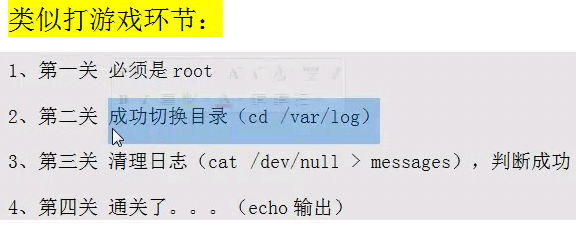
Shell Script

# 一．脚本基础知识

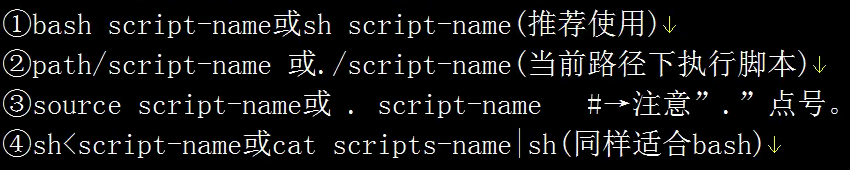


**注意：设置cron定时任务时，需要在脚本中重新定义相关的环境变量。**

**source xxx.sh 与 sh xxx.sh 的区别， 上下文环境不一样。前都会在一个shell环境中，后者是在一个子shell中。**

**引入其它函数脚本的时候，要用source(.) filefullpath**

## 脚本执行方式：



## 脚本开发规范

### 2.1脚本第一行指定脚本解释器

### 2.2 脚本开头添加版本版权等信息

#!/bin/bash

#Date: 10:08 2016-04-21

#Author: Create by leo.song

#Mail: haitao.song@foxmail.com

#Funciion: This scripts function is ...

#Version: 1.1

~

### 2.3不使用中文注释

### 2.4脚本以.sh结尾

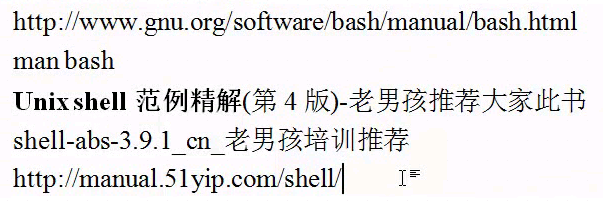
### 2.5结构体一次写完 缩进

if, for etc.

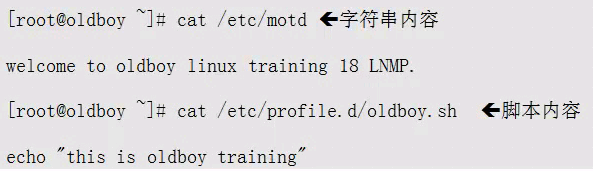
## Shell 帮助与资料

<http://www.gnu.org/software/bash/manual/bash.html>

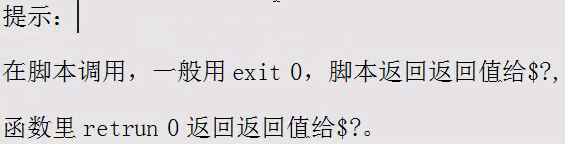
<http://manual.51yip.com/shell/>



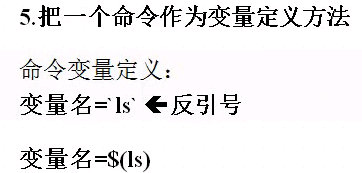
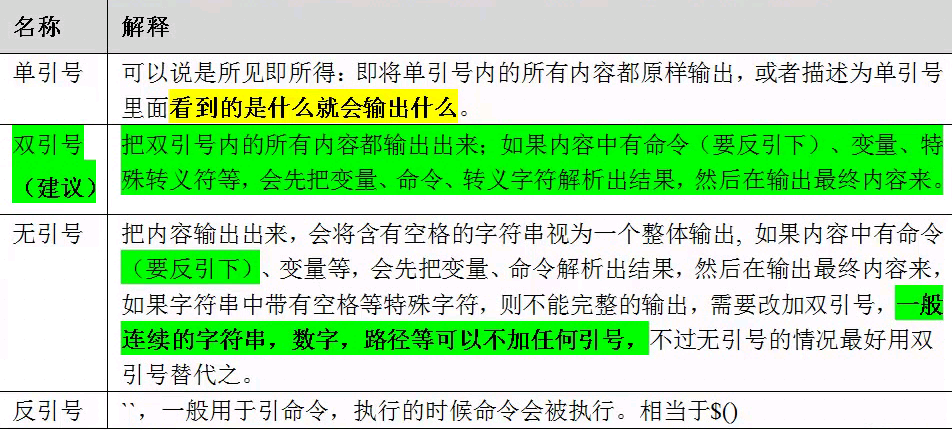
## 登录提示信息显示







