shell公开讲座

张义飞
QQ:44518838
Email:zyfforlinux@163.com
博客: http://blog.csdn.net/zhangyifei216

那些年我写过的脚本：

1.iptables实用工具

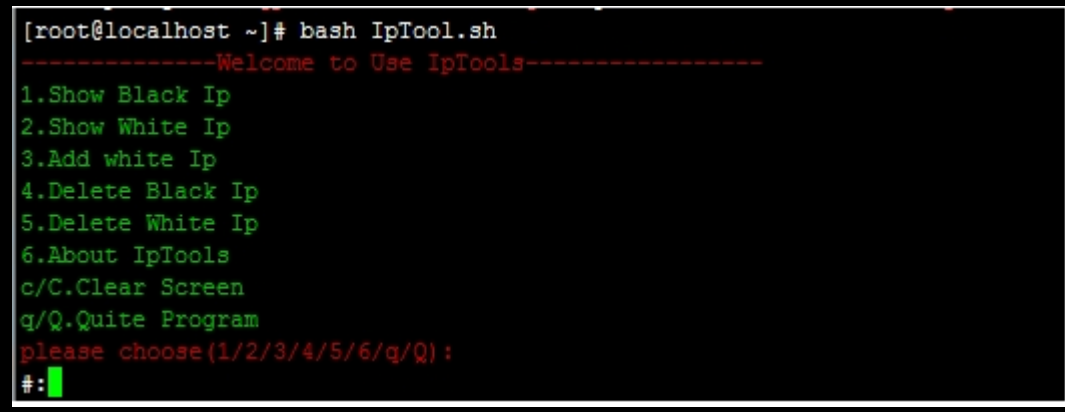
2.vps自动部署脚本

3.无线安全设计系统

4.git自动同步

5.数据备份，和日志滚动。。。。。

6.还有很多。。。。。 在那时写脚本是我最大的乐趣




```
#!/bin/bash

#CreateTime: 2014/11/22
# 更新 项目移动到coding.net
# 更新 修改对增加对不同分之的更新
# Usage: 如果什么参数都不加的话程序会判断指定目录是否有文件，有文件则pull,否则就是clone
#        加上 clone参数的话，则会强制覆指定目录的所有文件除了IgnoreFile中指定的文件


#要存放到哪个目录
DstDir="/alidata/www/telecom/"

#要忽略的文件要写相对于主目录的相对路径,因为同名文件的问题，防止误删除 空格分割
IgnoreFile="Common/Conf/config.php Admin/Common/Conf/config.php"

#github账户密码+仓库  账户中出现特殊字符需要html实体化 % ==> %40
GitUrl="[redacted]"

#仓库名称
GitStore=


#clone_an
```

```
#!/bin/bash

#author : jeff
#date   : 2015/01/29

# (1) set up all the mysqldump variables
FILE=ggcweb.sql.`date +"%Y-%m-%d"`
BACKUPDIR="/ggcweb/backup"
DBSERVER=[redacted]
DATABASE=[redacted]
USER="[redacted]
PASS=[redacted]
ERRORLOG="/ggcweb/errorlog"
CREATELOG="/ggcweb/sqllog"
DATE="14"

cd ${BACKUPDIR}
# (2) in case you run this more than once a day,
# remove the previous version of the file
unalias rm     2> /dev/null
rm -rf ${FILE}     2> /dev/null
```

```
chroot.sh
chrootstatus.sh
delhand.sh
deny_host.py
deny_site.py
deny_site.sh
getmumastatus.sh
getpwr.py
getqq.py
getsysinfo.sh
host1apd.sh
hostapd.sh
ssl1strip.sh
sslstrip-enhance
sslstrip_html.sh
sslstrip.log
startap.sh
starthostapd.sh
```

shell不是一个个体:
1.编程环境
2.实用命令
3.文本处理
4.正则表达式
构成了一个真正意义上的shell

shell的应用场景:
游戏运维，linux运维，自动化任务，文本处理，等等

课程安排:

1.bash相关

2.文本命令相关

3.文本处理命令相关

4.正则表达式相关

5.用户权限和组管理

6.shell编程

7.项目实战

    1.系统探针

    2.脚本语法检测和自动添加注释

    3.主机系统资源监控

扩展讲解(看时间): sed awk grep三剑客 vim技巧

什么是shell?

和OS进行交互

GUI:

GNOME KDE.....

CLI:

csh tcsh sh

bash:

1、命令历史、命令补全
2、管道、重定向
3、命令别名
4、命令行编辑
5、命令行展开
6、文件名通配
7、变量
8、编程

bash特性:
$
#

bash特性:
!n：执行命令历史中的第n条命令;
!-n:执行命令历史中的倒数第n条命令;
!!: 执行上一条命令;
!string：执行命令历史中最近一个以指定字符串开头的命令
!$:引用前一个命令的最后一个参数;
Esc, .
Alt+.

bash特性:

命令行编辑:

光标跳转:

Ctrl+a: 跳到命令行首

Ctrl+e: 跳到命令行尾

Ctrl+u:

删除光标至命令行首的内容

Ctrl+k: 删除光标至命令行尾的内容

Ctrl+l: 清屏

命令行展开：
创建a~z开头，.txt结尾的文件
touch {a..z}.txt

alias:
alias ipconfig="ifconfig"
unalias ipconfig

通配符:
字符匹配: * ? [] [^]
字符集: [:space:],[:lower:],[:punct:],[:alnum:],[:alpha:],[:upper:]

管道：前一个命令的输出，作为后一个命令的输入

命令1 | 命令2 | 命令3 | ...

