

Linux 常用命令指南 v 0.1

by Skywind 2004/5/25

目录

一. 文件和目录操作.....	2
命令名称: cd.....	2
命令名称: ls.....	2
命令名称: cp.....	3
命令名称: rm.....	3
命令名称: mv.....	4
命令名称: mkdir.....	4
命令名称: cat.....	5
命令名称: touch.....	5
命令名称: chmod.....	6
命令名称: chown.....	8
命令名称: find.....	8
命令名称: locate.....	9
命令名称: more.....	10
命令名称: less.....	11
命令名称: ln.....	11
命令名称: df.....	12
命令名称: du.....	13
二. 系统管理操作.....	14
命令名称: crontab.....	14
命令名称: date.....	15
命令名称: uptime.....	16
命令名称: last.....	17
命令名称: passwd.....	18
命令名称: useradd.....	18
命令名称: who.....	18
命令名称: ps.....	19
命令名称: top.....	20
命令名称: kill.....	21
三. 如何学习命令.....	22
方法一: man.....	22
方法二: Google.....	22
四. 编者附注.....	23

一. 文件和目录操作

命令名称：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 ../../..
```

命令名称：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）。

-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
```

命令名称：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
```

命令名称：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
```

命令名称：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
```

命令名称：mkdir

使用权限：所有使用者

使用方式：mkdir [选项] 目录...

说明：若目录不是已经存在则创建目录。

长选项必须用的参数在使用短选项时也是必须的。

-m, --mode=模式 设定权限<模式>（类似 chmod），而不是 rwxrwxrwx 减 umask

-p, --parents 需要时创建上层目录，如目录早已存在则不当作错误

-v, --verbose 每次创建新目录都显示信息

--help 显示此帮助信息并离开

--version 显示版本信息并离开

范例:

创建/root/123 目录, /root/目录已经存在

```
mkdir /root/123
```

创建目录层次/1/2/3, /1/并不存在

```
mkdir -p /1/2/3
```

命令名称: 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

命令名称: 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，将档案的时间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`，就是月日時与分。

```
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
```

命令名称：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。

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(即以递归的方式逐个变更)
--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

命令名称：chown

使用权限：root

使用方式：chown [-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 IDgroup：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(即以递归的方式逐个变更)--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 lampont：

```
chown -R lampont:users *
```

命令名称：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
```

命令名称: locate

使用权限: 所有使用者

使用方式: locate [-qi] [-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
```

的型式就可以了。 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

命令名称：more

使用权限：所有使用者

使用方式: `more [-dlfpsu] [-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。

命令名称: 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。

命令名称: ln

使用权限 : 所有使用者

使用方式 : `ln [options] source dest`, 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，而软连结则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连结则由参数决定。不论是硬连结或软连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。

-f：连结时先将与 dist 同档名的档案删除-d：允许系统管理者硬连结自己的目录-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：进行软连结(symbolic link)-v：在连结之前显示其档名-b：将在连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
```

命令名称：df

使用权限：所有使用者

用法：df [选项]... [文件]...

说明：显示每个<文件>所在的文件系统的信息，默认是显示所有文件系统。

参数：

- a, --all 包括大小为 0 个块的文件系统
- B, --block-size=大小 块以指定<大小>的字节为单位
- h, --human-readable 以容易理解的格式印出文件系统大小 (例如 1K 234M 2G)
- H, --si 类似 -h，但取 1000 的次方而不是 1024
- i, --inodes 显示 inode 信息而非块使用量
- k 即 --block-size=1K
- l, --local 只显示本机的文件系统
- no-sync 取得使用量数据前不进行 sync 动作 (默认)
- t, --type=类型 只印出指定<类型>的文件系统信息
- T, --print-type 印出文件系统类型
- x, --exclude-type=类型 只印出不是指定<类型>的文件系统信息

<大小>可以是以下的单位 (单位前可加上整数)：

kB=1000, K=1024, MB=1000000, M=1048576, 还有 G、T、P、E、Z、Y 如此类推。

范例：

用易懂的格式显示所有的文件系统：

```
df -h
```

命令名称：du

使用权限：所有使用者

用法：du [选项]... [文件]...

