# 一、Linux系统目录

/bin：

存放着最经常使用的命令。

/boot：ca

存放启动Linux时使用的一些核心文件，包括一些连接文件以及镜像文件。

/dev ：

存放Linux的外部设备，在Linux中访问设备的方式和访问文件的方式是相同的。

/etc：

存放所有的系统管理所需要的配置文件和子目录。

/home：

用户的主目录，在Linux中，每个用户都有一个自己的目录。

/lib：

存放着系统最基本的动态连接共享库，其作用类似于Windows里的DLL文件。

几乎所有的应用程序都需要用到这些共享库。

/media：

linux系统会自动识别一些设备，

例如U盘、光驱等等，当识别后，linux会把识别的设备挂载到这个目录下。

/mnt：

该目录是为了让用户临时挂载别的文件系统的，

我们可以将光驱挂载在/mnt/上，然后进入该目录就可以查看光驱里的内容了。

/opt：

给主机额外安装软件所摆放的目录。默认是空的。

/proc：

一个虚拟的目录，系统内存的映射，可以通过直接访问这目录来获取系统信息。

这个目录的内容是在内存里，我们可以直接修改里面的某些文件，

比如可以通过下面的命令来屏蔽主机的ping命令，使别人无法ping你的机器：

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

/root：

该目录为系统管理员，也称作超级权限者的用户主目录。

/run：

一个临时文件系统，存储系统启动以来的信息。

当系统重启时，这个目录下的文件应该被删掉或清除。

如果你的系统上有 /var/run 目录，应该让它指向 run。

/sbin：

存放的是系统管理员使用的系统管理程序。

/srv：

存放一些服务启动之后需要提取的数据。

/sys：

该目录下安装了2.6内核中新出现的一个文件系统 sysfs 。

sysfs文件系统集成了下面3种文件系统的信息：

针对进程信息的proc文件系统、针对设备的devfs文件系统以及针对伪终端的devpts文件系统。

该文件系统是内核设备树的一个直观反映。

当一个内核对象被创建的时候，对应的文件和目录也在内核对象子系统中被创建。

/tmp：

用来存放一些临时文件的。

/usr：

用户的很多应用程序和文件都放在这个目录下。

/usr/bin：

系统用户使用的应用程序。

/usr/sbin：

超级用户使用的比较高级的管理程序和系统守护程序。

/usr/src：

内核源代码默认的放置目录。

/var：

存放着在不断扩充着的东西。包括各种日志文件/var/log。

# 二、文件管理

## 1.处理文件目录

ls（列出目录）

-a ：全部的文件，包括隐藏档(开头为.的文件)

-lh ：长数据串列出，包含文件的属性与权限等等数据

-d ：仅列出目录本身，而不是列出目录内的文件数据

cd 切换目录

pwd 用于显示目前的工作目录的绝对路径

mkdir 用于建立名称为dirName至子目录

mkdir [-p] dirName

-p ：确保目录名称存在，不存在就创建

touch 修改文件或者目录的时间属性，若文件不存在，创建

-a 改变档案的读取时间记录

-m 改变档案的修改时间记录

-d 设定时间与日期

-t 设定档案的时间记录

rmdir 删除空目录

-p ：连同上一级[空的]目录也一起删除

rm 移除文件或目录

-f ：强制force，忽略不存在的文件，不会出现告警信息

-i ：互动模式，在删除前会询问使用者是否动作

-r ：递归删除

cp 复制文件或目录

-a ：在复制目录时使用，它保留链接、文件属性，并复制目录下的所有内容

-d ：复制时保留链接。这里所说的链接相当于Windows系统中的快捷方式。

-f ：强制(force)，若目标文件已经存在且无法开启，则移除后再尝试一次；

-i ：若目标档(destination)已经存在时，在覆盖时会先询问动作的进行(常用)

-p ：连同文件的属性一起复制过去，而非使用默认属性(备份常用)；

-r ：递归持续复制，用於目录的复制行为；(常用)

-l ：进行硬式连结(hard link)的连结档创建，而非复制文件本身；

scp命令用于 Linux 之间复制文件和目录。

scp [[user@]host1:]file1 [IP] [[user@]host2:]file2

mv [文件名] [文件名] 移动文件或目录，或 修改文件与目录的名称

-f ：强制force，如果目标文件已经存在，不会询问而直接覆盖

-i ：若目标文件存在时，询问是否覆盖

## 2.查看文件内容

cat [参数] [文件] 用于连接文件并打印到标准输出设备上

-n ：列出行号，空白行标行号；

-b ：列出行号，空白行不标行号；

tac 由最后一行开始显示

cut [参数] [文件] 用于每行从开头算起 num1 到 num2 的文字

命令从文件的每一行剪切字节、字符和字段并将这些字节、字符和字段写至标准输出

-b ：以字节为单位进行分割。字节位置将忽略多字节字符边界

-c ：以字符为单位进行分割。

-d ：自定义分隔符，默认为制表符。

-f ：与-d一起使用，指定显示哪个区域。

-n ：取消分割多字节字符。仅和 -b 标志一起使用。如果字符的最后一个字节落在由 -b 标志的 List 参数指示的范围之内，该字符将被写出；否则，该字符将被排除

nl 显示行号

-b ：指定行号指定的方式，主要有两种：

-b a ：列出行号；

-b t ：列出行号，空行无视；

-n ：列出行号表示的方法，主要有三种：

-n ln ：行号在荧幕的最左方显示；

-n rn ：行号在自己栏位的最右方显示，且不加 0 ；

-n rz ：行号在自己栏位的最右方显示，且加 0 ；

more 一页一页翻动

-n<行数> 一次显示的行数

操作命令

回车键 ：向下翻(一行)；

空格键 ：向下翻一页；

Ctrl+F ：向下翻一页；

Ctrl+B 或 b ：向上翻一页，对管线无用。

/字串 ：在这个显示的内容当中，向下搜寻『字串』这个关键字；

:f ：输出文档名以及目前行号；

q：退出

less 分页显示

回车键 ：向下翻动一行；

空白键 ：向下翻动一页；

[pagedown]：向下翻动一页；Ctrl+B，U半页

[pageup] ：向上翻动一页；Ctrl+F，D半页

/字串 ：向下搜寻『字串』的功能；

?字串 ：向上搜寻『字串』的功能；

n ：重复前一个搜寻 (与 / 或 ? 有关！)

N ：反向的重复前一个搜寻 (与 / 或 ? 有关！)

q ：退出 less

head 取出文件前几行

-n<行数>

tail [参数] [文件/日志] 取出文件后面几行

-f 循环读取，查阅正在改变的日志文件

-n<行数> 显示文件的尾部n行内容

file [参数] [文件/目录]用于辨别文件类型

-L 直接显示符号连接所指向的文件的类别。

## 3.检查文件

diff [参数] [文件1] [文件2] 比较文件的差异

-y 以并列方式显示文件的异同之处

-w 忽略全部的空格字符

| diffstat 进行比较结果的统计显示

cmp -参数 [文件1][文件2] 比较文件是否有差异

-c 除了标明差异处的十进制

-l 标出所以不一样的地方

cksum [文件] 输出CRC校验码

## 4.查找文件

find [路径] -option [ -print ] [ -exec -ok command ] {} \;

-name name 文件名称符合name的文件

-iname name 忽略大小写

-empty 空的文件

-type c/d/f 文件类型是C/目录/一般文件

find . -name “\*.c” #当前目录及子目录下所有.c文件列出来

find -iname "MyProgram.c" -exec md5sum {} \; #对找到的文件执行某个命令

find ~ -empty #查找home目录下的所有空文件

whereis [-bfmsu][-B <目录>...][-M <目录>...][-S <目录>...][文件...]

-b 　 只查找二进制文件。

-B<目录> 　只在设置的目录下查找二进制文件。

-f 　 不显示文件名前的路径名称。

-m 　 只查找说明文件。

-M<目录> 只在设置的目录下查找说明文件。

-s 　 只查找原始代码文件。

-S<目录> 　只在设置的目录下查找原始代码文件。

-u 　 查找不包含指定类型的文件。

locate 用于查找符合条件的文档，他会去保存文档和目录名称的数据库内，查找合乎范本样式条件的文档或目录。

-d 配置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，文档名为slocate.db，您可使用 这个参数另行指定。

查找passwd文件，输入以下命令：

locate passwd

which 命令用于查找文件，会在环境变量$PATH设置的目录里查找符合条件的文件

which [参数] [文件]

-n<文件名长度>指定文件名长度,指定的长度必须大于或等于文件中最长的文件名

-p<文件名长度> 与-n参数相同，但此处的<文件名长度>包括了文件的路径。

## 5.改变文件属性

ll或ls –l 命令来显示一个文件的属性以及文件所属的用户和组

当为[ d ]则是目录

当为[ - ]则是文件；

若是[ l ]则表示为链接文档(link file)；

若是[ b ]则表示为装置文件里面的可供储存的接口设备(可随机存取装置)；

若是[ c ]则表示为装置文件里面的串行端口设备，例如键盘、鼠标(一次性读取装置)。



chown -R 属主名 文件名（更改文件属主，文件属

chown -R 属主名**:**属组名 文件名

-R：递归更改文件属组

-v：显示指令执行过程

chgrp -R 属组名 文件名（更改文件属组）

chmod -R xyz 文件或目录（更改文件9个权限属性）

chmod u=rwx,g=rx,o=r 文件名

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| chmod | u  g  o  a | +(加入)  -(除去)  =(设定) | r  w  x | 文件或目录 |

chattr 用于改变文件属性

|  |  |  |  |
| --- | --- | --- | --- |
| chattr | -R 递归处理  -V 显示指令执行过程。  +<属性> 开启文件的该项属性。  -<属性> 关闭文件的该项属性。  =<属性> 指定文件的该项属性。 | a:让文件或目录仅供附加用途(日志)  c：将文件或目录压缩后存放。  d：将文件排除在倾倒操作之外  i：不得任意更动文件或目录。  u：预防意外删除 | 文件/  目录 |

举例：用chattr命令防止系统中某个关键文件被修改：

chattr +i /etc/resolv.conf

lsattr /etc/resolv.conf #查看文件属性

chattr +i /etc/{passwd,shadow,group,gshadow}

chattr -i /etc/{passwd,group,shadow,gshadow}

lsattr -a /etc/{passwd,shadow,group,gshadow}

## 6.分割文件

split命令用于将一个文件分割成数个.

