1% sort.txt
277 445 43.1% xx.com
$ ls
mm.txt.gz sort.txt.gz xx.com.gz
4：压缩一个 tar 备份文件，如 usr.tar，此时压缩文件的扩展名为.tar.gz
$ gzip usr.tar
$ ls
usr.tar.gz

unzip

作用

用 MS Windows 下的压缩软件 winzip 压缩的文件如何在 Linux 系统下展开呢？可以用 unzip 命令，该命令用于解扩展名为.zip 的压缩文件。

语法

unzip [选项] 压缩文件名.zip

参数

-x 文件列表 解压缩文件，但不包括指定的 file 文件。
-v 查看压缩文件目录，但不解压。
-t 测试文件有无损坏，但不解压。
-d 目录 把压缩文件解到指定目录下。
-z 只显示压缩文件的注解。
-n 不覆盖已经存在的文件。
-o 覆盖已存在的文件且不要求用户确认。
-j 不重建文档的目录结构，把所有文件解压到同一目录下。

例子

1：将压缩文件 text.zip 在当前目录下解压缩。
$ unzip text.zip
2：将压缩文件 text.zip 在指定目录/tmp 下解压缩，如果已有相同的文件存在，要求 unzip 命令不覆盖原先的文件。
$ unzip -n text.zip -d /tmp
3：查看压缩文件目录，但不解压。
$ unzip -v text.zip

zgrep

这个命令的功能是在压缩文件中寻找匹配的正则表达式，用法和 grep 命令一样，只不过操作的对象是压缩文件。如果用户想看看在某个压缩文件中有没有某一句话，便可用 zgrep 命令。

在 Linux 环境下运行 DOS 命令

Linux 系统提供了一组称为 mtools 的可移植工具，可以让用户轻松地从标准的 DOS 软盘上读、写文件和目录。它们对 DOS 和 Linux 环境之间交换文件非常有用。它们是不具备共同的文件系统格式的系统之间交换文件的有力手段。对于一个 MS-DOS 的软盘，只要把软盘放在软驱中，就可以利用 mtools 提供的命令来访问软盘上的文件。

mtools 的主要命令如下：

mcd 目录名 改变 MSDOS 目录；

mcopy 源文件 目标文件 在 MSDOS 和 Unix 之间复制文件；

mdel 文件名 删除 MSDOS 文件；

mdir 目录名 显示 MSDOS 目录；

mformat 驱动器号 在低级格式化的软盘上创建 MSDOS 文件系统；

rnlabel 驱动器号 产生 MSDOS 卷标；

mmd 目录名 建立 MSDOS 目录；

mrd 目录名 删除 MSDOS 目录；

mren 源文件 目标文件 重新命名已存在的 MSDOS 文件；

mtype 文件名 显示 MSDOS 文件的内容。

这些命令和对应的不加 m 的 MSDOS 命令非常相似。

例 1：在 Linux 环境下看 DOS 盘最上层的目录的内容：

```
$ mdir a:
Volume in drive A has no label
Volume Serial Number is 15F6-3362
Directory of A:\
SS6 CPP 331 09-24-99 7:41 ss6.cpp
CH9
11-20-99 16:22 ch9
XXQ 0 11-20-99 16:24 xxq
95CZXTA DOC 36,864 06-15-98 22:51 95czxta.doc
95CZXTB DOC 39,936 06-16-98 7:18 95czxtb.doc
HTCA DOC 27,136 01-08-99 0:13 htca.doc
HTCB DOC 27,136 01-08-99 0:12 htcb.doc
6 file（s） 131,403 bytes
1 dir（s） 1,295,872 bytes free
```

例 2：将 DOS 盘上的文件 xxq 复制到当前目录下，并用 ls 命令进行验证。

```
$ mcopy a:\htca.doc
$ ls －l htca.doc
-rw-r— -r- - 1 xxq xxq 27136 Jan 1 01:80 htca.doc
```

Linux 文件内容查询命令

grep/fgrep/egrep

这组命令以指定模式搜索文件，并通知用户在什么文件中搜索到与指定的模式匹配的字符串，并打印出所有包含该字符串的文本行，在该文本行的最前面是该行所在的文件名。grep 命令一次只能搜索一个指定的模式；egrep 命令检索扩展的正则表达式（包括表达式组和可选项）；fgrep 命令检索固定字符串，它不识别正则表达式，是快速搜索命令。

这组命令在搜索与定位文件中特定的主题方面非常有用。要搜索的模式可以被认为是一些关键词，您可以用它们来搜索文件中包含的这些关键词。编写程序时，可以用它来寻找某一个函数，或是相关的词组。grep 命令的搜索功能比 fgrep 强大，因为 grep 命令的搜索模式可以是正则表达式，而 fgrep 却不能。有关正则表达式请参见 shell 一章。

