

运维工程师必会的 109 个 Linux 命令

1 文件管理

1.1 basename

1.1.1 功能说明

从文件名中去掉路径和扩展名

例：basename include/stdio.h.h

Output "stdio".

1.1.2 选项

--help 显示此帮助信息并离开

--version 显示版本信息并离开

1.2 cat

1.2.1 功能说明

把档案串连接后传到基本输出（屏幕或加 > filename 到另一个档案）

1.2.2 语 法

cat [-AbeEnstTuv] [--help] [--version] filename

1.2.3 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` 里。

1.3 cd

1.3.1 功能说明

切换目录。

1.3.2 语法

`cd` [目的目录]

1.3.3 补充说明

`cd` 指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进入目的目录。

1.4 chgrp

1.4.1 功能说明

变更文件或目录的所属群组。

语法：

`chgrp` [-cfhRv] [--help] [--version] [所属群组] [文件或目录...]

或 `chgrp` [-cfhRv] [--help] [--reference=<参考文件或目录>] [--version] [文件或目录...]

1.4.2 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群组来管理。您可以使用 `chgrp` 指令去变更文件与目录的所属群组，设置方式采用群组名称或群组识别码皆可。

1.4.3 参数

`-c` 或 `--changes` 效果类似“`-v`”参数，但仅回报更改的部分。

`-f` 或 `--quiet` 或 `--silent` 不显示错误信息。

`-h` 或 `--no-dereference` 只对符号连接的文件作修改，而不更动其他任何相关文件。

`-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` 显示指令执行过程。

`--help` 在线帮助。

`--reference=<参考文件或目录>` 把指定文件或目录的所属群组全部设成和参考文件或目录的所属群组相同。

`--version` 显示版本信息。

1.5 chmod

1.5.1 功能说明

变更文件或目录的权限。

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...]`

1.5.2 补充说明

在 UNIX 系统家族里，文件或目录权限的控制分别以读取，写入，执行 3 种一般权限来区分，另有 3 种特殊权限可供运用，再搭配拥有者与所属群组管理权限范围。您可以使用 `chmod` 指令去变更文件与目录的权限，设置方式采用文字或数字代号皆可。符号连接的权限无法变更，如果您对符号连接修改权限，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：

u: User，即文件或目录的拥有者。

g: Group，即文件或目录的所属群组。

o: Other，除了文件或目录拥有者或所属群组之外，其他用户皆属于这个范围。

a: All，即全部的用户，包含拥有者，所属群组以及其他用户。

有关权限代号的部分，列表于下：

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”。

1.5.3 参数

-c 或 --changes 效果类似“-v”参数，但仅回报更改的部分。

-f 或 --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 显示指令执行过程。

--help 在线帮助。

--reference=<参考文件或目录> 把指定文件或目录的权限全部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> 开启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。

1.6 chown

1.6.1 功能说明

变更文件或目录的拥有者或所属群组。

1.6.2 语法

chown [-cfhRv][--dereference][--help][--version][拥有者.<所属群组>][文件或目录..]

或 chown [-chfRv][--dereference][--help][--version][.所属群组][文件或目录... ..]

或 chown [-cfhRv][--dereference][--help][--reference=<参考文件或目录>][--version][文件或目录...]

1.6.3 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群组来管理。您可以使用 chown 指令去变更文件与目录的拥有者或所属群组，设置方式采用用户名称或用户识别码皆可，设置群组则用群组名称或群组识别码。

1.6.4 参数

-c 或 --changes 效果类似“-v”参数，但仅回报更改的部分。

-f 或 --quite 或 --silent 不显示错误信息。

-h 或 --no-dereference 之对符号连接的文件作修改，而不更动其他任何相关文件。

-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 --version 显示指令执行过程。

`--dereference` 效果和“-h”参数相同。

`--help` 在线帮助。

`--reference=<参考文件或目录>` 把指定文件或目录的拥有者与所属群组全部设成和参考文件或目录的拥有者与所属群组相同。

`--version` 显示版本信息。

1.7 comm

1.7.1 功能说明

比较两个已排过序的文件。

1.7.2 语法

`comm [-123] [--help] [--version] [第 1 个文件] [第 2 个文件]`

1.7.3 补充说明

这项指令会一列列地比较两个已排序文件的差异，并将其结果显示出来，如果没有指定任何参数，则会把结果分成 3 行显示：第 1 行仅是在第 1 个文件中出现过的列，第 2 行是仅在第 2 个文件中出现过的列，第 3 行则是在第 1 与第 2 个文件里都出现过的列。若给予的文件名称为“-”，则 `comm` 指令会从标准输入设备读取数据。

1.7.4 参数

-1 不显示只在第 1 个文件里出现过的列。

-2 不显示只在第 2 个文件里出现过的列。

-3 不显示只在第 1 和第 2 个文件里出现过的列。

`--help` 在线帮助。

`--version` 显示版本信息。

1.8 cp

1.8.1 功能说明

复制文件或目录。

1.8.2 语法

```
cp [-abdfilpPrRsuvx] [-S <备份字尾字符串>] [-V <备份方式>] [--help] [--spares=<使用时机>] [--version] [源文件或目录] [目标文件或目录] [目的目录]
```

1.8.3 补充说明

cp 指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。

1.8.4 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，覆盖目标文件之前的备份，备份文件会在字尾加上一个备份字符串。

-d 或 --no-dereference 当复制符号连接时，把目标文件或目录也建立为符号连接，并指向与源文件或目录连接的原始文件或目录。

-f 或 --force 强行复制文件或目录，不论目标文件或目录是否已存在。

-i 或 --interactive 覆盖既有文件之前先询问用户。

-l 或 --link 对源文件建立硬连接，而非复制文件。

-p 或 --preserve 保留源文件或目录的属性。

-P 或 --parents 保留源文件或目录的路径。

-r 递归处理，将指定目录下的文件与子目录一并处理。

-R 或 --recursive 递归处理，将指定目录下的所有文件与子目录一并处理。

-s 或 --symbolic-link 对源文件建立符号连接，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”参数备份目标文件后，备份文件的字尾会被加上一个备份字符串，预设的备份字尾字符串是符号“~”。

`-u` 或 `--update` 使用这项参数后只会在源文件的更改时间较目标文件更新时或是 名称相互对应的目标文件并不存在，才复制文件。

`-v` 或 `--verbose` 显示指令执行过程。

`-V<备份方式>`或`--version-control=<备份方式>` 用“-b”参数备份目标文件后，备份文件的字尾会被加上一个备份字符串，这字符串不仅可用“-S”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。

`-x` 或 `--one-file-system` 复制的文件或目录存放的文件系统，必须与 cp 指令执行时所处的文件系统相同，否则不予复制。

`--help` 在线帮助。

`--sparse=<使用时机>` 设置保存稀疏文件的时机。

`--version` 显示版本信息。

1.9 cut

1.9.1 功能说明

显示每行从开头算起 num1 到 num2 的文字。

1.9.2 语法

```
cut -cnum1-num2 filename
```

1.9.3 举例

```
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

1.10 dd

1.10.1 功能说明

读取，转换并输出数据。

1.10.2 语法

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

1.10.3 补充说明

dd 可从标准输入或文件读取数据，依指定的格式来转换数据，再输出到文件，设备或标准输出。

1.10.4 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> 转换时，每次只转换指定的字节数。

conv=<关键字> 指定文件转换的方式。

count=<区块数> 仅读取指定的区块数。

ibs=<字节数> 每次读取的字节数。

if=<文件> 从文件读取。

obs=<字节数> 每次输出的字节数。

of=<文件> 输出到文件。

seek=<区块数> 一开始输出时，跳过指定的区块数。

skip=<区块数> 一开始读取时，跳过指定的区块数。

--help 帮助。

--version 显示版本信息。

1.11 diff

1.11.1 功能说明

比较文件的差异。

1.11.2 语法

```
diff [-abBcdefHilnNpQrstTuvwy][<行数>][-C <行数>][-D <巨集名称>][-I <字符  
或字符串>][-S <文件>][-W <宽度>][-x <文件或目录>][-X <文件>][--help][--left  
-column][--suppress-common-line][文件或目录 1][文件或目录 2]
```

1.11.3 补充说明

diff 以逐行的方式，比较文本文件的异同处。所是指定要比较目录，则 diff 会比较目录中相同文件名的文件，但不会比较其中子目录。

1.11.4 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 预设只会逐行比较文本文件。

-b 或--ignore-space-change 不检查空格字符的不同。

-B 或--ignore-blank-lines 不检查空白行。

-c 显示全部内文，并标出不同之处。

-C<行数>或--context<行数> 与执行"-c-<行数>"指令相同。

-d 或--minimal 使用不同的演算法，以较小的单位来做比较。

-D<巨集名称>或 ifdef<巨集名称> 此参数的输出格式可用于前置处理器巨集。

-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，但按照原来文件的顺序来显示不同处。

-H 或 --speed-large-files 比较大文件时，可加快速度。

-l<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，预设会显示：

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 语言的程序码文件时，显示差异所在的函数名称。

-P 或 --unidirectional-new-file 与 -N 类似，但只有当第二个目录包含了一个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做比较。

-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。

-t 或 --expand-tabs 在输出时，将 tab 字符展开。

-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 忽略全部的空格字符。

-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。

-X<文件>或--exclude-from<文件> 您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。

-y 或--side-by-side 以并列的方式显示文件的异同之处。

--help 显示帮助。

--left-column 在使用-y 参数时，若两个文件某一行内容相同，则仅在左侧的栏位显示该行内容。

--suppress-common-lines 在使用-y 参数时，仅显示不同之处。

1.12 dir

1.12.1 功能说明

列出<文件>的信息（默认为目前的目录）。

1.12.2 语法

dir [选项]... [文件]...

1.12.3 参数

如果不指定 -cftuSUX 或 --sort 任何一个选项，则根据字母大小排序。

-a, --all 不隐藏任何以 . 字符开始的项目

-A, --almost-all 列出除了 . 及 .. 以外的任何项目

--author 印出每个文件著作者

-b, --escape 以八进制溢出序列表示不可打印的字符

--block-size=大小 块以指定<大小>的字节为单位

-B, --ignore-backups 不列出任何以 ~ 字符结束的项目

-c	配合 -lt: 根据 ctime 排序及 显示 ctime (文件
	状态最后更改的时间)
	配合 -l: 显示 ctim
e 但根据名称排序	
	否则: 根据 ctime 排序
-C	按列列举。
-f	不进行排序, -aU 选项生效, -l
st 选项失效	
-F, --classify	加上文件类型的指示符号 (*/=@
其中一个)	
-i, --inode	印出每个文件的 inode 号
-I, --ignore=样式	不印出任何符合 shell 万用字符<样式>的项目
-k	即 --block-size=1K
-l	使用较长格式列出信息
-L, --dereference	当显示符号链接的文件信息时, 显示符
号链接所指示	
	的对象而并非符号链接本身的信息
-m	所有项目以逗号分隔, 并填满整
行行宽	
-n, --numeric-uid-gid	类似 -l, 但列出 UID 及 GID 号
-N, --literal	印出未经处理的项目名称 (例
如不特别处理控制字符)	
-o	类似 -l, 但不列出有关组的信
息	
-p, --file-type	加上文件类型的指示符号 (/=@
其中一个)	

`-q, --hide-control-chars` 以 ? 字符代替无法打印的字符

`--show-control-chars` 直接显示无法打印的字符（这是默认方式，除非调用

的程序名称是 'ls' 而且是在终端机画面输出结果）

`-Q, --quote-name` 将项目名称括上双引号

`--quoting-style=方式` 使用指定的 quoting <方式>显示项目的名称：

literal、locale、shell、shell-always、c、escape

`-r, --reverse` 依相反次序排列

`-R, --recursive` 同时列出所有子目录层

`-s, --size` 以块大小为单位列出所有文件的
大小

`-S` 按文件大小排序。

`-t` 按修改时间排序

`-u` 配合 `-lt`：显示访问时间而且依
访问时间排序

配合 `-l`：显示访问时间但根据名称排序

否则：根据访问时间排序

`-U` 不进行排序；依文件系统原有的
次序列出项目

`-v` 根据版本进行排序

`-w, --width=COLS` 自行指定萤幕宽度而不使用目前的数值

`-x` 逐行列出项目而不是逐栏列出

`-X` 根据扩展名排序

`-l` 每行只列出一个文件

`--help` 显示此帮助信息并离开

`--version` 显示版本信息并离开

1.13 dos2unix

1.13.1 功能说明

将一个 DOS 格式的文件转换为 UNIX 格式的文件。主要差别在换行符。

语法：

```
dos2unix [-hkqV] [-c convmode] [-o file ...] [-n infile outfile ...]
```

参数：

<code>-h --help</code>	帮助
<code>-k --keepdate</code>	保留文件输出日期
<code>-q --quiet</code>	安静模式，屏蔽所有警告。
<code>-V --version</code>	显示版本。
<code>-c --convmode</code> Mac, 默认为 ASCII	转换模式，可以是 ASCII, 7bit, ISO,
<code>-l --newline</code>	增加一行。
<code>-o --oldfile</code>	写到原始文件中
<code>file ...</code>	转换为旧文件的模式
<code>-n --newfile</code>	写入新文件
<code>infile</code>	输入文件名
<code>outfile</code>	输出文件名

1.14 egrep

1.14.1 功能说明

在文件内查找指定的字符串。

egrep 执行效果如 **grep -E**，使用的语法及参数可参照 **grep** 指令，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。

1.15 fgrep

1.15.1 功能说明

查找文件里符合条件的字符串。本指令相当于执行 **grep** 指令加上参数“-F”，详见 **grep** 指令说明。

1.15.2 语法

fgrep [范本样式][文件或目录...]

1.16 file

1.16.1 功能说明

辨识文件类型。

1.16.2 语法

file [-beLvz][-f <名称文件>][-m <魔法数字文件>...][文件或目录...]

1.16.3 补充说明

通过 **file** 指令，我们得以辨识该文件的类型。

1.16.4 参数

- b 列出辨识结果时，不显示文件名称。
- c 详细显示指令执行过程，便于排错或分析程序执行的情形。
- f<名称文件> 指定名称文件，其内容有一个或多个文件名称呢感，让 file 依序辨识这些文件，格式为每列一个文件名称。
- L 直接显示符号连接所指向的文件的类别。
- m<魔法数字文件> 指定魔法数字文件。
- v 显示版本信息。
- z 尝试去解读压缩文件的内容。

1.17 find

1.17.1 功能说明

查找文件或目录。

1.17.2 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][-depyh][-empty][-exec
ec <执行指令>]

[-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 <群组识别码>]

[-group <群组名称>][-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>][-mmi
n <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 <执行指令>][-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>]`

1. 17. 3 补充说明

find 指令用于查找符合条件的文件。任何位于参数之前的字符串都将被视为欲查找的目录。

1. 17. 4 参数

`-amin<分钟>` 查找在指定时间曾被存取过的文件或目录，单位以分钟计算。

`-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。

`-atime<24 小时数>` 查找在指定时间曾被存取过的文件或目录，单位以 24 小时计算。

`-cmin<分钟>` 查找在指定时间之时被更改的文件或目录。

`-cnewer<参考文件或目录>` 查找其更改时间较指定文件或目录的更改时间更接近现在的文件或目录。

`-ctime<24 小时数>` 查找在指定时间之时被更改的文件或目录，单位以 24 小时计算。

`-daystart` 从本日开始计算时间。

`-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 Byte 的文件，或目录下没有任何子目录或文件的空目录。

`-exec<执行指令>` 假设 find 指令的回传值为 True，就执行该指令。

- false 将 find 指令的回传值皆设为 False。
- fls<列表文件> 此参数的效果和指定“-ls”参数类似，但会把结果保存为指定的列表文件。
- follow 排除符号连接。
- fprint<列表文件> 此参数的效果和指定“-print”参数类似，但会把结果保存成指定的列表文件。
- fprint0<列表文件> 此参数的效果和指定“-print0”参数类似，但会把结果保存成指定的列表文件。
- fprintf<列表文件><输出格式> 此参数的效果和指定“-printf”参数类似，但会把结果保存成指定的列表文件。
- fstype<文件系统类型> 只寻找该文件系统类型下的文件或目录。
- gid<群组识别码> 查找符合指定之群组识别码的文件或目录。
- group<群组名称> 查找符合指定之群组名称的文件或目录。
- help 或 --help 在线帮助。
- ilname<范本样式> 此参数的效果和指定“-lname”参数类似，但忽略字符大小写的差别。
- iname<范本样式> 此参数的效果和指定“-name”参数类似，但忽略字符大小写的差别。
- inum<inode 编号> 查找符合指定的 inode 编号的文件或目录。
- 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“-regexe”参数类似，但忽略字符大小写的差别。
- links<连接数目> 查找符合指定的硬连接数目的文件或目录。
- lname<范本样式> 指定字符串作为寻找符号连接的范本样式。
- ls 假设 find 指令的回传值为 True，就将文件或目录名称列出到标准输出。
- maxdepth<目录层级> 设置最大目录层级。

- mindepth<目录层级> 设置最小目录层级。
- mmin<分钟> 查找在指定时间曾被更改过的文件或目录，单位以分钟计算。
- mount 此参数的效果和指定“-xdev”相同。
- mtime<24 小时数> 查找在指定时间曾被更改过的文件或目录，单位以 24 小时计算。
- name<范本样式> 指定字符串作为寻找文件或目录的范本样式。
- newer<参考文件或目录> 查找其更改时间较指定文件或目录的更改时间更接近现在的文件或目录。
- nogroup 找出不属于本地主机群组识别码的文件或目录。
- noleaf 不去考虑目录至少需拥有两个硬连接存在。
- nouser 找出不属于本地主机用户识别码的文件或目录。
- ok<执行指令> 此参数的效果和指定“-exec”参数类似，但在执行指令之前会先询问用户，若回答“y”或“Y”，则放弃执行指令。
- path<范本样式> 指定字符串作为寻找目录的范本样式。
- perm<权限数值> 查找符合指定的权限数值的文件或目录。
- print 假设 find 指令的回传值为 True，就将文件或目录名称列出到标准输出。格式为每列一个名称，每个名称之前皆有“./”字符串。
- print0 假设 find 指令的回传值为 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 指令的回传值为 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 不寻找字符串作为寻找文件或目录的范本样式。
- regex<范本样式> 指定字符串作为寻找文件或目录的范本样式。
- size<文件大小> 查找符合指定的文件大小的文件。
- true 将 find 指令的回传值皆设为 True。
- typ<文件类型> 只寻找符合指定的文件类型的文件。
- uid<用户识别码> 查找符合指定的用户识别码的文件或目录。

`-used<日数>` 查找文件或目录被更改之后在指定时间曾被存取过的文件或目录，单位以日计算。

`-user<拥有者名称>` 查找符合指定的拥有者名称的文件或目录。

`-version` 或 `--version` 显示版本信息。

`-xdev` 将范围局限在先行的文件系统中。

`-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连接检查。

1.18 grep

1.18.1 功能说明

查找文件里符合条件的字符串。

1.18.2 语法

`grep [-abcEFgHiIlLnqrsVwxy] [-A<显示列数>] [-B<显示列数>] [-C<显示列数>] [-d<进行动作>] [-e<范本样式>] [-f<范本文件>] [--help] [范本样式] [文件或目录...]`

1.18.3 补充说明

`grep` 指令用于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，预设 `grep` 指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` 指令会从标准输入设备读取数据。

1.18.4 参数

`-a` 或 `--text` 不要忽略二进制的文件。

`-A<显示列数>` 或 `--after-context=<显示列数>` 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。

`-b` 或 `--byte-offset` 在显示符合范本样式的那一行之前，标示出该行第一个字符的位编号。

`-B<显示列数>`或`--before-context=<显示列数>` 除了显示符合范本样式的那一列之外，并显示该列之前的内容。

`-c` 或 `--count` 计算符合范本样式的列数。

`-C<显示列数>`或`--context=<显示列数>`或`-<显示列数>` 除了显示符合范本样式的那一列之外，并显示该列之前的内容。

`-d<进行动作>`或`--directories=<进行动作>` 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` 指令将回报信息并停止动作。

`-e<范本样式>`或`--regexp=<范本样式>` 指定字符串做为查找文件内容的范本样式。

`-E` 或 `--extended-regexp` 将范本样式为延伸的普通表示法来使用。

`-f<范本文件>`或`--file=<范本文件>` 指定范本文件，其内容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，格式为每列一个范本样式。

`-F` 或 `--fixed-regexp` 将范本样式视为固定字符串的列表。

`-G` 或 `--basic-regexp` 将范本样式视为普通的表示法来使用。

`-h` 或 `--no-filename` 在显示符合范本样式的那一列之前，不标示该列所属的文件名称。

`-H` 或 `--with-filename` 在显示符合范本样式的那一列之前，表示该列所属的文件名称。

`-i` 或 `--ignore-case` 忽略字符大小写的差别。

`-l` 或 `--file-with-matches` 列出文件内容符合指定的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的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列之前，标示出该列的列数编号。

`-q` 或 `--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` 反转查找。

`-V` 或 `--version` 显示版本信息。

`-w` 或 `--word-regexp` 只显示全字符合的列。

`-x` 或 `--line-regexp` 只显示全列符合的列。

`-y` 此参数的效果和指定 “`-i`” 参数相同。

`--help` 在线帮助。

1.19 head

1.19.1 功能说明

先是文件的前几行（默认 10 行）。

1.19.2 语法

`head [options] [files]`

1.19.3 参数

`-c, --bytes=[-]N` 每个文件显示前 N 字节；

`-n, --lines=[-]N` 每个文件显示前 N 行；

`-q, --quiet, --silent` 不打印文件头

`-v, --verbose` 打印文件头

`--help` 显示此帮助信息并离开

`--version` 显示版本信息并离开

1.19.4 例子

显示文件 `phone_list` 前 20 行

