工作中采用的 linux 命令

1.从其他机器拷贝文件夹

格式: scp -r 文件夹名(源) 用户名@机器名:/路径（目的）
范例: scp -r search work@zjm-testing-ps23.zjm.baidu.com:/home/work/

2.查看哪个程序在用特定端口

格式: netstat -nap | grep 端口号
范例: netstat -nap | grep 8080

3.实时监控日志文件内容

格式: tail -f 日志文件名
范例: tail -f ui.log
说明:这显示 ui.log 文件的最后十行。tail 命令继续显示添加到 ui.log 文件中的行。显示会一直继续，直到您按下 Ctrl-C 按键顺序来停止

4.查看指定用户拥有的进程

格式: pstree 用户 id
范例: pstree work
说明:显示 work 用户正在运行的各进程之间的继承关系，以树状结构方式列出

5.查看磁盘空间使用情况

格式: df -h

6.文件搜索

1） which（whereis） 显示系统命令所在目录
which ls 可以查找 ls 命令文件所在目录。输出为：/bin/ls
whereis ls 可以查找 ls 命令所在目录，同时会显示该命令的帮助文档所在目录。
2） find 查找任何文件或目录 –name –size –ctime –atime –mtime –type -user
find [搜索路径] [查找方式] [搜索关键字]
1 根据文件名称查找
find ./ -name temp //在当前目录下，按照名字查找名字为 temp 的文件。可以使用通配符*和？，其中*匹配多个或零个字符，？匹配一个任意字符。
find ./ -name a.* find ./ -name a.tx? 都是查找 a.txt。
2 根据文件大小查找
find ./ -size +204800 //在当前目录下，按照文件大小来查找，其中后面的数字的单位是数据块，一个数据块是 512 字节。
204800*512 字节=204800*0.5KB=102400KB=100MB，因此上述命令是查找大于 100MB 的文件。
+是大于，-是小于，不加加减表示等于。
3 根据文件所有者来查找
find ./ -user yirenwei//在当前目录下，查找文件所有者为 yirenwei 的文件。
4 根据文件修改时间来查找
ctime（文件属性被修改过）atime（被访问过）mtime（内容被修改过） 单位是天
cmin、amin、mmin 单位是分钟
-之内，+超过
find ./ -ctime -1 查找 1 个小时内被属性被修改过的文件；
find ./ -cmin -10 查找 10 分钟之内属性被修改过的文件；
find ./ amin +10 查到已经访问了超过 10 分钟的文件。

5 连接符 –a（AND） -o（OR） -exec

5.1 find ./ -size +163800 –a -204800 //查找文件大小大于 80M 小于 100M 的文件。

5.2 find ./ -type f –exec ls {}\; 花括号，转义符，分号。 查找当前目录下所有的文件，然后再对查找结果进行 ls 操作。

又如： find ./ -user yirenwei –exec rm {}\;查找，并删除。

6 根据文件类型查找

find ./ -type f //查找二进制文件，还有 –type d/-type l，查找目录或者 link 文件。

3） locate （linux 特有的命令，UNIX 没有这个命令）

locate newfile //查找名称为 newfile 的文件。

注意：locate 是从系统文件的数据库中查找，不是在硬盘搜索，find 是在硬盘搜索。因此 locat 快，但是当新建的文件，可能找不到，因为系统的数据库还没有更新；可以使用 updatdb 命令来配合使用，手动更新数据库（只能 root 调用貌似）。

4） grep 在文件中查找需要的行

grep [指定字符串] [源文件]

-v 反选 例如：grep –v “^#” 不以#开始的行

7.查看内存使用情况

格式: free

8.查看本机系统内核

格式: uname -a

9.查看当前路径下的文件/文件夹大小

格式: du -hs 文件名/文件夹名

10.变更为其它使用者的身份

格式: su 使用者帐号
范例: su work

11.远程登陆

格式: ssh 用户名@机器名
范例: ssh rd@build01

12.不挂断地运行命令

格式: nohup command &
范例: nohup ./build_index.sh -d ../newdbi/ &

13.查看文件的行数

格式: wc -l 文件名
范例: wc -l as.conf

14.比较两个文件的不同之处

格式: vimdiff 文件 1 文件 2
范例: vimdiff 1.txt 2.txt

15.杀死进程

格式: killall -9 进程名
范例: killall -9 bs.se

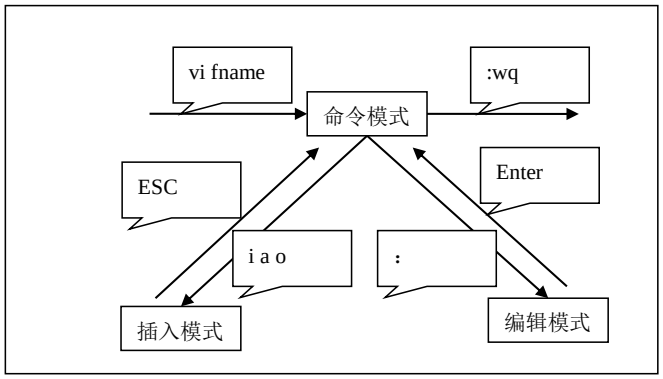
16.在 SecureCRT 中把文件传给本机的 SecureCRT 安装目录的 download 目录下