该组命令中的每一个命令都有一组选项，利用这些选项可以改变其输出方式。例如，可以在搜索到的文本行上加入行号，或者只输出文本行的行号，或者输出所有与搜索模式不匹配的文本行，或只简单地输出已搜索到指定模式的文件名，并且可以指定在查找模式时忽略大小写。

这组命令在指定的输入文件中查找与模式匹配的行。如果没有指定文件，则从标准输入中读取。正常情况下，每个匹配的行被显示到标准输出。如果要查找的文件是多个，则在每一行输出之前加上文件名。

语法

grep [选项] [查找模式] [文件名 1，文件名 2,……]

egrep [选项] [查找模式] [文件名 1，文件名 2,……]

fgrep [选项] [查找模式] [文件名 1，文件名 2,……]

参数

- E 每个模式作为一个扩展的正则表达式对待。

- F 每个模式作为一组固定字符串对待（以新行分隔），而不作为正则表达式。

- b 在输出的每一行前显示包含匹配字符串的行在文件中的字节偏移量。

- c 只显示匹配行的数量。

- i 比较时不区分大小写。

- h 在查找多个文件时，指示 grep 不要将文件名加入到输出之前。

- l 显示首次匹配串所在的文件名并用换行符将其隔开。当在某文件中多次出现匹配串时，不重复显示此文件名。

- n 在输出前加上匹配串所在行的行号（文件首行行号为 1）。

- v 只显示不包含匹配串的行。

- x 只显示整行严格匹配的行。

- e expression 指定检索使用的模式。用于防止以“-”开头的模式被解释为命令选项。

- f expfile 从 expfile 文件中获取要搜索的模式，一个模式占一行。

对该组命令的使用还需注意以下方面

在命令后键入搜索的模式，再键入要搜索的文件。其中，文件名列表中也可以使用特殊字符，如“*”等，用来生成文件名列表。如果想在搜索的模式中包含有空格的字符串，可以用单引号把要搜索的模式括起来，用来表明搜索的模式是由包含空格的字符串组成。否则，Shell 将把空格认为是命令行参数的定界符，而 grep 命令将把搜索模式中的单词解释为

文件名列表中的一部分。在下面的例子中，grep 命令在文件 example 中搜索模式“text file”。

$ grep ’ text file’ example

用户可以在命令行上用 Shell 特殊字符来生成将要搜索的文件名列表。在下面的例子中，特殊字符“*”用来生成一个文件名列表，该列表包含当前目录下所有的文件。该命令将搜索出当前目录下所有文件中与模式匹配的行。

$ grep data *

特殊字符在搜索一组指定的文件时非常有用。例如，如果想搜索所有的 C 程序源文件中特定的模式，您可以用“*.c”来指定文件名列表。假设用户的 C 程序中包含一些不必要的转向语句（goto 语句），想要找到这些语句，可以用如下的命令来搜索并显示所有包含 goto 语句的代码行：

$ grep goto *.c

用户可以在命令行上键入搜索模式，也可以使用-f 选项从指定文件中读取要搜索的模式。在文件中，每个搜索模式占一行。如果经常要搜索一组常见字符串时，这个功能非常有用。在下面的例子中，用户要在文件 exam 中搜索字符串“editor”和“create”，就把要搜索的模式放置在文件 mypats 中，然后，grep 命令从文件 mypats 中读取要搜索的模式。

$ cat mypats
editor
create
$ grep -f mypats exam

find

功能

在目录结构中搜索文件，并执行指定的操作。此命令提供了相当多的查找条件，功能很强大。

语法

find 起始目录 寻找条件 操作

说明

find 命令从指定的起始目录开始，递归地搜索其各个子目录，查找满足寻找条件的文件并对之采取相关的操作。

该命令提供的寻找条件可以是一个用逻辑运算符 not、and、or 组成的复合条件。逻辑运算符 and、or、not 的含义为：

（1）and：逻辑与，在命令中用“-a”表示，是系统缺省的选项，表示只有当所给的条件都满足时，寻找条件才算满足。例如：

$ find －name ’ tmp’ －xtype c -user ’ inin’

该命令寻找三个给定条件都满足的所有文件。

（2）or：逻辑或，在命令中用“-o”表示。该运算符表示只要所给的条件中有一个满足时，寻找条件就算满足。例如：

$ find －name ’ tmp’ －o －name ’ mina*’

该命令查询文件名为’ tmp’或是匹配’ mina*’的所有文件。

（3）not：逻辑非，在命令中用“!”表示。该运算符表示查找不满足所给条件的文件。例如：

$ find ! －name ’ tmp’

该命令查询文件名不是’tmp’的所有文件。
需要说明的是：当使用很多的逻辑选项时，可以用括号把这些选项括起来。为了避免 Shell 本身对括号引起误解，在话号前需要加转义字符“\\”来去除括号的意义。
例：$ find \\（ －name ’tmp’ －xtype c -user ’inin’ \\）