```
head -20 phone_list
```

显示有 202 区号的前 10 个号码

```
grep '(202)' phone_list | head
```

1.20 Less

1.20.1 功能说明

less 命令的功能几乎和 more 命令一样，也是用来按页显示文件，不同之处在于 less 命令在显示文件时允许用户既可以向前又可以向后翻阅文件。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。

1.20.2 语法

```
less [options] [filename]
```

1.21 ln

1.21.1 功能说明

连接文件或目录。

1.21.2 语法

```
ln [-bdfinsv] [-S <字尾备份字符串>] [-V <备份方式>] [--help] [--version] [源文件  
或目录] [目标文件或目录]
```

```
或 ln [-bdfinsv] [-S <字尾备份字符串>] [-V <备份方式>] [--help] [--version] [源文  
件或目录...][目的目录]
```

1.21.3 补充说明

ln 指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，且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。

1.21.4 参数

-b 或 --backup 删除，覆盖目标文件之前的备份。

-d 或 -F 或 --directory 建立目录的硬连接。

-f 或 --force 强行建立文件或目录的连接，不论文件或目录是否存在。

-i 或 --interactive 覆盖既有文件之前先询问用户。

-n 或 --no-dereference 把符号连接的目的目录视为一般文件。

-s 或 --symbolic 对源文件建立符号连接，而非硬连接。

-S<字尾备份字符串>或 --suffix=<字尾备份字符串> 用“-b”参数备份目标文件后，备份文件的字尾会被加上一个备份字符串，预设的字尾备份字符串是符号“~”，您可通过“-S”参数来改变它。

-v 或 --verbose 显示指令执行过程。

-V<备份方式>或 --version-control=<备份方式> 用“-b”参数备份目标文件后，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“-S”参数变更，当使用“-V”参数<备份方式>指定不同备份方式时，也会产生不同字尾的备份字符串。

--help 在线帮助。

--version 显示版本信息。

1.22 locate

1.22.1 功能说明

查找文件。

1.22.2 语法

locate [-d <数据库文件>][--help][--version][范本样式...]

1.22.3 补充说明

locate 指令用于查找符合条件的文件，它会去保存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。

1.22.4 参数

-d<数据库文件>或--database=<数据库文件> 设置 locate 指令使用的数据库。locate 指令预设的数据库位于 /var/lib/slocate 目录里，文件名为 slocate.db，您可使用这个参数另行指定。

--help 在线帮助。

--version 显示版本信息。

1.23 ls

1.23.1 功能说明

列出目录内容。

1.23.2 语法

```
ls [-laAbBcCdDfFgGhHiklLmnNopqQrRsStuUvxX] [-I <范本样式>] [-T <跳格字数>] [-w <每列字符数>] [--block-size=<区块大小>] [--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=<时间戳记>] [--version] [文件或目录...]
```

1.23.3 补充说明

执行 ls 指令可列出目录的内容，包括文件和子目录的名称。

1.23.4 参数

-l 每列仅显示一个文件或目录名称。

-a 或 --all 下所有文件和目录。

-A 或 --almost-all 显示所有文件和目录，但不显示现行目录和上层目录。

-b 或 --escape 显示脱离字符。

-B 或 --ignore-backups 忽略备份文件和目录。

-c 以更改时间排序，显示文件和目录。

-C 以又上至下，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`”参数相同，并关闭“`lst`”参数的效果。

`-F` 或 `--classify` 在执行文件，目录，Socket，符号连接，管道名称后面，各自加上“`*`”，“`/`”，“`=`”，“`@`”，“`|`”号。

`-g` 次参数将忽略不予处理。

`-G` 或 `--no-group` 不显示群组名称。

`-h` 或 `--human-readable` 用“`K`”，“`M`”，“`G`”来显示文件和目录的大小。

`-H` 或 `--si` 此参数的效果和指定“`-h`”参数类似，但计算单位是 1000Bytes 而非 1024Bytes。

`-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 `--kilobytes` 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。

`-L` 或 `--dereference` 如遇到性质为符号连接的文件或目录，直接列出该连接所指向的原始文件或目录。

`-m` 用“`,`”号区隔每个文件和目录的名称。

`-n` 或 `--numeric-uid-gid` 以用户识别码和群组识别码替代其名称。

`-N` 或 `--literal` 直接列出文件和目录名称，包括控制字符。

`-o` 此参数的效果和指定“`-l`”参数类似，但不列出群组名称或识别码。

`-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。

`-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，以区块为单位。

`-S` 用文件和目录的大小排序。

`-t` 用文件和目录的更改时间排序。

`-T<跳格字符>` 或 `--tabsize=<跳格字数>` 设置跳格字符所对应的空白字符数。

`-u` 以最后存取时间排序，显示文件和目录。

`-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<每列字符数>` 或 `--width=<每列字符数>` 设置每列的最大字符数。

`-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=<区块大小>` 指定存放文件的区块大小。

`--color=<列表格式>` 培植文件和目录的列表格式。

`--full-time` 列出完整的日期与时间。

`--help` 在线帮助。

`--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。

`--quoting-syte=<引号样式>` 把文件和目录名称以指定的引号样式标示起来。

`--show-control-chars` 在文件和目录列表时，使用控制字符。

`--sort=<排序方式>` 配置文件和目录列表的排序方式。

`--time=<时间戳记>` 用指定的时间戳记取代更改时间。

`--version` 显示版本信息。

1.24 `mkdir`

1.24.1 功能说明

建立目录

1.24.2 语法

```
mkdir [-p][--help][--version][-m <目录属性>][目录名称]
```

1.24.3 补充说明

mkdir 可建立目录并同时设置目录的权限。

1.24.4 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。

-p 或--parents 若所要建立目录的上层目录目前尚未建立，则会一并建立上层目录。

--help 显示帮助。

--verbose 执行时显示详细的信息。

--version 显示版本信息。

1.25 more

1.25.1 功能说明

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。

1.25.2 语法

```
more [-dlfpsu] [-num] [+/pattern] [+linenum] [fileNames..]
```

1.25.3 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

1.25.4 例子

`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。

1.26 mv

1.26.1 功能说明

移动或更名现有的文件或目录。

1.26.2 语法

`mv [-bfuiuv][--help][--version][-S <附加字尾>][-V <方法>][源文件或目录][目标文件或目录]`

1.26.3 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。

1.26.4 参数

-b 或--backup 若需覆盖文件，则覆盖前先行备份。

-f 或--force 若目标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 参数一并使用，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，若目标文件已存在，且其文件日期比源文件新，则不覆盖目标文件。

-v 或--verbose 执行时显示详细的信息。

-V=<方法>或

--version-control=<方法> 与-b 参数一并使用，可指定备份的方法。

--help 显示帮助。

--version 显示版本信息。

1.27 od

1.27.1 功能说明

输出文件内容。

1.27.2 语法

od [-abcdfhilovx][-A <字码基数>][-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...]

1.27.3 补充说明

od 指令会读取所给予的文件的内容，并将其内容以八进制字码呈现出来。

1.27.4 参数

- a 此参数的效果和同时指定“-ta”参数相同。
- A<字码基数> 选择要以何种基数计算字码。
- b 此参数的效果和同时指定“-toC”参数相同。
- c 此参数的效果和同时指定“-tC”参数相同。
- d 此参数的效果和同时指定“-tu2”参数相同。
- f 此参数的效果和同时指定“-tfF”参数相同。
- h 此参数的效果和同时指定“-tx2”参数相同。
- i 此参数的效果和同时指定“-td2”参数相同。
- j<字符数目>或--skip-bytes=<字符数目> 略过设置的字符数目。
- l 此参数的效果和同时指定“-td4”参数相同。
- N<字符数目>或--read-bytes=<字符数目> 到设置的字符数目为止。
- o 此参数的效果和同时指定“-to2”参数相同。
- s<字符串字符数>或--strings=<字符串字符数> 只显示符合指定的字符数目的字符串。
- t<输出格式>或--format=<输出格式> 设置输出格式。
- v 或--output-duplicates 输出时不省略重复的数据。
- w<每列字符数>或--width=<每列字符数> 设置每列的最大字符数。
- x 此参数的效果和同时指定“-h”参数相同。
- help 在线帮助。
- version 显示版本信息。

1.28 pwd

1.28.1 功能说明

显示工作目录。

1.28.2 语法

```
pwd [--help][--version]
```

1.28.3 补充说明

执行 pwd 指令可立刻得知您目前所在的工作目录的绝对路径名称。

1.28.4 参数

--help 在线帮助。

--version 显示版本信息。

1.29 rename

1.29.1 功能说明

文件批量改名。

1.29.2 语法

```
rename from to file
```

该命令的各选项含义如下：

from 源字符。

to 目标字符。

file 要改名的文件

1.29.3 例子

```
rename .rm .rmvb *
```

// 把所有文件的后缀由 rm 改为 rmvb

```
rename 'tr/A-Z/a-z/' *
```

// 把所有文件名中的大写改为小写

1.30 rm

1.30.1 功能说明

删除文件或目录。

1.30.2 语法

```
rm [-dfirv] [--help] [--version] [文件或目录...]
```

1.30.3 补充说明

执行 rm 指令可删除文件或目录，如欲删除目录必须加上参数“-r”，否则预设仅会删除文件。

1.30.4 参数

-d 或 --directory 直接把欲删除的目录的硬连接数据删成 0，删除该目录。

-f 或 --force 强制删除文件或目录。

-i 或 --interactive 删除既有文件或目录之前先询问用户。

-r 或 -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 显示指令执行过程。

--help 在线帮助。

`--version` 显示版本信息。

1.31 rmdir

1.31.1 功能说明

删除目录。

1.31.2 语法

`rmdir [-p][--help][--ignore-fail-on-non-empty][--verbose][--version][目录...]`

1.31.3 补充说明

当有空目录要删除时，可使用 `rmdir` 指令。

1.31.4 参数

`-p` 或 `--parents` 删除指定目录后，若该目录的上层目录已变成空目录，则将其一并删除。

`--help` 在线帮助。

`--ignore-fail-on-non-empty` 忽略非空目录的错误信息。

`--verbose` 显示指令执行过程。

`--version` 显示版本信息。

1.32 sed

1.32.1 功能说明

利用 `script` 来处理文本文件。

1.32.2 语法

`sed [-hnV][-e<script>][-f<script 文件>][文本文件]`

1.32.3 补充说明

sed 可依照 script 的指令，来处理、编辑文本文件。

1.32.4 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 或--quiet 或--silent 仅显示 script 处理后的结果。

-V 或--version 显示版本信息。

1.33 sort

1.33.1 功能说明

将文本文件内容加以排序。

1.33.2 语法

```
sort [-bcdfimMnr] [-o<输出文件>] [-t<分隔字符>] [+<起始栏位>-<结束栏位>] [--help]
[--version] [文件]
```

1.33.3 补充说明

sort 可针对文本文件的内容，以行为单位来排序。

1.33.4 参数

-b 忽略每行前面开始出的空格字符。

-c 检查文件是否已经按照顺序排序。

- d 排序时，处理英文字母、数字及空格字符外，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 之间的 ASCII 字符外，忽略其他的字符。
- m 将几个排序好的文件进行合并。
- M 将前面 3 个字母依照月份的缩写进行排序。
- 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。
- r 以相反的顺序来排序。
- t<分隔字符> 指定排序时所用的栏位分隔字符。
- +<起始栏位>-<结束栏位> 以指定的栏位来排序，范围由起始栏位到结束栏位的前一栏位。
- help 显示帮助。
- version 显示版本信息。

1.34 tail

1.34.1 功能说明

查看文件的后 N 行, 有实时监控的功能。

1.34.2 语法

```
tail [ -f ] [ -c Number | -n Number | -m Number | -b Number | -k Number ] [ File ]
```

或者

```
tail [ -r ] [ -n Number ] [ File ]
```

tail 命令从指定点开始将 File 参数指定的文件写到标准输出。如果没有指定文件，则会使用标准输入。Number 变量指定将多少单元写入标准输出。Number 变量的值可以是正的或负的整数。如果值的前面有 + (加号)，从文件开头指定的单元数开始将文

件写到标准输出。如果值的前面有 -（减号），则从文件末尾指定的单元数开始将文件写到

标准输出。如果值前面没有 +（加号）或 -（减号），那么从文件末尾指定的单元号开始读取文件。

1.34.3 参数

-b Number 从 Number 变量表示的 512 字节块位置开始读取指定文件。

-c Number 从 Number 变量表示的字节位置开始读取指定文件。

-f 如果输入文件是常规文件或如果 File 参数指定 FIFO（先进先出），那么 tail 命令不会在复制了输入文件的最后的指定单元后终止，而是继续从输入文件读取和复制额外的单元（当这些单元可用时）。如果没有指定 File 参数，并且标准输入是管道，则会忽略 -f 标志。**tail -f 命令可用于监视另一个进程正在写入的文件的生长。**

-k Number 从 Number 变量表示的 1KB 块位置开始读取指定文件。

-m Number 从 Number 变量表示的多字节字符位置开始读取指定文件。使用该标志提供在单字节和双字节字符代码集环境中的一致结果。

-n Number 从首行或末行位置来读取指定文件，位置由 Number 变量的符号（+ 或 - 或无）表示，并通过行号 Number 进行位移。

-r 从文件末尾以逆序方式显示输出。-r 标志的缺省值是以逆序方式显示整个文件。如果文件大于 20,480 字节，那么 -r 标志只显示最后的 20,480 字节。-r 标志只有与 -n 标志一起时才有效。否则，就会将其忽略

1.34.4 例子