格式: sz 文件名
范例: sz 1.txt

17.把本机文件传给 SecureCRT 中当前机器的当前目录

格式: rz

18.Vi 文本编辑器



开始进入是命令模式；i a o 进入插入模式，ESC 可以退出插入模式；：可以进入编辑模式，编辑完成回车退回命令模式。

1）插入命令

命令	作用

a	在光标后附加文本
A	在本行行末附加文本
i	在光标前插入文本
I	在本行开始插入文本
o	在光标下插入文本（下一行）
O	在光标上插入文本（上一行）

2） 定位

命令	作用
h	左
j	下
k	上
l	右
$	行首
0	行首
H	屏幕上端
M	中间
L	下端
:set nu	设置行号
:set nonu	取消行号
gg	第一行
G	最后一行
nG	第 n 行
:n	第 n 行

3） 删除

命令	作用
x	删除光标所在处的字符
nx	删除从光标开始起后面 n 个字符
dd	删除行
ndd	删除光标所在行开始一下 n 行
dG	删除光标所在行到到文件末尾的所有行
D	删除光标所在位置到行末的内容
:n1,n2d	删除 n1 到 n2 的行

4） 复制剪切

命令	作用
yy、Y	复制当前行
nyy、nY	复制当前行一下 n 行
dd、ndd	剪切当前行、剪切当前行一下 n 行
p、P	粘贴在当前行下面、粘贴在当前行上面

5） 替换和取消

命令	作用
r	取代光标所在处的字符（不用进入插入模式）
R	从光标所在处开始替换直到按 Esc 结束（不用进入插入模式）
u	取消上一步操作

6） 搜索和替换

命令	作用
/string	查找 string
n	光标跳到下一个找到的 string（从前往后，N 从后往前）
:set ic	忽视大小写、set noic 关闭忽略大小写
:%s/old/new/g	全文范围内，将 old 字符替换为 new
:n1,n2s/old/new/g	将 n1 到 n2 的 old 替换为 new

注意：:1,2s/^/#/g 将第一行和第二行开头加上#号；^代表行首、$代表行尾

7） 保存退出

:wq 保存退出 :wq!强行保存退出（文件的所有者以及 root 才可以用）

shift+ZZ 保存退出

:q! 退出不保存

8） vi 其他用法

:r filename 将 filename 文件的内容导入到当前文件中

:! ls /etc 在 vi 中编辑模式，键入:! 然后可以执行命令，不需要退出 vi

19.察看文件类型

格式：file 文件名

20.压缩命令

1） gzip 将文件压缩为.gz（只能压缩文件，不能压缩目录；不能保留源文件）

gzip [选项] 文件名

gzip newfile 可以得到 newfile.gz 而原来的 newfile 文件不存在了。

解压缩：gunzip newfile.gz 解压 newfile.gz 得到 newfile 文件

2） tar 打包文件，将一个目录打包为一个 tar 包，可以与 gzip 结合压缩目录。

tar [选项] [打包以后的文件名] [对那个目录进行打包]

选项：-c 产生打包文件 -v 显示详细信息 –f 指定打包后的文件名 –z 打包的同时压缩

tar –zcvf dir.tar.gz dir 将 dir 打包并压缩为名为 dir.tar.gz

解压缩：将-c 换成-x 即可 tar –zxvf dir.tar.gz

3） zip 将文件打包为 zip 文件，windows 和 linux 默认统一支持的压缩格式。

zip file.zip file（压缩文件）

zip –r dir.zip dir（压缩目录）

解压缩：unzip dir.zip

4） bzip2 将文件压缩为.bz2 文件，只能压缩文件，跟 gzip 类似，但是可以添加-k 选项，从而保留原文件;

bzip2 –k a.bz2 a.txt 得到：a.bz2

解压缩：bunzip2 a.bz2

总结：linux 下一共有四种压缩方式，得到的后缀分别为：.gz/.tar.gz/.zip/.bz2；其中 tar.gz 是先打包再压缩的。对应的四种命令是：gzip/tar -c/zip/bzip2

对应的解压缩为：gunzip/tar –x/unzip/bunzip2

21.非交互性文本流编辑器

格式：sed [选项] ‘命令’ 输入文件名

参数：

-n 不打印，sed 不写编辑行到标准输出，缺省为打印所有行

-c 下一命令是编辑命令，使用多项编辑时加入此选项。如果只用到一条 sed 命令，此选项无用，但指定它也没有关系。

-f 如果正在调用 sed 脚本文件，使用此选项。此选项通知 sed 一个脚本文件支持所有的 sed 命令，例如：sed -f myscript.sed input_file，这里 myscript.sed 即为支持 sed 命令的文件。