寻找条件有以下选项：
首先，下列各个选项中的 n 值可以有三种输入方式，假设 n 为 20，则：
+20 表示 20 以后（21，22，23 等）
-20 表示 20 以前（19，18，17 等）
20 表示正好是 20
1. 以名称和文件属性查找。
- name ’字串’ 查找文件名匹配所给字串的所有文件，字串内可用通配符*、?、[ ]。
- lname ’字串’查找文件名匹配所给字串的所有符号链接文件，字串内可用通配符*、?、[ ]。
-gid n 查找属于 ID 号为 n 的用户组的所有文件。
-uid n 查找属于 ID 号为 n 的用户的所有文件。
-group ’字串’ 查找属于用户组名为所给字串的所有的文件。
-user ’字串’ 查找属于用户名为所给字串的所有的文件。
-empty 查找大小为 0 的目录或文件。
-path ’字串’ 查找路径名匹配所给字串的所有文件，字串内可用通配符*、?、[ ]。
-perm 权限 查找具有指定权限的文件和目录，权限的表示可以如 711，644。
-size n[bckw] 查找指定文件大小的文件，n 后面的字符表示单位，缺省为 b，代表 512 字节的块。
-type x 查找类型为 x 的文件，x 为下列字符之一：
b 块设备文件
c 字符设备文件
d 目录文件
p 命名管道（FIFO）
f 普通文件
l 符号链接文件（symbolic links）
s socket 文件
-xtype x 与-type 基本相同，但只查找符号链接文件。
2. 以时间为条件查找
- amin n 查找 n 分钟以前被访问过的所有文件。
- atime n 查找 n 天以前被访问过的所有文件。
- cmin n 查找 n 分钟以前文件状态被修改过的所有文件。
- ctime n 查找 n 天以前文件状态被修改过的所有文件。
- mmin n 查找 n 分钟以前文件内容被修改过的所有文件。
- mtime n 查找 n 天以前文件内容被修改过的所有文件。
3. 可执行的操作
- exec 命令名称 {} 对符合条件的文件执行所给的 Linux 命令，而不询问用户是否需要执行该命令。{}表示命令的参数即为所找到的文件；命令的末尾必须以“\\;”结束。
- ok 命令名称 { } 对符合条件的文件执行所给的 Linux 命令，与 exec 不同的是，它会询问用户是否需要执行该命令。

- ls 详细列出所找到的所有文件。
- fprintf 文件名 将找到的文件名写入指定文件。
- print 在标准输出设备上显示查找出的文件名。
- printf 格式 格式的写法请参考有关 C 语言的书。

例 1：查找当前目录中所有以 main 开头的文件，并显示这些文件的内容。

$ find . - name ‘main*’ - exec more {} \\;

例 2：删除当前目录下所有一周之内没有被访问过的 a .out 或*.o 文件。

$ find . \\（- name a.out - o - name ‘*.o’ \\）\\

> - atime +7 - exec rm {} \\;

说明如下： 命令中的“.”表示当前目录，此时 find 将从当前目录开始，逐个在其子目录中查找满足后面指定条件的文件。\\（和\\）表示括号（），其中的“\\”称为转义符。之所以这样写是由于对 Shell 而言，（和）另有不同的含义，而不是这里的用于组合条件的用途。“-name a.out”是指要查找名为 a.out 的文件；“- name ‘*.o’”是指要查找所有名字以 .o 结尾的文件。这两个- name 之间的-o 表示逻辑或（or），即查找名字为 a.out 或名字以 .o 结尾的文件，find 在当前目录及其子目录下找到这样的文件之后，再进行判断，看其最后访问时间是否在 7 天以前（条件－atime +7），若是，则对该文件执行命令 rm（- exec rm{ }\\;）。其中{ }代表当前查到的符合条件的文件名，\\；则是语法所要求的。上述命令中第一行的最后一个\\是续行符。当命令太长而在一行写不下时，可输入一个\\，之后系统将显示一个>，指示用户继续输入命令。

locate

功能

locate 命令用于查找文件，它比 find 命令的搜索速度快，它需要一个数据库，这个数据库由每天的例行工作（crontab）程序来建立。当我们建立好这个数据库后，就可以方便地来搜寻所需文件了。

语法

locate 相关字

例如

查找相关字 issue

$ locate issue

/etc/issue
/etc/issue.net
/usr/man/man5/issue.5
/usr/man/man5/issue.net.5 。

Linux 文件的复制、删除和移动命令

cp

功能

将给出的文件或目录拷贝到另一文件或目录中，就如同 DOS 下的 copy 命令一样，功能非常强大。

语法

cp [选项] 源文件或目录 目标文件或目录

说明

该命令把指定的源文件复制到目标文件或把多个源文件复制到目标目录中。

参数

- a 该选项通常在拷贝目录时使用。它保留链接、文件属性，并递归地拷贝目录，其作用等于 dpR 选项的组合。
- d 拷贝时保留链接。
- f 删除已经存在的目标文件而不提示。
- i 和 f 选项相反，在覆盖目标文件之前将给出提示要求用户确认。回答 y 时目标文件将被覆盖，是交互式拷贝。
- p 此时 cp 除复制源文件的内容外，还将把其修改时间和访问权限也复制到新文件中。
- r 若给出的源文件是一目录文件，此时 cp 将递归复制该目录下所有的子目录和文件。此时目标文件必须为一个目录名。
- l 不作拷贝，只是链接文件。