说明：总结每个<文件>的磁盘用量，目录则取总用量。

参数：

```
-a, --all           write counts for all files, not just directories
-B, --block-size=SIZE use SIZE-byte blocks
-b, --bytes         print size in bytes
-c, --total         produce a grand total
-D, --dereference-args dereference FILEs that are symbolic links
-h, --human-readable 以容易理解的格式印出文件大小 (例如 1K 234M 2G)
-H, --si            类似 -h, 但取 1000 的次方而不是 1024
-k                  即 --block-size=1K
-l, --count-links    连硬链接的大小也计算在内
-L, --dereference     找出任何符号链接指示的真正目的地
-S, --separate-dirs   不包括子目录的占用量
-s, --summarize       只分别计算命令列中每个参数所占的总用量
-x, --one-file-system skip directories on different filesystems
-X FILE, --exclude-from=FILE Exclude files that match any pattern in FILE.
    --exclude=PATTERN Exclude files that match PATTERN.
    --max-depth=N      print the total for a directory (or file, with --all)
                        only if it is N or fewer levels below the command
                        line argument; --max-depth=0 is the same as
                        --summarize
```

<大小>可以是以下的单位 (单位前可加上整数)：

kB=1000, K=1024, MB=1000000, M=1048576, 还有 G、T、P、E、Z、Y 如此类推。

范例：

用易读懂的格式、比较少的输出方式显示/tmp 目录的大小，并给出一个统计：

```
du /tmp -hsc
```

查看当前目录下所有目录以及子目录的大小

```
du -h
```

二. 系统管理操作

命令名称: crontab

使用权限 : 所有使用者

使用方式 :

```
crontab [ -u user ] file
```

```
crontab [ -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进程式之用, 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进程式, 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 7 点第 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 :

```
*/20 6-12 * * * /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。

命令名称: 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。

命令名称: 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

命令名称：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
```

命令名称：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：

-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。

命令名称：useradd

使用权限：所有使用者

使用方式：

usage: useradd [-u uid [-o]] [-g group] [-G group,...]

[-d home] [-s shell] [-c comment] [-m [-k template]]

[-f inactive] [-e expire] [-p passwd] [-M] [-n] [-r] name

说明：用来创建系统用户（帐号）

参数：

-u 指定用户的 uid 号

-g 指定用户的默认组

-G 指定用户的所属组（可以为多个）

-d 指定用户的家目录

-s 指定用户的 shell

name 系统用户（帐号）名

命令名称：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

命令名称：ps

使用权限：所有使用者

使用方式：ps [options] [--help]

说明：显示瞬间进程(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 列出所有的进程
- 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进程

au(x) 输出格式：

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

USER：进程拥有者

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：该进程的状态：

- D：不可中断的静止（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
- <：高优先序的进程

N: 低优先序的进程

L: 有记忆体分页分配并锁在记忆体内 (即时系统或握 A I/O)

START: 进程开始时间

TIME: 执行的时间

COMMAND: 所执行的指令

范例:

```
% ps
```

```
PID TTY TIME CMD
```

```
2791 tty0 00:00:00 tcsh
```

```
3092 tty0 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]
```

命令名称: 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进程 (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) 的进程
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进程下命令:

```
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
```

命令名称: 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) 给进程 id 为 pid 的进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 的进程砍掉 (kill) :

```
kill -9 323
```

将 pid 为 456 的进程重跑 (restart) :

```
kill -HUP 456
```

三. 如何学习命令

方法一: **man**

对一条你不知道作用或者不清楚具体参数的命令,你可以使用 **man** 命令来查看命令的文档。**man** 是 **manual** 的缩写。几乎每一个命令都有相应的文档。要深入了解 **man**, 请输入以下命令:

```
man man
```

一般格式:

man 命令

快捷键:

ctrl+f 向下翻页

ctrl+b 向上翻页

q 退出

方法二: **Google**

几乎任何你想获得的信息都可以在 www.google.com 上找到。学习 Linux 要学会使用网络和搜索引擎。

四. 编者附注

Linux 之命令成百上千，浩如烟海，这篇短短的指南无法逐一列举，也着实无此必要，既为“指南”，重在提点重要及指明学习之方向，所以本篇在罗列若干命令之后有“如何学习命令”的第三节，其实此节当属本篇之核心，是谓“授之以鱼，不如授之以渔”，学习的方法至关重要。

此份《Linux 常用命令指南》现在还是内部测试版，命令的解说将来定会有所增减，文中大半为整理编辑得来，加之编纂仓促，其中不当不妥之处还望诸位多加指正。

大汉网络
研究部
徐刚
2004-6-7