## 定义变量的时候引号的区别



## 6．只取脚本名称或者只取脚本路径

[root@mysql-db-server ~]# basename /etc/init.d/functions

functions

[root@mysql-db-server ~]# dirname /etc/init.d/functions

/etc/init.d

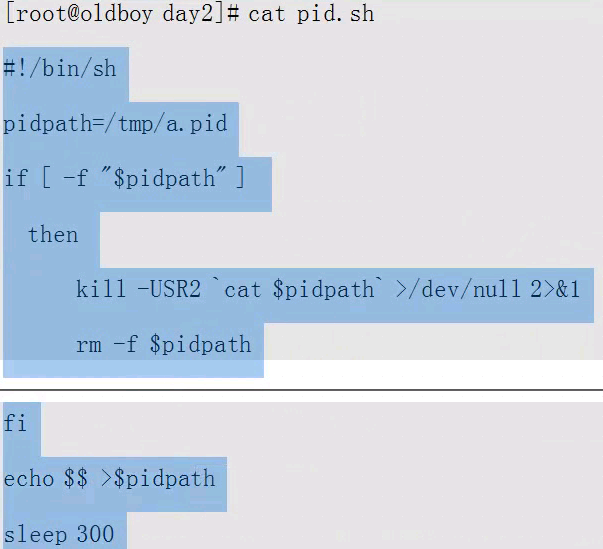
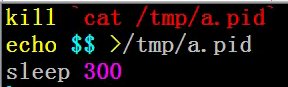
[root@mysql-db-server ~]#

在脚本中：

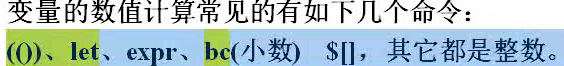
basename $0

dirname $0

## 同一时刻只运行同一个脚本



## 变量的数值运算

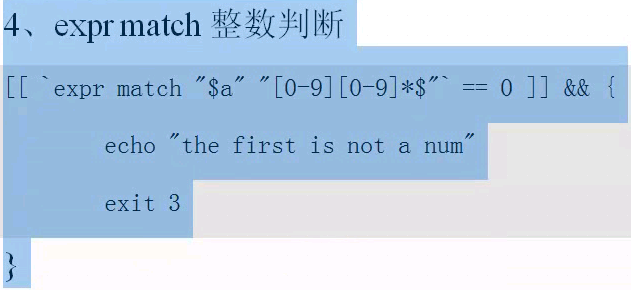


((i=i+9))

a = $((i+98))

如果双括号带：$，将获得表达式值，赋值给左边变量。

## expr







文件类型：

#!/bin/sh

if expr "$1" : ".\*\.pub" >/dev/null

#if expr "$1" : ".\*\.pub" &>/dev/null

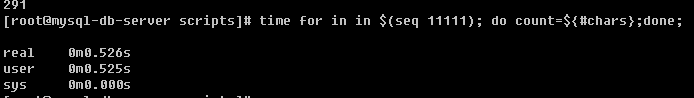
then

echo "You are using $1"

else

echo "Pls use \*.pub file."