需要说明的是，为防止用户在不经意的情况下用 cp 命令破坏另一个文件，如用户指定的目标文件名是一个已存在的文件名，用 cp 命令拷贝文件后，这个文件就会被新拷贝的源文件覆盖，因此，建议用户在使用 cp 命令拷贝文件时，最好使用 i 选项。

$ cp - i exam1.c /usr/wang/shiyan1.c

该命令将文件 exam1.c 拷贝到/usr/wang 这个目录下，并改名为 shiyan1.c。若不希望重新命名，可以使用下面的命令：

$ cp exam1.c /usr/ wang/

$ cp - r /usr/xu/ /usr/liu/ 将/usr/xu 目录中的所有文件及其子目录拷贝到目录/usr/liu 中。

mv

功能

为文件或目录改名或将文件由一个目录移入另一个目录中。该命令如同 DOS 下的 ren 和 move 的组合。

语法

mv [选项] 源文件或目录 目标文件或目录

说明

视 mv 命令中第二个参数类型的不同（是目标文件还是目标目录），mv 命令将文件重命名或

将其移至一个新的目录中。当第二个参数类型是文件时，mv 命令完成文件重命名，此时，源文件只能有一个（也可以是源目录名），它将所给的源文件或目录重命名为给定的目标文件名。当第二个参数是已存在的目录名称时，源文件或目录参数可以有多个，mv 命令将各参数指定的源文件均移至目标目录中。在跨文件系统移动文件时，mv 先拷贝，再将原有文件删除，而链至该文件的链接也将丢失。

参数

- I 交互方式操作。如果 mv 操作将导致对已存在的目标文件的覆盖，此时系统询问是否重写，要求用户回答 y 或 n，这样可以避免误覆盖文件。

- f 禁止交互操作。在 mv 操作要覆盖某已有的目标文件时不给任何指示，指定此选项后，i 选项将不再起作用。

如果所给目标文件（不是目录）已存在，此时该文件的内容将被新文件覆盖。为防止用户在不经意的情况下用 mv 命令破坏另一个文件，建议用户在使用 mv 命令移动文件时，最好使用 i 选项。

需要注意的是，mv 与 cp 的结果不同。mv 好象文件“搬家”，文件个数并未增加，而 cp 对文件进行复制，文件个数增加了。

例 1：将/usr/xu 中的所有文件移到当前目录（用“.”表示）中：

$ mv /usr/xu/ * .

例 2：将文件 wch.txt 重命名为 wjz.doc

$ mv wch.txt wjz.doc

rm

功能

在 linux 中创建文件很容易，系统中随时会有文件变得过时且毫无用处。用户可以用 rm 命令将其删除。该命令的功能为删除一个目录中的一个或多个文件或目录，它也可以将某个目录及其下的所有文件及子目录均删除。对于链接文件，只是删除了链接，原有文件均保持不变。

语法

rm [选项] 文件…

如果没有使用- r 选项，则 rm 不会删除目录。

参数

- f 忽略不存在的文件，从不给出提示。

- r 指示 rm 将参数中列出的全部目录和子目录均递归地删除。

- i 进行交互式删除。

使用 rm 命令要格外小心。因为一旦一个文件被删除，它是不能被恢复的。例如，用户在输入 cp，mv 或其他命令时，不小心误输入了 rm 命令，当用户按了回车键并认识到自己的错误时，已经太晚了，文件已经没有了。为了防止此种情况的发生，可以使用 rm 命令中的 i 选项来确认要删除的每个文件。如果用户输入 y，文件将被删除。如果输入任何其他东西，文件将被保留。在下一个例子中，用户要删除文件 test 和 example。然后会被要求对每个文件进行确认。用户最终决定删除 example 文件，保留 test 文件。

$ rm - ii test example Remove test ?n

Remove example ?y

Linux 与用户有关的命令

passwd

出于系统安全考虑，Linux 系统中的每一个用户除了有其用户名外，还有其对应的用户口令。因此使用 useradd 命令增加时，还需使用 passwd 命令为每一位新增加的用户设置口令；用户以后还可以随时用 passwd 命令改变自己的口令。
该命令的一般格式为： passwd [用户名]
其中用户名为需要修改口令的用户名。只有超级用户可以使用“passwd 用户名”修改其他用户的口令，普通用户只能用不带参数的 passwd 命令修改自己的口令。
该命令的使用方法如下：
输入
passwd< Enter>；
在（current） UNIX passwd:下输入当前的口令
在 new password:提示下输入新的口令（在屏幕上看不到这个口令）：
系统提示再次输入这个新口令。
输入正确后，这个新口令被加密并放入/etc/shdow 文件.
选取一个不易被破译的口令是很重要的.
选取口令应遵守如下规则：
口令应该至少有六位（最好是八位）字符；
口令应该是大小写字母、标点符号和数字混杂的。
超级用户修改其他用户（xxq）的口令的过程如下，

```
# passwd xxq
    New UNIX password:
    Retype new UNIX password:
    passwd: all authentication tokens updated successfully
#
```