```
tail -f /var/log/messages
```

参数-f 使 tail 不停地去读最新的内容，这样有实时监视的效果，用 Ctrl+c 来终止！

1.35 touch

1.35.1 功能说明

改变文件或目录时间。

1.35.2 语法

`touch [-acfm] [-d <日期时间>] [-r <参考文件或目录>] [-t <日期时间>] [--help] [--version] [文件或目录...]` 或 `touch [-acfm] [--help] [--version] [日期时间] [文件或目录...]`

1.35.3 补充说明

使用 `touch` 指令可更改文件或目录的日期时间，包括存取时间和更改时间。

1.35.4 参数

`-a` 或 `--time=atime` 或 `--time=access` 或 `--time=use` 只更改存取时间。

`-c` 或 `--no-create` 不建立任何文件。

`-d<时间日期>` 使用指定的日期时间，而非现在的时间。

`-f` 此参数将忽略不予处理，仅负责解决 BSD 版本 `touch` 指令的兼容性问题。

`-m` 或 `--time=mtime` 或 `--time=modify` 只更改变动时间。

`-r<参考文件或目录>` 把指定文件或目录的日期时间，统统设成和参考文件或目录的日期时间相同。

`-t<日期时间>` 使用指定的日期时间，而非现在的时间。

`--help` 在线帮助。

`--version` 显示版本信息。

1.36 uniq

1.36.1 功能说明

检查及删除文本文件中重复出现的行列。

1.36.2 语法

`uniq [-cdu] [-f<栏位>] [-s<字符位置>] [-w<字符位置>] [--help] [--version] [输入文件] [输出文件]`

1.36.3 补充说明

uniq 可检查文本文件中重复出现的行列。

1.36.4 参数

- c 或--count 在每列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<栏位>或--skip-fields=<栏位> 忽略比较指定的栏位。
- s<字符位置>或--skip-chars=<字符位置> 忽略比较指定的字符。
- u 或--unique 仅显示出一次的行列。
- w<字符位置>或--check-chars=<字符位置> 指定要比较的字符。
- 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。

1.37 unix2dos

1.37.1 功能说明

将 UNIX 格式的文件转化为 DOS 格式。

1.37.2 语法

unix2dos [-hkqV] [-o file ...] [-c convmode] [-n infile outfile ...]

1.37.3 参数

- h --help 帮助。

<code>-k --keepdate</code>	保留文件日期。
<code>-q --quiet</code>	安静模式，屏蔽所有警告。
<code>-V --version</code>	显示版本。
<code>-c --convmode</code> Mac, 默认为 ASCII	转换模式, 可以是 ASCII, 7bit, ISO,
<code>-l --newline</code>	增加一行。
<code>-o --oldfile</code>	写到原始文件中
<code>file ...</code>	转换为旧文件的模式
<code>-n --newfile</code>	写入新文件
<code>infile</code>	输入文件名
<code>outfile</code>	输出文件名

1.38 vdir

`vdir [options] [files]`

列举目录内容, 功能同 `ls -lb`

1.39 vi

1.39.1 功能说明

文本编辑。

1.39.2 语法

`vim [参数] [文件 ...]` 编辑指定的文件

或:`vim [参数] -` 从标准输入(stdin)读取文本

或:`vim [参数] -t tag` 编辑时使用指定的 tag

或:`vim [参数] -q [errorfile]` 编辑时加载第一个错误

1.39.3 参数

--	只有在这之后的文件
-v	Vi 模式 (同 "vi")
-e	Ex 模式 (同 "ex")
-s (只能与 "ex" 一起使用)	安静 (batch) 模式
-d mdiff", 可迅速比较两文件不同处)	Diff 模式 (同 "vi mdiff", 可迅速比较两文件不同处)
-y m", modeless)	简易模式 (同 "evi m", modeless)
-R w")	只读模式 (同 "vie w")
-Z m")	限制模式 (同 "rvi m")
-m	不可修改 (写入文件)
-M	文本不可修改
-b	二进制模式
-l	Lisp 模式
-C 兼容模式	'compatible' 传统 Vi 兼容模式
-N 与传统 Vi 兼容, 可使用 Vim 加强能力	'nocompatible' 不完全 与传统 Vi 兼容, 可使用 Vim 加强能力
-V[N]	Verbose 等级
-D	调试模式
-n 用内存	不使用交换文件, 只使 用内存
-r	列出交换文件后退出

-r (加文件名)	恢复上次崩溃的资料 (Recover crashed session)
-L	与 -r 一样
-A	按阿拉伯模式启动
-H	按犹太模式启动
-F	按波斯模式启动
-T <terminal>	设定终端为 <terminal>
-u <vimrc>	使用 <vimrc> 替换任何 .vimrc
--noplugin	不加载任何 plugin
-o[N]	打开 N 个窗口 (预设是每个文件一个)
-O[N]	同 -o 窗体垂直分割。
+	启动后跳到文件结尾
++<lnum>	启动后跳到第 <lnum> 行
--cmd <command>	加载任何 vimrc 前执行 <command>
-c <command>	加载第一个文件后执行 <command>
-S <session> n>	加载第一个文件后载入 Session 文件<session>
-s <scriptin>	从 <scriptin> 读入一般模式命令
-w <scriptout>	对文件 <scriptout> 附加 (append) 所有输入的命令
-W <scriptout>	对文件 <scriptout> 写入所有输入的命令
-x	编辑编码过的文件
-i <viminfo>	使用 <viminfo> 而非 .viminfo
-h or --help	帮助。
--version	版本。

1.39.4 命令

1. 39. 4. 1 进入输入模式

新增 (append)

a : 从光标所在位置後面开始新增资料，光标後的资料随新增资料向後移动。

A: 从光标所在列最後面的地方开始新增资料。

插入 (insert)

i: 从光标所在位置前面开始插入资料，光标後的资料随新增资料向後移动。

I : 从光标所在列的第一个非空白字元前面开始插入资料。

开始 (open)

o : 在光标所在列下新增一行并进入输入模式。

O: 在光标所在列上方新增一行并进入输入模式。

1. 39. 4. 2 退出 vi

在指令模式下键入:q, :q!, :wq 或:x(注意:号)，就会退出 vi。其中:wq 和:x 是存盘退出，而:q 是直接退出，如果文件已有新的变化，vi 会提示你保存文件而:q 命令也会失效，这时你可以用:w 命令保存文件后再用:q 退出，或用:wq 或:x 命令退出，如果你不想保存改变后的文件，你就需要用:q!命令，这个命令将不保存文件而直接退出 vi。

1. 39. 4. 3 删除与修改文件的命令:

x: 删除光标所在字符。

dd : 删除光标所在的行。

r : 修改光标所在字元，r 後接著要修正的字符。

R: 进入取替换状态，新增文字会覆盖原先文字，直到按 [ESC] 回到指令模式下为止。

s: 删除光标所在字元，并进入输入模式。

S: 删除光标所在的行，并进入输入模式。

1. 39. 4. 4 屏幕翻滚类命令

Ctrl+u: 向文件首翻半屏

Ctrl+d: 向文件尾翻半屏

Ctrl+f: 向文件尾翻一屏

Ctrl+b: 向文件首翻一屏

nz: 将第 n 行滚至屏幕顶部, 不指定 n 时将当前行滚至屏幕顶部。

1. 39. 4. 5 删除命令

ndw 或 ndW: 删除光标处开始及其后的 n-1 个字

do: 删至行首

d\$: 删至行尾

ndd: 删除当前行及其后 n-1 行

x 或 X: 删除一个字符, x 删除光标后的, 而 X 删除光标前的

Ctrl+u: 删除输入方式下所输入的文本

1. 39. 4. 6 搜索及替换命令

/pattern: 从光标开始处向文件尾搜索 pattern

?pattern: 从光标开始处向文件首搜索 pattern

n: 在同一方向重复上一次搜索命令

N: 在反方向上重复上一次搜索命令

:s/p1/p2/g: 将当前行中所有 p1 均用 p2 替代

:n1,n2s/p1/p2/g: 将第 n1 至 n2 行中所有 p1 均用 p2 替代

:g/p1/s//p2/g: 将文件中所有 p1 均用 p2 替换

1. 39. 4. 7 复制, 黏贴

(1) 选定文本块, 使用 v 进入可视模式; 移动光标键选定内容

(2) 复制选定块到缓冲区, 用 y; 复制整行, 用 yy

(3) 剪切选定块到缓冲区, 用 d; 剪切整行用 dd

(4) 粘贴缓冲区中的内容，用 p

1.39.4.8 其他

在同一编辑窗打开第二个文件，用:sp [filename]

在多个编辑文件之间切换，用 Ctrl+w

1.40 wc

1.40.1 功能说明

计算字数。

1.40.2 语法

wc [-clw][--help][--version][文件...]

1.40.3 补充说明

利用 wc 指令我们可以计算文件的 Byte 数、字数、或是列数，若不指定文件名称、或是所给予的文件名为“-”，则 wc 指令会从标准输入设备读取数据。

1.40.4 参数

-c 或--bytes 或--chars 只显示 Bytes 数。

-l 或--lines 只显示列数。

-w 或--words 只显示字数。

--help 在线帮助。

--version 显示版本信息。

2 磁盘管理

2.1 dd

2.1.1 功能说明

读取，转换并输出数据。

2.1.2 语法

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

2.1.3 补充说明

dd 可从标准输入或文件读取数据，依指定的格式来转换数据，再输出到文件，设备或标准输出。

2.1.4 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> 转换时，每次只转换指定的字节数。

conv=<关键字> 指定文件转换的方式。

conv = ascii 把 EBCDIC 码转换为 ASCII 码。

conv = ebcdic 把 ASCII 码转换为 EBCDIC 码。

conv = ibm 把 ASCII 码转换为 alternate EBCDIC 码。

conv = block 把变动位转换成固定字符。

conv = ublock 把固定位转换成变动位。

conv = ucase 把字母由小写转换为大写。

conv = lcase 把字母由大写转换为小写。

conv = notrunc 不截短输出文件。

conv = swab 交换每一对输入字节。

conv = noerror 出错时不停止处理。

conv = sync 把每个输入记录的大小都调到 ibs 的大小（用 NUL 填充）。

count=<区块数> 仅读取指定的区块数。

ibs=<字节数> 每次读取的字节数。

if=<文件> 从文件读取。

obs=<字节数> 每次输出的字节数。

of=<文件> 输出到文件。

seek=<区块数> 一开始输出时，跳过指定的区块数。

skip=<区块数> 一开始读取时，跳过指定的区块数。

--help 帮助。

--version 显示版本信息。

2.1.5 例子

1 整盘数据备份与恢复

备份：

```
dd if=/dev/hdx of=/dev/hdy
```

将本地的/dev/hdx 整盘备份到/dev/hdy

```
dd if=/dev/hdx of=/path/to/image
```

将/dev/hdx 全盘数据备份到指定路径的 image 文件

```
dd if=/dev/hdx | gzip >/path/to/image.gz
```

备份/dev/hdx 全盘数据，并利用 gzip 工具进行压缩，保存到指定路径

恢复：


```
dd if=/path/to/image of=/dev/hdx
```

将备份文件恢复到指定盘

```
gzip -dc /path/to/image.gz | dd of=/dev/hdx
```

将压缩的备份文件恢复到指定盘

2. 利用 netcat 远程备份

```
dd if=/dev/hda bs=16065b | netcat < targethost-IP > 1234
```

在源主机上执行此命令备份/dev/hda

```
netcat -l -p 1234 | dd of=/dev/hdc bs=16065b
```

在目的主机上执行此命令来接收数据并写入/dev/hdc

```
netcat -l -p 1234 | bzip2 > partition.img
```

```
netcat -l -p 1234 | gzip > partition.img
```

以上两条指令是目的主机指令的变化分别采用 bzip2 gzip 对数据进行压缩, 并将备份文件保存在当前目录。

3. 备份 MBR

备份:

```
dd if=/dev/hdx of=/path/to/image count=1 bs=512
```

备份磁盘开始的 512Byte 大小的 MBR 信息到指定文件

恢复:

```
dd if=/path/to/image of=/dev/hdx
```

将备份的 MBR 信息写到磁盘开始部分

4. 备份软盘

```
dd if=/dev/fd0 of=disk.img count=1 bs=1440k
```

将软驱数据备份到当前目录的 disk.img 文件

5. 拷贝内存资料到硬盘

```
dd if=/dev/mem of=/root/mem.bin bs=1024
```

将内存里的数据拷贝到 root 目录下的 mem.bin 文件

6. 从光盘拷贝 iso 镜像

```
dd if=/dev/cdrom of=/root/cd.iso
```

拷贝光盘数据到 root 文件夹下，并保存为 cd.iso 文件

7. 增加 Swap 分区文件大小

```
dd if=/dev/zero of=/swapfile bs=1024 count=262144
```

创建一个足够大的文件（此处为 256M）

```
mkswap /swapfile
```

把这个文件变成 swap 文件

```
swapon /swapfile
```

启用这个 swap 文件

```
/swapfile swap swap defaults 0 0
```

在每次开机的时候自动加载 swap 文件，需要在 `/etc/fstab` 文件中增加一行

8. 销毁磁盘数据

```
dd if=/dev/urandom of=/dev/hda1
```

利用随机的数据填充硬盘，在某些必要的场合可以用来销毁数据。执行此操作以后，`/dev/hda1` 将无法挂载，创建和拷贝操作无法执行。

9. 得到最恰当的 block size

```
dd if=/dev/zero bs=1024 count=1000000 of=/root/1Gb.file
```

```
dd if=/dev/zero bs=2048 count=500000 of=/root/1Gb.file
```

```
dd if=/dev/zero bs=4096 count=250000 of=/root/1Gb.file
```

```
dd if=/dev/zero bs=8192 count=125000 of=/root/1Gb.file
```

通过比较 `dd` 指令输出中所显示的命令执行时间，即可确定系统最佳的 `block size` 大小

10. 测试硬盘读写速度

```
dd if=/root/1Gb.file bs=64k | dd of=/dev/null
```

```
dd if=/dev/zero of=/root/1Gb.file bs=1024 count=1000000
```

通过上两个命令输出的执行时间，可以计算出测试硬盘的读 / 写速度

11. 修复硬盘

```
dd if=/dev/sda of=/dev/sda
```

当硬盘较长时间（比如 1，2 年）放置不使用后，磁盘上会产生 `magnetic flux point`。当磁头读到这些区域时会遇到困难，并可能导致 I/O 错误。当这种情况影响到硬盘的第一个扇

区时，可能导致硬盘报废。上边的命令有可能使这些数据起死回生。且这个过程是安全，高效的。

2.2 df

2.2.1 功能说明

显示磁盘的相关信息。

2.2.2 语法

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

2.2.3 补充说明

df 可显示磁盘的文件系统与使用情形。

2.2.4 参数

-a 或 --all 包含全部的文件系统。

--block-size=<区块大小> 以指定的区块大小来显示区块数目。

-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 1000 Bytes 为换算单位而非 1024 Bytes。

-i 或 --inodes 显示 inode 的信息。

-k 或 --kilobytes 指定区块大小为 1024 字节。

-l 或 --local 仅显示本地端的文件系统。

-m 或 --megabytes 指定区块大小为 1048576 字节。

--no-sync 在取得磁盘使用信息前，不要执行 sync 指令，此为预设值。

-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` 指令。

`-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。

`-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。

2.3 du

2.3.1 功能说明

显示目录或文件的大小。

2.3.2 语法

`du [-abcDhHklmsSx] [-L <符号连接>] [-X <文件>] [--block-size] [--exclude=<目录或文件>] [--max-depth=<目录层数>] [--help] [--version] [目录或文件]`

2.3.3 补充说明

`du` 会显示指定的目录或文件所占用的磁盘空间。

2.3.4 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 `byte` 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` 显示指定符号连接的源文件大小。

`-h` 或 `--human-readable` 以 K, M, G 为单位, 提高信息的可读性。

`-H` 或 `--si` 与 `-h` 参数相同, 但是 K, M, G 是以 1000 为换算单位。

`-k` 或 `--kilobytes` 以 1024 bytes 为单位。

`-l` 或 `--count-links` 重复计算硬件连接的文件。

`-L<符号连接>` 或 `--dereference<符号连接>` 显示选项中所指定符号连接的源文件大小。

`-m` 或 `--megabytes` 以 1MB 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, 并不含其子目录的大小。

`-x` 或 `--one-file-system` 以一开始处理时的文件系统为准, 若遇上其它不同的文件系统目录则略过。

`-X<文件>` 或 `--exclude-from=<文件>` 在<文件>指定目录或文件。

`--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。

2.4 fdisk

2.4.1 功能说明

磁盘分区。

2.4.2 语法

`fdisk [-b <分区大小>][-uv][外围设备代号]` 或 `fdisk [-l][-b <分区大小>][-uv][外围设备代号...]` 或 `fdisk [-s <分区编号>]`

2.4.3 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。

2.4.4 参数

- b<分区大小> 指定每个分区的大小。
- l 列出指定的外围设备的分区表状况。
- s<分区编号> 将指定的分区大小输出到标准输出上，单位为区块。
- u 搭配“-l”参数列表，会用分区数目取代柱面数目，来表示每个分区的起始地址。
- v 显示版本信息。

2.5 format

2.5.1 功能说明

格式化磁盘。最好是对未格式过化的磁盘使用。

2.5.2 语法

```
format [-n] device
```

-n 软盘格式化后不作检验。

2.5.3 补充说明

DOS 的 FORMAT A: 实际上作了以下工作：

- (1) 物理格式化磁盘；
- (2) 建立 A: 这个目录 (=Linux 中的建立一个文件系统)；
- (3) 使这个磁盘能被使用者使用 (=Linux 中的 mount 一个驱动器)。

这三步在 Linux 中是分开的，在 Linux 中可以建立和使用 MS-DOS 格式的磁盘，还可用其它格式的（例如最常用的 ext2），以下是做一张可用的磁盘方法：

（1）先 su 为 root

（2）格式化一张 1.44M 的软盘(A:)

```
# fdformat /dev/fd0H1440
```

（3）建立一个文件系统：

```
# mkfs -t ext2 -c /dev/fd0H1440（建立一个 ext2 的文件系统）
```

```
或 # mformat a:（建立一个 MS-DOS 的文件系统）
```

在使用这张磁盘前，需要先将这个磁盘连通驱动器 mount，mount 一个驱动器：

```
# mount -t ext2 /dev/fd0 /mnt
```

```
或 # mount -t msdos /dev/fd0 /mnt
```

现在可以使用这张磁盘了，当要拿出磁盘前，一定要 umount！

```
# umount /mnt
```

现在可以把磁盘拿出来了。

fdformat 和 mkfs 最好只对未格式过化的磁盘用、已作过的就不需要用这两条命令了。

如要使用 B 驱动器，用 fd1H1440 和 fd1 替换前文中的 fd0H1440 和 fd0。这样，以前对 A:，B: 的工作现在都移转对 /mnt 操作了，例如：

DOS Linux

```
C:GUIDO>dir a: $ ls /mnt
```

```
C:GUIDO>copy a:*. * docs emp $ cp /mnt/* /docs/temp
```

```
C:GUIDO>copy *.zip a:zip $ cp *.zip /mnt/zip
```

```
C:GUIDO>a: $ cd /mnt
```

```
A:> /mnt$
```

使用硬盘的方法类似于软盘，例如可以 mount 一个硬盘或光驱，稍微变一下 /dev 下的 device，例如 mount 一个光驱：

```
# mount -t iso9660 /dev/cdrom /cdrom
```


2.6 free

2.6.1 功能说明

显示内存状态。

2.6.2 语法

```
free [-bkmotV] [-s <间隔秒数>]
```

2.6.3 补充说明

free 指令会显示内存的使用情况，包括实体内存，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。

2.6.4 参数

- b 以 Byte 为单位显示内存使用情况。
- k 以 KB 为单位显示内存使用情况。
- m 以 MB 为单位显示内存使用情况。
- o 不显示缓冲区调节列。
- s<间隔秒数> 持续观察内存使用状况。
- t 显示内存总和列。
- V 显示版本信息。

2.7 mount

2.7.1 功能说明

挂载文件系统。

2.7.2 语法

```
mount [-t vfstype] [-o options] device dir
```

其中：

1. `-t vfstype` 指定文件系统的类型，通常不必指定。`mount` 会自动选择正确的类型。常用类型有：

光盘或光盘镜像：iso9660

DOS fat16 文件系统：msdos

Windows 9x fat32 文件系统：vfat

Windows NT ntfs 文件系统：ntfs

Mount Windows 文件网络共享：smbfs

UNIX (LINUX) 文件网络共享：nfs

2. `-o options` 主要用来描述设备或档案的挂接方式。常用的参数有：

`loop`：用来把一个文件当成硬盘分区挂接上系统

`ro`：采用只读方式挂接设备

`rw`：采用读写方式挂接设备

`iocharset`：指定访问文件系统所用字符集

3. `device` 要挂接(`mount`)的设备。

4. `dir` 设备在系统上的挂接点(`mount point`)。

2.7.3 例子

2.7.3.1 制作并挂载光盘 ISO 文件

1、从光盘制作光盘镜像文件。将光盘放入光驱，执行下面的命令。

```
#cp /dev/cdrom /home/sunky/mydisk.iso 或
```

```
#dd if=/dev/cdrom of=/home/sunky/mydisk.iso
```

注：执行上面的任何一条命令都可将当前光驱里的光盘制作成光盘镜像文件/home/sunky/mydisk.iso

2、将文件和目录制作成光盘镜像文件，执行下面的命令。

```
#mkisofs -r -J -V mydisk -o /home/sunky/mydisk.iso /home/sunky/ mydir
```

注：这条命令将/home/sunky/mydir 目录下所有的目录和文件制作成光盘镜像文件/home/sunky/mydisk.iso，光盘卷标为：mydisk

3、光盘镜像文件的挂载(mount)

```
#mkdir /mnt/vcdrom
```

注：建立一个目录用来作挂载点(mount point)

```
#mount -o loop -t iso9660 /home/sunky/mydisk.iso /mnt/vcdrom
```

注：使用/mnt/vcdrom 就可以访问盘镜像文件 mydisk.iso 里的所有文件了。

2.7.3.2 挂载移动硬盘

对 linux 系统而言，USB 接口的移动硬盘是当作 SCSI 设备对待的。插入移动硬盘之前，应先用 fdisk -l 或 more /proc/partitions 查看系统的硬盘和硬盘分区情况。