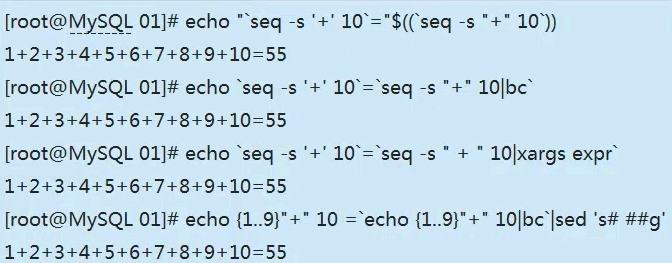
fi

[root@mysql-db-server scripts]# time for in in $(seq 11111); do count=${#chars};done;

real 0m0.526s

user 0m0.525s

sys 0m0.000s



## 10.参数个数判断

#!/bin/bash

[ $# -ne 2 ] && {

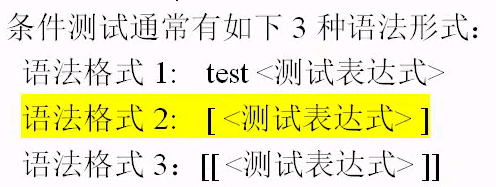
echo "$0 ARG1 ARG2"

exit 1

}

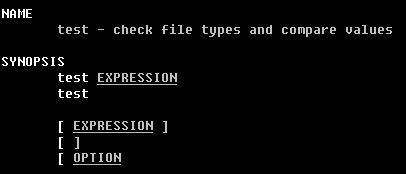
echo "The number of parameters is $#"

## 11.条件测试与比较



推荐使用2.

[root@mysql-db-server scripts]# man test



[root@mysql-db-server scripts]# test -f /etc/ && echo 1||echo 0

0

[root@mysql-db-server scripts]# test -d /etc/ && echo 1||echo 0

1

[root@mysql-db-server scripts]# [ -z "" ] && echo 0 ||echo 1

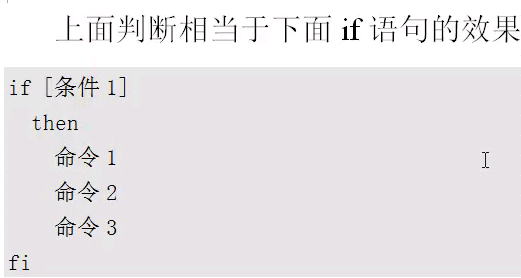
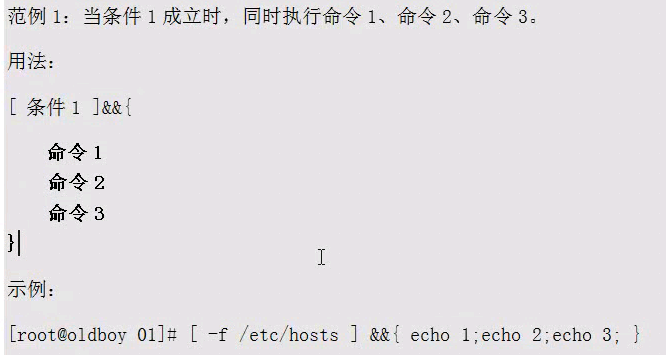
0

[root@mysql-db-server scripts]# [ -z "abc" ] && echo 0 ||echo 1

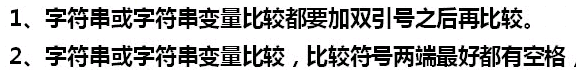
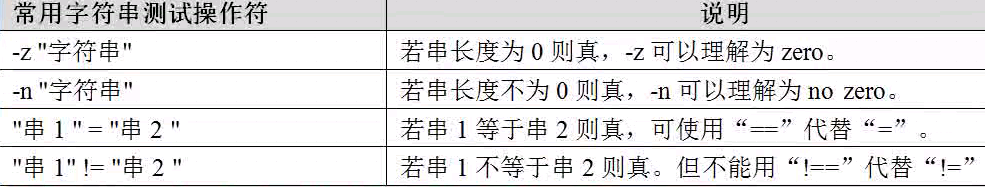
1