su

这个命令非常重要。它可以让一个普通用户拥有超级用户或其他用户的权限，也可以让超级用户以普通用户的身份做一些事情。普通用户使用这个命令时必须有超级用户或其他用户的口令。如要离开当前用户的身份，可以打 exit。
该命令的一般形式为
su [选项] [?] [使用者帐号]
说明
若没有指定使用者帐号，则系统预设值为超级用户 root。
参数
? c 执行一个命令后就结束。
? 加了这个减号的目的是使环境变量和欲转换的用户相同。
? m 保留环境变量不变。

例 1： 变成 root 用户
$ su ?
password: 【输入超级用户的密码】
例 2： 变成 xu 使用者，并执行一个命令就结束。
$ su -xu ? c “rmdir cat1”

Linux 系统管理命令

wall

功能
这个命令的功能是对全部已登录的用户发送信息，用户可以先把要发送的信息写好存入一个文件中，然后输入：
wall < 文件名
这样就能对所有的用户发送信息了。
在上面的例子中符号“<”表示输入重定向，有关它的含义和用法请参阅第十章的有关内容。
例如：
wall ‘Thank you!’
Broadcast message from root （tty1） Fri Nov 26 14：15：07 1999…
Thank you!
#
执行以上命令后，用户的屏幕上显示出“Thank you!”信息后，并不出现系统提示符$（#），再次按回车键后，屏幕出现系统提示符。

write

功能
write 命令的功能是向系统中某一个用户发送信息。
格式
write 用户帐号 [终端名称]
例如：
$ write xxq hello
此时系统进入发送信息状态，用户可以输入要发送的信息，输入完毕，希望退出发送状态时，按组合键< Ctrl+c>即可。
上述命令执行的结果是，用户 xxq 的屏幕上会显示：
message from test@test.tlc.com.cn tty1 at 15:51…
hello
EOF

mesg

mesg 命令设定是否允许其他用户用 write 命令给自己发送信息。如果允许别人给自己发送信息，输入命令：
mesg y
否则，输入：
mesg n
对于超级用户，系统的默认值为 n；而对于一般用户系统的默认值为 y。 如果 mesg 后不带任何参数，则显示当前的状态是 y 还是 n，如：
$ mesg is
y
或：
mesg is
n

sync

功能
sync 命令是在关闭 Linux 系统时使用的。 用户需要注意的是，不能用简单的关闭电源的方法关闭系统，因为 Linux 象其他 Unix 系统一样，在内存中缓存了许多数据，在关闭系统时需要进行内存数据与硬盘数据的同步校验，保证硬盘数据在关闭系统时是最新的，只有这样才能确保数据不会丢失。一般正常的关闭系统的过程是自动进行这些工作的，在系统运行过程中也会定时做这些工作，不需要用户干预。
sync 命令是强制把内存中的数据写回硬盘，以免数据的丢失。用户可以在需要的时候使用此命令。
格式
sync

shutdown

功能
shutdown 命令可以安全地关闭或重启 Linux 系统，它在系统关闭之前给系统上的所有登录用户提示一条警告信息。该命令还允许用户指定一个时间参数，可以是一个精确的时间，也可以是从现在开始的一个时间段。精确时间的格式是 hh:mm，表示小时和分钟；时间段由“+”和分钟数表示。系统执行该命令后，会自动进行数据同步的工作。
格式
shutdown [选项] [时间] [警告信息]
参数
- k 并不真正关机，而只是发出警告信息给所有用户。
- r 关机后立即重新启动。
- h 关机后不重新启动。

- f 快速关机，重启动时跳过 fsck。
- n 快速关机，不经过 init 程序。
- c 取消一个已经运行的 shutdown。
需要特别说明的是，该命令只能由超级用户使用。
例 1：系统在十分钟后关机，并且马上重新启动。
shutdown – r +10
例 2：系统马上关机，并且不重新启动。
shutdown – h now

free

功能
free 命令的功能是查看当前系统内存的使用情况，它显示系统中剩余及已用的物理内存和交换内存，以及共享内存和被核心使用的缓冲区。
格式
free [-b | -k | -m]
参数
-b 以字节为单位显示。
-k 以 K 字节为单位显示。
-m 以兆字节为单位显示。
例：
$ free
total used free shared buffers cached
Mem: 63076 32020 31056 8204 16360 6048
-/+ buffers/cache: 9612 53464
Swap: 64476 2240 62236

uptime

功能
uptime 命令显示系统已经运行了多长时间，它依次显示下列信息：现在时间、系统已经运行了多长时间、目前有多少登录用户、系统在过去的 1 分钟、5 分钟和 15 分钟内的平均负载。
格式
uptime
例：
uptime
4:43pm up 1 day, 5:51, 2 user, load average: 0.01, 0.01, 0.00