范例：

sed –n ‘2p’ quota.txt 打印文本的第二行

sed –n ‘/Neave/’p quota.txt 匹配单词 Neave，并打印此行

sed ‘1d’ quota.txt 删除文本的第一行

22.将文件中的记录分类

格式：sort [选项] –o 输出文件名 [其他选项] +pos1 +pos2 输入文件名

参数：

-c 测试文件是否已经分类

-m 合并两个分类文件

-u 删除所有复制行

-o 存储 sort 结果的输出文件名

其它参数有：

-b 使用域进行分类时，忽略第一个空格。

-n 指定分类是域上的数字分类。

-t 域分隔符；用非空格或 t a b 键分隔域。

-r 对分类次序或比较求逆。

+n n 为域号。使用此域号开始分类。

n n 为域号。在分类比较时忽略此域，一般与+ n 一起使用。

pos1，pos2 传递到 m，n。m 为域号，n 为开始分类字符数；例如 4，6 意即以第 5 域分类，从第 7 个字符开始。

范例：sort -t: +2n video.txt 以“：”为域分割符，按第三个域数字分类 video.txt

23.去除文本文件的重复行

格式：uniq [选项] 输入文件 [输出文件]

参数：

-u 只显示不重复行。

-d 只显示有重复数据行，每种重复行只显示其中一行

-c 打印每一重复行出现次数。

-f n 为数字，前 n 个域被忽略。一些系统不识别- f 选项，这时替代使用- n。

范例：uniq –n2 parts.txt

24.连接两个文本文件

格式：join [选项] 文件 1 文件 2

说明：join 用来将来自两个文本文件的行连在一起，两个输入文件必须已经分过类，每个文件里都有一些元素与另一个文件相关，由于这种关系，join 将两个文件连在一起。

参数：

-an n 为一数字，用于连接时从文件 n 中显示不匹配行。例如， - a 1 显示第一个文件的不匹配行，- a 2 为从第二个文件中显示不匹配行。

-o n.m n 为文件号，m 为域号。1 . 3 表示只显示文件 1 第三域，每个 n，m 必须用逗号分隔，如 1 . 3，2 . 1。

-j n m n 为文件号，m 为域号。使用其他域做连接域。

-t 域分隔符。用来设置非空格或 t a b 键的域分隔符。例如，指定冒号做域分隔符- t：

范例： join –o 1.1,2.2 names.txt town.txt

25.从文本文件中剪切列或域

格式：cut [选项] 文件名

参数：

-c list 指定剪切字符数。

-f field 指定剪切域数。

-d 指定与空格和 tab 键不同的域分隔符。

- c 用来指定剪切范围，如下所示：

- c 1，5-7 剪切第 1 个字符，然后是第 5 到第 7 个字符。

-f 格式与- c 相同。

-f 1，5 剪切第 1 域，第 5 域。

范例： cut –d: -f3 pers

26.将两个文本文件粘贴在一起

格式：paste [options] file1 file2

说明：粘贴两个不同来源的数据时，首先需将其分类，并确保两个文件行数相同。paste 将按行将不同文件行信息放在一行。缺省情况下， paste 连接时，用空格或 tab 键分隔新行中不同文本，除非指定-d 选项，它将成为域分隔符。

参数：

-d 指定不同于空格或 t a b 键的域分隔符。例如用@分隔域，使用- d @。

-s 将每个文件合并成行而不是按行粘贴。

范例： paste –d: pas2 pas1 用冒号作分割符

27.将大文件进行分割

格式：split [选项] 输入文件名 输出文件名前缀
参数：-a 指定后缀的长度（缺省为 2）
-b 每个输出文件的字节大小
-l 每个输出文件的行数
范例：split –l 1000 bigfile.txt outfile

28.linux 常用命令

1. cd

格式 : cd [dirName]
说明 : 变换工作目录至 dirName。 其中 dirName 可为绝对路径或相对路径。若目录命令省略，则变换至使用者登录时所在的目录（ home directory ）。另外，"~" 也表示为 home directory 的意思，"." 则是表示当前所在的目录，".." 则表示当前目录位置的上一层目录。
范例 :

1. 跳到当前目录的子目录（如：usr/bin）当中:
cd /usr/bin

2. 跳到自己的 home directory :
cd ~ （或 cd ）

3. 跳到当前目录的上上两层 :
cd ../..