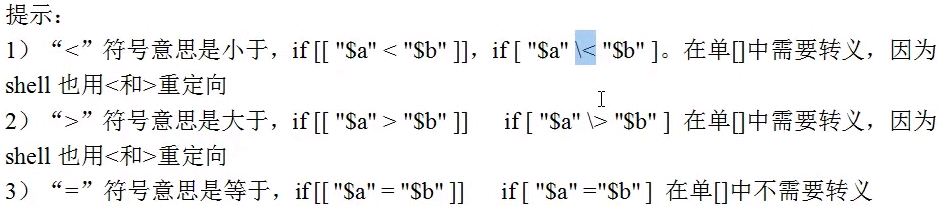
## 12.条件成立执行语句

[root@mysql-db-server scripts]# [ 3 -ne 3 ] || { echo "ehllo"; echo "kkk"; }

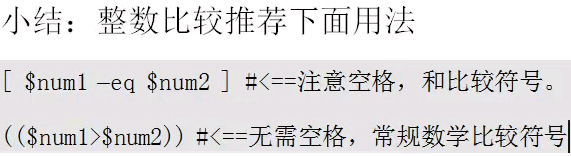
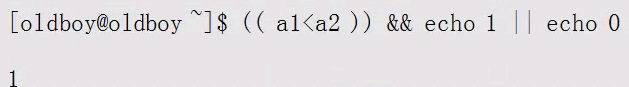
## 13.字符串比较



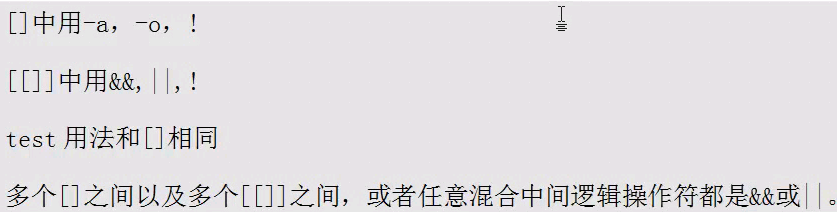
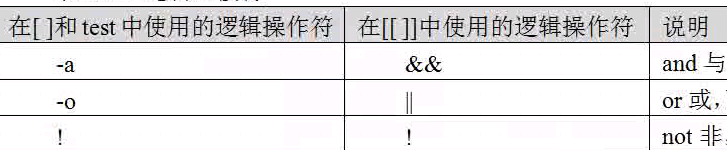
## 14.整数比较



整数比较也可以使用双括号：



## 15.逻辑操作符



## 示例：比较两个数的大小：

#!/bin/sh

#no1 judge args

[ $# -ne 2 ] && {

echo "Usage: $0 num1 num2"

exit 1

}

#no.2 juges if int

expr $1 + $2 &>/dev/null

[ $? -ne 0 ] && {

echo "Pls input two numbers."

exit 2;

}

#3 compare

[ $1 -lt $2 ] && {

echo "$1 < $2"

exit 0

}

[ $1 -eq $2 ] && {

echo "$1 == $2"

exit 0

}

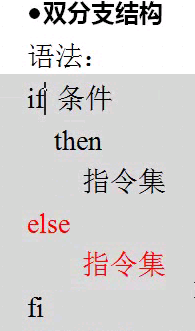
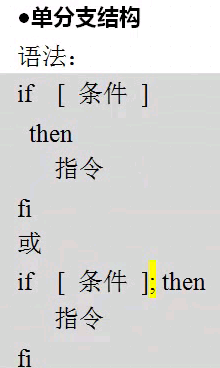
[ $1 -gt $2 ] && {

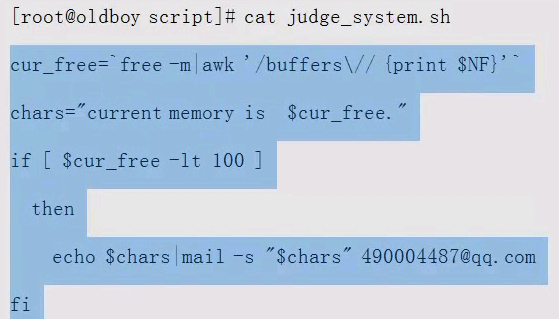
echo "$1 > $2"

exit 0

}

## 16.if 条件句





检查mysql 运行状态 ：

#!/bin/sh

if [ `netstat -tunpl|grep 3306|grep mysqld|wc -l` -ge 1 ]