Linux 的常用网络命令

计算机网络的主要优点是能够实现资源和信息的共享，并且用户可以远程访问信息。Linux 提供了一组强有力的网络命令来为用户服务，这些工具能够帮助用户登录到远程计算机上、传输文件和执行远程命令等。
本章介绍下列几个常用的有关网络操作的命令：
ftp 传输文件
telnet 登录到远程计算机上
r - 使用各种远程命令
netstat 查看网络的状况
nslookup 查询域名和 IP 地址的对应
finger 查询某个使用者的信息
ping 查询某个机器是否在工作
使用 ftp 命令进行远程文件传输

ftp

是标准的文件传输协议的用户接口。ftp 是在 TCP/IP 网络上的计算机之间传输文件的简单有效的方法。它允许用户传输 ASCII 文件和二进制文件。
在 ftp 会话过程中，用户可以通过使用 ftp 客户程序连接到另一台计算机上。从此，用户可以在目录中上下移动、列出目录内容、把文件从远程机拷贝到本地机上、把文件从本地机传输到远程系统中。
需要注意的是，如果用户没有那个文件的存取权限，就不能从远程系统中获得文件或向远程系统传输文件。为了使用 ftp 来传输文件，用户必须知道远程计算机上的合法用户名和口令。这个用户名/口令的组合用来确认 ftp 会话，并用来确定用户对要传输的文件可以进行什么样的访问。另外，用户显然需要知道对其进行 ftp 会话的计算机的名字或 IP 地址。
Ftp 命令的功能是在本地机和远程机之间传送文件。该命令的一般格式如下：
$ ftp 主机名/IP
其中“主机名/IP”是所要连接的远程机的主机名或 IP 地址。在命令行中，主机名属于选项，如果指定主机名，ftp 将试图与远程机的 ftp 服务程序进行连接；如果没有指定主机名，ftp 将给出提示符，等待用户输入命令：
$ ftp ftp >
此时在 ftp>提示符后面输入 open 命令加主机名或 IP 地址，将试图连接指定的主机。 不管使用哪一种方法，如果连接成功，需要在远程机上登录。用户如果在远程机上有帐号，就可以通过 ftp 使用这一帐号并需要提供口令。
在远程机上的用户帐号的读写权限决定该用户在远程机上能下载什么文件和将上载文件放到哪个目录中。 如果没有远程机的专用登录帐号，许多 ftp 站点设有可以使用的特殊帐号。这个帐号的登录名为 anonymous（也称为匿名 ftp），当使用这一帐号时，要求输入 email 地址作为口令。
如果远程系统提供匿名 ftp 服务，用户使用这项服务可以登录到特殊的，供公开使用的目录。一般专门提供两个目录：pub 目录和 incoming 目录。pub 目录包含该站点供公众使用的所

有文件，incoming 目录存放上载到该站点的文件。
一旦用户使用 ftp 在远程站点上登录成功，将得到“ftp>”提示符。现在可以自由使用 ftp 提供的命令，可以用 help 命令取得可供使用的命令清单，也可以在 help 命令后面指定具体的命令名称，获得这条命令的说明。
最常用的命令有：

ls 列出远程机的当前目录
cd 在远程机上改变工作目录
lcd 在本地机上改变工作目录
ascii 设置文件传输方式为 ASCII 模式
binary 设置文件传输方式为二进制模式
close 终止当前的 ftp 会话
hash 每次传输完数据缓冲区中的数据后就显示一个#号
get（mget） 从远程机传送指定文件到本地机
put（mput） 从本地机传送指定文件到远程机
open 连接远程 ftp 站点
quit 断开与远程机的连接并退出 ftp
? 显示本地帮助信息
! 转到 Shell 中
下面简单将 ftp 常用命令作一简介。

启动 ftp 会话

open 命令用于打开一个与远程主机的会话。该命令的一般格式是： open 主机名/IP
如果在 ftp 会话期间要与一个以上的站点连接，通常只用不带参数的 ftp 命令。如果在会话期间只想与一台计算机连接，那么在命令行上指定远程主机名或 IP 地址作为 ftp 命令的参数。
终止 ftp 会话
close、disconnect、quit 和 bye 命令用于终止与远程机的会话。close 和 disronnect 命令关闭与远程机的连接，但是使用户留在本地计算机的 ftp 程序中。quit 和 bye 命令都关闭用户与远程机的连接，然后退出用户机上的 ftp 程序。