```
[root at pldyrouter /]# fdisk -l
```

```
Disk /dev/sda: 73 dot 4 GB, 73407820800 bytes
```

```
255 heads, 63 sectors/track, 8924 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Device Boot Start End Blocks Id System
```

```
/dev/sda1 1 4 32098+ de Dell Utility
```

```
/dev/sda2 * 5 2554 20482875 7 HPFS/NTFS
```

```
/dev/sda3 2555 7904 42973875 83 Linux
```

```
/dev/sda4 7905 8924 8193150 f Win95 Ext'd (LBA)
```

```
/dev/sda5 7905 8924 8193118+ 82 Linux swap
```

在这里可以清楚地看到系统有一块 SCSI 硬盘/dev/sda 和它的四个磁盘分区/dev /sda1 — /dev/sda4, /dev/sda5 是分区/dev/sda4 的逻辑分区。接好移动硬盘后, 再用 `fdisk -l` 或 `more /proc/partitions` 查看系统的硬盘和硬盘分区情况

```
[root at pldyrouter /]# fdisk -l
```

```
Disk /dev/sda: 73 dot 4 GB, 73407820800 bytes
```

```
255 heads, 63 sectors/track, 8924 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Device Boot Start End Blocks Id System
```

```
/dev/sda1 1 4 32098+ de Dell Utility
```

```
/dev/sda2 * 5 2554 20482875 7 HPFS/NTFS
```

```
/dev/sda3 2555 7904 42973875 83 Linux
```

```
/dev/sda4 7905 8924 8193150 f Win95 Ext'd (LBA)
```

```
/dev/sda5 7905 8924 8193118+ 82 Linux swap
```

```
Disk /dev/sdc: 40.0 GB, 40007761920 bytes
```

```
255 heads, 63 sectors/track, 4864 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Device Boot Start End Blocks Id System
```

```
/dev/sdc1 1 510 4096543+ 7 HPFS/NTFS
```

```
/dev/sdc2 511 4864 34973505 f Win95 Ext'd (LBA)
```

```
/dev/sdc5 511 4864 34973473+ b Win95 FAT32
```

大家应该可以发现多了一个 SCSI 硬盘/dev/sdc 和它的两个磁盘分区/dev/sdc1?、/dev/sdc2, 其中/dev/sdc5 是/dev/sdc2 分区的逻辑分区。我们可以使用下面的命令挂载/dev/sdc1 和/dev/sdc5。

```
#mkdir -p /mnt/usbhd1
```

```
#mkdir -p /mnt/usbhd2
```

注：建立目录用来作挂载点(mount point)

```
#mount -t ntfs /dev/sdc1 /mnt/usbhd1
```

```
#mount -t vfat /dev/sdc5 /mnt/usbhd2
```

注：对 ntfs 格式的磁盘分区应使用 -t ntfs 参数，对 fat32 格式的磁盘分区应使用 -t vfat 参数。若汉字文件名显示为乱码或不显示，可以使用下面的命令格式。

```
#mount -t ntfs -o iocharset=cp936 /dev/sdc1 /mnt/usbhd1
```

```
#mount -t vfat -o iocharset=cp936 /dev/sdc5 /mnt/usbhd2
```

linux 系统下使用 fdisk 分区命令和 mkfs 文件系统创建命令可以将移动硬盘的分区制作成 linux 系统所特有的 ext2、ext3 格式。这样，在 linux 下使用就更方便了。使用下面的命令直接挂载即可。

```
#mount /dev/sdc1 /mnt/usbhd1
```

2.7.3.3 挂载 U 盘

和 USB 接口的移动硬盘一样对 linux 系统而言 U 盘也是当作 SCSI 设备对待的。使用方法和移动硬盘完全一样。插入 U 盘之前，应先用 fdisk -l 或 more /proc/partitions 查看系统的硬盘和硬盘分区情况。

```
[root at pldyrouter root]# fdisk -l
```

```
Disk /dev/sda: 73 dot 4 GB, 73407820800 bytes
```

```
255 heads, 63 sectors/track, 8924 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Device Boot Start End Blocks Id System
```

```
/dev/sda1 1 4 32098+ de Dell Utility
```

```
/dev/sda2 * 5 2554 20482875 7 HPFS/NTFS  
  
/dev/sda3 2555 7904 42973875 83 Linux  
  
/dev/sda4 7905 8924 8193150 f Win95 Ext'd (LBA)  
  
/dev/sda5 7905 8924 8193118+ 82 Linux swap
```

插入U盘后,再用 `fdisk -l` 或 `more /proc/partitions` 查看系统的硬盘和硬盘分区情况。

```
[root at pldyrouter root]# fdisk -l  
  
Disk /dev/sda: 73 dot 4 GB, 73407820800 bytes  
  
255 heads, 63 sectors/track, 8924 cylinders  
  
Units = cylinders of 16065 * 512 = 8225280 bytes  
  
Device Boot Start End Blocks Id System  
  
/dev/sda1 1 4 32098+ de Dell Utility  
  
/dev/sda2 * 5 2554 20482875 7 HPFS/NTFS  
  
/dev/sda3 2555 7904 42973875 83 Linux  
  
/dev/sda4 7905 8924 8193150 f Win95 Ext'd (LBA)  
  
/dev/sda5 7905 8924 8193118+ 82 Linux swap  
  
Disk /dev/sdd: 131 MB, 131072000 bytes  
  
9 heads, 32 sectors/track, 888 cylinders  
  
Units = cylinders of 288 * 512 = 147456 bytes  
  
Device Boot Start End Blocks Id System  
  
/dev/sdd1 * 1 889 127983+ b Win95 FAT32  
  
Partition 1 has different physical/logical endings:  
  
phys=(1000, 8, 32) logical=(888, 7, 31)
```

系统多了一个 SCSI 硬盘/dev/sdd 和一个磁盘分区/dev/sdd1, /dev/sdd1 就是我们要挂接的 U 盘。

```
#mkdir -p /mnt/usb
```

注：建立一个目录用来作挂接点 (mount point)

```
#mount -t vfat /dev/sdd1 /mnt/usb
```

注：现在可以通过/mnt/usb 来访问 U 盘了，若汉字文件名显示为乱码或不显示，可以使用下面的命令。

```
#mount -t vfat -o iocharset=cp936 /dev/sdd1 /mnt/usb
```

2.7.3.4 挂接 Windows 文件共享

Windows 网络共享的核心是 SMB/CIFS，在 linux 下要挂接 (mount) windows 的磁盘共享，就必须安装和使用 samba 软件包。现在流行的 linux 发行版绝大多数已经包含了 samba 软件包，如果安装 linux 系统时未安装 samba 请首先安装 samba。当然也可以到 www.samba.org 网站下载.....新的版本是 3.0.10 版。

当 windows 系统共享设置好以后，就可以在 linux 客户端挂接 (mount) 了，具体操作如下：

```
# mkdir -p /mnt/samba
```

注：建立一个目录用来作挂接点 (mount point)

```
# mount -t smbfs -o username=administrator,password=pldy123 //10.140.133.23/c$ /mnt/samba
```

注：administrator 和 pldy123 是 ip 地址为 10.140.133.23 windows 计算机的一个用户名和密码，c\$是这台计算机的一个磁盘共享。如此就可以在 linux 系统上通过/mnt/samba 来访问 windows 系统磁盘上的文件了。

2.7.3.5 挂接 UNIX 系统 NFS 文件共享

类似于 windows 的网络共享，UNIX (Linux) 系统也有自己的网络共享，那就是 NFS (网络文件系统)，下面我们就以 SUN Solaris2.8 和 REDHAT as server 3 为例简单介绍一下在 linux 下如何 mount nfs 网络共享。

在 linux 客户端挂接 (mount) NFS 磁盘共享之前，必须先配置好 NFS 服务端。

1、Solaris 系统 NFS 服务端配置方法如下：

(1) 修改 /etc/dfs/dfstab, 增加共享目录

```
share -F nfs -o rw /export/home/sunky
```

(2) 启动 nfs 服务

```
# /etc/init.d/nfs.server start
```

(3) NFS 服务启动以后, 也可以使用下面的命令增加新的共享

```
# share /export/home/sunky1
```

```
# share /export/home/sunky2
```

注: /export/home/sunky 和 /export/home/sunky1 是准备共享的目录

2、linux 系统 NFS 服务端配置方法如下：

(1) 修改 /etc/exports, 增加共享目录

```
/export/home/sunky 10.140.133.23(rw)
```

```
/export/home/sunky1 *(rw)
```

```
/export/home/sunky2 linux-client(rw)
```

注: /export/home/目录下的 sunky、sunky1、sunky2 是准备共享 的目录, 10.140.133.23、*、linux-client 是被允许挂接此共享 linux 客户机的 IP 地址或主机名。如果要使用主机名 linux-client 必须在服务端主机/etc/hosts 文件里增加 linux-client 主机 ip 定义。格式如下:

```
10.140.133.23 linux-client
```

(2) 启动与停止 NFS 服务

```
/etc/rc.d/init.d/portmap start (在 REDHAT 中 PORTMAP 是默认启动的)
```

```
/etc/rc.d/init.d/nfs start 启动 NFS 服务
```

```
/etc/rc.d/init.d/nfs stop 停止 NFS 服务
```


注：若修改/etc/export 文件增加新的共享，应先停止 NFS 服务，再启动 NFS 服务方能使新增加的共享起作用。使用命令 `exportfs -rv` 也可以达到同样的效果。

3、linux 客户端挂接(mount)其他 linux 系统或 UNIX 系统的 NFS 共享

```
# mkdir -p /mnt/nfs
```

注：建立一个目录用来作挂接点(mount point)

```
#mount -t nfs -o rw 10.140.133.9:/export/home/sunky /mnt/nfs
```

注：这里我们假设 10.140.133.9 是 NFS 服务端的主机 IP 地址，当然这里也可以使用主机名，但必须在本机/etc/hosts 文件里增加服务端 ip 定义。/export/home/sunky 为服务端共享的目录。

2.8 umount

2.8.1 功能说明

卸除文件系统。

2.8.2 语法

```
umount [-ahnrV] [-t <文件系统类型>][文件系统]
```

2.8.3 补充说明

umount 可卸除目前挂在 Linux 目录中的文件系统。

2.8.4 参数

- a 卸除/etc/mtab 中记录的所有文件系统。
- 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挂入文件系统。

-t<文件系统类型> 仅卸除选项中所指定的文件系统。

-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，也可以用设备名称或挂入点来表示文件系统。

3 压缩管理

3.1 bunzip2

3.1.1 功能说明

.bz2 文件的解压缩程序。

3.1.2 语法

bunzip2 [-fkLsvV] [.bz2 压缩文件]

3.1.3 补充说明

bunzip2 可解压缩.bz2 格式的压缩文件。bunzip2 实际上是 bzip2 的符号连接，执行 bunzip2 与 bzip2 -d 的效果相同。

3.1.4 参数

-f 或 --force 解压缩时，若输出的文件与现有文件同名时，预设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，预设会删除原来的压缩文件。若要保留压缩文件，请使用此参数。

-s 或 --small 降低程序执行时，内存的使用量。

-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。

3.2 bzip2

3.2.1 功能说明

.bz2 文件的压缩程序。

3.2.2 语法

bzip2 [-cdfhklstvVz][--repetitive-best][--repetitive-fast][- 压缩等级][要压缩的文件]

3.2.3 补充说明

bzip2 采用新的压缩演算法，压缩效果比传统的 LZ77/LZ78 压缩演算法来得好。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，并删除原始的文件。

3.2.4 参数

-c 或 --stdout 将压缩与解压缩的结果送到标准输出。

-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，预设不会覆盖现有文件。若要覆盖，请使用此参数。

-h 或 --help 显示帮助。

-k 或 --keep bzip2 在压缩或解压缩后，会删除原始的文件。若要保留原始文件，请使用此参数。

-s 或 --small 降低程序执行时内存的使用量。

-t 或 --test 测试 .bz2 压缩文件的完整性。

-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。

`--repetitive-best` 若文件中有重复出现的资料时，可利用此参数提高压缩效果。

`--repetitive-fast` 若文件中有重复出现的资料时，可利用此参数加快执行速度。

`-压缩等级` 压缩时的区块大小。

3.3 cpio

3.3.1 功能说明

备份文件。

3.3.2 语法

`cpio [-0aBckLovV][-C <输入/输出大小>][-F <备份档>][-H <备份格式>][-O <备份档>][--block-size=<区块大小>][--force-local][--help][--quiet][--version]`

或 `cpio [-bBcdfikmnrSstuvV][-C <输入/输出大小>][-E <范本文件>][-F <备份档>][-H <备份格式>][-I <备份档>][-M <回传信息>][-R <拥有者><:/.><所属群组>][--block-size=<区块大小>][--force-local][--help][--no-absolute-filenames][--no-preserve-owner][--only-verify-crc][--quiet][--sparse] [--version][范本样式...]`

或 `cpio [-0adkiLmpuvV][-R <拥有者><:/.><所属群组>][--help][--no-preserve-owner][--quiet] [--sparse][--version][目的目]`

3.3.3 补充说明

`cpio` 是用来建立，还原备份档的工具程序，它可以加入，解开 `cpio` 或 `tra` 备份档内的文件。

3.3.4 参数

`-0` 或 `--null` 接受新增列控制字符，通常配合 `find` 指令的 `"-print0"` 参数使用。

`-a` 或 `--reset-access-time` 重新设置文件的存取时间。

`-A` 或 `--append` 附加到已存在的备份档中，且这个备份档必须存放在磁盘上，而不能放置于磁带机里。

`-b` 或 `--swap` 此参数的效果和同时指定 `"-sS"` 参数相同。

`-B` 将输入/输出的区块大小改成 5210 Bytes。

-c 使用旧 ASCII 备份格式。

-C<区块大小>或--io-size=<区块大小> 设置输入/输出的区块大小，单位是 Byte。

-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，格式为每列一个范本样式。

-f 或--nonmatching 让 cpio 解开所有不符合范本条件的文件。

-F<备份档>或--file=<备份档> 指定备份档的名称，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保存设备存取备份档。

-H<备份格式> 指定备份时欲使用的文件格式。

-i 或--extract 执行 copy-in 模式，还原备份档。

-l<备份档> 指定备份档的名称，用来取代标准输入，也能借此通过网络使用另一台主机的保存设备读取备份档。

-k 此参数将忽略不予处理，仅负责解决 cpio 不同版本间的兼容性问题。

-l 或--link 以硬连接的方式取代复制文件，可在 copy-pass 模式下运用。

-L 或--dereference 不建立符号连接，直接复制该连接所指向的原始文件。

-m 或 preserve-modification-time 不去更换文件的更改时间。

-M<回传信息>或--message=<回传信息> 设置更换保存媒体的信息。

-n 或--numeric-uid-gid 使用“-tv”参数列出备份档的内容时，若再加上参数“-n”，则会以用户识别码和群组识别码替代拥有者和群组名称列出文件清单。

-o 或--create 执行 copy-out 模式，建立备份档。

-O<备份档> 指定备份档的名称，用来取代标准输出，也能借此通过网络 使用另一台主机的保存设备存放备份档。

-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动时，采用互动模式。

-R<拥有者><:/./><所属群组>或

----owner<拥有者><:/.><所属群组> 在 copy-in 模式还原备份档,或 copy-pass 模式复制文件时,可指定这些备份,复制的文件的拥有者与所属群组。

-s 或--swap-bytes 交换每对字节的内容。

-S 或--swap-halfwords 交换每半个字节的内容。

-t 或--list 将输入的内容呈现出来。

-u 或--unconditional 置换所有文件,不论日期时间的新旧与否,皆不予询问而直接覆盖。

-v 或--verbose 详细显示指令的执行过程。

-V 或--dot 执行指令时,在每个文件的执行程序前面加上"."号

--block-size=<区块大小> 设置输入/输出的区块大小,假如设置数值为 5,则区块大小为 2500,若设置成 10,则区块大小为 5120,依次类推。

--force-local 强制将备份档存放在本地主机。

--help 在线帮助。

--no-absolute-filenames 使用相对路径建立文件名称。

--no-preserve-owner 不保留文件的拥有者,谁解开了备份档,那些文件就归谁所有。

-only-verify-crc 当备份档采用 CRC 备份格式时,可使用这项参数检查备份档内的每个文件是否正确无误。

--quiet 不显示复制了多少区块。

--sparse 倘若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。

--version 显示版本信息。

3.4 gunzip

3.4.1 功能说明

解压文件。

3.4.2 语法

`gunzip [-acfhLLnNqrtvV][-s <压缩字尾字符串>][文件...]` 或 `gunzip [-acfhLLnNqrtvV][-s <压缩字尾字符串>][目录]`

3.4.3 补充说明

`gunzip` 是个使用广泛的解压缩程序，它用于解开被 `gzip` 压缩过的文件，这些压缩文件预设最后的扩展名为“.gz”。事实上 `gunzip` 就是 `gzip` 的硬连接，因此不论是压缩或解压缩，都可通过 `gzip` 指令单独完成。

3.4.4 参数

- `-a` 或 `--ascii` 使用 ASCII 文字模式。
- `-c` 或 `--stdout` 或 `--to-stdout` 把解压后的文件输出到标准输出设备。
- `-f` 或 `-force` 强行解开压缩文件，不理睬文件名称或硬连接是否存在以及该文件是否为符号连接。
- `-h` 或 `--help` 在线帮助。
- `-l` 或 `--list` 列出压缩文件的相关信息。
- `-L` 或 `--license` 显示版本与版权信息。
- `-n` 或 `--no-name` 解压缩时，若压缩文件内含有远来的文件名称及时间戳记，则将其忽略不予处理。
- `-N` 或 `--name` 解压缩时，若压缩文件内含有原来的文件名称及时间戳记，则将其回存到解开的文件上。
- `-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。
- `-t` 或 `--test` 测试压缩文件是否正确无误。
- `-v` 或 `--verbose` 显示指令执行过程。
- `-V` 或 `--version` 显示版本信息。

3.5 gzip

3.5.1 功能说明

压缩文件。

3.5.2 语法

```
gzip [-acdfhlLnNqrtvV] [-S <压缩字尾字符串>] [-<压缩效率>] [--best/fast] [文件...] 或 gzip [-acdfhlLnNqrtvV] [-S <压缩字尾字符串>] [-<压缩效率>] [--best/fast] [目录]
```

3.5.3 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。

3.5.4 参数

-a 或--ascii 使用 ASCII 文字模式。

-c 或--stdout 或--to-stdout 把压缩后的文件输出到标准输出设备，不去更动原始文件。

-d 或--decompress 或--uncompress 解开压缩文件。

-f 或--force 强行压缩文件。不理睬文件名称或硬连接是否存在以及该文件是否为符号连接。

-h 或--help 在线帮助。

-l 或--list 列出压缩文件的相关信息。

-L 或--license 显示版本与版权信息。

-n 或--no-name 压缩文件时，不保存原来的文件名称及时间戳记。

-N 或--name 压缩文件时，保存原来的文件名称及时间戳记。

-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。

-t 或--test 测试压缩文件是否正确无误。

-v 或--verbose 显示指令执行过程。

-V 或--version 显示版本信息。

-<压缩效率> 压缩效率是一个介于 1—9 的数值，预设值为“6”，指定愈大的数值，压缩效率就会愈高。

--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。

3.6 split

3.6.1 功能说明

切割文件。

3.6.2 语法

split [--help][--version][-<行数>][-b <字节>][-C <字节>][-l <行数>][要切割的文件][输出文件名]

3.6.3 补充说明

split 可将文件切成较小的文件，预设每 1000 行会切成一个小文件。

3.6.4 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> 指定每多少字就要切成一个小文件。

-C<字节> 与-b 参数类似，但切割时尽量维持每行的完整性。

--help 显示帮助。

--version 显示版本信息。

[输出文件名] 设置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。

3.7 tar

3.7.1 功能说明

备份文件。

3.7.2 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 <卷册名称>] [-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=<群组名称>]
```

```
[--help] [--ignore-failed-read] [--new-volume-script=<Script 文件>]
```

```
[--newer-mtime] [--no-recursion] [--null] [--numeric-owner] [--owner=<用户名称>]
```

```
[--posix] [--serve] [--preserve-order] [--preserve-permissions] [--record-size=<区块数目>]
```

```
[--recursive-unlink] [--remove-files] [--rsh-command=<执行指令>]
```

```
[--same-owner] [--suffix=<备份字尾字符串>] [--totals] [--use-compress-program=<执行指令>]
```

```
[--version] [--volno-file=<编号文件>] [文件或目录...]
```