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

-<行数> 指定每多少行切成一个小文件，默认1000行

-b<字节> 指定每多少字节切成一个小文件

-C<字节> 切割时尽量维持每行的完整性

## 7.加密解密文件

rhmask命令用于对文件进行加密和解密操作

rhmask [加密文件][输出文件] 或 rhmask [-d][加密文件][源文件][输出文件]

-d 产生加密过的文件

## 8.链接

ln [参数][源文件/目录][目标文件或目录]

必要参数：

-b 删除，覆盖以前建立的链接

-d 允许超级用户制作目录的硬链接

-f 强制执行

-i 交互模式，文件存在则提示用户是否覆盖

-n 把符号链接视为一般目录

-s 软链接(符号链接)

-v 显示详细的处理过程

选择参数：

-S "-S<字尾备份字符串> "或 "--suffix=<字尾备份字符串>"

-V "-V<备份方式>"或"--version-control=<备份方式>"

软链接：

• 1.软链接，以路径的形式存在。类似于Windows操作系统中的快捷方式

• 2.软链接可以 跨文件系统 ，硬链接不可以

• 3.软链接可以对一个不存在的文件名进行链接

• 4.软链接可以对目录进行链接

硬链接：

• 1.硬链接，以文件副本的形式存在。但不占用实际空间。

• 2.不允许给目录创建硬链接

• 3.硬链接只有在同一个文件系统中才能创建

给文件创建硬链接，为log2013.log创建硬链接ln2013，log2013.log与ln2013的各项属性相同

* ln log2013.log ln2013

## 9.读取标准输入

read 命令用于从标准输入读取数值

read 内部命令被用来从标准输入读取单行数据。

这个命令可以用来读取键盘输入，当使用重定向的时候，可以读取文件中的一行数据。

-a 后跟一个变量，该变量会认为是个数组，然后给其赋值，默认是以空格为分割符。

-d 后面跟一个标志符，其实只有其后的第一个字符有用，作为结束的标志。

-p 后面跟提示信息，即在输入前打印提示信息。

-e 在输入的时候可以使用命令补全功能。

-n 后跟一个数字，定义输入文本的长度，很实用。

-r 屏蔽\，如果没有该选项，则\作为一个转义字符，有的话 \就是个正常的字符了。

-s 安静模式，在输入字符时不再屏幕上显示，例如login时输入密码。

-t 后面跟秒数，定义输入字符的等待时间。

-u 后面跟fd，从文件描述符中读入，该文件描述符可以是exec新开启的。

tee 命令用于读取标准输入的数据，并将其内容输出成文件。

tee [参数] [文件名]

-a 附加到既有文件的后面，而非覆盖它

-i 忽略中断信号

# 三、文档编辑

## 1.过滤

col 命令过滤控制字符

在许多UNIX说明文件里，都有RLF控制字符。当我们运用shell特殊字符">"和">>"，把说明文件的内容输出成纯文本文件时，控制字符会变成乱码，col指令则能有效滤除这些控制字符。

col [-bfx][-l<缓冲区列数>]

-b 过滤所有的控制字符

-f 滤除RLF字符，保留HRLF字符。

-x 以多个空格字符来表示跳格字符

-l<缓冲区列数> 预设的内存缓冲区有128列

colrm 命令滤除指定的行

colrm指令从标准输入设备读取书记，转而输出到标准输出设备。

如果不加任何参数，则该指令不会过滤任何一行。

colrm [开始行数编号<结束行数编号>]

## 2.切割

csplit用于分割文件

将文件依照指定的范本样式予以切割后，分别保存成名称为xx00,xx01,xx02...的文件。

若给予的文件名称为"-"，则csplit指令会从标准输入设备读取数据。

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

• -b<输出格式>预设的输出格式其文件名称为xx00,xx01...等，您可以通过改变<输出格式>来改变输出的文件名。

• -f<输出字首字符串> 预设的输出字首字符串其文件名为xx00,xx01...等，如果你指定输出字首字符串为"hello"，则输出的文件名称会变成hello00,hello01...等。

• -k 保留文件，就算发生错误或中断执行，也不能删除已经输出保存的文件。

• -n<输出文件名位数>预设的输出文件名位数其文件名称为xx00,xx01...等，如果你指定输出文件名位数为"3"，则输出的文件名称会变成xx000,xx001...等。

• -q或-s 不显示指令执行过程。

• -z 删除长度为0 Byte文件。

将文本文件testfile以第 2 行为分界点切割成两份，使用如下命令：

* csplit testfile 2

## 3.查找

look 命令用于查询单词

look指令用于英文单字的查询。您仅需给予它欲查询的字首字符串，它会显示所有开头字符串符合该条件的单字。

-f 忽略字符大小写差别

## 4.排序

sort 用于将文本文件内容加以排序

-n 依照数值的大小排序。

-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。

sort 命令将以默认的方式将文本文件的第一列以ASCII 码的次序排列，并将结果输出到标准输出。

fmt命令用于编排文本文件

fmt指令会从指定的文件里读取内容，将其指定格式重新编排后，输出到标准输出设备。若指定的文件名为"-"，则fmt指令会从标准输入设备读取数据。

• -c 每段前两列缩排。

• -p<列起始字符串> 仅合并含有指定字符串的列，通常运用在程序语言的注解方面。

• -s只拆开字数超出每列字符数的列，但不合并字数不足每列字符数的列。

• -t 每列前两列缩排，但第1列和第2列的缩排格式不同。

• -u 每个字符之间都以一个空格字符间隔，每个句子之间则两个空格字符分隔。

• -w<每列字符数> >或-<每列字符数> 设置每列的最大字符数。

fold命令限制文件列宽

fold指令会从指定的文件里读取内容，将超过限定列宽的列加入增列字符后，输出到标准输出设备。若不指定任何文件名称，或是所给予的文件名为"-"，则fold指令会从标准输入设备读取数据。

fold [-参数][-w<每列行数>] [文件...]

-b 以Byte为单位计算列宽

-s 以空格字符作为换列点

-w<每列行数> 设置每列的最大行数

## 5.比较

comm命令用于两个已排过序的文件，按列比较文件差异，并显示结果

comm [-123] [第1个文件][第2个文件]

-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。

join命令用于将两个文件中，指定栏位内容相同的行连接起来。

找出两个文件中，指定栏位内容相同的行，并加以合并，再输出到标准输出设备。

join [参数] [文件1][文件2]

-a<1或2> 除了显示原来的输出内容之外，还显示指令文件中没有相同栏位的行。

-e<字符串> 若[文件]中找不到指定的栏位，则在输出中填入选项中的字符串。

-i 比较栏位内容时，忽略大小写的差异。

-o<格式> 按照指定的格式来显示结果。

-t<字符> 使用栏位的分隔字符。

-v<1或2> 跟-a相同，但是只显示文件中没有相同栏位的行。

-1<栏位> 连接[文件1]指定的栏位。

-2<栏位> 连接[文件2]指定的栏位。

## 6.检查/删除

uniq命令用于检查及删除文本文件中重复出现的行列，一般与sort命令结合使用。

uniq [参数] [输入文件][输出文件]

-c 在每列旁边显示该行重复出现的次数。

-d 仅显示重复出现的行列。

-f<栏位> 忽略比较指定的栏位。

-s<字符位置> 忽略比较指定的字符。

-u 仅显示出一次的行列。

-w<字符位置> 指定要比较的字符。

[输入文件] 指定已排序好的文本文件。如果不指定此项，则从标准读取数据；

[输出文件] 指定输出的文件。如果不指定此选项，则将内容显示到标准输出设备（显示终端）。

tr 命令用于转换或者删除文件中的字符

tr 指令从标准输入设备读取数据，经过字符串转译后，将结果输出到标准输出设备。

tr [参数] [第一字符集][第二字符集]

tr [OPTION]…SET1[SET2]

-c：反选设定字符。符合 SET1 部份不做处理，不符合部份进行转换

-d：删除指令字符

-s：缩减连续重复的字符成指定的单个字符

-t：削减 SET1 指定范围，使之与 SET2 设定长度相等

字符集合的范围：

• \NNN 八进制值的字符 NNN (1 to 3 为八进制值的字符)