4. 跳到父目录下的另外一个目录（如：conf）:
cd ../conf

2. pwd

格式 : pwd
说明 : 显示当前所在目录

3. ls

格式 : ls [-alrtAFR] [name...]
说明 : 显示指定工作目录下之内容（列出目前工作目录所含之档案及子目录)。

-a 显示所有文件及目录 (包括"."开头的文件)
-l 除文件名外，还将文件类型、权限、拥有者、文件大小等信息详细列出
-r 将文件以相反次序显示(原定依英文字母次序)
-t 将文件依建立时间之先后次序列出
-A 同 -a ，但不列出 "." (当前目录) 及 ".." (父目录)
-F 在列出的文件命令后加一符号；例如可执行文件则加 "*", 目录则加 "/"
-R 若目录下有文件，则以下之档案亦皆依序列出

范例：
1. 列出当前工作目录下所有命令是 s 开头的文件，愈新的排愈后面 ：
ls -ltr s*
2. 将 /bin 目录以下所有目录及文件详细资料列出 ：
ls -lR /bin

3. 列出当前工作目录下所有文件及目录；目录后加 "/", 可执行文件后加"*" :
ls -AF

4. passwd

格式：passwd [-l|-u] [username]
格式：passwd [-x max] [-n min] [-w warn] [-i inact] [username]
格式：passwd [username]
说明：用来更改使用者的密码。当具备 root 权限的使用者时，可以设置[username]的密码、更改使用者密码的有效期、锁定账户和解除锁定等。

-l:锁定账户[username]，不更改原有密码，使账户[username]不能登录
-u:解除锁定
-x:要求更改密码的最大天数
-n:允许更改密码的最小天数
-w:密码失效之前的警告天数（即提前几天警告账户密码将失效）
-i:密码失效之后多少天，账户失效
-d 关闭使用者的密码认证功能, 使用者在登入时将可以不用输入密码,
-S 显示指定使用者的密码认证种类,
[username] 指定帐号名称.

范例：
1. 锁定账户: lijiangong ， 使他不能登录

passwd -l lijiangong

2. 解除锁定:

passwd -u lijiangong

3. 设定密码有效期

passwd -x10 -n3 -w2 -i5 lijiangong

即：3 天之后才能更改密码、10 天之后必须更改密码、第八天提醒修改密码、密码失效 5 天之后账户失效

5. who

格式 : who - [husfV] [user]
说明 ： 显示有哪些用户登录到系统中，显示的信息包含用户 ID，使用的终端，上线时间，呆滞时间，CPU 使用量，动作等等。
参数说明 ：

-H : 显示标题列
-u : 显示用户的闲置时间
-s : 使用简短的格式来显示
--version : 显示程式版本

相关命令 : whoami
说明 ： 显示当前用户是谁

6. cat

格式：cat [-AbeEnstTuv] [--help] [--version] fileName
说明：把文件串连接后输出到荧幕或加 > fileName 到另一个档案
参数：

-A 等价于 -vET
-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
-e 等价于 –vE
-E 每行末尾显示一个$符号
-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
-t 等价于 –vT
-T 显示制表符为 ^I
-v 或 --show-nonprinting, dos 格式的回车换行显示为^M

范例：

1. 把 textfile1 的文件内容加上行号后输入到 textfile2 文件里：

cat -n textfile1 > textfile2

2. 把 textfile1 和 textfile2 的文件内容加上行号（空白行不加）之后将内容附加到 textfile3 ：

cat -b textfile1 textfile2 >> textfile3

（ > 为重定向操作符， >>为重定向追加操作符 ）

7. mkdir

格式： mkdir dirName
说明： 创建目录。

范例：

1. 在当前目录下创建一子目录，名为 AAA:

mkdir AAA

8. rmdir

格式： rmdir [-p] dirName
说明： 删除空的目录。
参数： -p 是当子目录被删除后使它也成为空目录时，则顺便一并删除。
范例：

1. 将当前目录下，名为 AAA 的子目录删除 ：

rmdir AAA

2. 在当前目录下的 BBB 目录中，删除名为 Test 的子目录。若 Test 删除后，BBB 目录成为空目录，则 BBB 亦予删除。

rmdir -p BBB/Test

9. chmod

格式 ：chmod [-cfvR] [--help] [--version] mode file...
说明 ： Linux/Unix 的文件存取权限分为三级 ： 文件拥有者、组、其他。利用 chmod 控制文件的存取权限。
参数 ：

mode ： 权限设定字串，格式如下 ： [ugoa...][[+-=][rwx]...][,...]，其中 u 表示该文件的拥有者，g 表示与该文件的拥有者属于同一个组(group)者，o 表示其他以外的人，a 表示这三者皆是。

+ 表示增加权限、- 表示取消权限、= 表示唯一设定权限。

r 表示可读取，w 表示可写入，x 表示可执行。X 表示只有当该档案是个子目录

或者该档案已经被设定过为可执行。

-c：若该档案权限确实已经更改，才显示其更改动作

-f：若该档案权限无法被更改也不要显示错误讯息

-v：显示权限变更的详细资料

-R：对目前目录下的所有档案与子目录进行相同的权限变更(即以递回的方式逐个变更)