3.7.3 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。

3.7.4 参数

-A 或--catenate 新增温暖件到已存在的备份文件。

-b<区块数目>或--blocking-factor=<区块数目> 设置每笔记录的区块数目，每个区块大小为 12Bytes。

-B 或--read-full-records 读取数据时重设区块大小。

-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 或--diff 或--compare 对比备份文件内和文件系统上的文件的差异。

-f<备份文件>或--file=<备份文件> 指定备份文件。

-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。

-g 或--listed-incremental 处理 GNU 格式的大量备份。

-G 或--incremental 处理旧的 GNU 格式的大量备份。

-h 或--dereference 不建立符号连接，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 复制的文件或目录存放的文件系统，必须与 tar 指令执行时所处的文件系统相同，否则不予复制。

-L<媒体容量>或--tape-length=<媒体容量> 设置存放每体的容量，单位以 1024 Bytes 计算。

-m 或--modification-time 还原文件时，不变更文件的更改时间。

-M 或--multi-volume 在建立，还原备份文件或列出其中的内容时，采用多卷册模式。

-N<日期格式>或--newer=<日期时间> 只将较指定日期更新的文件保存到备份文件里。

-o 或--old-archive 或--portability 将资料写入备份文件时使用 V7 格式。

-O 或--stdout 把从备份文件里还原的文件输出到标准输出设备。

-p 或--same-permissions 用原来的文件权限还原文件。

-P 或--absolute-names 文件名使用绝对名称，不移除文件名称前的"/"号。

-r 或--append 新增文件到已存在的备份文件的结尾部分。

-R 或--block-number 列出每个信息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 倘若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。

-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 解开或建立符合设置条件的文件。

-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 显示指令执行过程。

-V<卷册名称>或--label=<卷册名称> 建立使用指定的卷册名称的备份文件。

-w 或--interactive 遭遇问题时先询问用户。

-W 或--verify 写入备份文件后，确认文件正确无误。

-x 或--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 ar 排除符合设置条件的文件。

-z 或--gzip 或--ungzip 通过 gzip 指令处理备份文件。

-Z 或--compress 或--uncompress 通过 compress 指令处理备份文件。

-<设备编号><存储密度> 设置备份用的外围设备编号及存放数据的密度。

--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>或--backup 移除文件前先进行备份。

--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。
- `--exclude=<范本样式>` 排除符合范本样式的问家。
- `--group=<群组名称>` 把加入设备文件中的文件的所属群组设成指定的群组。
- `--help` 在线帮助。
- `--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` 只保存更改过的文件。
- `--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` 以用户识别码及群组识别码取代用户名称和群组名称。
- `--owner=<用户名称>` 把加入备份文件中的文件的拥有者设成指定的用户。
- `--posix` 将数据写入备份文件时使用 POSIX 格式。
- `--preserve` 此参数的效果和指定“-ps”参数相同。
- `--preserve-order` 此参数的效果和指定“-A”参数相同。
- `--preserve-permissions` 此参数的效果和指定“-p”参数相同。
- `--record-size=<区块数目>` 此参数的效果和指定“-b”参数相同。
- `--recursive-unlink` 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。
- `--remove-files` 文件加入备份文件后，就将其删除。
- `--rsh-command=<执行指令>` 设置要在远端主机上执行的指令，以取代 rsh 指令。
- `--same-owner` 尝试以相同的文件拥有者还原问家你。
- `--suffix=<备份字尾字符串>` 移除文件前先行备份。

- totals 备份文件建立后，列出文件大小。
- use-compress-program=<执行指令> 通过指定的指令处理备份文件。
- version 显示版本信息。
- volno-file=<编号文件> 使用指定文件内的编号取代预设的卷册编号。

3.8 unzip

3.8.1 功能说明

解压缩 zip 文件

3.8.2 语法

unzip [-cflptuvz][-agCjLMnoqsVX][-P <密码>][.zip 文件][文件][-d <目录>][-x <文件>] 或 unzip [-Z]

3.8.3 补充说明

unzip 为 .zip 压缩文件的解压缩程序。

3.8.4 参数

- c 将解压缩的结果显示到屏幕上，并对字符做适当的转换。
- f 更新现有的文件。
- 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到目录中。
- v 执行是时显示详细的信息。

- z 仅显示压缩文件的备注文字。
- 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。
- L 将压缩文件中的全部文件名改为小写。
- M 将输出结果送到 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 执行后覆盖原有文件。
- 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。
- d<目录> 指定文件解压缩后所要存储的目录。
- x<文件> 指定不要处理 .zip 压缩文件中的哪些文件。
- Z unzip -Z 等于执行 zipinfo 指令。

3.9 zgrep

3.9.1 功能说明

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` 命令相同，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。

3.9.2 语法

```
zgrep [options] [files]
```

[options]同 `grep` 对应选项。

3.10 zip

3.10.1 功能说明

压缩文件。

3.10.2 语法

```
zip [-options][-b <工作目录>][-l1][-n <字尾字符串>][-t <日期时间>][-<压缩效率>][压缩文件][文件...][-i <范本样式>][-x <范本样式>]
```

3.10.3 补充说明

`zip` 是个使用广泛的压缩程序，文件经它压缩后会另外产生具有“.zip”扩展名的压缩文件。

3.10.4 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。

`-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。

`-f` 此参数的效果和指定“-u”参数类似，但不仅更新既有文件，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。
- g 将文件压缩后附加在既有的压缩文件之后，而非另行建立新的压缩文件。
- h 在线帮助。
- i<范本样式> 只压缩符合条件的文件。
- j 只保存文件名称及其内容，而不存放任何目录名称。
- J 删除压缩文件前面不必要的数据。
- k 使用 MS-DOS 兼容格式的文件名称。
- l 压缩文件时，把 LF 字符置换成 LF+CR 字符。
- ll 压缩文件时，把 LF+CR 字符置换成 LF 字符。
- 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。
- n<字尾字符串> 不压缩具有特定字尾字符串的文件。
- o 以压缩文件内拥有最新更改时间的文件为准，将压缩文件的更改时间设成和该文件相同。
- q 不显示指令执行过程。
- r 递归处理，将指定目录下的所有文件和子目录一并处理。
- S 包含系统和隐藏文件。
- t<日期时间> 把压缩文件的日期设成指定的日期。
- T 检查备份文件内的每个文件是否正确无误。
- u 更换较新的文件到压缩文件内。
- v 显示指令执行过程或显示版本信息。
- V 保存 VMS 操作系统的文件属性。
- w 在文件名称里假如版本编号，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。

- X 不保存额外的文件属性。
- y 直接保存符号连接，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。
- \$ 保存第一个被压缩文件所在磁盘的卷册名称。
- <压缩效率> 压缩效率是一个介于 1-9 的数值。

4 进程管理

4.1 crontab

4.1.1 功能说明

设置计时器。

4.1.2 语法

crontab [-u <用户名称>][配置文件] 或 crontab [-u <用户名称>][-elr]

4.1.3 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预设的指令或程序。只要用户会编辑计时器的配置文件，就可以使用计时器的功能。其配置文件格式如下：

Minute	Hour	Day	Month	DayOfWeek	Command
--------	------	-----	-------	-----------	---------

4.1.4 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> 指定要设定计时器的用户名称。

4.2 kill

4.2.1 功能说明

删除执行中的程序或工作。

4.2.2 语法

kill [-s <信息名称或编号>][程序] 或 kill [-l <信息编号>]

4.2.3 补充说明

kill 可将指定的信息送至程序。预设的信息为 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 指令或 jobs 指令查看。

4.2.4 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。

-s <信息名称或编号> 指定要送出的信息。

[程序] [程序]可以是程序的 PID 或是 PGID，也可以是工作编号。

4.3 nohup

4.3.1 功能说明

不挂断地运行命令。该命令可以在你退出帐户/关闭终端之后继续运行相应的进程。

4.3.2 语法

nohup Command [Arg ...] [&]

nohup 命令运行由 Command 参数和任何相关的 Arg 参数指定的命令,忽略所有挂断(SIGHUP)信号。在注销后使用 nohup 命令运行后台中的程序。要运行后台中的 nohup 命令,添加& (表示“and”的符号)到命令的尾部。

无论是否将 nohup 命令的输出复位向到终端,输出都将附加到当前目录的 nohup.out 文件中。如果当前目录的 nohup.out 文件不可写,输出复位向到 \$HOME/nohup.out 文件中。如果没有文件能创建或打开以用于追加,那么 Command 参数指定的命令不可调用。如果标准错误是一个终端,那么把指定的命令写给标准错误的所有输出作为标准输出复位向到相同的文件描述符。

退出状态: 该命令返回下列出口值:

126 可以查找但不能调用 Command 参数指定的命令。

127 nohup 命令发生错误或不能查找由 Command 参数指定的命令。

否则, nohup 命令的退出状态是 Command 参数指定命令的退出状态。

如果使用 nohup 命令提交作业,那么在缺省情况下该作业的所有输出都被复位向到一个名为 nohup.out 的文件中,除非另外指定了输出文件:

nohup command > myout.file 2>&l &在上面的例子中,输出被复位向到 myout.file 文件中。

4.3.3 例子

我们有个 test.php 需要在后台运行,并且希望在后台能够定期运行,那么就使用 nohup:

```
nohup /root/test.php &
```

```
(nohup sh make.sh &)
```

4.4 ps

4.4.1 功能说明

报告程序状况。

4.4.2 语法

```
ps [-aAcdefHjlmNVwy][acefghLnrsSTuvxX][-C <指令名称>][-g <群组名称>][-G <群组识别码>][-p <程序识别码>][p <程序识别码>][-s <阶段作业>][-t <终端机编号>][t <终端机编号>][-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][--deselect][--forest][--headers][--help][--info][--lines <显示列数>][--no-headers][--group <群组名称>][-Group <群组识别码>][--pid <程序识别码>][--rows <显示列数>][-sid <阶段作业>][--tty <终端机编号>][--user <用户名称>][--User <用户识别码>][--version][--width <每列字符数>]
```

4.4.3 补充说明

ps 是用来报告程序执行状况的指令，您可以搭配 kill 指令随时中断，删除不必要的程序。

4.4.4 参数

- a 显示所有终端机下执行的程序，除了阶段作业领导者之外。
- a 显示现行终端机下的所有程序，包括其他用户的程序。
- A 显示所有程序。
- c 显示 CLS 和 PRI 栏位。
- c 列出程序时，显示每个程序真正的指令名称，而不包含路径，参数或常驻服务的标示。
- C<指令名称> 指定执行指令的名称，并列出该指令的程序的状况。
- d 显示所有程序，但不包括阶段作业领导者的程序。
- 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, PPIP, 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<群组名称> 此参数的效果和指定“-G”参数相同，当亦能使用阶段作业领导者的名称来指定。

g 显示现行终端机下的所有程序，包括群组领导者的程序。

-G<群组识别码> 列出属于该群组的程序的状态，也可使用群组名称来指定。

h 不显示标题列。

-H 显示树状结构，表示程序间的相互关系。

-j 或 j 采用工作控制的格式显示程序状态。

-l 或 l 采用详细的格式来显示程序状态。

L 列出栏位的相关信息。

-m 或 m 显示所有的执行绪。

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 指令终端机下的程序之外。

-p<程序识别码> 指定程序识别码，并列出该程序的状态。

p<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。

r 只列出现行终端机正在执行中的程序。

-s<阶段作业> 指定阶段作业的程序识别码，并列出隶属该阶段作业的程序的状态。

s 采用程序信号的格式显示程序状态。

S 列出程序时，包括已中断的子程序资料。

-t<终端机编号> 指定终端机编号，并列出属于该终端机的程序的状态。

t<终端机编号> 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 显示现行终端机下的所有程序。

-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态。

-U<用户识别码> 列出属于该用户的程序的状态，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的状态。

v 采用虚拟内存的格式显示程序状态。

-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，不以终端机来区分。

X 采用旧式的 Linux i386 登陆格式显示程序状况。

-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> 设置每列的最大字符数。

--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。

--deselect 此参数的效果和指定“-N”参数相同。

--forest 此参数的效果和指定“-f”参数相同。

--headers 重复显示标题列。

--help 在线帮助。

--info 显示排错信息。

--lines<显示列数> 设置显示画面的列数。

--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<群组名称> 此参数的效果和指定“-G”参数相同。

--Group<群组识别码> 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<阶段作业> 此参数的效果和指定“-s”参数相同。

--tty<终端机编号> 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。

`--widthy<每列字符数>` 此参数的效果和指定“-cols”参数相同。

4.5 renice

4.5.1 功能说明

调整优先权。

4.5.2 语法

`renice` [`优先等级`][`-g` `<程序群组名称>...`][`-p` `<程序识别码>...`][`-u` `<用户名称>...`]

4.5.3 补充说明

`renice` 指令可重新调整程序执行的优先权等级。预设是以程序识别码指定程序调整其优先权，您亦可以指定程序群组或用户名称调整优先权等级，并修改所有隶属于该程序群组或用户的程序的优先权。等级范围从-20—19，只有系统管理者可以改变其他用户程序的优先权，也仅有系统管理者可以设置负数等级。

4.5.4 参数

`-g` `<程序群组名称>` 使用程序群组名称，修改所有隶属于该程序群组的程序的优先权。

`-p` `<程序识别码>` 改变该程序的优先权等级，此参数为预设值。

`-u` `<用户名称>` 指定用户名称，修改所有隶属于该用户的程序的优先权。

4.6 top

4.6.1 功能说明

显示，管理执行中的程序。

4.6.2 语法

`top [bcirqsS][d <间隔秒数>][n <执行次数>]`

4.6.3 补充说明

执行 `top` 指令可显示目前正在系统中执行的程序，并通过它所提供的互动式界面，用热键加以管理。

4.6.4 参数

- b 使用批处理模式。
- c 列出程序时，显示每个程序的完整指令，包括指令名称，路径和参数等相关信息。
- d<间隔秒数> 设置 `top` 监控程序执行状况的间隔时间，单位以秒计算。
- i 执行 `top` 指令时，忽略闲置或是已成为 Zombie 的程序。
- n<执行次数> 设置监控信息的更新次数。
- q 持续监控程序执行的状况。
- s 使用保密模式，消除互动模式下的潜在危机。
- S 使用累计模式，其效果类似 `ps` 指令的“-S”参数。

4.7 who

4.7.1 功能说明

显示目前登入系统的用户信息。

4.7.2 语法

`who [-Himqsw][--help][--version][am i][记录文件]`

4.7.3 补充说明

执行这项指令可得知目前有那些用户登入系统，单独执行 `who` 指令会列出登入帐号，使用的终端机，登入时间以及从何处登入或正在使用哪个 X 显示器。

4.7.4 参数

`-H` 或 `--heading` 显示各栏位的标题信息列。

`-i` 或 `-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` 只显示登入系统的帐号名称和总人数。

`-s` 此参数将忽略不予处理，仅负责解决 who 指令其他版本的兼容性问题。

`-w` 或 `-T` 或 `--mesg` 或 `--message` 或 `--writable` 显示用户的信息状态栏。

`--help` 在线帮助。

`--version` 显示版本信息。

5 网络管理

5.1 Curl

Curl 是 Linux 下一个很强大的 http 命令行工具，其功能十分强大。

5.1.1 读取网页

```
$ curl http://www.linuxidc.com
```

5.1.2 保存网页

```
$ curl http://www.linuxidc.com > page.html
```

```
$ curl -o page.html http://www.linuxidc.com
```

5.1.3 使用的 proxy 服务器及其端口： `-x`

```
$ curl -x 123.45.67.89:1080 -o page.html http://www.linuxidc.com
```

5.1.4 使用 cookie 来记录 session 信息

```
$ curl -x 123.45.67.89:1080 -o page.html -D cookie0001.txt http://www.linuxidc.com
```

这个 option: -D 是把 http 的 response 里面的 cookie 信息存到一个特别的文件中，

这样，当页面被存到 page.html 的同时，cookie 信息也被存到了 cookie0001.txt 里面了

5.1.5 下一次访问的时候，继续使用上次留下的 cookie 信息

使用 option 来把上次的 cookie 信息追加到 http request 里面去: -b

```
$ curl -x 123.45.67.89:1080 -o page1.html -D cookie0002.txt -b cookie0001.txt http://www.linuxidc.com
```

5.1.6 浏览器信息

```
$ curl -A "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)" -x 123.45.67.89:1080 -o page.html -D cookie0001.txt http://www.linuxidc.com
```

5.1.7 referer

```
$ curl -A "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)" -x 123.45.67.89:1080 -e "mail.linuxidc.com" -o page.html -D cookie0001.txt http://www.linuxidc.com
```

这样就可以骗对方的服务器，你是从 mail.linuxidc.com 点击某个链接过来的

5.1.8 下载文件

```
$ curl -o 1.jpg http://cgi2.tky.3web.ne.jp/~zzh/screen1.JPG
```

```
$ curl -O http://cgi2.tky.3web.ne.jp/~zzh/screen1.JPG
```

-O 可以按照服务器上的文件名，自动存在本地

```
$ curl -O http://cgi2.tky.3web.ne.jp/~zzh/screen[1-10].JPG
```

5.1.9 批量下载

```
$ curl -O http://cgi2.tky.3web.ne.jp/~{zzh,nick}/[001-201].JPG
```

这样产生的下载，就是

```
~zzh/001.JPG
```

```
~zzh/002.JPG
```

```
...
```

```
~zzh/201.JPG
```

```
~nick/001.JPG
```

```
~nick/002.JPG
```

```
...
```

```
~nick/201.JPG
```

5.1.10 自定义文件名的下载

```
curl -o #2_#1.jpg http://cgi2.tky.3web.ne.jp/~{zzh,nick}/[001-201].JPG
```