• \\ 反斜杠

• \a Ctrl-G 铃声

• \b Ctrl-H 退格符

• \f Ctrl-L 走行换页

• \n Ctrl-J 新行

• \r Ctrl-M 回车

• \t Ctrl-I tab键

• \v Ctrl-X 水平制表符

• CHAR1-CHAR2 ：字符范围从 CHAR1 到 CHAR2 的指定，范围的指定以 ASCII 码的次序为基础，只能由小到大，不能由大到小。

• [CHAR\*] ：这是 SET2 专用的设定，功能是重复指定的字符到与 SET1 相同长度为止

• [CHAR\*REPEAT] ：这也是 SET2 专用的设定，功能是重复指定的字符到设定的 REPEAT 次数为止(REPEAT 的数字采 8 进位制计算，以 0 为开始)

• [:alnum:] ：所有字母字符与数字

• [:alpha:] ：所有字母字符

• [:blank:] ：所有水平空格

• [:cntrl:] ：所有控制字符

• [:digit:] ：所有数字

• [:graph:] ：所有可打印的字符(不包含空格符)

• [:lower:] ：所有小写字母

• [:print:] ：所有可打印的字符(包含空格符)

• [:punct:] ：所有标点字符

• [:space:] ：所有水平与垂直空格符

• [:upper:] ：所有大写字母

• [:xdigit:] ：所有 16 进位制的数字

• [=CHAR=] ：所有符合指定的字符(等号里的 CHAR，代表你可自订的字符)

## 7.计算

wc 命令用于计算字数

利用wc指令我们可以计算文件的Byte数、字数、或是列数。

wc [参数] [文件...]

-c 只显示Bytes数。

-l 只显示行数。

-w 只显示字数。

# 四、文件传输

## 1.FTP

ftp命令设置文件系统相关功能

ftp [参数][主机名称或IP地址]

-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。

bye命令用于中断FTP连线并结束程序

在ftp模式下，输入bye即可中断目前的连线作业，并结束ftp的执行。

ftpshut命令在指定的时间关闭FTP服务器

指令提供系统管理者在设置的时间关闭FTP服务器，能在关闭前发出警告信息通知用户。

关闭时间若设置后为"none"，则会马上关闭服务器。采用"+30"的方式来设置表示服务器在30分钟之后关闭。使用"1130"的格式则代表服务器会在每日的11时30分关闭，时间格式为24小时制。

FTP服务器关闭后，在/etc目录下会产生一个名称为shutmsg的文件，把它删除后即可再度启动FTP服务器的功能。

ftpshut [-d<分钟>][-l<分钟>][关闭时间]["警告信息"]

-d<分钟> 切断所有FTP连线时间

-l<分钟> 停止接受FTP登入的时间

在晚上11:00 关闭FTP服务器，并在关闭前5 分钟拒绝新的FTP登录，前3 分钟关闭所有ftp的链接，且给出警告信息，可使用如下命令：

ftpshut-d 3 -1 5 1100 "Server will be shutdown at 23:00:00"

ftpwho命令用于显示目前所有以FTP登入的用户信息

指令可得知目前用FTP登入系统的用户有那些人，以及他们正在进行的操作。

ftpcount命令用于显示目前以FTP登入的用户人数

指令可得知目前用FTP登入系统的人数以及FTP登入人数的上限。

## 2.TFTP

tftp命令用于传输文件

tftp [主机名称或IP地址]

操作说明：

connect：连接到远程tftp服务器

mode：文件传输模式

put：上传文件

get：下载文件

quit：退出

verbose：显示详细的处理信息

trace：显示包路径

status：显示当前状态信息

binary：二进制传输模式

ascii：ascii 传送模式

rexmt：设置包传输的超时时间

timeout：设置重传的超时时间

help：帮助信息

? ：帮助信息

## 3.NCFTP

ncftp命令用于传输文件。

NcFTP是文字模式FTP程序的佼佼者，它具备多样特色， 包括显示传输速率，下载进度，自动续传，标住书签，可通过防火墙和代理服务器等。

当不指定用户名时，ncftp 命令会自动尝试使用匿名账户anonymous 去连接远程FTP 服 务器，不需要用户输入账号和密码。

ncftp [主机或IP地址]

-u<用户名> 指定登录FTP服务器的用户名

-p<密码> 设置用户密码

-P<端口号> 指定FTP端口号，默认为21

-j<账号> 指定账号

-h 帮助信息

-v 版本信息

# 五、磁盘管理

## 1.查询磁盘使用量

df 命令用于显示系统上的文件系统的磁盘使用情况统计(列出文件系统的整体磁盘使用量)

df [参数]... [文件]...

-a ：列出所有的文件系统，包括系统特有的 /proc 等文件系统；

-k ：以 KBytes 的容量显示各文件系统；

-m ：以 MBytes 的容量显示各文件系统；

-h ：以人们较易阅读的 GBytes, MBytes, KBytes 等格式自行显示；

-T ：显示文件系统类型, 连同该 partition 的 filesystem 名称 (例如 ext3) 也列出；

-i ：不用硬盘容量，而以 inode 的数量来显示

du 命令用于显示目录或文件的大小 (检查磁盘使用量)

du [参数] [目录或文件]

-a ：列出所有的文件与目录容量

-h ：以人们较易读的容量格式 (G/M) 显示；

-s ：列出总量而已，而不列出每个各别的目录占用容量；

-S ：显示个别目录的大小，不包括子目录下的总计

-k ：以 KBytes 列出容量显示；

-m ：以 MBytes 列出容量显示；

-L<符号连接> 显示选项中所指定符号连接的源文件大小

-X<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。

stat命令用于以文字的格式来显示inode的内容

stat [文件或目录]

mlabel命令用于设定磁盘的标签 (Label)

如果磁盘上设定过标签，mlabel 会将他显示给使用者。

如果没有指定新标签并且没有指定 c 或 s 选项，mlabel 会提示使用者输入新的标签。如果直接按下 Enter ，就会将原本的标签删除。

mlabel [参数] drive:[new\_label]

-v 更多的讯息。

-c 清除原有的标签，不出现提示讯息。

-s 显示目前的标签，不出现提示讯息。

dirs命令用于显示目录记录

显示目录堆叠中的记录

dirs [+/-n -l] [路径]

+n 显示从左边算起第n笔的目录。

-n 显示从右边算起第n笔的目录。

-l 显示目录完整的记录。

lndir命令用于连接目录内容

执行lndir指令，可一口气把源目录底下的文件和子目录统统建立起相互对应的符号连接。

lndir [-ignorelinks][-silent][源目录][目的目录]

-ignorelinks 直接建立符号连接的符号连接。

-silent 不显示指令执行过程。

## 2.系统挂载

mount 用于挂载Linux系统外的文件系统

[-t 文件系统] [-L 标签名] [-o 额外选项] [-n] 装置文件名 挂载点 (磁盘挂载)

mount [-hV]

mount -a [-fFnrsvw] [-t vfstype]

mount [-fnrsvw] [-o options [,...]] device | dir

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

-V：显示程序版本

-h：显示辅助讯息

-a：将 /etc/fstab 中定义的所有档案系统挂上。

-F：这个命令通常和 -a 一起使用，它会为每一个 mount 的动作产生一个行程负责执行。在系统需要挂上大量 NFS 档案系统时可以加快挂上的动作。

-v：显示较讯息，通常和 -f 用来除错。

-f：通常用在除错的用途。它会使 mount 并不执行实际挂上的动作，而是模拟整个挂上的过程。通常会和 -v 一起使用。

-n：一般而言，mount 在挂上后会在 /etc/mtab 中写入一笔资料。但在系统中没有可写入档案系统存在的情况下可以用这个选项取消这个动作。

-s-r：等于 -o ro

-w：等于 -o rw

-L：将含有特定标签的硬盘分割挂上。

-U：将档案分割序号为 的档案系统挂下。-L 和 -U 必须在/proc/partition 这种档案存在时才有意义。

-o async：打开非同步模式，所有的档案读写动作都会用非同步模式执行。

-o sync：在同步模式下执行。

-o atime、-o noatime：当 atime 打开时，系统会在每次读取档案时更新档案的『上一次调用时间』。当我们使用 flash 档案系统时可能会选项把这个选项关闭以减少写入的次数。

-o auto、-o noauto：打开/关闭自动挂上模式。

-o defaults:使用预设的选项 rw, suid, dev, exec, auto, nouser, and async.

-o dev、-o nodev-o exec、-o noexec允许执行档被执行。

-o suid、-o nosuid：允许执行档在 root 权限下执行。

-o user、-o nouser：使用者可以执行 mount/umount 的动作。

-o remount：将一个已经挂下的档案系统重新用不同的方式挂上。。

-o ro：用唯读模式挂上。

-o rw：用可读写模式挂上。

-o loop=：使用 loop 模式用来将一个档案当成硬盘分割挂上系统。

umount 命令用于卸载文件系统(装置文件名或挂载点) (磁盘卸载)

umount [参数][-t <文件系统类型>][文件系统/挂载点/设备名称]

-f ：强制卸除！可用在类似网络文件系统 (NFS) 无法读取到的情况下；

-n ：卸除时不要将信息存入/etc/mtab文件中。

-a 卸除/etc/mtab中记录的所有文件系统。

-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。

## 3.磁盘配额

quota命令用于显示磁盘已使用的空间与限制(查询用户或用户组的配额)

quota [-quvV][用户名称...]

或

quota [-gqvV][群组名称...]