--help：显示辅助说明

--version：显示版本

范例：

1. 将文件 file1.txt 设为所有人皆可读取：

chmod ugo+r file1.txt 或 chmod a+r file1.txt

2. 将文件 file1.txt 与 file2.txt 设为该文件拥有者，与其所属同一个组的人可写入，但其他以外的人则不可写入：

chmod ug+w,o-w file1.txt file2.txt

3. 将 ex1.py 设定为只有该文件拥有者可以执行：

chmod u+x ex1.py

4. 将目前目录下的所有档案与子目录皆设为任何人可读取：

chmod -R a+r *

10. chown

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

说明：利用 chown 可以将文件的拥有者加以改变。

参数：

user：新的档案拥有者的使用者

IDgroup：新的档案拥有者的使用者群体(group)

-c：若该档案拥有者确实已经更改，才显示其更改动作

-f：若该档案拥有者无法被更改也不要显示错误讯息

-h：只对于连结(link)进行变更，而非该 link 真正指向的档案

-v：显示拥有者变更的详细资料

-R：对目前目录下的所有档案与子目录进行相同的拥有者变更(即以递回的方式逐个变更)

--help：显示辅助说明

--version：显示版本

范例：

1. 将文件 file1.txt 的拥有者设为 users 组的使用者 jessie：

chown jessie:users file1.txt

2. 将当前目录下的所有文件与子目录的拥有者都设为 users 组的使用者 lamport :
chmod -R lamport:users *

11. chgrp

格式 : chgrp [-R] group file...
说明 : 改变文件的所属的组。
参数 :
-R : 对当前目录下的所有文件与子目录（包括子目录下的文件）进行相同的变更

范例 :
1. 将文件 file1.txt 的所属组设为 users 组:
chgrp users file1.txt

2. 将当前目录下的所有文件与子目录（包括子目录下的文件）都设为 bin 组:
chmod -R bin *

批注 [h1]: chgrp

12. touch

格式: touch [-acfm]
[-r reference-file] [--file=reference-file]
[-t MMDDhhmm[[CC]YY][.ss]]
[-d time] [--date=time] [--time={atime,access,use,mtime,modify}]
[--no-create] [--help] [--version]
file1 [file2 ...]
说明：touch 指令改变文件的时间记录。 ls -l 可以显示文件的时间记录。
参数:
-a 改变文件的读取时间记录。
-m 改变文件的修改时间记录。
-c 假如目的文件不存在，不会建立新的文件。与 --no-create 的效果一样。
-r 使用参考文件的时间记录，与 reference-file 的效果一样。
-d 设定时间与日期，可以使用各种不同的格式。
-f 不使用，是为了与其他 unix 系统的相容性而保留。
-t 设定档案的时间记录，格式与 date 指令相同。
--no-create 不会建立新档案。
--help 列出指令格式。
--version 列出版本讯息。

范例：

1. 最简单的命令格式，将文件的创建时间改为现在的时间。若文件不存在，系统会建立一
个新的文件。

touch file

touch file1 file2

2. 将 file 的时间记录改变成与 referencefile 一样。

touch -r referencefile file

3. 将 file 的时间记录改成 5 月 6 日 18 点 3 分，公元两千年。时间可以使用 am, pm 或是 24 小时的格式，日期可以使用其他格式如 6 May 2000 。

touch -d "6:03pm" file

touch -d "05/06/2000" file

touch -d "6:03pm 05/06/2000" file

13. cp

格式：cp [-arf] source dest

cp [-arf] source... directory

说明：将一个文件拷贝至另一文件，或将数个文件拷贝至另一目录。

参数:

-a 将文件状态、权限等信息都照原状予以复制。

-r 若 source 中含有目录名，则将目录下的文件顺序拷贝至目的地。

-f 若目的地已经有相同的文件名存在，则在复制前先予以删除再行复制。

范例：

1. 将文件 aaa 复制一份名字为 bbb 的文件:

cp aaa bbb

2. 将当前目录下的所有 C 程序拷贝到当前目录下的 Finished 子目录中 ：

cp *.c Finished

此外，还有一条命令是远程拷贝 scp

14. mv

格式： mv [-i] source dest

mv [-i] source... directory

说明：将一个文件改名为另一文件，或将数个文件移至另一目录。

参数：-i 若目的地已有同名文件，则先询问是否覆盖旧文件。

范例：

1. 将文件 aaa 改名为 bbb :
mv aaa bbb

2. 将所有的 C 程序移至 Finished 子目录中 :
mv -i *.c Finished

15. rm

格式：rm [-ifr] name...
说明：删除文件及目录。
参数:
-i 删除前逐一询问确认。
-f 即使原文件属性设为只读，也直接删除，无需逐一确认。
-r 将目录及以下之文件逐一删除。