这样，自定义出来下载下来的文件名，就变成了这样：

原来： ~zzh/001.JPG → 下载后： 001-zzh.JPG 原来： ~nick/001.JPG → 下载后：
001-nick.JPG

这样一来就不怕文件重名啦

5.1.11 断点续传

```
$ curl -c -O http://cgi2.tky.3wb.ne.jp/~zzh/screen1.JPG
```

分块下载，我们使用这个 option 就可以了： -r

举例说明

比如我们有一个 `http://cgi2.tky.3web.ne.jp/~zzh/zhao1.MP3` 要下载（赵老师的电话朗诵 :D ）我们就可以用这样的命令：

```
$ curl -r 0-10240 -o "zhao.part1" http://cgi2.tky.3web.ne.jp/~zzh/zhao1.MP3 &\

$ curl -r 10241-20480 -o "zhao.part1" http://cgi2.tky.3web.ne.jp/~zzh/zhao1.MP3
&\

$ curl -r 20481-40960 -o "zhao.part1" http://cgi2.tky.3web.ne.jp/~zzh/zhao1.MP3
&\

$ curl -r 40961- -o "zhao.part1" http://cgi2.tky.3web.ne.jp/~zzh/zhao1.MP3
```

这样就可以分块下载啦。不过你需要自己把这些破碎的文件合并起来如果你用 UNIX 或苹果，用 `cat zhao.part* > zhao.MP3` 就可以如果用的是 Windows，用 `copy /b` 来解决吧，呵呵

5.1.12 浏览 FTP

```
$ curl -u name:passwd ftp://ip:port/path/file
```

或者大家熟悉的

```
$ curl ftp://name:passwd@ip:port/path/file
```

5.1.13 FTP 上传

上传的 option 是 `-T`

比如我们向 ftp 传一个文件：

```
$ curl -T localfile -u name:passwd ftp://upload_site:port/path/
```

5.1.14 HTTP 上传

```
$ curl -T localfile http://cgi2.tky.3web.ne.jp/~zzh/abc.cgi
```

注意，这时候，使用的协议是 HTTP 的 PUT method

5.1.15 POST 模式读取网页

POST 模式的 option 则是 `-d`

比如，

```
$ curl -d "user=nickwolfe&password=12345" http://www.linuxidc.com/login.cgi
```

5.1.16 POST 模式下的文件上传

比如

```
<form method="POST" enctype="multipart/form-data" action="http://cgi2.tky.3web.ne.jp/~zzh/up_file.cgi">
```

```
<input type="file" name="upload">
```

```
<input type="submit" name="nick" value="go">
```

```
</form>
```

这样一个 HTTP 表单，我们要用 curl 进行模拟，就该是这样的语法：

```
$ curl -F upload=@localfile -F nick=go http://cgi2.tky.3web.ne.jp/~zzh/up_file.cgi
```

5.2 finger

5.2.1 功能说明

查找并显示用户信息。

5.2.2 语法

```
finger [-lmsp][帐号名称...]
```

5.2.3 补充说明

finger 指令会去查找，并显示指定帐号的用户相关信息，包括本地与远端主机的用户皆可，帐号名称没有大小写的差别。单独执行 finger 指令，它会显示本地主机现在所有的用户的登陆信息，包括帐号名称，真实姓名，登入终端机，闲置时间，登入时间以及地址和电话。

5.2.4 参数

-l 列出该用户的帐号名称，真实姓名，用户专属目录，登入所用的 Shell，登入时间，转信地址，电子邮件状态，还有计划文件和方案文件内容。

-m 排除查找用户的真实姓名。

-s 列出该用户的帐号名称，真实姓名，登入终端机，闲置时间，登入时间以及地址和电话。

-p 列出该用户的帐号名称，真实姓名，用户专属目录，登入所用的 Shell，登入时间，转信地址，电子邮件状态，但不显示该用户的计划文件和方案文件内容。

5.3 ftp

5.3.1 功能说明

设置文件系统相关功能。

5.3.2 语法

ftp [-dignv][主机名称或 IP 地址]

5.3.3 补充说明

FTP 是 ARPANet 的标准文件传输协议，该网络就是现今 Internet 的前身。

5.3.4 参数

-d 详细显示指令执行过程，便于排错或分析程序执行的情形。

-i 关闭互动模式，不询问任何问题。

-g 关闭本地主机文件名称支持特殊字符的扩充特性。

-n 不使用自动登陆。

-v 显示指令执行过程。

5.4 ifconfig

5.4.1 功能说明

显示或设置网络设备。

5.4.2 语法

```
ifconfig [网络设备][down up -allmulti -arp -promisc][add<地址>][del<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<子网掩码>][tunnel<地  
址>][-broadcast<地址>][pointopoint<地址>][IP 地址]
```

5.4.3 补充说明

ifconfig 可设置网络设备的状态，或是显示目前的设置。

如果提示找不到命令，可用/sbin/ifconfig。

5.4.4 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。

<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。

io_addr<I/O 地址> 设置网络设备的 I/O 地址。

irq<IRQ 地址> 设置网络设备的 IRQ。

media<网络媒介类型> 设置网络设备的媒介类型。

mem_start<内存地址> 设置网络设备在主内存所占用的起始地址。

metric<数目> 指定在计算数据包的转送次数时，所要加上的数目。

mtu<字节> 设置网络设备的 MTU。

netmask<子网掩码> 设置网络设备的子网掩码。

tunnel<地址> 建立 IPv4 与 IPv6 之间的隧道通信地址。

up 启动指定的网络设备。

-broadcast<地址> 将要送往指定地址的数据包当成广播数据包来处理。

-pointopoint<地址> 与指定地址的网络设备建立直接连线，此模式具有保密功能。

-promisc 关闭或启动指定网络设备的 promiscuous 模式。

[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。

5.5 ip

5.5.1 功能说明

ip 是 iproute2 软件包里面的一个强大的网络配置工具，它能够替代一些传统的网络管理工具。例如：ifconfig、route 等。

5.5.2 语法

ip [选项] [动作] [指令]

5.5.3 参数

动作 := { link | addr | route | rule | neigh | tunnel | maddr | mroute | monitor }

选项 := { -V[ersion] | -s[tatistics] | -r[esolve] | -f[amily] { inet | inet6 | ipx | dnet | link } | -o[neline] }

5.5.4 装置介面 (device) 的相关设定：ip link

ip link 可以设定与装置 (device) 有关的相关设定，包括 MTU 以及该网路介面的 MAC 等等，当然也可以启动 (up) 或关闭 (down) 某个网路介面啦！整个语法是这样的：

[root@linux ~]# ip [-s] link show <== 单纯的查阅该装置相关的资讯

[root@linux ~]# ip link set [device] [动作与参数]

参数：

show: 仅显示出这个装置的相关内容, 如果加上 -s 会显示更多统计数据;

set : 可以开始设定项目, device 指的是 eth0, eth1 等等介面代号;

动作与参数: 包括有底下的这些动作:

up|down : 启动 (up) 或关闭 (down) 某个介面, 其他参数使用预设的以太网路;

address : 如果这个装置可以更改 MAC 的话, 用这个参数修改!

name : 给予这个装置一个特殊的名字;

mtu : 就是最大传输单元啊!

范例一: 显示出所有的介面资讯

```
[root@linux ~]# ip link show
```

```
1: lo: <LOOPBACK,UP,10000> mtu 16436 qdisc noqueue
```

```
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

```
2: eth0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc pfifo_fast qlen 1000
```

```
    link/ether 00:50:fc:22:9a:cb brd ff:ff:ff:ff:ff:ff
```

```
3: sit0: <NOARP> mtu 1480 qdisc noop
```

```
    link/sit 0.0.0.0 brd 0.0.0.0
```

```
[root@linux ~]# ip -s link show eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc pfifo_fast qlen 1000
```

```
    link/ether 00:50:fc:22:9a:cb brd ff:ff:ff:ff:ff:ff
```

```
    RX: bytes  packets  errors  dropped  overrun  mcast
```

```
    484011792  2247372  0          0          0          0
```

```
    TX: bytes  packets  errors  dropped  carrier  collsns
```

```
    2914104290  2867753  0          0          0          0
```

使用 `ip link show` 可以显示出整个装置介面的硬体相关资讯，如上所示，包括网卡位址(MAC)、MTU 等等， 比较有趣的应该是那个 `sit0` 的介面了，那个 `sit0` 的介面是用在 IPv4 及 IPv6 的封包转换上的， 对我们仅使用 IPv4 的网路是没有作用的。 `lo` 及 `sit0` 都是主机内部所自行设定的。 而如果加上 `-s` 的参数后，则这个网路卡的相关统计资讯就会被列出来， 包括接收 (RX) 及传送 (TX) 的封包数量等等，详细的内容与 `ifconfig` 所输出的结果相同的。

范例二：启动、关闭与设定装置的相关资讯

```
[root@linux ~]# ip link set eth0 up
```

启动 eth0 这个装置介面；

```
[root@linux ~]# ip link set eth0 down
```

阿就关闭啊！简单的要命～

```
[root@linux ~]# ip link set eth0 mtu 1000
```

更改 MTU 的值，达到 1000 bytes，单位就是 bytes 啊！

更新网路卡的 MTU 使用 `ifconfig` 也可以达成啊！没啥了不起，不过，如果是要更改『网路卡代号、MAC 位址的资讯』的话，那可就得使用 `ip` 罗～不过，设定前得要先关闭该网路卡，否则会不成功。 如下所示：

范例三：修改网路卡代号、MAC 等参数

```
[root@linux ~]# ip link set eth0 name vbird
```

SIOCSIFNAME: Device or resource busy

因为该装置目前是启动的，所以不能这样做设定。你应该要这样做：

```
[root@linux ~]# ip link set eth0 down <==关闭介面
```

```
[root@linux ~]# ip link set eth0 name vbird <==重新设定
```

```
[root@linux ~]# ip link show <==观察一下
```

```
2. vbird: <BROADCAST,MULTICAST> mtu 900 qdisc pfifo_fast qlen 1000
```

```
    link/ether 00:40:d0:13:c3:46 brd ff:ff:ff:ff:ff:ff
```

怕了吧！连网路卡代号都可以改变！不过，玩玩后记得改回来啊！

因为我们的 `ifcfg-eth0` 还是使用原本的装置代号！避免有问题，要改回来

```
[root@linux ~]# ip link set vbird name eth0 <==介面改回来
```

```
[root@linux ~]# ip link set eth0 address aa:aa:aa:aa:aa:aa
```

```
[root@linux ~]# ip link show eth0
```

如果你的网路卡支援硬体位址 (MAC) 可以更改的话,

那么上面这个动作就可以更改你的网路卡位址了! 厉害吧!

不过, 还是那句老话, 测试完之後请立刻改回来啊!

在这个装置的硬体相关资讯设定上面, 包括 MTU, MAC 以及传输的模式等等, 都可以在这里设定。有趣的是那个 address 的项目, 那个项目後面接的可是硬体位址 (MAC) 而不是 IP 喔! 很容易搞错啊! 切记切记! 更多的硬体参数可以使用 `man ip` 查阅一下与 `ip link` 有关的设定。

5.5.5 關於额外的 IP 相关设定: ip address

如果说 `ip link` 是与 OSI 七层协定的第二层资料连阶层有关的话, 那么 `ip address` (`ip addr`) 就是与第三层网路层有关的参数啦! 主要是在设定与 IP 有关的各项参数, 包括 `netmask`, `broadcast` 等等。

```
[root@linux ~]# ip address show <==就是查阅 IP 参数啊!
```

```
[root@linux ~]# ip address [add|del] [IP 参数] [dev 装置名] [相关参数]
```

参数:

`show` : 单纯的显示出介面的 IP 资讯啊;

`add|del` : 进行相关参数的增加 (`add`) 或删除 (`del`) 设定, 主要有:

IP 参数: 主要就是网域的设定, 例如 `192.168.100.100/24` 之类的设定喔;

`dev` : 这个 IP 参数所要设定的介面, 例如 `eth0`, `eth1` 等等;

相关参数: 主要有底下这些:

`broadcast`: 设定广播位址, 如果设定值是 `+` 表示『让系统自动计算』

`label` : 亦即是这个装置的别名, 例如 `eth0:0` 就是了!

`scope` : 这个介面的领域, 通常是这几个大类:

global : 允许来自所有来源的连线;

site : 仅支援 IPv6 , 仅允许本主机的连线;

link : 仅允许本装置自我连线;

host : 仅允许本主机内部的连线;

所以当然是使用 global 罗! 预设也是 global 啦!

范例一: 显示出所有的介面之 IP 参数:

```
[root@linux ~]# ip address show
```

```
1: lo: <LOOPBACK,UP,10000> mtu 16436 qdisc noqueue
```

```
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

```
    inet 127.0.0.1/8 scope host lo
```

```
    inet6 ::1/128 scope host
```

```
        valid_lft forever preferred_lft forever
```

```
2: eth0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc pfifo_fast qlen 1000
```

```
    link/ether 00:50:fc:22:9a:cb brd ff:ff:ff:ff:ff:ff
```

```
    inet 192.168.1.2/24 brd 192.168.1.255 scope global eth0
```

```
    inet6 fe80::250:fcff:fe22:9acb/64 scope link
```

```
        valid_lft forever preferred_lft forever
```

```
3: sit0: <NOARP> mtu 1480 qdisc noop
```

```
    link/sit 0.0.0.0 brd 0.0.0.0
```

看到上面那个特殊的字体吗? 没错! 那就是 IP 参数啦! 也是 ip address 最主要的功能。

底下我们进一步来新增虚拟的网路介面试看看:

范例二: 新增一个介面, 名称假设为 eth0:vbird

```
[root@linux ~]# ip address add 192.168.50.50/24 broadcast + \
```

```
> dev eth0 label eth0:vbird

[root@linux ~]# ip address show eth0

2: eth0: mtu 1500 qdisc pfifo_fast qlen 1000

    link/ether 00:40:d0:13:c3:46 brd ff:ff:ff:ff:ff:ff

    inet 192.168.1.100/24 brd 192.168.1.255 scope global eth0

    inet 192.168.50.50/24 brd 192.168.50.255 scope global eth0:vbird

    inet6 fe80::240:d0ff:fe13:c346/64 scope link

        valid_lft forever preferred_lft forever

# 看到上面的特殊字体了吧？多出了一行新的介面，且名称是 eth0:vbird

# 至於那个 broadcast + 也可以写成 broadcast 192.168.50.255 啦！

[root@linux ~]# ifconfig

eth0:vbir Link encap:Ethernet HWaddr 00:40:D0:13:C3:46

    inet addr:192.168.50.50 Bcast:192.168.50.255 Mask:255.255.25
5.0

    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

    Interrupt:5 Base address:0x3e00

# 如果使用 ifconfig 就能够看到这个怪东西了！
```

范例三：将刚刚的介面删除

```
[root@linux ~]# ip address del 192.168.50.50/24 dev eth0
```

5.5.6 關於路由的相关设定：ip route

ip route 的功能几乎与 route 这个指令差不多，但是，他还可以进行额外的参数设计，例如 MTU 的规划等等，相当的强悍啊！

```
[root@linux ~]# ip route show <==单纯的显示出路由的设定而已
```

```
[root@linux ~]# ip route [add|del] [IP 或网域] [via gateway] [dev 装置]
```

参数：

show ：单纯的显示出路由表，也可以使用 list ；

add|del ：增加（add）或删除（del）路由的意思。

IP 或网域：可使用 192.168.50.0/24 之类的网域或者是单纯的 IP ；

via ：从那个 gateway 出去，不一定需要；

dev ：由那个装置连出去，这就需要了！

mtu ：可以额外的设定 MTU 的数值喔！

范例一：显示出目前的路由资料

```
[root@linux ~]# ip route show
```

```
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.2
```

```
169.254.0.0/16 dev eth1 scope link
```

```
default via 192.168.1.254 dev eth1
```

如上表所示，最简单的功能就是显示出目前的路由资讯，其实跟 route 这个指令相同啦！
指示必须要注意几个小东西：

proto：此路由的路由协定，主要有 redirect, kernel, boot, static, ra 等，其中 kernel 指的是直接由核心判断自动设定。

scope：路由的范围，主要是 link ，亦即是与本装置有关的直接连线。

再来看一下如何进行路由的增加与删除吧！

范例二：增加路由，主要是本机直接可沟通的网域

```
[root@linux ~]# ip route add 192.168.5.0/24 dev eth0
```

针对本机直接沟通的网域设定好路由，不需要透过外部的路由器

```
[root@linux ~]# ip route show
```

```
192.168.5.0/24 dev eth0 scope link
```

.... 以下省略....

范例三：增加可以通往外部的路由，需透过 router 喔！

```
[root@linux ~]# ip route add 192.168.10.0/24 via 192.168.5.100 dev eth0
```

```
[root@linux ~]# ip route show
```

```
192.168.5.0/24 dev eth0 scope link
```

.... 其他省略....

```
192.168.10.0/24 via 192.168.5.100 dev eth0
```

仔细看喔，因为我有 192.168.5.0/24 的路由存在（我的网卡直接联系），

所以才可以将 192.168.10.0/24 的路由丢给 192.168.5.100

那部主机来帮忙传递喔！与之前提到的 route 指令是一样的限制！

范例四：增加预设路由

```
[root@linux ~]# ip route add default via 192.168.1.2 dev eth0
```

那个 192.168.1.2 就是我的预设路由器（gateway）的意思啊！ ^_^

真的记得，只要一个预设路由就 OK ！

范例五：删除路由

```
[root@linux ~]# ip route del 192.168.10.0/24
```

```
[root@linux ~]# ip route del 192.168.5.0/24
```

5.6 netstat

5.6.1 功能说明

显示网络状态。

5.6.2 语法

```
netstat [-acCeFghilMnNoprstuvVwx] [-A<网络类型>] [--ip]
```


5.6.3 补充说明

利用 netstat 指令可让你得知整个 Linux 系统的网络情况。

5.6.4 参数

- a 或--all 显示所有连线中的 Socket。
- A<网络类型>或--<网络类型> 列出该网络类型连线中的相关地址。
- c 或--continuous 持续列出网络状态。
- C 或--cache 显示路由器配置的快取信息。
- e 或--extend 显示网络其他相关信息。
- F 或--fib 显示 FIB。
- g 或--groups 显示多重广播功能群组组员名单。
- h 或--help 在线帮助。
- 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 显示伪装的网络连线。
- n 或--numeric 直接使用 IP 地址，而不通过域名服务器。
- N 或--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--statistic 显示网络工作信息统计表。
- t 或--tcp 显示 TCP 传输协议的连线状况。
- u 或--udp 显示 UDP 传输协议的连线状况。
- v 或--verbose 显示指令执行过程。

- V 或--version 显示版本信息。
- w 或--raw 显示 RAW 传输协议的连线状况。
- 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。

5.7 nslookup

5.7.1 功能说明

nslookup 命令的功能是查询一台机器的 IP 地址和其对应的域名。使用权限所有用户。它通常需要一台域名服务器来提供域名服务。如果用户已经设置好域名服务器，就可以用这个命令查看不同主机的 IP 地址对应的域名。

5.7.2 语法

nslookup [IP 地址/域名]

5.7.3 例子

(1) 在本地计算机上使用 nslookup 命令

```
$ nslookup
```

```
Default Server: name.cao.com.cn
```

```
Address: 192.168.1.9
```

```
>
```

在符号“>”后面输入要查询的 IP 地址域名，并回车即可。如果要退出该命令，输入“exit”，并回车即可。

(2) 使用 nslookup 命令测试 named

输入下面命令：

```
nslookup
```

然后就进入交互式 nslookup 环境。如果 named 正常启动，则 nslookup 会显示当前 DNS 服务器的地址和域名，否则表示 named 没能正常启动。