-u 列出用户的磁盘空间限制。

-g 列出群组的磁盘空间限制。

-q 简明列表，只列出超过限制的部分。

-v 显示该用户或群组，在所有挂入系统的存储设备的空间限制。

-s 以习惯单位显示容量大小，如M、G。

quotaon命令用于开启磁盘空间限制

执行quotaon指令可开启用户和群组的才磅秒年空间限制，各分区的文件系统根目录必须有quota.user和quota.group配置文件。

quotaon [参数][文件系统...]

-a 开启在/ect/fstab文件里，有加入quota设置的分区的空间限制。

-g 开启群组的磁盘空间限制。

-u 开启用户的磁盘空间限制。

-v 显示指令指令执行过程。

quotaoff命令关闭磁盘空间限制

执行quotaoff指令可关闭用户和群组的磁盘空间限制。

quotaoff [参数][文件系统...]

-a 关闭在/etc/fstab文件里，有加入quota设置的分区的空间限制。

-g 关闭群组的磁盘空间限制。

-u 关闭用户的磁盘空间限制。

-v 显示指令执行过程。

edquota命令用于编辑用户或群组的磁盘配额

edquota预设会使用vi来编辑使用者或群组的磁盘配额设置

edquota [-p <源用户名称>][-ug][用户或群组名称...]

或

edquota [-ug] -t

-u 设置用户的磁盘配额，这是预设的参数。

-g 设置群组的磁盘配额。

-p<源用户名称> 将源用户的磁盘配额设置套用至其他用户或群组。

-t 设置宽限期限。

quotacheck命令用于检查磁盘的使用空间与限制

执行quotacheck指令，扫描挂入系统的分区，并在各分区的文件系统根目录下产生quota.user和quota.group文件，设置用户和群组的磁盘空间限制。

quotacheck [参数][文件系统...]

-a 扫描在/etc/fstab文件里，有加入quota设置的分区。

-d 详细显示指令执行过程，便于排错或了解程序执行的情形。

-u 扫描磁盘空间时，计算每个用户识别码所占用的目录和文件数目。

-g 扫描磁盘空间时，计算每个群组识别码所占用的目录和文件数目。

-R 排除根目录所在的分区。

-v 显示指令执行过程。

-s 以习惯单位显示容量大小，如M、G；

repquota命令用于检查磁盘空间限制的状态(查询整个分区的配额情况)

执行repquota指令，可报告磁盘空间限制的状况，清楚得知每位用户或每个群组已使用多少空间。

-a 列出在/etc/fstab文件里，有加入quota设置的分区的使用状况，包括用户和群组。

-u 列出所有用户的磁盘空间限制。

-g 列出所有群组的磁盘空间限制。

-v 显示该用户或群组的所有空间限制。

-s 以习惯单位显示容量大小，如M、G。

# 六、磁盘维护

## 1.Mkfs建立系统

mkfs命令用于在特定的分区上建立 linux 文件系统

mkfs [参数] 文件系统/装置文件名 [block]

-t <文件系统格式>：例 ext3, ext2, vfat ,reiserfs ,fat32 ,msdos等(系统支持才会生效)

device ： 预备检查的硬盘分区，例如：/dev/sda1

-V : 详细显示模式

-c : 在制做档案系统前，检查该partition 是否有坏轨

-l bad\_blocks\_file : 将有坏轨的block资料加到 bad\_blocks\_file 里面

block : 给定 block 的大小

mke2fs命令用于建立ext2文件系统== mkfs.ext2

mke2fs [参数][ 设备名称][区块数]

-b<区块大小> 指定区块大小，单位为字节。

-c 检查是否有损坏的区块。

-f<不连续区段大小> 指定不连续区段的大小，单位为字节。

-F 不管指定的设备为何，强制执行mke2fs。

-i<字节> 指定"字节/inode"的比例。

-N<inode数> 指定要建立的inode数目。

-l<文件> 从指定的文件中，读取文件西中损坏区块的信息。

-L<标签> 设置文件系统的标签名称。

-m<百分比值> 指定给管理员保留区块的比例，预设为5%。

-M 记录最后一次挂入的目录。

-q 执行时不显示任何信息。

-r 指定要建立的ext2文件系统版本。

-R=<区块数> 设置磁盘阵列参数。

-S 仅写入superblock与group descriptors，而不更改inode able inode bitmap以及block bitmap。

-v 执行时显示详细信息。

-V 显示版本信息。

mkdosfs命令用于建立DOS文件系统==mkfs.msdos

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

mkdosfs [参数] device [ block\_count ]

-c 建立档案系统之前先检查是否有坏轨。

-l 从得定的档案中读取坏轨记录。

-f 指定档案配置表（FAT , File Allocation Table)的数量。预设值为 2 。目前 Linux 的 FAT 档案系统不支援超过 2 个 FAT 表。通常这个不需要改。

-F 指定 FAT 表的大小，通常是 12 或是 16 个位元组。12 位元组通常用于磁碟片，16 位元组用于一般硬盘的分割区，也就是所谓的 FAT16 格式。这个值通常系统会自己选定适当的值。在磁碟片上用 FAT16 通常不会发生作用，反之在硬盘上用 FAT12 亦然。

-i 指定 Volume ID。一般是一个 4 个位元组的数字，像是 2e203a47 。如果不给系统会自己产生。

-m 当使用者试图用这片磁片或是分割区开机，而上面没有操作系统时，系统会给使用者一段警告讯息。这个参数就是用来变更这个讯息的。你可以先用档案编辑好，然后用这个参数指定，或是用

-m -

这样系统会要求你直接输入这段文字。要特别注意的是，档案里的字串长度不要超过 418 个字，包括展开的跳栏符号（TAB）和换行符号（换行符号在 DOS 底下算两个字元！）

-n 指定 Volume Name，就是磁碟标签。如同在 DOS 底下的 format 指令一样，给不给都可以。没有预设值。

-r 指定根目录底下的最大档案数。这里所谓的档案数包括目录。预设值是在软碟上是 112 或是 224 ，在硬盘上是 512。没事不要改这个数字。

-s 每一个磁丛（cluster）的磁区数。必须是 2 的次方数。不过除非你知道你在作什么，这个值不要乱给。

-v 提供额外的讯息

mkfs.minix命令用于建立Minix文件系统

mkfs.minix [参数][设备名称][区块数]

-c 检查是否有损坏的区块。

-i<inode数目> 指定文件系统的inode总数。

-l<文件> 从指定的文件中，读取文件系统中损坏区块的信息。

-n<文件名长度> 指定文件名称长度的上限。

-v 建立第2版的Minix文件系统。

fsconf命令用于设置文件系统相关功能

fsconf是Red Hat Linux发行版专门用来调整Linux各项设置的程序。

fsconf [--check]

--chedk 检查特定文件的权限。

## 2.Disk磁盘分区

fdisk是一个创建和维护分区表的程序

它兼容DOS类型的分区表、BSD或者SUN类型的磁盘列表。

fdisk [必要参数][选择参数]

必要参数：

-l 列出素所有分区表

-u 与"-l"搭配使用，显示分区数目

选择参数：

-s<分区编号> 指定分区

-v 版本信息

菜单操作说明

m ：显示菜单和帮助信息

a ：活动分区标记/引导分区

d ：删除分区

l ：显示分区类型

n ：新建分区

p ：显示分区信息

q ：退出不保存

t ：设置分区号

v ：进行分区检查

w ：保存修改

x ：扩展应用，高级功能

cfdisk命令用于磁盘分区

cfdisk是用来磁盘分区的程序，它十分类似DOS的fdisk，具有互动式操作界面而非传统fdisk的问答式界面，您可以轻易地利用方向键来操控分区操作。

cfdisk [参数][外围设备代号]

-a 在程序里不用反白代表选取，而以箭头表示。

-c<柱面数目> 忽略BIOS的数值，直接指定磁盘的柱面数目。

-h<磁头数目> 忽略BIOS的数值，直接指定磁盘的磁头数目。

-P<r,s,t> 显示分区表的内容，附加参数"r"会显示整个分区表的详细资料，附加参数"s"会依照磁区的顺序显示相关信息，附加参数"t"则会以磁头，磁区，柱面的方式来显示资料。

-s<磁区数目> 忽略BIOS的数值，直接指定磁盘的磁区数目。

-v 显示版本信息。

-z 不读取现有的分区，直接当作没有分区的新磁盘使用。

sfdisk命令是硬盘分区工具程序

sfdisk为硬盘分区工具程序，可显示分区的设置信息，并检查分区是否正常。

sfdisk [-?Tvx][-d <硬盘>][-g <硬盘>][-l <硬盘>][-s <分区>][-V <硬盘>]

-d<硬盘> 显示硬盘分区的设置。

-g<硬盘>或--show-geometry<硬盘> 显示硬盘的CHS参数。

-l<硬盘> 显示后硬盘分区的相关设置。

-s<分区> 显示分区的大小，单位为区块。

-T或--list-types 显示所有sfdisk能辨识的文件系统ID。

-v或--version 显示版本信息。

-V<硬盘>或--verify<硬盘> 检查硬盘分区是否正常。

-x或--show-extend 显示扩展分区中的逻辑分区。

hdparm命令用于显示与设定硬盘的参数

hdparm可检测，显示与设定IDE或SCSI硬盘的参数。

gdisk/parted

## 3.Swap交换

swapon命令用于激活Linux系统中交换空间

Linux系统的内存管理必须使用交换区来建立虚拟内存。

/sbin/swapon -a [-v]

/sbin/swapon [-v] [-p priority] specialfile ...