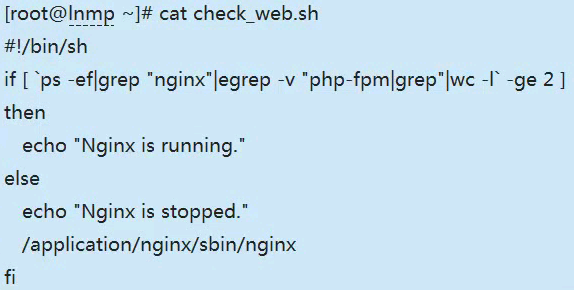
then

echo "Mysql is running."

else

echo "Mysql is stopped."

fi





## 17.服务状态检查



[leo@mysql-db-server 3306]$ curl -s -I http://www.baidu.com|head -1

HTTP/1.1 200 OK

[leo@mysql-db-server 3306]$ curl -s -I http://www.baidu.com|head -1|grep "\b200\b"|wc –l

[leo@mysql-db-server 3306]$ curl -s -I -w "%{http\_code}\n" http://www.baidu.com -o /dev/null

200

echo -e "\n"|telnet baidu.com 80|grep Connected

其它思路：

服务启动时，生成一个文件， 以此文件来判断服务是否启动。

正常停止的时候，删除该文件。

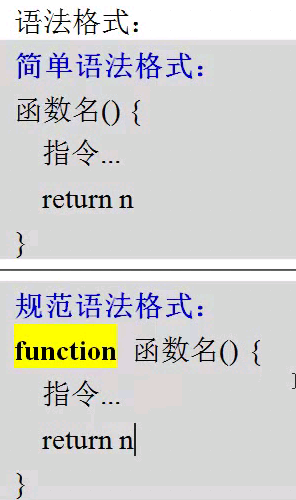




检查远端的端口是否开放

<http://oldboy.blog.51cto.com/2561410/942530>

## 18.函数



大型shell 程序组织结构 ：

[leo@mysql-db-server scripts]$ mkdir shell\_dir

[leo@mysql-db-server scripts]$ cd shell\_dir/

[leo@mysql-db-server shell\_dir]$ mkdir bin fun logs conf

[leo@mysql-db-server shell\_dir]$ tree .

.