下面简单介绍一些基本的 DNS 诊断。

◆ 检查正向 DNS 解析，在 nslookup 提示符下输入带域名的主机名，如 hp712.my.com，nslookup 应能显示该主机名对应的 IP 地址。如果只输入 hp712，nslookup 会根据/etc/resolv.conf 的定义，自动添加 my.com 域名，并回答对应的 IP 地址。

◆检查反向 DNS 解析，在 nslookup 提示符下输入某个 IP 地址，如 192.22.33.20，nslookup 应能回答该 IP 地址所对应的主机名。

◆检查 MX 邮件地址记录在 nslookup 提示符下输入：

```
set q=mx
```

然后输入某个域名，输入 my.com 和 mail.my.com，nslookup 应能够回答对应的邮件服务器地址，即

support.my.com 和 support2.my.com。

◆检查 TXT 记录：

```
set type=txt
```

然后输入某个域名，nslookup 会显示对应的记录。

5.8 ping

5.8.1 功能说明

检测主机。

5.8.2 语法

```
ping [-dfnqrRv] [-c<完成次数>] [-i<间隔秒数>] [-I<网络界面>] [-l<前置载入>] [-p<范  
本样式>] [-s<数据包大小>] [-t<存活数值>] [主机名称或 IP 地址]
```

5.8.3 补充说明

执行 ping 指令会使用 ICMP 传输协议，发出要求回应的信息，若远端主机的网络功能没有问题，就会回应该信息，因而得知该主机运作正常。

5.8.4 参数

- d 使用 Socket 的 SO_DEBUG 功能。
- c<完成次数> 设置完成要求回应的次数。
- f 极限检测。
- i<间隔秒数> 指定收发信息的间隔时间。
- I<网络界面> 使用指定的网络界面送出数据包。
- l<前置载入> 设置在送出要求信息之前，先行发出的数据包。
- n 只输出数值。
- p<范本样式> 设置填满数据包的范本样式。
- q 不显示指令执行过程，开头和结尾的相关信息除外。
- r 忽略普通的 Routing Table，直接将数据包送到远端主机上。
- R 记录路由过程。
- s<数据包大小> 设置数据包的大小。
- t<存活数值> 设置存活数值 TTL 的大小。
- v 详细显示指令的执行过程。

5.9 rcp

5.9.1 功能说明

远端复制文件或目录。

5.9.2 语法

rcp [-pr][源文件或目录][目标文件或目录] 或 rcp [-pr][源文件或目录...][目标文件]

5.9.3 补充说明

rcp 指令用在远端复制文件或目录，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，则它灰把前面指定的所有文件或目录复制到该目录中。

5.9.4 参数

- p 保留源文件或目录的属性，包括拥有者，所属群组，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。

5.10 route

5.10.1 功能说明

route 表示手工产生、修改和查看路由表。

5.10.2 语法

```
#route [-add][-net|-host] targetaddress [-netmask Nm][dev]If]
```

```
#route [-delete][-net|-host] targetaddress [gw Gw] [-netmask Nm] [dev]If]
```

5.10.3 参数

- add: 增加路由。
- delete: 删除路由。
- net: 路由到达的是一个网络，而不是一台主机。
- host: 路由到达的是一台主机。
- netmask Nm: 指定路由的子网掩码。
- gw: 指定路由的网关。

[dev]If: 强迫路由链指定接口。

5.10.4 例子

```
route add -net 202.96.96.0 -netmask 255.255.255.0 gw 192.168.1.10 dev eth0
```

表示到 202.96.96.0/255.255.255.0 这个网络的信息通过你的第一块网卡发送，默认网关的地址是 192.168.1.10

5.11 tcpdump

5.11.1 功能说明

倾倒网络传输数据。

5.11.2 语法

```
tcpdump [-adeflnNOpqStvx][-c<数据包数目>][-dd][-ddd][-F<表达文件>][-i<网络界面>][-r<数据包文件>][-s<数据包大小>][-tt][-T<数据包类型>][-vv][-w<数据包文件>]  
[输出数据栏位]
```

5.11.3 补充说明

执行 tcpdump 指令可列出经过指定网络界面的数据包文件头，在 Linux 操作系统中，你必须是系统管理员。

5.11.4 参数

- a 尝试将网络和广播地址转换成名称。
- c<数据包数目> 收到指定的数据包数目后，就停止进行倾倒操作。
- d 把编译过的数据包编码转换成可阅读的格式，并倾倒到标准输出。
- dd 把编译过的数据包编码转换成 C 语言的格式，并倾倒到标准输出。
- ddd 把编译过的数据包编码转换成十进制数字的格式，并倾倒到标准输出。
- e 在每列倾倒资料上显示连接层级的文件头。

- f 用数字显示网际网络地址。
- F<表达文件> 指定内含表达方式的文件。
- i<网络界面> 使用指定的网络截面送出数据包。
- l 使用标准输出列的缓冲区。
- n 不把主机的网络地址转换成名字。
- N 不列出域名。
- O 不将数据包编码最佳化。
- p 不让网络界面进入混杂模式。
- q 快速输出，仅列出少数的传输协议信息。
- r<数据包文件> 从指定的文件读取数据包数据。
- s<数据包大小> 设置每个数据包的大小。
- S 用绝对而非相对数值列出 TCP 关联数。
- t 在每列倾倒资料上不显示时间戳记。
- tt 在每列倾倒资料上显示未经格式化的时间戳记。
- T<数据包类型> 强制将表达方式所指定的数据包转译成设置的数据包类型。
- v 详细显示指令执行过程。
- vv 更详细显示指令执行过程。
- x 用十六进制字码列出数据包资料。
- w<数据包文件> 把数据包数据写入指定的文件。

5.12 telnet

5.12.1 功能说明

远端登入。

5.12.2 语法

telnet [-8acdEfFKLrx][-b<主机别名>][-e<脱离字符>][-k<域名>][-l<用户名称>][-n<记录文件>][-S<服务类型>][-X<认证形态>][主机名称或 IP 地址<通信端口>]

5.12.3 补充说明

执行 telnet 指令开启终端机阶段作业，并登入远端主机。

5.12.4 参数

- 8 允许使用 8 位字符资料，包括输入与输出。
- a 尝试自动登入远端系统。
- b<主机别名> 使用别名指定远端主机名称。
- c 不读取用户专属目录里的 .telnetrc 文件。
- d 启动排错模式。
- e<脱离字符> 设置脱离字符。
- 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到远端主机。
- k<域名> 使用 Kerberos 认证时，加上此参数让远端主机采用指定的领域名，而非该主机的域名。
- K 不自动登入远端主机。
- l<用户名称> 指定要登入远端主机的用户名称。
- L 允许输出 8 位字符资料。
- n<记录文件> 指定文件记录相关信息。
- r 使用类似 rlogin 指令的用户界面。

-S<服务类型> 设置 telnet 连线所需的 IP TOS 信息。

-x 假设主机有支持数据加密的功能，就使用它。

-X<认证形态> 关闭指定的认证形态。

5.13 traceroute

5.13.1 功能说明

显示数据包到主机间的路径。

5.13.2 语法

```
traceroute [-dFlrvx] [-f<存活数值>] [-g<网关>...] [-i<网络界面>] [-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] [数据包大小]
```

5.13.3 补充说明

traceroute 指令让你追踪网络数据包的路由途径，预设数据包大小是 40Bytes，用户可另行设置。

5.13.4 参数

-d 使用 Socket 层级的排错功能。

-f<存活数值> 设置第一个检测数据包的存活数值 TTL 的大小。

-F 设置勿离断位。

-g<网关> 设置来源路由网关，最多可设置 8 个。

-i<网络界面> 使用指定的网络界面送出数据包。

-I 使用 ICMP 回应取代 UDP 资料信息。

-m<存活数值> 设置检测数据包的最大存活数值 TTL 的大小。

-n 直接使用 IP 地址而非主机名称。

- p<通信端口> 设置 UDP 传输协议的通信端口。
- r 忽略普通的 Routing Table，直接将数据包送到远端主机上。
- s<来源地址> 设置本地主机送出数据包的 IP 地址。
- t<服务类型> 设置检测数据包的 TOS 数值。
- v 详细显示指令的执行过程。
- w<超时秒数> 设置等待远端主机回报的时间。
- x 开启或关闭数据包的正确性检验。

6 系统管理

6.1 cal

6.1.1 功能说明

日历命令，可以显示当月日历、当年日历、指定月份年份的日历等。

6.1.2 语法

```
cal [options] [[month] year]
```

6.1.3 参数

-j

以一月一日起的天数显示本月的月历。

-m

以星期一为每周的第一天方式，显示本月的月历。

-y

显示全年日历

-jy

以一月一日起的天数显示全年日历。

6.2 cat

使用方式: `cat [-AbeEnstTuv] [--help] [--version] filename`

说明: 把档案串连接后传到基本输出 (屏幕或加 `> 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 里。

6.3 chsh

6.3.1 功能说明

更换登入系统时使用的 shell。

6.3.2 语法

`chsh [-luv] [-s <shell 名称>] [用户名称]`

6.3.3 补充说明

每位用户在登入系统时, 都会拥有预设的 shell 环境, 这个指令可更改其预设值。若不指定

任何参数与用户名称，则 chsh 会以应答的方式进行设置。

6.3.4 参数

-s<shell 名称>或--shell<shell 名称> 更改系统预设的 shell 环境。

-l 或--list-shells 列出目前系统可用的 shell 清单。

-u 或--help 在线帮助。

-v 或--version 显示版本信息。

6.4 clear

清除终端显示，与按 Ctrl-L 效果相同。

6.5 date

6.5.1 功能说明

显示或设置系统时间与日期。

6.5.2 语法

date [-d <字符串>][-u][+%H%I%K%l%M%P%r%s%S%T%X%Z%a%A%b%B%c%d%D%j%m%U%w%x%y%Y%n%t]

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

6.5.3 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 会显示目前的日期与时间。

6.5.4 参数

%H 小时(以 00-23 来表示)。

%I 小时(以 01-12 来表示)。

%K 小时(以 0-23 来表示)。

%l 小时(以 0-12 来表示)。

%M 分钟(以 00-59 来表示)。

%P AM 或 PM。

%r 时间(含时分秒, 小时以 12 小时 AM/PM 来表示)。

%s 总秒数。起算时间为 1970-01-01 00:00:00 UTC。

%S 秒(以本地的惯用法来表示)。

%T 时间(含时分秒, 小时以 24 小时制来表示)。

%X 时间(以本地的惯用法来表示)。

%Z 市区。

%a 星期的缩写。

%A 星期的完整名称。

%b 月份英文名的缩写。

%B 月份的完整英文名称。

%c 日期与时间。只输入 date 指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)。

%j 该年中的第几天。

%m 月份(以 01-12 来表示)。

%U 该年中的周数。

%w 该周的天数, 0 代表周日, 1 代表周一, 异词类推。

%x 日期(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)。

%n 在显示时, 插入新的一行。

%t 在显示时, 插入 tab。

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> 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。

-s<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。

-u 显示 GMT。

--help 在线帮助。

--version 显示版本信息。

6.6 echo

6.6.1 功能说明

显示文字。

6.6.2 语法

echo [-ne][字符串]或 echo [--help][--version]

6.6.3 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符隔开，并在最后加上换行号。

6.6.4 参数

-n 不要在最后自动换行

-e 若字符串中出现以下字符，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 换行且光标移至行首；

\r 光标移至行首，但不换行；

\t 插入 tab；

\v 与\f 相同；

\\ 插入\字符；

\nnn 插入 nn (八进制) 所代表的 ASCII 字符；

-help 显示帮助

-version 显示版本信息

6.7 ethtool

6.7.1 功能说明

ethtool 是用于查询及设置网卡参数的命令。

6.7.2 概要

```

ethtool ethX          //查询 ethX 网口基本设置

ethtool -h             //显示 ethtool 的命令帮助(help)

ethtool -i ethX        //查询 ethX 网口的相关信息

ethtool -d ethX        //查询 ethX 网口注册性信息

ethtool -r ethX        //重置 ethX 网口到自适应模式

ethtool -S ethX        //查询 ethX 网口收发包统计

ethtool -s ethX [speed 10|100|1000]\           //设置网口速率 10/100/100
OM

[duplex half|full]\           //设置网口半/全双工

[autoneg on|off]\           //设置网口是否自协商

[port tp|au|bnc|mii]\           //设置网口类型

[phyad N]\

[xcvr internal|external]\

[wol p|u|m|b|a|g|s|d...]\

[sopass xx:yy:zz:aa:bb:cc]\

[msglvl N]

```

6.7.3 举例

1) [root@linux /]# ethtool eth1

Settings for eth1:

Supported ports: [TP]

Supported link modes: 10baseT/Half 10baseT/Full

100baseT/Half 100b

aseT/Full

1000baseT/Full

Supports auto-negotiation: Yes

Advertised link modes: 10baseT/Half 10baseT/Full

100baseT/Half 100b

aseT/Full

1000baseT/Full

Advertised auto-negotiation: Yes

Speed: 1000Mb/s

Duplex: Full

Port: Twisted Pair

PHYAD: 0

Transceiver: internal

Auto-negotiation: on

Supports Wake-on: umbg

Wake-on: d

Link detected: yes

2) [root@linux /]# ethtool -i eth1

driver: e1000

version: 5.0.43-k1

firmware-version: N/A

bus-info: 06:08.1

3) [root@linux /]# ethtool -S eth1

NIC statistics:

rx_packets: 58068300

tx_packets: 87124083

```
rx_bytes: 1589713008
tx_bytes: 2165825901
rx_errors: 0
tx_errors: 0
rx_dropped: 0
tx_dropped: 0
multicast: 0
collisions: 0
rx_length_errors: 0
rx_over_errors: 0
rx_crc_errors: 0
rx_frame_errors: 0
rx_fifo_errors: 0
rx_missed_errors: 0
tx_aborted_errors: 0
tx_carrier_errors: 0
tx_fifo_errors: 0
tx_heartbeat_errors: 0
tx_window_errors: 0
```

```
4) [root@linux ~]# ethtool -s eth1 autoneg off speed 100 duplex full
```

6.7.4 相关

```
1) [root@linux ~]# which ethtool
```

`/sbin/ethtool`

2) `[root@linux ~]# rpm -qf /sbin/ethtool`

`ethtool-1.6-5`

3) 怎样使 `ethtool` 设置永久保存在网络设备中?

解决方法一:

`ethtool` 设置可通过 `/etc/sysconfig/network-scripts/ifcfg-ethX` 文件保存, 从而在设备下次启动时激活选项。

例如: `ethtool -s eth0 speed 100 duplex full autoneg off`

此指令将 `eth0` 设备设置为全双工自适应, 速度为 100Mbps。若要 `eth0` 启动时设置这些参数, 修改文件 `/etc/sysconfig/network-scripts/ifcfg-eth0` , 添加如下行:

`ETHTOOL_OPTS="speed 100 duplex full autoneg off"`

解决方法二:

将 `ethtool` 设置写入 `/etc/rc.d/rc.local` 之中。

6.8 free

6.8.1 功能说明

显示内存状态。

6.8.2 语法

`free [-bkmotV] [-s <间隔秒数>]`

6.8.3 补充说明

`free` 指令会显示内存的使用情况, 包括实体内存, 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。

6.8.4 参数

- b 以 Byte 为单位显示内存使用情况。
- k 以 KB 为单位显示内存使用情况。
- m 以 MB 为单位显示内存使用情况。
- o 不显示缓冲区调节列。
- s<间隔秒数> 持续观察内存使用状况。
- t 显示内存总和列。
- V 显示版本信息。

6.9 getconf

6.9.1 功能说明

将系统配置变量值写入标准输出。

6.9.2 语法

```
getconf [ -v specification ] [ SystemwideConfiguration | PathConfiguration Path  
name ] [ DeviceVariable Devicename ]
```

或 `getconf -a`

6.9.3 描述

用 `SystemwideConfiguration` 参数调用 `getconf` 命令，将 `SystemwideConfiguration` 参数所指定的变量值写到标准输出。

用 `PathConfiguration` 和 `Pathname` 参数调用 `getconf` 命令，写入 `PathConfiguration` 参数为 `Pathname` 参数指定路径指定的变量值，到标准输出。

`getconf` 命令，用 `-a` 标志调用，并写入全部系统配置变量值到标准输出。

`getconf` 命令，用 `DeviceVariable` 和 `Devicename` 参数调用，写磁盘设备名或位置的值到标准输出，设备路径由 `Devicename` 参数指定。

如果在系统中定义指定的变量且其值描述为可从 `confstr` 子例程得到，则指定变量的值按以下格式写入：

“%s\n”， < 值 >

否则，如果在系统中定义指定变量，其值按以下格式写入：

“%d\n”， < 值 >

如果指定变量是有效的但在系统中未定义，则将以下内容写入标准输出：

“undefined\n”

如果变量名无效或出现错误，就会有一个诊断消息写入标准错误。

标记 `-a specification` 显示了指定规格及版本，其配置变量等待确定。如果该标志未被指定，返回值将响应一个实现缺省值 XBS5 的相应的编辑环境。

`-v` 将全部系统配置变量值写入标准输出。

6.9.4 参数

`Pathname` 为 `PathConfiguration` 参数指定路径名。

`SystemwideConfiguration` 指定一个系统配置变量。

`PathConfiguration` 指定一个系统路径配置变量。

`Devicename` 指定一个设备路径名。

`DeviceVariable` 指定一个设备变量。

当列入以下的表格中的第一列符号被用作 `system_var` 操作数时，`getconf` 将产生与用第二列的值调用 `confstr` 时相同的值。

6.10 groupadd

6.10.1 功能说明

添加群组。

6.10.2 语法

groupadd [群组名称]

6.11 groupdel

6.11.1 功能说明

删除群组。

6.11.2 语法

groupdel [群组名称]

6.11.3 补充说明

需要从系统上删除群组时，可用 groupdel 指令来完成这项工作。倘若该群组中仍包括某些用户，则必须先删除这些用户后，方能删除群组。

6.12 groupmod

6.12.1 功能说明

更改群组识别码或名称。

6.12.2 语法

groupmod [-g <群组识别码> [-o>][-n <新群组名称>][群组名称]

6.12.3 补充说明

需要更改群组的识别码或名称时，可用 groupmod 指令来完成这项工作。

6.12.4 参数

-g <群组识别码> 设置欲使用的群组识别码。

-o 重复使用群组识别码。

`-n <新群组名称>` 设置欲使用的群组名称。

6.13 id

6.13.1 功能说明

显示用户的 ID，以及所属群组的 ID。

6.13.2 语法

`id [-gGnru] [--help] [--version] [用户名称]`

6.13.3 补充说明

`id` 会显示用户以及所属群组的实际与有效 ID。若两个 ID 相同，则仅显示实际 ID。若仅指定用户名称，则显示目前用户的 ID。

6.13.4 参数

`-g` 或 `--group` 显示用户所属群组的 ID。

`-G` 或 `--groups` 显示用户所属附加群组的 ID。

`-n` 或 `--name` 显示用户，所属群组或附加群组的名称。

`-r` 或 `--real` 显示实际 ID。

`-u` 或 `--user` 显示用户 ID。

`-help` 显示帮助。

`-version` 显示版本信息。

6.14 init

6.14.1 功能说明

`init` 是 Linux 系统操作中不可缺少的程序之一。

所谓的 `init` 进程，它是一个由内核启动的用户级进程。

内核自行启动（已经被载入内存，开始运行，并已初始化所有的设备驱动程序和数据结构等）之后，就通过启动一个用户级程序 `init` 的方式，完成引导进程。所以，`init` 始终是第一个进程（其进程编号始终为 1）。