/sbin/swapon [-s]

-V 显示版本讯息

-s 显示简短的装置讯息

-a 自动启动所有SWAP装置

-p 设定优先权，你可以在0到32767中间选一个数字给他。或是在 /etc/fstab 里面加上 pri=[value] ([value]就是0~32767中间一个数字)，然后你就可以很方便的直接使用 swapon -a 来启动他们，而且有优先权设定。

swapoff命令用于关闭系统交换区(swap area)

swapoff实际上为swapon的符号连接，可用来关闭系统的交换区。

swapoff [设备]

-a 将/etc/fstab文件中所有设置为swap的设备关闭

-h 帮助信息

-V 版本信息

mkswap命令用于设置交换区(swap area)

mkswap可将磁盘分区或文件设为Linux的交换区。

mkswap [-cf][-v0][-v1][设备名称或文件][交换区大小]

-c 建立交换区前，先检查是否有损坏的区块。

-f 在SPARC电脑上建立交换区时，要加上此参数。

-v0 建立旧式交换区，此为预设值。

-v1 建立新式交换区。

[交换区大小] 指定交换区的大小，单位为1024字节。

## 4.Fsck检查修复

badblocks命令用于检查磁盘装置中损坏的区块

执行指令时须指定所要检查的磁盘装置，及此装置的磁盘区块数

fsck命令用于 检查与修复 Linux 档案系统，可以同时检查一个或多个 Linux 档案系统

fsck [-t 文件系统] [-ACay] 装置名称 (磁盘检验)

fsck [参数] 文件系统 [...]

filesys ： device 名称(eg./dev/sda1)，mount 点 (eg. / 或 /usr)

-t <类型>: 给定档案系统的型式，若在 /etc/fstab 中已有定义或 kernel 本身已支援的则不需加上此参数

-s : 依序一个一个地执行 fsck 的指令来检查

-A : 对/etc/fstab 中所有列出来的 partition 做检查

-C : 显示完整的检查进度

-d : 列印 e2fsck 的 debug 结果

-p : 同时有 -A 条件时，同时有多个 fsck 的检查一起执行

-R : 同时有 -A 条件时，省略 / 不检查

-V : 详细显示模式

-a : 如果检查有错则自动修复

-r : 如果检查有错则由使用者回答是否修复

fsck.ext2命令用于检查文件系统并尝试修复错误

当ext2文件系统发生错误时，可用fsck.ext2指令尝试加以修复。

fsck.ext2 [参数][外围设备代号]

-a 自动修复文件系统，不询问任何问题。

-b<分区第一个磁区地址> 指定分区的第一个磁区的起始地址，也就是Super Block。

-B<区块大小> 设置该分区每个区块的大小。

-c 检查指定的文件系统内，是否存在有损坏的区块。

-C<反叙述器> 指定反叙述器，fsck.ext2指令会把全部的执行过程，都交由其逆向叙述，便于排错或监控程序执行的情形。

-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。

-f 强制对该文件系统进行完整检查，纵然该文件系统在慨略检查下没有问题。

-F 检查文件系统之前，先清理该保存设备块区内的数据。

-I<inode缓冲区块数> 设置欲检查的文件系统，其inode缓冲区的区块数目。

-l<损坏区块文件> 把文件中所列出的区块，视为损坏区块并将其标示出来，避免应用程序使用该区块。

-L<损坏区块文件> 此参数的效果和指定"-l"参数类似，但在参考损坏区块文件标示损坏区块之前，会先将原来标示成损坏区块者统统清楚，即全部重新设置，而非仅是加入新的损坏区块标示。

-n 把欲检查的文件系统设成只读，并关闭互动模式，否决所有询问的问题。

-p 此参数的效果和指定"-a"参数相同。

-P<处理inode大小> 设置fsck.ext2指令所能处理的inode大小为多少。

-r 此参数将忽略不予处理，仅负责解决兼容性的问题。

-s 检查文件系统时，交换每对字节的内容。

-S 此参数的效果和指定"-s"参数类似，但不论该文件系统是否已是标准位顺序，一律交换每对字节的内容。

-t 显示fsck.ext2指令的时序信息。

-v 详细显示指令执行过程。

-V 显示版本信息。

-y 关闭互动模式，且同意所有询问的问题。

e2fsck命令用于检查使用 Linux ext2 档案系统的 partition 是否正常工作

e2fsck [参数] device

device ： 预备检查的硬盘 partition，例如：/dev/sda1

-a : 对 partition 做检查，若有问题便自动修复，等同 -p 的功能

-b : 设定存放 superblock 的位置

-B : 设定单位 block 的大小

-c : 检查该partition 是否有坏轨

-C file : 将检查的结果存到 file 中以便查看

-d : 列印 e2fsck 的 debug 结果

-f : 强制检查

-F : 在开始检查前，将device 的 buffer cache 清空，避免有错误发生

-l bad\_blocks\_file : 将有坏轨的block资料加到 bad\_blocks\_file 里面

-L bad\_blocks\_file : 设定坏轨的block资料存到 bad\_blocks\_file 里面，若无该档则自动产生

-n : 将档案系统以[唯读]方式开启

-p : 对 partition 做检查，若有问题便自动修复

-v : 详细显示模式

-V : 显示出目前 e2fsck 的版本

-y : 预先设定所有检查时的问题均回答[是]

fsck.minix命令用于检查文件系统并尝试修复错误

当minix文件系统发生错误时，可用fsck.minix指令尝试加以参考。

fsck.minix [参数][外围设备代号]

-a 自动修复文件系统，不询问任何问题。

-f 强制对该文件系统进行完整检查，纵然该文件系统在慨略检查下没有问题。

-l 列出所有文件名称。

-m 使用类似MINIX操作系统的警告信息。

-r 采用互动模式，在执行修复时询问问题，让用户得以确认并决定处理方式。

-s 显示该分区第一个磁区的相关信息。

-v 显示指令执行过程。

## 5.读取数据

dd命令用于读取、转换并输出数据

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

if=文件名：输入文件名，默认为标准输入。即指定源文件。

of=文件名：输出文件名，默认为标准输出。即指定目的文件。

ibs=bytes：一次读入bytes个字节，即指定一个块大小为bytes个字节。

obs=bytes：一次输出bytes个字节，即指定一个块大小为bytes个字节。

bs=bytes：同时设置读入/输出的块大小为bytes个字节。

cbs=bytes：一次转换bytes个字节，即指定转换缓冲区大小。

skip=blocks：从输入文件开头跳过blocks个块后再开始复制。

seek=blocks：从输出文件开头跳过blocks个块后再开始复制。

count=blocks：仅拷贝blocks个块，块大小等于ibs指定的字节数。

conv=<关键字>，关键字可以有以下11种：

conversion：用指定的参数转换文件。

ascii：转换ebcdic为ascii

ebcdic：转换ascii为ebcdic

ibm：转换ascii为alternate ebcdic

block：把每一行转换为长度为cbs，不足部分用空格填充

unblock：使每一行的长度都为cbs，不足部分用空格填充

lcase：把大写字符转换为小写字符

ucase：把小写字符转换为大写字符

swab：交换输入的每对字节

noerror：出错时不停止

notrunc：不截短输出文件

sync：将每个输入块填充到ibs个字节，不足部分用空（NUL）字符补齐。

## 6.写入数据

sync命令用于数据同步

sync命令是在关闭Linux系统时使用的。

Linux 系统中欲写入硬盘的资料有的时候会了效率起见，会写到 filesystem buffer 中，这个 buffer 是一块记忆体空间，如果欲写入硬盘的资料存于此 buffer 中，而系统又突然断电的话，那么资料就会流失了，sync 指令会将存于 buffer 中的资料强制写入硬盘中。

# 七、网络通讯

ifconfig 显示或者设置网络设备

启动关闭指定网卡

# ifconfig eth0 down

# ifconfig eth0 up

为网卡配置和删除IPv6地址

# ifconfig eth0 add 33ffe:3240:800:1005::2/ 64 //为网卡添加IPv6地址

# ifconfig eth0 del 33ffe:3240:800:1005::2/ 64 //为网卡删除IPv6地址

用ifconfig修改MAC地址

# ifconfig eth0 down //关闭网卡

# ifconfig eth0 hw ether 00:AA:BB:CC:DD:EE //修改MAC地址

# ifconfig eth0 up //启动网卡

# ifconfig eth1 hw ether 00:1D:1C:1D:1E //关闭网卡并修改MAC地址

# ifconfig eth1 up //启动网卡

配置IP地址

# ifconfig eth0 192.168.1.56

//给eth0网卡配置IP地址

# ifconfig eth0 192.168.1.56 netmask 255.255.255.0

// 给eth0网卡配置IP地址,并加上子掩码

# ifconfig eth0 192.168.1.56 netmask 255.255.255.0 broadcast 192.168.1.255

// 给eth0网卡配置IP地址,加上子掩码,加上个广播地址

启用和关闭ARP协议

# ifconfig eth0 arp //开启

# ifconfig eth0 -arp //关闭

设置最大传输单元

# ifconfig eth0 mtu 1500

//设置能通过的最大数据包大小为 1500 bytes

nc命令用于设置路由器

执行本指令可设置路由器的相关参数。

nc [参数][主机名称][通信端口...]

