1）Linux查看系统版本

     cat /proc/version

2) 文件末尾追加

     echo >> filename

    文件覆盖

     echo > filename

3) 网络流量：

sar -n DEV interval count

sar -n DEV 1 4

4)搜索

grep -rn "text" ./\* | grep -v "pretext"

find . \* | xargs grep "aaaaa"就是在当前目录下查找包含字符串aaaaa的文件；其中.表示查找的目录是当前目录，

        \*表示查找所有类型的文件，|是管道命令，表示将前面的输出作为后面的输入，xargs表示参数，grep是过滤；

和上面类似的有个sed命令，一般用在shell脚本里面查找，替换，删除或者按照某种正则来匹配字符串：比如sed -i "/aaa/d" test.xml 表示将test.xml中的aaa这行删掉；

 sed -i "/first line/a second line"表示在first line后加一行second line；

find . -name CA.sh 2>/dev/null

5)

变量说明:

$$

Shell本身的PID（ProcessID）

$!

Shell最后运行的后台Process的PID

$?

最后运行的命令的结束代码（返回值）

$-

使用Set命令设定的Flag一览

$\*

所有参数列表。如"$\*"用「"」括起来的情况、以"$1 $2 … $n"的形式输出所有参数。

$@

所有参数列表。如"$@"用「"」括起来的情况、以"$1" "$2" … "$n" 的形式输出所有参数。

$#

添加到Shell的参数个数

$0

Shell本身的文件名

$1～$n

添加到Shell的各参数值。$1是第1参数、$2是第2参数…。

**1 字符串判断**

str1 = str2　　　　　　当两个串有相同内容、长度时为真

str1 != str2　　　　　 当串str1和str2不等时为真

-n str1　　　　　　　 当串的长度大于0时为真(串非空)

-z str1　　　　　　　 当串的长度为0时为真(空串)

str1　　　　　　　　   当串str1为非空时为真

**2 数字的判断**

int1 -eq int2　　　　两数相等为真

int1 -ne int2　　　　两数不等为真

int1 -gt int2　　　　int1大于int2为真

int1 -ge int2　　　　int1大于等于int2为真

int1 -lt int2　　　　int1小于int2为真

int1 -le int2　　　　int1小于等于int2为真

**3 文件的判断**

-r file　　　　　用户可读为真

-w file　　　　　用户可写为真

-x file　　　　　用户可执行为真

-f file　　　　　文件为正规文件为真

-d file　　　　　文件为目录为真

-c file　　　　　文件为字符特殊文件为真

-b file　　　　　文件为块特殊文件为真

-s file　　　　　文件大小非0时为真

-t file　　　　　当文件描述符(默认为1)指定的设备为终端时为真

**3 复杂逻辑判断**

-a 　 　　　　　 与

-o　　　　　　　 或

!　　　　　　　　非

函数调用：

fSum 3 2;

 function fSum()

 {

     echo $1,$2;

     return $(($1+$2));

 }

 fSum 5 7;

 total=$(fSum 3 2);

 echo $total,$?;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1、查找

find  -name ap\* -o -name may\*  查找以ap或may开头的文档

grep:

-I ：忽略大小写

-v查找不包含匹配项的行

过滤掉Permission denied的文件或目录：

find . -name CA.sh 2>/dev/null

查找最近30分钟内修改过的文件：

find . name CA.sh -type f -mmin -30

查找最近30分钟内修改过的文件的详细情况：

find . name CA.sh -type f -mmin -30 ls

查找最近一天内修改过的常规文件：

find . name CA.sh -type f -mtime -1

最近一天前（2天内）修改过的：

find . name CA.sh -type f -mtime +1

find -name aaa.txt就是查找文件名为aaa.txt的文件；

find . \* | xargs grep "aaaaa"就是在当前目录下查找包含字符串aaaaa的文件；其中.表示查找的目录是当前目录，

        \*表示查找所有类型的文件，|是管道命令，表示将前面的输出作为后面的输入，xargs表示参数，grep是过滤；

和上面类似的有个sed命令，一般用在shell脚本里面查找，替换，删除或者按照某种正则来匹配字符串：比如sed -i "/aaa/d" test.xml 表示将test.xml中的aaa这行删掉；

 sed -i "/first line/a second line"表示在first line后加一行second line；

查看内存使用情况：free

查看磁盘占用情况：df

查看某个目录大小：du -h /home/testdir

查找并kill进程：

ps -ef | grep wsgi | grep -v grep | cut -c 9-15 | xargs kill -s 9

2、挂载：mount  基本用法：

mount － t 文件系统类型 设备文件名 装载目录

3、

sftp -oPort=300 username@ip或hostname

4、排序，文本处理（注意这个不是文件排序而是文本内容的处理）：

sort -参数 1.txt  表示对1.txt中的所有行排序

sort -f 1.txt -f是指大小写不敏感

 -r 表示逆序

5、

more让画面在显示满一页时暂停，此时可按空格健继续显示下一个画面，或按Q键停止显示；

less与 more 十分相似，不同的是 less 允许使用者往回卷动以浏览已经看过的部份，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