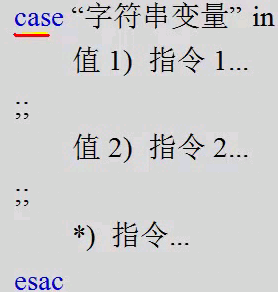
├── bin

├── conf

├── fun

└── logs

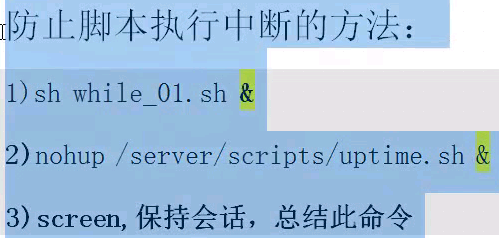
## 19.case

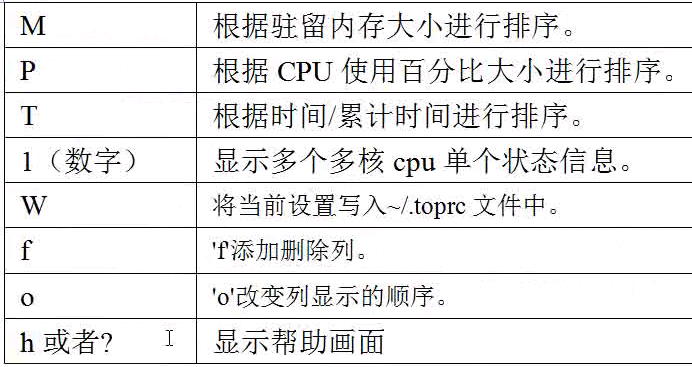
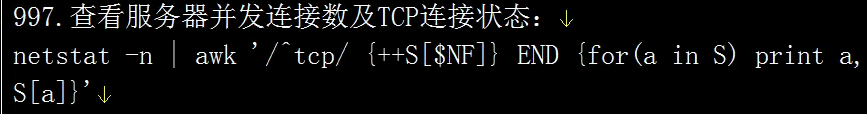
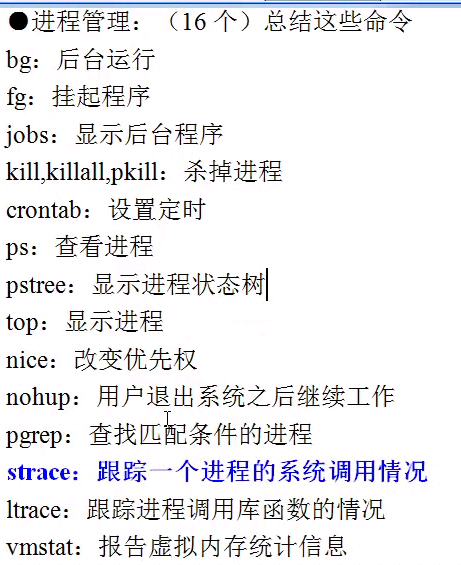


## 20.while



## 21.防止脚本中断的方法





22.日志流量统计



#!/bin/sh

sum=0

while read line

do

value=`echo $line|awk '{$10}'`

expr $value + 1 &>/dev/null

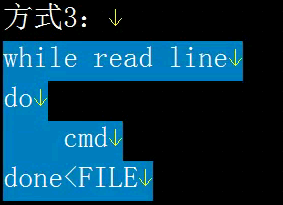
[ $? -ne 0 ] && continue

((sum+=value))

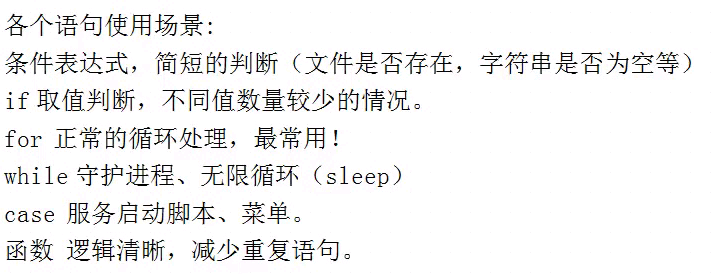
done < /usr/local/nginx/logs/access.log

echo "Total flow is : $sum"