-g<网关> 设置路由器跃程通信网关，最多可设置8个。

-G<指向器数目> 设置来源路由指向器，其数值为4的倍数。

-i<延迟秒数> 设置时间间隔，以便传送信息及扫描通信端口。

-l 使用监听模式，管控传入的资料。

-n 直接使用IP地址，而不通过域名服务器。

-o<输出文件> 指定文件名称，把往来传输的数据以16进制字码倾倒成该文件保存。

-p<通信端口> 设置本地主机使用的通信端口。

-r 乱数指定本地与远端主机的通信端口。

-s<来源位址> 设置本地主机送出数据包的IP地址。

-u 使用UDP传输协议。

-v 显示指令执行过程。

-w<超时秒数> 设置等待连线的时间。

-z 使用0输入/输出模式，只在扫描通信端口时使用。

netstat命令用于显示网络状态

利用netstat指令可让你得知整个Linux系统的网络情况。

netstat [参数]

-a 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。

-g 显示多重广播功能群组组员名单。

-i 显示网络界面信息表单。

-l 显示监控中的服务器的Socket。

-n 直接使用IP地址，而不通过域名服务器。

-p 显示正在使用Socket的程序识别码和程序名称。

-s 显示网络工作信息统计表。

-t 显示TCP传输协议的连线状况。

-u 显示UDP传输协议的连线状况。

-v 显示指令执行过程。

losf命令

ss命令

ping命令用于检测主机

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。

例 ping一个远程主机，只发5个数据包

$ ping -c 5 gmail.com

telnet命令用于远端登入

执行telnet指令开启终端机阶段作业，并登入远端主机。

telnet [参数][主机名称或IP地址<通信端口>]

traceroute命令用于显示数据包到主机间的路径

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。

traceroute [参数][主机名称或IP地址][数据包大小]

-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。

tcpdump命令用于倾倒网络传输数据。

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。

tcpdump [参数][输出数据栏位]

-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。

# 八、系统管理

## 1.关机

shutdown -h now #立马关机

shutdown -h +10 #10分钟后关机

shutdown -h 20:25 #系统会在今天20:25关机

shutdown -r now #系统立马重启

shutdown -r 10 #系统十分钟后重启

reboot #重启 (出问题/强制重启)

init 6 #重启

## 2.用户和用户组管理

### 1)用户账号的管理

useradd 参数 用户名 (添加新的用户账号)

-d 目录 指定用户主目录，如果此目录不存在，使用-m选项，可以创建主目录。

-g 用户组 指定用户所属的用户组。

-G 用户组，用户组 指定用户所属的附加组。

-u 用户号 指定用户的用户号，如果有-o选项，则可以重复使用其他用户的标识号。

userdel 参数 用户名 (删除账号)

-r 把用户的主目录一起删除

usermod 参数 用户名 (修改账号)

-c, -d, -m, -g, -G, -s, -u以及-o等

-l 新用户名 指定一个新的账号，即将原来的用户名改为新的用户名。

passwd 参数 用户名

-l 锁定口令，即禁用账号。

-u 口令解锁。

-d 使账号无口令。

-f 强迫用户下次登录时修改口令。

### 2)用户组的管理

groupadd 选项 用户组(增加新的用户组)

-g GID 指定新用户组的组标识号（GID）。

-o可以与系统已有用户组的GID相同。

groupdel 用户组(删除用户组)

groupmod 选项 用户组(修改用户组的属性)

-g GID 为用户组指定新的组标识号。

-o可以与系统已有用户组的GID相同。

-n新用户组 将用户组的名字改为新名字

如果一个用户同时属于多个用户组，那么用户可以在用户组之间切换，以便具有其他用户组的权限。

用户可以在登录后，使用命令newgrp切换到其他用户组，这个命令的参数就是目的用户组。例如：newgrp root

ps 显示当前进程的状态

-aux 显示所有包含其他使用者的进程

-ef 显示所有命令，连带命令行

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

USER 进程拥有者

VSZ 占用的虚拟内存大小

RSS 占用的物理内存大小

TTY 终端的次要装置号码

STAT 该进程的状态

D: 无法中断的休眠状态 (通常 IO 的进程)

R: 正在执行中

S: 静止状态

T: 暂停执行

Z: 不存在但暂时无法消除

W: 没有足够的记忆体分页可分配

<: 高优先序的行程

N: 低优先序的行程

L: 有记忆体分页分配并锁在记忆体内 (实时系统或捱A I/O)

START 进程开始时间

TIME 执行的时间

COMMAND 所执行的指令

free 显示内存状态

-b 以为单位显示内存使用情况

-k以为单位显示内存使用情况

-m以为单位显示内存使用情况

-h以合适的单位显示内存使用情况

-s<间隔秒数> 持续观察内存使用状况

-t 显示内存总和列

top 实时显示process的动态

-d<时间> 改变显示的更新速度

-n<更新次数> 完成后将会推出top

top命令会显示当前系统中占用资源最多的一些进程（默认以CPU占用率排序）如果你想改变排序方式，可以在结果列表中点击O（大写字母O）会显示所有可用于排序的列，这个时候你就可以选择你想排序的列

Current Sort Field: P for window 1:Def

Select sort field via field letter, type any other key to return

a: PID = Process Id v: nDRT = Dirty Pages count

d: UID = User Id y: WCHAN = Sleeping in Function

e: USER = User Name z: Flags = Task Flags

如果只想显示某个特定用户的进程，可以使用-u选项

$ top -u oracle

kill 用于删除执行中的程序或者工作

-l <信息编号>

kill -kill 12345 #强制杀死进程

kill -9 12345 #彻底杀死进程

#杀死指定用户所有进程

#kill -9 $(ps -ef | grep hnlinux) //方法一 过滤出hnlinux用户进程

#kill -u hnlinux //方法二

uname 显示系统信息

-a 显示全部信息

-m 显示电脑类型

-n 显示计算机名

-r 显示操作系统的发行编号,内核

-s 显示操作系统名称

-v 显示系统时间

date命令可以用来显示或设定系统的日期与时间

设置系统日期

# date -s "01/31/2010 23:59:53"

当你修改了系统时间，你需要同步硬件时间和系统时间

# hwclock –systohc

# hwclock --systohc –utc

date +%Y/%m/%d

date +%H:%M

cal 显示日历

cal [month] [year]

Su

su命令用于切换用户账号，超级用户使用命令可以切换到任何其他用户而不用输入密码

$ su - USERNAME

# 九、系统设置

export 用于设置或显示环境变量

-p 列出所有的shell赋予程序的环境变量

-n 删除指定的变量。变量实际为删除，只是不会输出到后续指令的执行环境中。

crontab 用于定期执行程序的命令(定时任务)

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

-e 执行文字编辑器来设定时程表，内定的文字编辑器是VI

-r 删除目前的时程表

-l 列出目前的时程表

linux任务调度的工作主要分为以下两类：

* 1、系统执行的工作：系统周期性所要执行的工作，如备份系统数据、清理缓存
* 2、个人执行的工作：某个用户定期要做的工作，例如每隔10分钟检查邮件服务器是否有新信，这些工作可由每个用户自行设置

时程表的格式如下：

f1 f2 f3 f4 f5 program

* 其中 f1 是表示分钟，f2 表示小时，f3 表示一个月份中的第几日，f4 表示月份，f5 表示一个星期中的第几天。program 表示要执行的程序。
* Min hour day month weekday
* 当 f1 为 \* 时表示每分钟都要执行 program，f2 为 \* 时表示每小时都要执行程序，其馀类推
* 当 f1 为 a-b 时表示从第 a 分钟到第 b 分钟这段时间内要执行，f2 为 a-b 时表示从第 a 到第 b 小时都要执行，其馀类推
* 当 f1 为 \*/n 时表示每 n 分钟个时间间隔执行一次，f2 为 \*/n 表示每 n 小时个时间间隔执行一次，其馀类推
* 当 f1 为 a, b, c,... 时表示第 a, b, c,... 分钟要执行，f2 为 a, b, c,... 时表示第 a, b, c...个小时要执行，其馀类推

使用者也可以将所有的设定先存放在文件中，用 crontab file 的方式来设定时程表。

0 \*/2 \* \* \* /sbin/service httpd restart 意思是每两个小时重启一次apache

50 7 \* \* \* /sbin/service sshd start 意思是每天7：50开启ssh服务

50 22 \* \* \* /sbin/service sshd stop 意思是每天22：50关闭ssh服务

rpm 用来管理套件

使用rpm安装apache

# rpm -ivh httpd-2.2.3-22.0.1.el5.i386.rpm

更新apache

# rpm -uvh httpd-2.2.3-22.0.1.el5.i386.rpm

卸载/删除apache

# rpm -ev httpd

1、安装

rpm -i 需要安装的包名

rpm -ivh example.rpm 安装example.rpm包并在安装过程中显示正在安装的文件信息及安装进度；

rpm -Uvh [zabbix-release-4.0-2.el7.noarch.rpm](https://repo.zabbix.com/zabbix/4.0/rhel/7/x86_64/zabbix-release-4.0-1.el7.noarch.rpm)

2、查看安装完成的软件

[root@jacky zookeeper]# rpm -qa | grep jdk

java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.i686

java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.i686

3、卸载软件

rpm -e --nodeps 卸载的软件包

[root@jacky zookeeper]# rpm -e --nodeps java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.i686