改变目录

“cd [目录]”命令用于在 ftp 会话期间改变远程机上的目录，lcd 命令改变本地目录，使用户能指定查找或放置本地文件的位置。

远程目录列表

ls 命令列出远程目录的内容，就像使用一个交互 shell 中的 ls 命令一样。ls 命令的一般格式是： ls [目录] [本地文件]
如果指定了目录作为参数，那么 ls 就列出该目录的内容。如果给出一个本地文件的名字，那么这个目录列表被放入本地机上您指定的这个文件中。

从远程系统获取文件

get 和 mget 命令用于从远程机上获取文件。get 命令的一般格式为： get 文件名
您还可以给出本地文件名，这个文件名是这个要获取的文件在您的本地机上创建时的文件名。如果您不给出一个本地文件名，那么就使用远程文件原来的名字。
mget 命令一次获取多个远程文件。mget 命令的一般格式为： mget 文件名列表
使用用空格分隔的或带通配符的文件名列表来指定要获取的文件，对其中的每个文件都要求用户确认是否传送。

向远程系统发送文件

put 和 mput 命令用于向远程机发送文件。Put 命令的一般格式为： put 文件名
mput 命令一次发送多个本地文件，mput 命令的一般格式为：mput 文件名列表
使用用空格分隔的或带通配符的文件名列表来指定要发送的文件。对其中的每个文件都要求用户确认是否发送。

改变文件传输模式

默认情况下，ftp 按 ASCII 模式传输文件，用户也可以指定其他模式。ascii 和 brinary 命令的功能是设置传输的模式。用 ASCII 模式传输文件对纯文本是非常好的，但为避免对二进制文件的破坏，用户可以以二进制模式传输文件。

检查传输状态

传输大型文件时，可能会发现让 ftp 提供关于传输情况的反馈信息是非常有用的。hash 命令使 ftp 在每次传输完数据缓冲区中的数据后，就在屏幕上打印一个#字符。本命令在发送和接收文件时都可以使用。

ftp 中的本地命令

当您使用 ftp 时，字符“!”用于向本地机上的命令 shell 传送一个命令。如果用户处在 ftp 会话中，需要 shell 做某些事，就很有用。例如用户要建立一个目录来保存接收到的文件。如果输入!mkdir newdir，那么 Linux 就在用户当前的本地目录中创建一个名为 new_dir 的目录。

telnet

用户使用 telnet 命令进行远程登录。该命令允许用户使用 telnet 协议在远程计算机之间进行通信，用户可以通过网络在远程计算机上登录，就像登录到本地机上执行命令一样。
为了通过 telnet 登录到远程计算机上，必须知道远程机上的合法用户名和口令。虽然有些系统确实为远程用户提供登录功能，但出于对安全的考虑，要限制来宾的操作权限，因此，这种情况下能使用的功能是很少的。当允许远程用户登录时，系统通常把这些用户放在一个受限制的 shell 中，以防系统被怀有恶意的或不小心的用户破坏。
用户还可以使用 telnet 从远程站点登录到自己的计算机上，检查电子邮件、编辑文件和运行程序，就像在本地登录一样。
但是，用户只能使用基于终端的环境而不是 X Wndows 环境，telnet 只为普通终端提供终端仿真，而不支持 X Wndow 等图形环境。
telnet 命令的一般形式为：
telnet 主机名/IP
其中“主机名/IP”是要连接的远程机的主机名或 IP 地址。如果这一命令执行成功，将从远程机上得到 login：提示符。
使用 telnet 命令登录的过程如下： $ telnet 主机名/IP 启动 telnet 会话。 一旦 telnet 成功地连接到远程系统上，就显示登录信息并提示用户输人用户名和口令。如果用户名和口令输入正确，就能成功登录并在远程系统上工作。
在 telnet 提示符后面可以输入很多命令，用来控制 telnet 会话过程，在 telnet 联机帮助手册中对这些命令有详细的说明。
下面是一台 Linux 计算机上的 telnet 会话举例：
$ telnet server. somewhere. com Trying 127.0.0.1… Connected to serve.
somewhere. com. Escape character is \'?]\'. “TurboLinux release 4. 0 （Colgate）
kernel 2.0.18 on an I486 login: bubba password: Last login:Mon Nov

l5 20:50:43 for localhost Linux 2. 0.6. （Posix）. server: ~$ server: ~$
logout Connection closed by foreign host $
用户结束了远程会话后，一定要确保使用 logout 命令退出远程系统。然后 telnet 报告远程会话被关闭，并返回到用户的本地机的 Shell 提示符下。

r-系列命令

除 ftp 和 telnet 以外，还可以使用 r-系列命令访问远程计算机和在网络上交换文件。 使用 r-系列命令需要特别注意，因为如果用户不小心，就会造成严重的安全漏洞。用户发出一个 r-系列命令后，远程系统检查名为/etc/hosts.equiv 的文件，以查看用户的主机是否列在这个文件中。如果它没有找到用户的主机，就检查远程机上同名用户的主目录中名为．rhosts 的文件，看是否包括该用户的主机。如果该用户的主机包括在这两个文件中的任何一个之中，该用户执行 r-系列命令就不用提供口令。
虽然用户每次访问远程机时不用键入口令可能是非常方便的，但是它也可能会带来严重的安全问题。我们建议用户在建立/etc/hosts.equiv 和.rhosts 文件之前，仔细考虑 r-命令隐含的安全问题。