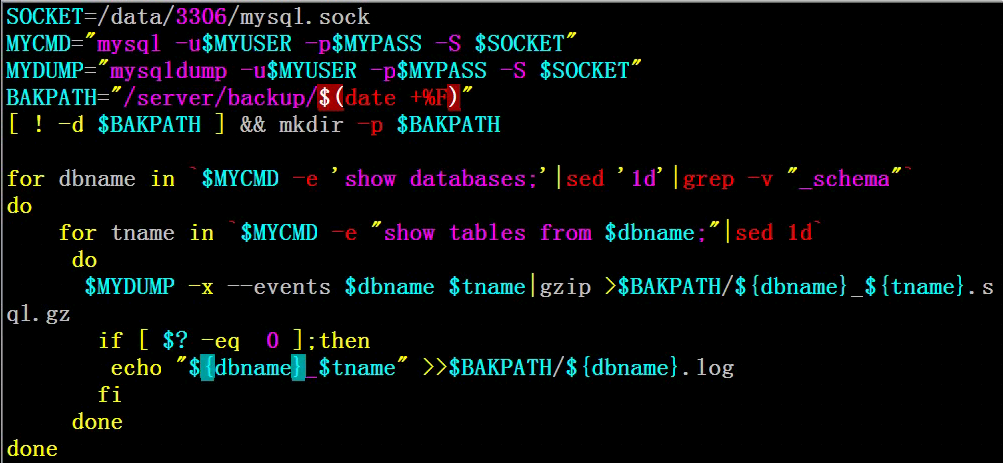
## 22.while 读文件方式

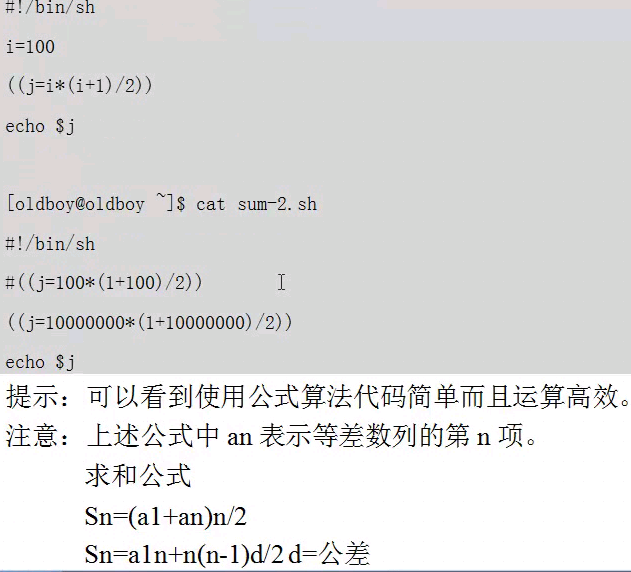
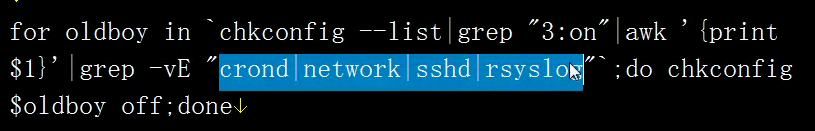


小结：



## 23.for 循环

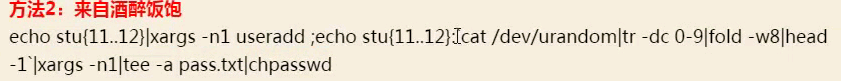




## 24.批量添加用户随机密码

[root@mysql-db-server oldboy]# for n in `seq -w 10`; do pass=`echo $RANDOM|md5sum|cut -c 2-9`; useradd newuser$n;echo $pass|passwd --stdin newuser$n; echo "addedd user : newuser$n

with password : $pass" >> /tmp/createUser.log; done



## 25.产生随机数

#### $RANDOM

#### openssl rand –base64 65

#### date +%s%N

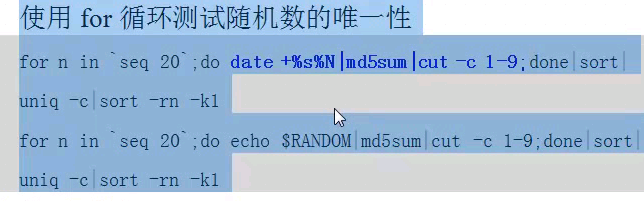
#### head /dev/urandom|cksum

#### cat /proc/sys/kernel/random/uuid

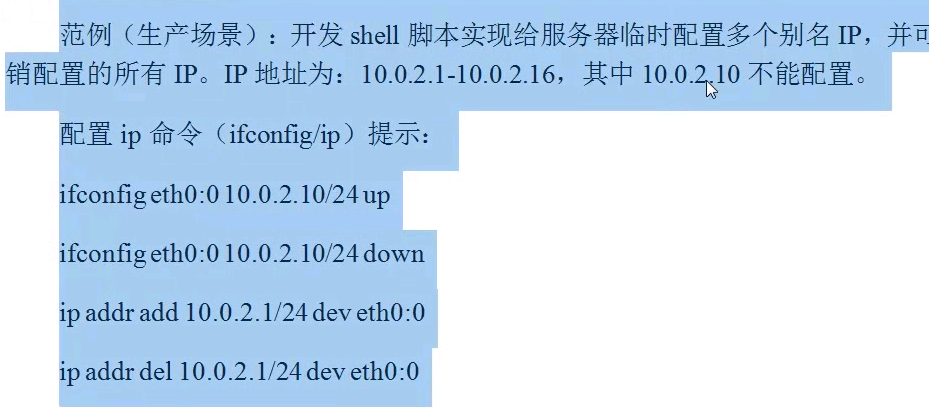
[root@mysql-db-server oldboy]# which expect