范例：
1. 删除所有 C 程序文件并删除前逐一询问确认 :
rm -i *.c

2. 将 Finished 子目录及子目录中所有文件删除 :
rm -r Finished

16. find

各式 : find [path...] [expression]
说明 : 将符合 expression 的文件列出来。
-amin n : 在过去 n 分钟内被读取过的文件
-anewer file : 比文件 file 更晚被读取过的文件
-atime n : 在过去 n 天被读取过的文件
-cmin n : 在过去 n 分钟内被修改过的文件
-cnewer file :比文件 file 更新的文件
-ctime n : 在过去 n 天过修改过的文件
-name filename, -iname filename : 符合 filename 的文件。iname 会忽略大小写
-size n : 档案大小 是 n 单位，b 代表 512 位元组的区块，c 表示字元数，k 表示 kilo bytes，w 是二个位元组。-type c : 档案类型是 c 的档案。

范例:
1. 将当前目录及其子目录下所有扩展名是 c 的文件列出来。
find . -name "*.c"

2. 将当前目录及其子目录下所有最近 20 分钟内更新过的文件列出
find . -cmin -20

17. grep

格式：grep [-no] pattern files
说明：搜索字符串命令
参数：
-n 显示行号
-o 只显示匹配的串
范例：
1. grep printf *
file1.c: printf("\nHello\n");
file2.c: printf("\nSample\n");

grep -n printf *
file1.c:4 printf("\nHello\n");
file2.c:9 printf("\nSample\n");

grep -o printf *
file1.c: printf
file2.c: printf

2. 如果搜索的串中有空格，则用引号括起来
grep "asd abc" *

18. more

格式：more [-num] [+linenum] [fileNames..]
说明：类似 cat ，不过是以一页一页的方式显示。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。
参数：-num 一次显示的行数
+linenum 从第 num 行开始显示
fileNames 欲显示内容的文件，可为多个文件
范例：
1. 从第 20 行开始显示 testfile 之文件内容。
more +20 testfile

19. less

格式： less [Option] filename
说明： less 的作用与 more 十分相似，都可以用来浏览文本文件的内容，不同的是 less 允许使用者往回卷动（PageUp PageDown）以浏览已经看过的部份，同时因为 less 并未在一开始就读入整个文件，因此在遇上大型文件的开启时，会比一般的文本编辑器(如 vi) 来 的快速。

20. head

格式：head [-n|c num] fileName
说明：显示文件头部内容。没有参数时，显示最前 10 行
参数：
-n num 显示最前 num 行
-c num 显示最前 num 字符

21. tail

格式：tail [-n|c num] fileName
说明：显示文件尾部内容。没有参数时，显示最后 10 行
参数：
-n num 显示最后 num 行
-c num 显示最后 num 字符
-f 跟踪，随文件增长显示新的内容。（用 Ctrl+C 退出）

22. cut

格式：cut -cnum1-num2 filename
说明：显示每行从开头算起第 num1 到 num2 的字符。

范例：

```
1. shell>> cat example
test2
this is test1
```

显示每行开头算起前 6 个字符
shell>> cut -c0-6 example
test2
this i

23. at

格式 : at TIME command
说明 : 指定在 TIME 这个特定时刻执行某个程序或指令，TIME 的格式是 HH:MM，其中的 HH 为小时，MM 为分钟，甚至你也可以指定 am, pm, midnight, noon, teatime(就是下午 4 点)等口语词。

如果想要指定超过一天内的时间，则可以用 MMDDYY 或者 MM/DD/YY 的格式，其中 MM 是月份，DD 是第几日，YY 是指年份。另外，使用者甚至也可以使用是 now + 时间间隔来弹性指定时间，其中的时间间隔可以是 minutes, hours,days, weeks

另外，使用者也可指定 today 或 tomorrow 来表示今天或明天。当指定了时间并按下 enter 之后，at 会进入交谈模式并要求输入指令或程式，当你输入完后按下 ctrl+D 即可完成所有动作。

范例 :

1. 三天后的下午 5 点钟执行 /bin/ls :
at 5pm + 3 days /bin/ls
2. 三个星期后的下午 5 点执行 /bin/ls :
at 5pm + 2 weeks /bin/ls
3. 明天的 17:20 执行 /bin/date :
at 17:20 tomorrow /bin/date
4. 1999 年的最后一天的最后一分钟印出 hello world !
at 23:59 12/31/1999 hello world !