6、解压压缩

filename.tar.gz:

解压：tar -zxvf filename.tar.gz

压缩：tar -zcvf filename.tar.gz

只打包不压缩：

tar -cvf code.tar code/

压缩：

tar -zcvf code.tar.gz code/

tar -jcvf code.tar.bz2 code/

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

7、

chmod 777 a.txt

777的意思：7就是111，rwx读写执行的意思，第一个7是当前用户，第二个7是用户所在的group，第三个表示others就是其他用户了；

      比如某个文件没有执行权限，要给它加上时可以这么写 chmod a+x a.sh

                                     加写和执行权限    chmod a+wx a.sh

                                     去除它的执行权限  chmod a-x a.sh

                                     去除group的执行权限 chmod g-x a.sh

                                     去除other执行权限     chmod o-x a.sh

                                 要给某个目录所有文件都加上写权限就可以这样：chmod -R a+w yourdir  （前面说过了-R表示递归的意思）

 改变文件的所属：

 chown -参数 所有者:组 文件名

chown -v root:users file1.txt

 -v表示显示详细处理信息

查看文件行数：

awk '{print NR}' liushi.txt | tail -n1

8、创建链接：

ln  链接的意思：在Linux中有些文件它实质上不是一个真正存在的文件，而是一个指向其他地方的某个文件的链接，使用这个链接就和使用那个原始文件一样的效果。

       ln -参数 源文件 链接名字

       比如参数 -f就是指在创建链接时先将源文件删除；

       这个当然还分软连接，硬链接：

【硬连接】

硬连接指通过索引节点来进行连接。在Linux的文件系统中，保存在磁盘分区中的文件不管是什么类型都给它分配一个编号，称为索引节点号(Inode Index)。在Linux中，多个文件名指向同一索引节点是存在的。一般这种连接就是硬连接。硬连接的作用是允许一个文件拥有多个有效路径名，这样用户就可以建立硬连接到重要文件，以防止“误删”的功能。其原因如上所述，因为对应该目录的索引节点有一个以上的连接。只删除一个连接并不影响索引节点本身和其它的连接，只有当最后一个连接被删除后，文件的数据块及目录的连接才会被释放。也就是说，文件真正删除的条件是与之相关的所有硬连接文件均被删除。

硬连接的2个限制：

l  不允许给目录创建硬链接

l  只有在同一文件系统中的文件之间才能创建链接。 即不同硬盘分区上的两个文件之间不能够建立硬链接。这是因为硬链接是通过结点指向原始文件的，而文件的i-结点在不同的文件系统中可能会不同。

【软连接】

另外一种连接称之为符号连接（Symbolic Link），也叫软连接。软链接文件有类似于Windows的快捷方式。它实际上是一个特殊的文件。在符号连接中，文件实际上是一个文本文件，其中包含的有另一文件的位置信息。

这就允许符号链接（经常简写为symlinks）指向位于其他分区、甚至是其他网络硬盘上的某个文件

9、shell读入参数

read -p "input your first name:" firstname

read -p "input your last name:" lastname

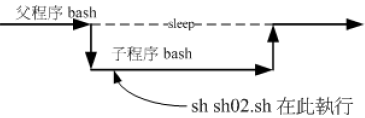
echo -e "\n your name:$firstname $lastname"

运算：var=$((运算内容))

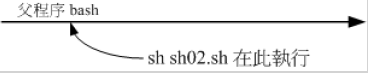
total=$(($firstnu\*$secnu))

10、shell执行：

直接执行方式：



source方式：在父程序中执行的，所以各项动作会在原本的bash内生效（比如不注销系统而让写入bashrc的内容生效：source ~/.bashrc）



11、

test -e test.cert && echo "exist"||echo "not exist"

方式：command1 && command2

  如果command1执行成功，则执行command2

方式：command1 || command2

如果command1执行失败，则执行command2

12、文件相关

load 数据到mysqldb：

#!/bin/bash

time awk -F ',' '{ind=$1%100;print $0>>"./loadfile/load-"ind".txt"}' liushi.txt

#把数据导入到DB

for((loopTimes=0;loopTimes<100;loopTimes++))

do

this\_file="./loadfile/load-"$loopTimes".txt"

/usr/bin/mysql -h10.206.30.110 -uoss -poss\_da dba20180306watering -P3346 --default-character-set=gbk --local-infile=1 -e "load data local infile '$this\_file' IGNORE into table tb\_vanishedUsers\_$loopTimes FIELDS TERMINATED BY ',';"

done

13.