/usr/bin/expect

#### [root@mysql-db-server oldboy]# mkpasswd -l 8



## 26.IP 配置：

[root@mysql-db-server scripts]# cat ip-admin.sh

#!/bin/sh

usage() {

echo $#

if [ $# -ne 1 ]; then

echo "$0 {start|stop}"

fi

exit

}

start() {

for ip in `seq 16`

do

if [ $ip -eq 16 ]; then

continue

fi

ip addr add 192.168.138.$ip/24 dev eth0:$ip

done

}

stop() {

for ip in `seq 16`

do

if [ $ip -eq 16 ]; then

continue

fi

ip addr del 192.168.138.$ip/24 dev eth0 label eth0:$ip

done

}

op() {

case "$1" in

start)

start

;;

stop)

stop

;;

\*)

usage

esac

}

op $1

## 26数组



[root@vm-master ~]#array=(1 2 3)

[root@vm-master ~]#echo ${#array[\*]}

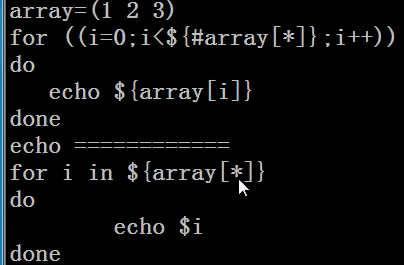
3

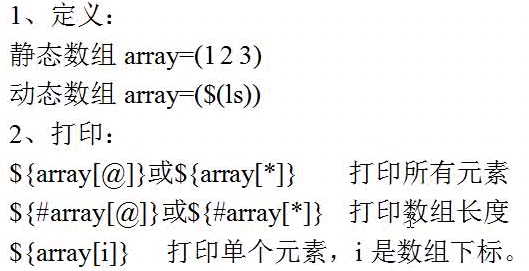
[root@vm-master ~]#echo ${array[@]}

1 2 3

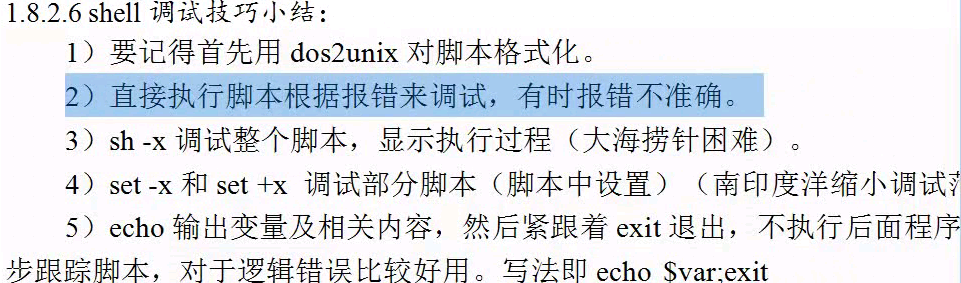
[root@vm-master ~]#echo ${array[\*]}

1 2 3

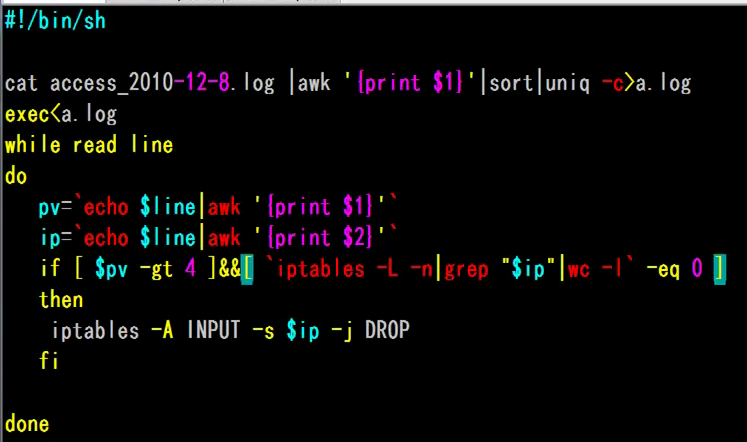




## 27.脚本调试小技巧



DDOS 攻击 封IP



三分钟一次