24. crontab

格式 :
crontab [-u user] filecrontab [-u user] { -l | -r | -e } cmd
说明 :
crontab 是用来在固定时间或固定间隔执行程序。
-u user 设定指定 user 的时间表，这个前提是你必须要有其权限(比如说是 root)才能够指定他人的时程表。如果不使用 -u user ，就是表示设定自己的时间表。

参数 :
-e : 执行文字编辑器来设定时程表，默认的文字编辑器是 VI
-r : 删除当前的时程表

-l：列出当前的时程表

时程表的格式如下 ：

f1 f2 f3 f4 f5 program

其中 f1 是表示分钟，f2 表示小时，f3 表示一个月份中的第几日，f4 表示月份，f5 表示一个星期中的第几天（0~6，0 为星期天）。cmd 表示要执行的程序。

当 f1 为 * 时表示每分钟都要执行，f2 为 * 时表示每小时都要执行，其余类推；

当 f1 为 a-b 时，表示从第 a 分钟到第 b 分钟这段时间内要执行，f2 为 a-b 时，表示从第 a 到第 b 小时都要执行，其余类推；

当 f1 为 */n 时，表示每 n 分钟个时间间隔执行一次，f2 为 */n 表示，每 n 小时个时间间隔执行一次，其余类推；

当 f1 为 a, b, c,... 时，表示第 a, b, c,... 分钟要执行，f2 为 a, b, c,... 时表示，第 a, b, c...个小时要执行，其余类推；

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

范例 ：

1. 每月每天每小时的第 0 分钟执行一次 /bin/ls :

0 * * * * /bin/ls

2. 在 12 月内, 每天的早上 6 点到 12 点中，每隔 20 分钟执行一次 /usr/bin/backup :

0/20 6-12 * 12 * /usr/bin/backup

3. 周一到周五每天下午 5:00 执行一次 /bin/ls :

0 17 * * 1-5 /bin/ls

4. 每月每天的午夜 0 点 20 分, 2 点 20 分, 4 点 20 分....执行 echo "haha"

20 0-23/2 * * * echo "haha"

25. sleep

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

说明 : sleep 可以用来将当前动作延迟一段时间

参数 ：

--help : 显示辅助讯息

--version : 显示版本编号

number : 时间长度，后面可接 s、m、h 或 d, 其中 s 为秒，m 为 分钟，h 为小时，d 为日数

范例 ：

1. 显示当前时间后延迟 1 分钟，之后再次显示时间 ：

date;sleep 1m;date

26. mesg

格式： mesg [y|n]
说明： 决定是否允许其他人传讯息到自己的终端机介面
参数：
y：允许讯息传到终端机介面上。
n：不允许讯息传到终端机介面上 。

例子 ：
1. 改变当前讯息设定，改成不允许讯息传到终端机介面上 ：
mesg n

27. wall

格式 ：
wall [message]
说明：
wall 会将信息传给每一个 mesg 设定为 yes 的上线使用者。当使用终端界面做为标准输入时，信息结束时需加 Ctrl+D

范例 ：
1. 传讯息"hi" 给每一个使用者 ：
wall hi

28. write

格式 ：write user
说明 ：给其他使用者写信息
参数 ：user：接收信息者
范例：
1. 传信息给 Rollaend
write Rollaend

接下来就是将讯息打上去，结束按 ctrl+c
注： 若对方设定 mesg n，则此时信息将无法传给对方

29. kill

格式： kill [-s signal] pid ...
kill -l [signal]
说明：kill 送出一个特定的信号 (signal) 给进程号为 pid 的进程。根据该信号而做特定的动作，若没有指定,默认是送出终止 (TERM) 信号

参数：
-s (signal)：其中常用的一个信号(9) 杀死进程；详细的信号可以用 kill -l
-l (signal)：列出所有可用的信号名称

范例：
1. 将 pid 为 323 的进程杀死 ：
kill -9 323
2. 将 pid 为 456 的行程重跑 (restart)：
kill -HUP 456

30. ps

格式：ps [options] [--help]
说明：显示进程的名称、占用资源、状态等
参数：
ps 的参数非常多，在此仅列出 3 个
-A 列出所有的行程
-e 列出所有的进程
-f 显示详细的信息（包括命令行参数）

范例：
ps
PID TTY TIME CMD
2791 ttyp0 00:00:00 tcsh
3092 ttyp0 00:00:00 ps
% ps -A
PID TTY TIME CMD
1 ? 00:00:03 init
2 ? 00:00:00 kflushd
2 ? 00:00:00 kflushd
3 ? 00:00:00 kpiod
4 ? 00:00:00 kswapd
5 ? 00:00:00 mdrecoveryd

.....

pstree

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

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

参数：

-a 显示该行程的完整指令及参数，如果是被记忆体置换出去的行程则会加上括号

-c 如果有重覆的行程名，则分开列出 (预设值是会在前面加上 *

-c 如果有重覆的行程名，则分开列出 (预设值是会在前面加上 *

范例：

1. pstree

```
init-+-amd
|-apmd
|-atd
|-httpd---10*[httpd]
%pstree -p
init(1)-+-amd(447)
|-apmd(105)
|-atd(339)
%pstree -c
init-+-amd
|-apmd
|-atd
|-httpd-+-httpd
| |-httpd
| |-httpd
| |-httpd
....
```