rlogin
功能：rlogin 是“remote login”（远程登录）的缩写。该命令与 telnet 命令很相似，允许用户启动远程系统上的交互命令会话。
格式： rlogin [-8EKLdx] [-e char] [-k realm] [- l username] host
参数：
-8 此选项始终允许 8 位输入数据通道。该选项允许发送格式化的 ANSI 字符和其他的特殊代码。如果不用这个选项，除非远端的终止和启动字符不是或，否则就去掉奇偶校验位。
-E 停止把任何字符当作转义字符。当和-8 选项一起使用时，它提供一个完全的透明连接。
-K 关闭所有的 Kerberos 确认。只有与使用 Kerberos 确认协议的主机连接时才使用这个选项。
-L 允许 rlogin 会话在 litout 模式中运行。要了解更多信息，请查阅 tty 联机帮助。
-d 打开与远程主机进行通信的 TCP sockets 的 socket 调试。要了解更多信息，请查阅 setsockopt 的联机帮助。
-e 为 rlogin 会话设置转义字符，默认的转义字符是“~”，用户可以指定一个文字字符或一个\\nnn 形式的八进制数。
-k 请求 rlogin 获得在指定区域内的远程主机的 Kerberos 许可，而不是获得由 krb_realmofhost（3）确定的远程主机区域内的远程主机的 Kerberos 许可。
-x 为所有通过 rlogin 会话传送的数据打开 DES 加密。这会影响响应时间和 CPU 利用率，但是可以提高安全性。

rsh
功能：rsh 是“remote shell”（远程 shell）的缩写。 该命令在指定的远程主机上启动一个 shell 并执行用户在 rsh 命令行中指定的命令。如果用户没有给出要执行的命令，rsh 就用 rlogin 命令使用户登录到远程机上。
格式：rsh [-Kdnx] [-k realm] [-l username] host [command]
command 可以是从 shell 提示符下键人的任何 Linux 命令。

参数：
-K 关闭所有的 Kerbero 确认。该选项只在与使用 Kerbero 确认的主机连接时才使用。
-d 打开与远程主机进行通信的 TCP sockets 的 socket 调试。要了解更多的信息，请查阅 setsockopt 的联机帮助。
-k 请求 rsh 获得在指定区域内的远程主机的 Kerberos 许可，而不是获得由 krb_relmofhost（3）确定的远程主机区域内的远程主机的 Kerberos 许可。
-l 缺省情况下，远程用户名与本地用户名相同。本选项允许指定远程用户名，如果指定了远程用户名，则使用 Kerberos 确认，与在 rlogin 命令中一样。
-n 重定向来自特殊设备/dev/null 的输入。
-x 为传送的所有数据打开 DES 加密。这会影响响应时间和 CPU 利用率，但是可以提高安全性。
Linux 把标准输入放入 rsh 命令中,并把它拷贝到要远程执行的命令的标准输入中。它把远程命令的标准输出拷贝到 rsh 的标准输出中。它还把远程标准错误拷贝到本地标准错误文件中。任何退出、中止和中断信号都被送到远程命令中。当远程命令终止了，rsh 也就终止了。

rcp

功能：rcp 代表“remote file copy”（远程文件拷贝）。该命令用于在计算机之间拷贝文件。rcp 命令有两种格式。第一种格式用于文件到文件的拷贝；第二种格式用于把文件或目录拷贝到另一个目录中。
格式：
rcp [-px] [-k realm] file1 file2 rcp [-px] [-r] [-k realm] file
directory 每个文件或目录参数既可以是远程文件名也可以是本地文件名。远程文件名具有如下形式：rname@rhost：path，其中 rname 是远程用户名，rhost 是远程计算机名，path 是这个文件的路径。
参数：
-r 递归地把源目录中的所有内容拷贝到目的目录中。要使用这个选项，目的必须是一个目录。
-p 试图保留源文件的修改时间和模式，忽略 umask。
-k 请求 rcp 获得在指定区域内的远程主机的 Kerberos 许可，而不是获得由 krb_relmofhost（3）确定的远程主机区域内的远程主机的 Kerberos 许可。
-x 为传送的所有数据打开 DES 加密。这会影响响应时间和 CPU 利用率，但是可以提高安全性。
如果在文件名中指定的路径不是完整的路径名，那么这个路径被解释为相对远程机上同名用户的主目录。如果没有给出远程用户名，就使用当前用户名。如果远程机上的路径包含特殊 shell 字符，需要用反斜线（\\）、双引号（”）或单引号（’）括起来，使所有的 shell 元字符都能被远程地解释。
需要说明的是，rcp 不提示输入口令，它通过 rsh 命令来执行拷贝。 - Turbolinux 提供稿件