4、参数说明

-a 查询所有包

-i 显示套件的相关信息

-i <包名> 安装指定的包

-l 显示包的文件列表

-h包安装时列出标记

-p<包名> 查询指定的RPM包

-q 使用询问模式，遇到问题先问用户

-R 显示包的关联性信息

-s 显示文件状态，本参数需配合”-l”参数使用

-U<包名> 升级指定的包

-v 显示指令执行过程

-vv 显示详细指令执行过程，用于排错

--nodeps 　不验证套件档的相互关联性。

--nofiles 　 不验证文件的属性。

# 十、备份压缩

tar 文件2 文件1

-c：压缩

-x：解压

-t：查看内容

-r：向压缩归档文件末尾追加文件

-u：更新原压缩包中的文件

-z：有gzip属性的 tar.gz

-j：有bz2属性的 tar.bz2

-Z：有compress属性的 tar.Z

-v：显示所有过程

-f: 使用档案名字，后面只能接档案名。

总结

1、\*.tar 用 tar –xvf 解压

2、\*.gz 用 gzip -d或者gunzip 解压

3、\*.tar.gz和\*.tgz 用 tar –xzf 解压

4、\*.bz2 用 bzip2 -d或者用bunzip2 解压

5、\*.tar.bz2用tar –xjf 解压

6、\*.Z 用 uncompress 解压

7、\*.tar.Z 用tar –xZf 解压

8、\*.rar 用 unrar e解压

9、\*.zip 用 unzip 解压

# 11、vi/vim



## 1.vi/vim 的使用

基本上 vi/vim 共分为三种模式，分别是**命令模式（Command mode）**，**输入模式（Insert mode）**和**底线命令模式（Last line mode）**。 这三种模式的作用分别是：

### 命令模式：

用户刚刚启动 vi/vim，便进入了命令模式。

此状态下敲击键盘动作会被Vim识别为命令，而非输入字符。

常用的几个命令：

* **i** 切换到输入模式，以输入字符。
* **x** 删除当前光标所在处的字符。
* **:** 切换到底线命令模式，以在最底一行输入命令。

若想要编辑文本：启动Vim，进入了命令模式，按下i，切换到输入模式。

命令模式只有一些最基本的命令，因此仍要依靠底线命令模式输入更多命令。

### 输入模式

在命令模式下按下i就进入了输入模式。

在输入模式中，可以使用以下按键：

* 字符按键以及Shift组合，输入字符
* **ENTER**，回车键，换行
* **BACK SPACE**，退格键，删除光标前一个字符
* **DEL**，删除键，删除光标后一个字符
* **方向键**，在文本中移动光标
* **HOME**/**END**，移动光标到行首/行尾
* Page Up/Page Down，上/下翻页
* **Insert**，切换光标为输入/替换模式，光标将变成竖线/下划线
* **ESC**，退出输入模式，切换到命令模式

### 底线命令模式

在命令模式下按下:（英文冒号）就进入了底线命令模式。

底线命令模式可以输入单个或多个字符的命令，可用的命令非常多。

在底线命令模式中，基本的命令有（已经省略了冒号）：

* q 退出程序
* w 保存文件

按ESC键可随时退出底线命令模式。



## 2. vi/vim 按键说明

### 命令模式

|  |  |
| --- | --- |
| 移动光标的方法 | |
| h 或 (←) | 光标向左移动一个字符 |
| j 或 (↓) | 光标向下移动一个字符 |
| k 或 (↑) | 光标向上移动一个字符 |
| l 或 (→) | 光标向右移动一个字符 |
| [Ctrl] + [f] | 屏幕『向下』移动一页，相当于 [Page Down]按键 (常用) |
| [Ctrl] + [b] | 屏幕『向上』移动一页，相当于 [Page Up] 按键 (常用) |
| [Ctrl] + [d] | 屏幕『向下』移动半页 |
| [Ctrl] + [u] | 屏幕『向上』移动半页 |
| + | 光标移动到非空格符的下一行 |
| - | 光标移动到非空格符的上一行 |
| n<space> | 那个 n 表示『数字』，例如 20 。按下数字后再按空格键，  光标会向右移动这一行的 n 个字符。 |
| 0 或[Home] | 这是数字『 0 』：移动到这一行的最前面字符处 (常用) |
| $ 或[End] | 移动到这一行的最后面字符处(常用) |
| H | 光标移动到这个屏幕的最上方那一行的第一个字符 |
| M | 光标移动到这个屏幕的中央那一行的第一个字符 |
| L | 光标移动到这个屏幕的最下方那一行的第一个字符 |
| G | 移动到这个档案的最后一行(常用) |
| nG | n 为数字。移动到这个档案的第 n 行。 (可配合 :set nu) |
| gg | 移动到这个档案的第一行，相当于 1G 啊！ (常用) |
| n<Enter> | n 为数字。光标向下移动 n 行(常用) |

|  |  |
| --- | --- |
| 搜索替换 | |
| /word | 向光标之下寻找一个名称为 word 的字符串。  例如要在档案内搜寻 vbird 这个字符串，就输入 /vbird 即可！ (常用) |
| ?word | 向光标之上寻找一个字符串名称为 word 的字符串。 |
| n | 这个 n 是英文按键。代表重复前一个搜寻的动作。  举例来说， 如果刚刚我们执行 /vbird 去向下搜寻 vbird 这个字符串，  则按下 n 后，会向下继续搜寻下一个名称为 vbird 的字符串。  如果是执行 ?vbird 的话，那么按下 n 则会向上继续搜寻名称为 vbird 的字符串！ |
| N | 这个 N 是英文按键。与 n 刚好相反，为『反向』进行前一个搜寻动作。  例如 /vbird 后，按下 N 则表示『向上』搜寻 vbird 。 |
| :n1,n2s/word1/word2/g | n1 与 n2 为数字。在第 n1 与 n2 行之间寻找 word1 这个字符串，  并将该字符串取代为 word2 ！  举例来说，在 100 到 200 行之间搜寻 vbird 并取代为 VBIRD 则： 『:100,200s/vbird/VBIRD/g』。(常用) |
| :1,$s/word1/word2/g  或 :%s/word1/word2/g | 从第一行到最后一行寻找 word1 字符串，  并将该字符串取代为 word2 ！(常用) |
| :1,$s/word1/word2/gc  或 :%s/word1/word2/gc | 从第一行到最后一行寻找 word1 字符串，并将该字符串取代为 word2 ！  且在取代前显示提示字符给用户确认 (confirm) 是否需要取代！(常用) |

|  |  |
| --- | --- |
| 删除、复制与贴上 | |
| x, X | 在一行字当中，x 为向后删除一个字符 (相当于 [del] 按键)， X 为向前删除一个字符(相当于 [backspace] 亦即是退格键) (常用) |
| nx | n 为数字，连续向后删除 n 个字符。举例来说，我要连续删除 10 个字符， 『10x』。 |
| dd | 删除游标所在的那一整行(常用) |
| ndd | n 为数字。删除光标所在的向下 n 行，例如 20dd 则是删除 20 行 (常用) |
| d1G | 删除光标所在到第一行的所有数据 |
| dG | 删除光标所在到最后一行的所有数据 |
| d$ | 删除游标所在处，到该行的最后一个字符 |
| d0 | 那个是数字的 0 ，删除游标所在处，到该行的最前面一个字符 |
| yy | 复制游标所在的那一行(常用) |
| nyy | n 为数字。复制光标所在的向下 n 行，例如 20yy 则是复制 20 行(常用) |
| y1G | 复制游标所在行到第一行的所有数据 |
| yG | 复制游标所在行到最后一行的所有数据 |
| y0 | 复制光标所在的那个字符到该行行首的所有数据 |
| y$ | 复制光标所在的那个字符到该行行尾的所有数据 |
| p, P | p 为将已复制的数据在光标下一行贴上，P 则为贴在游标上一行！ 举例来说，我目前光标在第 20 行，且已经复制了 10 行数据。则按下 p 后， 那 10 行数据会贴在原本的 20 行之后，亦即由 21 行开始贴。但如果是按下 P 呢？ 那么原本的第 20 行会被推到变成 30 行。 (常用) |
| J | 将光标所在行与下一行的数据结合成同一行 |
| c | 重复删除多个数据，例如向下删除 10 行，[ 10cj ] |
| u | 复原前一个动作。(常用) |
| [Ctrl]+r | 重做上一个动作。(常用) |
| . | 小数点！意思是重复前一个动作的意思。 (常用) |

### 输入模式

|  |  |
| --- | --- |
| 进入输入或取代的编辑模式 | |
| i, I | 进入输入模式(Insert mode)： i 为『从目前光标所在处输入』，  I 为『在目前所在行的第一个非空格符处开始输入』。 (常用) |
| a, A | 进入输入模式(Insert mode)： a 为『从目前光标所在的下一个字符处开始输入』，  A 为『从光标所在行的最后一个字符处开始输入』。(常用) |
| o, O | 进入输入模式(Insert mode)：  o 为『在目前光标所在的下一行处输入新的一行』；  O 为在目前光标所在处的上一行输入新的一行！(常用) |
| r, R | 进入取代模式(Replace mode)： r 只会取代光标所在的那一个字符一次；  R会一直取代光标所在的文字，直到按下 ESC 为止；(常用) |
| [Esc] | 退出编辑模式，回到一般模式中(常用) |