31. top

格式：top

说明：显示 CPU 的使用率、内存大小、内存使用率、进程状态等

32. expr

格式： expr [option] expresion
说明：字符串处理命令
范例：

1 求字串长度
shell>> expr length "this is a test"
14

2. 数字运算
shell>> expr 14 +[-*/%] 9
13[5、126、1、5]

3. 从位置处抓取字串
shell>> expr substr "this is a test" 3 5
is is

4. 字符串首次出现的索引值
shell>> expr index "testforthegame" e
2

5. 字串真实重现
shell>> expr quote thisisatestformela
thisisatestformela

33. locate

格式： locate [-q] [-d] [--database=]
locate [-r] [--regexp=]
locate [-qv] [-o] [--output=]
locate [-Vh] [--version] [--help]
说明：
locate 让使用者可以很快速的搜寻档案系统内是否有指定的档案。其方法是先建立一个包括系统内所有档案名称及路径的资料库，之后当寻找时就只需查询这个资料库，而不必实际深入档案系统之中了。在一般的 distribution 之中，资料库的建立都被放在 contab 中自动执行。一般使用者在使用时只要用
locate your_file_name 的型式就可以了。
参数：
-n 至多显示 个输出。

范例：

1. locate chdrv： 寻找所有叫 chdrv 的档案
2. locate -n 100 a.out： 寻找所有叫 a.out 的档案，但最多只显示 100 个

34. split

格式：split [OPTION] [INPUT [PREFIX]]

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

参数：

-b, --bytes=SIZE

-b, --bytes=SIZE SIZE 值为每一输出档案的大小，单位为 byte。

-C, --line-bytes=SIZE 每一输出档中，单行的最大 byte 数。

-l, --lines=NUMBER NUMBER 值为每一输出档的列数大小。

-NUMBER 与 -l NUMBER 相同。

--verbose 于每个输出档被开启前，列印出侦错资讯到标准错误输出。

--help 显示辅助资讯然后离开。

--version 列出版本资讯然后离开。

SIZE 可加入单位: b 代表 512， k 代表 1K， m 代表 1 Meg。

范例：

1. PostgresSQL 大型资料库备份与回存：

因 Postgres 允许表格大过你系统档案的最大容量，所以要将表格 dump 到单一的档案可能会有问题，使用 split 进行档案分割。

% pg_dump dbname | split -b 1m - filename.dump.

重新载入

% createdb dbname

% cat filename.dump.* | pgsql dbname

% cat filename.dump.* | pgsql dbname

35. ln

格式： ln [options] source dist，

其中 option 的格式为：

[-bdfinsvF] [-S backup-suffix] [-V {numbered,existing,simple}] [--help] [--version] [--]

说明：Linux/Unix 档案系统中，有所谓的连结(link)，我们可以将其视为档案的别名，而连结又可分为两种：硬连结(hard link)与软连结(symbolic link)的别名，而连结又可分为两种：硬连结(hard link)与软连结(symbolic link)，硬连结的意思是一个档案可以有多个名称，而软连结的方式则是产生一个特殊的档案，该档案的内容是指向另一个档案的位置。硬连结是存在同一个档案系统中，而软连结却可以跨越不同的档案系统。ln source dist 是产生一个连结(dist)到 source，至于使用硬连结或软链结则由参数决定。不论是硬连结或软链结都不会将原本的档案复制一份，只会占用非常少量的磁碟空间。

参数：

-f : 链结时先将与 dist 同档名的档案删除
-d : 允许系统管理者硬链结自己的目录
-i : 在删除与 dist 同档名的档案时先进行询问
-n : 在进行软连结时，将 dist 视为一般的档案
-s : 进行软链结(symbolic link)
-v : 在连结之前显示其档名
-b : 将在链结时会被覆写或删除的档案进行备份
-S SUFFIX : 将备份的档案都加上 SUFFIX 的字尾
-V METHOD : 指定备份的方式
--help : 显示辅助说明
--version : 显示版本

范例 ：

1. 将档案 yy 产生一个 symbolic link : zz

ln -s yy zz

2. 将档案 yy 产生一个 hard link : zz

ln yy xx

36. login 、logout

37. exit

38. man，info

man、info ：在线手册命令
格式：man command
范例：man cat [info cat]

39. alias

说明：建别名
参数：alias aaa="ls –l"

39. unalias

说明：取消别名
参数：unalias aaa

40. halt

说明：停机命令

41. shutdown

格式：shutdown [-t sec] [-hrk] time [message]
参数：
-t sec 告诉 init 守护进程等待 sec 秒后才发警告 信息、终止其他进程
-h 停机
-r 重起
-k 仅发警告信息，不停机
time 停机前等待时间。格式 1 hh:mm (绝对时间 小时:分钟)
格式 2 +m (m 是分钟数)
格式 3 now (=+0)
warning message 警告信息

43. reboot

说明：重起命令

44. clear

说明：清屏命令