内核会在过去曾使用过 `init` 的几个地方查找它，它的正确位置（对 Linux 系统来说）是 `/sbin/init`。如果内核找不到 `init`，它就会试着运行 `/bin/sh`，如果运行失败，系统的启动也会失败。

6.14.2 语法

```
init [bootflags] [runlevel]
```

6.14.3 参数

6.14.3.1 Boot flags

- s 切换到单用户模式

q 重新加载 `inittab` 文件

a b c 只载入在 `runlevel` 中有 a b c 的程序。

u 重建自身，并保留状态

6.14.3.2 Runlevels

（可以参考 Red Hat Linux 里面的 `/etc/inittab`）

0 - 停机（千万不能把 `initdefault` 设置为 0）

1 - 单用户模式

2 - 多用户，没有 NFS

3 - 完全多用户模式（标准的运行级）

4 - 没有用到

5 - X11（xwindow）

6 - 重新启动（千万不要把 `initdefault` 设置为 6）

6.15 ipcs

6.15.1 功能说明

显示进程间通信信息。

6.15.2 语法

ipcs [-asmq] [-tclup]

ipcs [-smq] -i id

ipcs -h

6.15.3 参数

- m 显示共享内存
- q 显示消息队列
- s 显示信号灯
- a 显示所有信息，默认为该选项

6.15.4 例子

\$ipcs #显示所有信息，包括消息队列、共享内存、信号灯，与命令 ipcs -a 等价

\$ ipcs

----- Shared Memory Segments -----

key	shmid	owner	perms	byte
s	nattch	status		

----- Semaphore Arrays -----

key	semid	owner	perms	nsem
s				

----- Message Queues -----

key	msqid	owner	perms	used-byte
s	messages			

\$ipcs -m #显示共享内存信息

\$ ipcs -m

----- Shared Memory Segments -----

key	shmid	owner	perms	byte
s	nattch	status		

\$ipcs -s #显示信号灯信息

\$ ipcs -s

----- Semaphore Arrays -----

key	semid	owner	perms	nsem
s				

\$ipcs -q #显示消息队列信息

\$ ipcs -q

----- Message Queues -----

key	msqid	owner	perms	used-byte
s	messages			

6.16 mesg

6.16.1 功能说明

设置终端机的写入权限。

6.16.2 语法

`mesg [ny]`

6.16.3 补充说明

将 `mesg` 设置 `y` 时，其他用户可利用 `write` 指令将信息直接显示在您的屏幕上。

6.16.4 参数

`n` 不允许将信息直接显示在你的屏幕上。

`y` 允许将信息直接显示在你的屏幕上。

6.17 passwd

6.17.1 功能说明

设置密码。

6.17.2 语法

`passwd [-dklS][-u <-f>][用户名称]`

6.17.3 补充说明

`passwd` 指令让用户可以更改自己的密码，而系统管理者则能用它管理系统用户的密码。只有管理者可以指定用户名称，一般用户只能变更自己的密码。

6.17.4 参数

`-d` 删除密码。本参数仅有系统管理者才能使用。

`-f` 强制执行。

`-k` 设置只有在密码过期失效后，方能更新。

`-l` 锁住密码。

-s 列出密码的相关信息。本参数仅有系统管理者才能使用。

-u 解开已上锁的帐号。

6.18 rpm

6.18.1 功能说明

管理套件。

6.18.2 语法

rpm [-acdhilqRsv] [-b<完成阶段><套件档>+] [-e<套件档>] [-f<文件>+] [-i<套件档>] [-p<套件档>+]

[-U<套件档>] [-vv] [--addsign<套件档>+] [--allfiles] [--allmatches] [--badreloc] [--buildroot<根目录>]

[--changelog] [--checksig<套件档>+] [--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>] [--http port<通信端口>] [--ignorearch] [--ignoreos]

[--ignoresize] [--includedocs] [--initdb] [justdb] [--nobulid] [--nodeps] [--nofiles] [--nogpg] [--nomd5] [--nogpg]

[--noorder] [--noscripts] [--notriggers] [--oldpackage] [--percent] [--pipe<执行指令>] [--prefix<目的目录>]

[--provides] [--queryformat<档头格式>] [--querytags] [--rcfile<配置档>] [--rebulid<套件档>] [--rebulidb]

[--recompile<套件档>] [--relocate<原目录>=<新目录>] [--replacefiles] [--replacepkgs] [--requires]

[--resign<套件档>+] [--rmsource] [--rmsource<文件>] [--root<根目录>] [--scripts] [--setperms] [--setugids]

[--short-circuit][--sign][--target=<安装平台>+][--test][--timecheck<检查秒数>]
[--triggeredby<套件档>]

[--triggers][--verify][--version][--whatprovides<功能特性>][--whatrequires<功能特性>]

6.18.3 补充说明

rpm 原本是 Red Hat Linux 发行版专门用来管理 Linux 各项套件的程序，由于它遵循 GPL 规则且功能强大方便，因而广受欢迎。逐渐受到其他发行版的采用。RPM 套件管理方式的出现，让 Linux 易于安装，升级，间接提升了 Linux 的适用度。

6.18.4 参数

-a 查询所有套件。

-b<完成阶段><套件档>+或-t <完成阶段><套件档>+ 设置包装套件的完成阶段，并指定套件档的文件名称。

-c 只列出组态配置文件，本参数需配合“-l”参数使用。

-d 只列出文本文件，本参数需配合“-l”参数使用。

-e<套件档>或--erase<套件档> 删除指定的套件。

-f<文件>+ 查询拥有指定文件的套件。

-h 或--hash 套件安装时列出标记。

-i 显示套件的相关信息。

-i<套件档>或--install<套件档> 安装指定的套件档。

-l 显示套件的文件列表。

-p<套件档>+ 查询指定的 RPM 套件档。

-q 使用询问模式，当遇到任何问题时，rpm 指令会先询问用户。

-R 显示套件的关联性信息。

-s 显示文件状态，本参数需配合“-l”参数使用。

-U<套件档>或--upgrade<套件档> 升级指定的套件档。

-v 显示指令执行过程。

-vv 详细显示指令执行过程，便于排错。

-addsign<套件档>+ 在指定的套件里加上新的签名认证。

--allfiles 安装所有文件。

--allmatches 删除符合指定的套件所包含的文件。

--badreloc 发生错误时，重新配置文件。

--buildroot<根目录> 设置产生套件时，欲当作根目录的目录。

--changelog 显示套件的更改记录。

--checksig<套件档>+ 检验该套件的签名认证。

--clean 完成套件的包装后，删除包装过程中所建立的目录。

--dbpath<数据库目录> 设置欲存放 RPM 数据库的目录。

--dump 显示每个文件的验证信息。本参数需配合“-l”参数使用。

--excludedocs 安装套件时，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 强行置换套件或文件。

--ftp proxy<主机名称或 IP 地址> 指定 FTP 代理服务器。

--ftpport<通信端口> 设置 FTP 服务器或代理服务器使用的通信端口。

--help 在线帮助。

--http proxy<主机名称或 IP 地址> 指定 HTTP 代理服务器。

--httpport<通信端口> 设置 HTTP 服务器或代理服务器使用的通信端口。

--ignorearch 不验证套件档的结构正确性。

--ignoreos 不验证套件档的结构正确性。

--ignoresize 安装前不检查磁盘空间是否足够。

--includedocs 安装套件时，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，当不变动任何文件。

--nobulid 不执行任何完成阶段。

--nodeps 不验证套件档的相互关联性。

--nofiles 不验证文件的属性。

--nogpg 略过所有 GPG 的签名认证。

--nomd5 不使用 MD5 编码演算确认文件的大小与正确性。

--nopgp 略过所有 PGP 的签名认证。

--noorder 不重新编排套件的安装顺序，以便满足其彼此间的关联性。

--noscripts 不执行任何安装 Script 文件。

--notriggers 不执行该套件包装内的任何 Script 文件。

--oldpackage 升级成旧版本的套件。

--percent 安装套件时显示完成度百分比。

--pipe<执行指令> 建立管道，把输出结果转为该执行指令的输入数据。

--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 查询该套件所提供的兼容度。

--queryformat<档头格式> 设置档头的表示方式。

--querytags 列出可用于档头格式的标签。

--rcfile<配置文件> 使用指定的配置文件。

--rebulid<套件档> 安装原始代码套件，重新产生二进制文件的套件。

--rebulidb 以现有的数据库为主，重建一份数据库。

--recompile<套件档> 此参数的效果和指定"--rebulid"参数类似，当不产生套件档。

--relocate<原目录>=<新目录> 把本来会放到原目录下的文件改放到新目录。

--replacefiles 强行置换文件。

`--replacepkgs` 强行置换套件。

`--requires` 查询该套件所需要的兼容度。

`--resing<套件档>+` 删除现有认证，重新产生签名认证。

`--rmsource` 完成套件的包装后，删除原始代码。

`--rmsource<文件>` 删除原始代码和指定的文件。

`--root<根目录>` 设置欲当作根目录的目录。

`--scripts` 列出安装套件的 Script 的变量。

`--setperms` 设置文件的权限。

`--setugids` 设置文件的拥有者和所属群组。

`--short-circuit` 直接略过指定完成阶段的步骤。

`--sign` 产生 PGP 或 GPG 的签名认证。

`--target=<安装平台>+` 设置产生的套件的安装平台。

`--test` 仅作测试，并不真的安装套件。

`--timecheck<检查秒数>` 设置检查时间的计时秒数。

`--triggeredby<套件档>` 查询该套件的包装者。

`--triggers` 展示套件档内的包装 Script。

`--verify` 此参数的效果和指定“-q”参数相同。

`--version` 显示版本信息。

`--whatprovides<功能特性>` 查询该套件对指定的功能特性所提供的兼容度。

`--whatrequires<功能特性>` 查询该套件对指定的功能特性所需要的兼容度。

6.19 service

6.19.1 功能说明

启动、停止、重启服务

6.19.2 语法

`service [service name] [start|stop|restart]`

6.19.3 参数说明

`service name`: 这个服务名称必须要实际存在 `/etc/rc.d/init.d` 这个目录下才行

`start|stop|restart`: 基本上, 这个与 `/etc/rc.d/init.d/service_name` 所允许的启动形式有关.

6.20 setup

6.20.1 功能说明

`setup` 是一个设置公用程序, 提供图形界面的操作方式。在 `setup` 中可设置 7 类的选项:

1. 登陆认证方式
2. 键盘组态设置
3. 鼠标组态设置
4. 开机时所要启动的系统服务
5. 声卡组态设置
6. 时区设置
7. X Windows 组态设置

6.20.2 语法

`setup`

6.21 shutdown

6.21.1 功能说明

shutdown 可以用来进行关机程序,并且在关机以前传送讯息给所有使用者正在执行的程序,shutdown 也可以用来重开机。

6.21.2 语法

```
shutdown [-t seconds] [-rkhncfF] time [message]
```

6.21.3 参数

-t seconds : 设定在几秒钟之后进行关机程序

-k : 并不会真的关机,只是将警告讯息传送给所有只用户

-r : 关机后重新开机

-h : 关机后停机

-n : 不采用正常程序来关机,用强迫的方式杀掉所有执行中的程序后自行关机

-c : 取消目前已经进行中的关机动作

-f : 关机时,不做 fck 动作(检查 Linux 档系统)

-F : 关机时,强迫进行 fsck 动作

time : 设定关机的时间

message : 传送给所有使用者的警告讯息

6.22 su

6.22.1 功能说明

变更用户身份。

6.22.2 语法

```
su [-flmp][--help][--version][-][-c <指令>][-s <shell>][用户帐号]
```

6.22.3 补充说明

su 可让用户暂时变更登入的身份。变更时须输入所要变更的用户帐号与密码。

6.22.4 参数

-c<指令>或--command=<指令> 执行完指定的指令后，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。
此外，也会变更 PATH 变量。

-m, -p 或--preserve-environment 变更身份时，不要变更环境变量。

-s<shell>或--shell=<shell> 指定要执行的 shell。

--help 显示帮助。

--version 显示版本信息。

[用户帐号] 指定要变更的用户。若不指定此参数，则预设变更为 root。

6.23 sync

6.23.1 功能说明

将内存缓冲区内的数据写入磁盘。

6.23.2 语法

sync [--help][--version]

6.23.3 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。

6.23.4 参数

- `--help` 显示帮助。
- `--version` 显示版本信息。

6.24 uname

6.24.1 功能说明

显示系统信息。

6.24.2 语法

`uname [-amnrsv] [--help] [--version]`

6.24.3 补充说明

`uname` 可显示电脑以及操作系统的相关信息。

6.24.4 参数

- `-a` 或 `--all` 显示全部的信息。
- `-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编号。
- `-s` 或 `--sysname` 显示操作系统名称。
- `-v` 显示操作系统的版本。
- `--help` 显示帮助。
- `--version` 显示版本信息。

6.25 uptime

6.25.1 功能说明

uptime 命令是用来查询 linux 系统负载的，可以显示

- 1 当前时间
- 2 系统已运行时间。
- 3 在线用户数
- 4 最近 1 分钟、5 分钟、15 分钟系统的平均负载。

6.25.2 语法

```
uptime [option]
```

6.25.3 Option

-V

显示版本信息。

6.26 useradd

6.26.1 功能说明

建立用户帐号。

6.26.2 语法

```
useradd [-mMnr][-c <备注>][-d <登入目录>][-e <有效期限>][-f <缓冲天数>][-g <群组>][-G <群组>][-s <shell>][-u <uid>][用户帐号] 或 useradd -D [-b] [-e <有效期限>][-f <缓冲天数>][-g <群组>][-G <群组>][-s <shell>]
```

6.26.3 补充说明

useradd 可用来建立用户帐号。帐号建好之后，再用 passwd 设定帐号的密码。而可用 userdel 删除帐号。使用 useradd 指令所建立的帐号，实际上是保存在/etc/passwd 文本文件中。

6.26.4 参数

-c<备注> 加上备注文字。备注文字会保存在 passwd 的备注栏位中。

-d<登入目录> 指定用户登入时的启始目录。

-D 变更预设值。

-e<有效期限> 指定帐号的有效期限。

-f<缓冲天数> 指定在密码过期后多少天即关闭该帐号。

-g<群组> 指定用户所属的群组。

-G<群组> 指定用户所属的附加群组。

-m 自动建立用户的登入目录。

-M 不要自动建立用户的登入目录。

-n 取消建立以用户名称为名的群组。

-r 建立系统帐号。

-s<shell> 指定用户登入后所使用的 shell。

-u<uid> 指定用户 ID。

6.27 userdel

6.27.1 功能说明

删除用户帐号。

6.27.2 语法

userdel [-r][用户帐号]

6.27.3 补充说明

userdel 可删除用户帐号与相关的文件。若不加参数，则仅删除用户帐号，而不删除相关文件。

6.27.4 参数

`-f` 删除用户登入目录以及目录中所有文件。

6.28 usermod

6.28.1 功能说明

修改用户帐号。

6.28.2 语法

```
usermod [-LU][-c <备注>][-d <登入目录>][-e <有效期限>][-f <缓冲天数>][-g <群组>][-G <群组>][-l <帐号名称>][-s <shell>][-u <uid>][用户帐号]
```

6.28.3 补充说明

usermod 可用来修改用户帐号的各项设定。

6.28.4 参数

`-c<备注>` 修改用户帐号的备注文字。

`-d 登入目录` 修改用户登入时的目录。

`-e<有效期限>` 修改帐号的有效期限。

`-f<缓冲天数>` 修改在密码过期后多少天即关闭该帐号。

`-g<群组>` 修改用户所属的群组。

`-G<群组>` 修改用户所属的附加群组。

`-l<帐号名称>` 修改用户帐号名称。

`-L` 锁定用户密码，使密码无效。

`-s<shell>` 修改用户登入后所使用的 shell。

`-u<uid>` 修改用户 ID。

-U 解除密码锁定。

6.29 vmstat

6.29.1 功能说明

vmstat 是 Virtual Memory Statistics（虚拟内存统计）的缩写，可对操作系统的虚拟内存、进程、CPU 活动进行监视。它是对系统的整体情况进行统计，不足之处是无法对某个进程进行深入分析。

6.29.2 语法

vmstat [-V] [-n] [delay [count]]

6.29.3 参数

-V 表示打印出版本信息；

-n 表示在周期性循环输出时，输出的头部信息仅显示一次；

delay 是两次输出之间的延迟时间；

count 是指按照这个时间间隔统计的次数。

6.29.4 输出信息

类别	项目	含义	说明
Procs	r	等待执行的任务数	展示了正在执行和等待 CPU 资源的任务个数。当这个值超过了 CPU 数目，就会出现 CPU 瓶颈了
	b		
Memory	swpd	正在使用的 swap 大小单位 K	
	free	空闲的内存空间	
	buff	已使用的 buff 大小，对块设备的读写进行缓冲	
	cache	已使用的 cache 大小，文件系统的 cache	

	inact		
	active		
Swap	si	交换内存使用,由磁盘调入内存	
	so	交换内存使用,由内存调入磁盘	
IO	bi	从块设备读入的数据总量(读磁盘) (KB/s),	
	bo	写入到块设备的数据总量(写磁盘) (KB/s)	
System	in	每秒产生的中断次数	
	cs	每秒产生的上下文切换次数	上面这 2 个值越大, 会看到由内核消耗的 CPU 时间会越多
CPU	us	用户进程消耗的 CPU 时间百分比	us 的值比较高时, 说明用户进程消耗的 CPU 时间多, 但是如果长期超过 50% 的使用, 那么我们就该考虑优化程序算法或者进行加速了
	sy	内核进程消耗的 CPU 时间百分比	sy 的值高时, 说明系统内核消耗的 CPU 资源多, 这并不是良性的表现, 我们应该检查原因。
	id	空闲	
	wa	IO 等待消耗的 CPU 时间百分比	wa 的值高时, 说明 IO 等待比较严重, 这可能是由于磁盘大量作随机访问造成, 也有可能是磁盘的带宽出现瓶颈(块操作)。

6.30 wall

6.30.1 功能说明

传送信息。

6.30.2 语法

wall [公告信息]

6.30.3 补充说明

通过 wall 指令可将信息发送给每位同意接收公众信息的终端机用户，若不给予其信息内容，则 wall 指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机用户。

6.31 write

6.31.1 功能说明

传送信息。

6.31.2 语法

write [用户名称][终端机编号]

6.31.3 补充说明

通过 write 指令可传递信息给另一位登入系统的用户，当输入完毕后，键入 EOF 表示信息结束，write 指令就会将信息传给对方。如果接收信息的用户不只登入本地主机一次，你可以指定接收信息的终端机编号。

7 更多参考

《Linux in a Nutshell, 5th Edition》<http://www.oreillynet.com/linux/cmd/>

《Linux Command Reference》http://www.perpetualpc.net/srtd_commands_rev.html