### 底线命令模式

|  |  |
| --- | --- |
| 指令行的储存、离开等指令 | |
| :w | 将编辑的数据写入硬盘档案中(常用) |
| :w! | 若文件属性为『只读』时，强制写入该档案。 |
| :q | 离开 vi (常用) |
| :q! | 若曾修改过档案，又不想储存，使用 ! 为强制离开不储存档案。 |
| :wq | 储存后离开，若为 :wq! 则为强制储存后离开 (常用) |
| ZZ | 这是大写的 Z 喔！若档案没有更动，则不储存离开，  若档案已经被更动过，则储存后离开！ |
| :w [filename] | 将编辑的数据储存成另一个档案（类似另存新档） |
| :r [filename] | 在编辑的数据中，读入另一个档案的数据。  亦即将 『filename』 这个档案内容加到游标所在行后面 |
| :n1,n2 w [filename] | 将 n1 到 n2 的内容储存成 filename 这个档案。 |
| :! command | 暂时离开 vi 到指令行模式下执行 command 的显示结果！例如 『:! ls /home』即可在 vi 当中察看 /home 底下以 ls 输出的档案信息！ |

|  |  |
| --- | --- |
| vim 环境的变更 | |
| :set nu | 显示行号，设定之后，会在每一行的前缀显示该行的行号 |
| :set nonu | 与 set nu 相反，为取消行号！ |

# 12、sed

sed 命令是利用脚本来处理文本文件。

sed 可依照脚本的指令来处理、编辑文本文件。

Sed 主要用来自动编辑一个或多个文件、简化对文件的反复操作、编写转换程序等。

-e<脚本> 以选项中指定的script来处理输入的文本文件。

-f<脚本文件> 以选项中指定的script文件来处理输入的文本文件。

-n 仅显示script处理后的结果

a :新增，a的后面可以接字串，，这些字串会在(目前的下一行)出现

c :取代，c 的后面可以接字串，这些字串可以取代 n1,n2 之间的行！

d :删除

i :插入，i 的后面可以接字串，而这些字串会在(目前的上一行)出现；

p :打印，需配合-n

s :取代,可以搭配正则，例如 1,20s/old/new/g 就是啦

举例：

sed -e 4a\newline testfile #使用sed 在第四行后添加新字符串

nl /etc/passwd | sed '2,5d' #将/etc/passwd 的内容列出并列印行号，将第 2~5 行删除

nl /etc/passwd | sed '2,5c No 2-5 number'#将第2-5行的内容取代成为『No 2-5 number』

nl /etc/passwd | sed -n '5,7p' #列出文件内的5-7行

nl /etc/passwd | sed '/root/p' #搜索有root关键字的行

nl /etc/passwd | sed -n '/root/{s/bash/blueshell/;p;q}'#找到对应的行，执行花括号的命令，每个命令用分号分隔，这里把bash替换为blueshell，再输出这行：

**数据的搜寻并替换**

sed 's/要被取代的字串/新的字串/g'

# 13、awk

AWK是一种处理文本文件的语言，是一个强大的文本分析工具。

awk [选项参数] 'script' var=value file(s)

或

awk [选项参数] -f scriptfile var=value file(s)

**选项参数说明：**

-F fs or --field-separator fs  
 指定输入文件折分隔符，fs是一个字符串或者是一个正则表达式，如-F:。

-v var=value or --asign var=value  
 赋值一个用户定义变量。

-f scripfile or --file scriptfile  
 从脚本文件中读取awk命令。

运算符

| **运算符** | **描述** |
| --- | --- |
| = += -= \*= /= %= ^= \*\*= | 赋值 |
| ?: | C条件表达式 |
| || | 逻辑或 |
| && | 逻辑与 |
| ~ 和 !~ | 匹配正则表达式和不匹配正则表达式 |
| < <= > >= != == | 关系运算符 |
| 空格 | 连接 |
| + - | 加，减 |
| \* / % | 乘，除与求余 |
| + - ! | 一元加，减和逻辑非 |
| ^ \*\*\* | 求幂 |
| ++ -- | 增加或减少，作为前缀或后缀 |
| $ | 字段引用 |
| in | 数组成员 |

举例

内建变量

| **变量** | **描述** |
| --- | --- |
| $n | 当前记录的第n个字段，字段间由FS分隔 |
| $0 | 完整的输入记录 |
| ARGC | 命令行参数的数目 |
| ARGIND | 命令行中当前文件的位置(从0开始算) |
| ARGV | 包含命令行参数的数组 |
| CONVFMT | 数字转换格式(默认值为%.6g)ENVIRON环境变量关联数组 |
| ERRNO | 最后一个系统错误的描述 |
| FIELDWIDTHS | 字段宽度列表(用空格键分隔) |
| FILENAME | 当前文件名 |
| FNR | 各文件分别计数的行号 |
| FS | 字段分隔符(默认是任何空格) |
| IGNORECASE | 如果为真，则进行忽略大小写的匹配 |
| NF | 一条记录的字段的数目 |
| NR | 已经读出的记录数，就是行号，从1开始 |
| OFMT | 数字的输出格式(默认值是%.6g) |
| OFS | 输出记录分隔符（输出换行符），输出时用指定的符号代替换行符 |
| ORS | 输出记录分隔符(默认值是一个换行符) |
| RLENGTH | 由match函数所匹配的字符串的长度 |
| RS | 记录分隔符(默认是一个换行符) |
| RSTART | 由match函数所匹配的字符串的第一个位置 |
| SUBSEP | 数组下标分隔符(默认值是/034) |

打印九九乘法表

seq 9 | sed 'H;g' | awk -v RS='' '{for(i=1;i<=NF;i++)printf("%dx%d=%d%s", i, NR, i\*NR, i==NR?"\n":"\t")}'

# 14、grep

grep 用于查找文件里符合条件的字符串

grep [-参数][-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>] [范本样式][文件或目录...]

-A<显示行数> 显示符合条件的一行，并显示该行后的内容

-i 忽略字符大小写的差别

-n 显示符合条件的行，标识出该行的列数编号

-v 显示不包含匹配文本的所有行

-E 将样式为延伸的正则表达式来使用

-F 将样式视为固定字符串的列表。

-o 只显示匹配PATTERN 部分。

ip addr | grep "inet" | head -3 | awk &apos;{print $2}&apos; | cut -d/ -f1

ifconfig ens33 | grep "inet" | head -1 | awk &apos;{ print $2}&apos;

# 15、let

<https://www.runoob.com/linux/linux-comm-let.html>

# 16、expr

<https://www.runoob.com/linux/linux-comm-expr.html>

# 17、bc

<https://www.runoob.com/linux/linux-comm-bc.html>

# 18、xargs

xargs 可以将管道或标准输入（stdin）数据转换成命令行参数，也能够从文件的输出中读取数据。

xargs 也可以将单行或多行文本输入转换为其他格式，例如多行变单行，单行变多行。

xargs 默认的命令是 echo，这意味着通过管道传递给 xargs 的输入将会包含换行和空白，不过通过 xargs 的处理，换行和空白将被空格取代。

xargs 是一个强有力的命令，它能够捕获一个命令的输出，然后传递给另外一个命令。

Xargs一般是和管道一起使用。

somecommand |xargs -item command

-n<次数> 表示命令在执行的时候一次用的argument的个数，默认是用所有的。

-d<定界符> 分隔符，xargs分隔符是回车，argument分隔符是空格，修改的是xargs的分隔符。

-L<num> 从标准输入一次读取 num 行送给 command 命令。

将所有图片文件拷贝到外部驱动器

$ ls \*.jpg | xargs -n1 -i cp {} /external-hard-drive/directory

将系统中所有jpd文件压缩打包

$ find / -name \*.jpg -type f -print | xargs tar -cvzf images.tar.gz

下载文件中列出的所有url对应的页面

$ cat url-list.txt | xargs wget –c

# 19、帮助

whatis [命令] 显示命令的描述信息

man [命令] 查看该命令帮助文档

[命令] –help

man页面一般可以分为8种命令类型

1.用户命令

2.系统调用

3.c库函数

4.设备与网络接口

5.文件格式

6.游戏与屏保

7.环境、表、宏

8.系统管理员命令和后台运行命令

例如，我们执行whatis crontab，你可以看到crontab有两个命令类型1和5，所以我们可以通过下面的命令查看命令类型5的man页面

$ whatis crontab

crontab (1) - maintain crontab files for individual users (V3)

crontab (5) - tables for driving cron

$ man 5 crontab

# 20、yum

* 1.列出所有可更新的软件清单命令：yum check-update
* 2.更新所有软件命令：yum update
* 3.仅安装指定的软件命令：yum install <package\_name>
* 4.仅更新指定的软件命令：yum update <package\_name>
* 5.列出所有可安装的软件清单命令：yum list
* 6.删除软件包命令：yum remove <package\_name>
* 7.查找软件包命令：yum search <keyword>
* 8.清除缓存命令：
  + yum clean package：清除缓存目录下的软件包
  + yum clean headers：清除缓存目录下的标题
  + yum clean oldheaders：清除缓存目录下旧的标题
  + yum clean，yum clean all（= yum clean package; yum clean oldheaders）：清除缓存目录下的软件包及旧的header
* 9.生成缓存：yum makecache

# 21、wget

使用wget从网上下载软件、音乐、视频

$ wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.2.1.tar.gz

下载文件并以指定的文件名保存文件

$ wget -O taglist.zip http://www.vim.org/scripts/download\_script.php?src\_id=7701





