编程功能:

shell编程，本次讲座的重点。

文本相关命令:

cat tac more less head tail

文本处理相关:

cut:

- -d -f

sort:

- -n -r -t -k

uniq

- -c
- -i
- -d

wc

tr

grep, egrep,
grep: 根据模式搜索文本，并将符合模式的文本行显示出来。

grep [options] PATTERN [FILE...]
Pattern: 文本字符和正则表达式的元字符组合而成匹配条件

什么是元字符?

字符匹配:
字数匹配:
位置锚定:
分组引用:

字符匹配:

.: 匹配任意单个字符
[]: 匹配指定范围内的任意单个字符
[^]: 匹配指定范围外的任意单个字符

匹配次数（贪婪模式）：

*: 匹配其前面的字符任意次

a, b, ab, aab, acb, adb, amnb

a*b， a?b

a.*b

.*: 任意长度的任意字符

\?: 匹配其前面的字符1次或0次

\{m,n\}:匹配其前面的字符至少m次，至多n次

\{1,\}

\{0,3\}

位置锚定：

^: 锚定行首，此字符后面的任意内容必须出现在行首
$: 锚定行尾，此字符前面的任意内容必须出现在行尾
^$: 空白行
\<或\b: 锚定词首，其后面的任意字符必须作为单词首部出现
\>或\b: 锚定词尾，其前面的任意字符必须作为单词的尾部出现

分组：

\(\)

\(ab\)*

后向引用

\1: 引用第一个左括号以及与之对应的右括号所包括的所有内容

\2:

\3:

grep Options

-i

--color

-v: 显示没有被模式匹配到的行

-o：只显示被模式匹配到的字符串

扩展正则表达式
字符匹配：

.
[]
[^]

次数匹配：
*:
?:
+: 匹配其前面的字符至少1次
{m,n}

扩展正则表达式
位置锚定：
^
$
\<
\>
分组：
()：分组
\1, \2, \3, ...
或者
|: or
C|cat: Cat或cat, C或cat

Filesystem Hierarchy Standard（文件系统目录标准）的缩写，多数Linux版本采用这种文件组织形式，类似于Windows操作系统中c盘的文件目录，FHS采用树形结构组织文件。FHS定义了系统中每个区域的用途、所需要的最小构成的文件和目录，同时还给出了例外处理与矛盾处理。--百度百科

/boot
/dev
/etc:
/home: /home/USERNAME
/root:
/lib: /lib/modules:
/media:
/mnt:
/opt:
/proc:
/sys
/tmp:
/var:
/bin:
/sbin

```
#!/bin/bash
echo "Hello World"/printf "Hello World"

[root@localhost shell]# bash hello.sh
Hello World
[root@localhost shell]# chmod +x hello.sh
[root@localhost shell]# ./hello.sh
Hello World
```

1、统计/usr/bin/目录下的文件个数；
2、取出当前系统上所有用户的shell，要求，每种shell只显示一次，并且按顺序进行显示；
3、思考：如何显示/var/log目录下每个文件的内容类型？
4、取出/etc/inittab文件的第6行；
5、取出/etc/passwd文件中倒数第9个用户的用户名和shell，显示到屏幕上
6、显示/etc目录下所有以pa开头的文件，并统计其个数；

通配符练习：

1、创建a123, cd6, c78m, c1 my, m.z, k 67, 8yu, 789等文件；注意，以上文件是以逗号隔开的，其它符号都是文件名的组成部分；
2、显示所有以a或m开头的文件；
3、显示所有文件名中包含了数字的文件；
4、显示所有以数字结尾且文件名中不包含空白的文件；
5、显示文件名中包含了非字母或数字的特殊符号的文件；

正则表达式练习：

1、显示/proc/meminfo文件中以不区分大小的s开头的行；

grep -i "^s"

2、显示/etc/passwd中以nologin结尾的行;

取出默认shell为/sbin/nologin的用户列表

取出默认shell为bash，且其用户ID号最小的用户的用户名

3、显示/etc/inittab中以#开头，且后面跟一个或多个空白字符，而后又跟了任意非空白字符的行；

4、显示/etc/inittab中包含了:一个数字:(即两个冒号中间一个数字)的行；

5、显示/boot/grub/grub.conf文件中以一个或多个空白字符开头的行；

项目实战

补充知识: 管道 重定向
1.判断网络是否通 ping
2.获取网卡地址 ifconfig
3.获取内存使用率 free
4.获取CPU的负载情况 w
5.获取CPU的空闲率 vmstat

项目实战

```
[root@localhost shell]# bash getsysinfo.sh
eth0网卡:172.16.0.134
网络畅通
内存总量:442
内存使用量:320
内存剩余量:122
CPU空闲率:98%
CPU5分钟负载情况: 0.00
CPU10分钟负载情况: 0.00
CPU15分钟负载情况: 0.00
```

/etc/passwd:
用户名：密码：UID:GID：注释：家目录：默认SHELL
/etc/group:
组名：密码：GID:以此组为其附加组的用户列表
/etc/shadow：
用户名：密码：最近一次修改密码的时间：最短使用期限：最长使用期限：警告时间：非活动时间：过期时间：

用户管理：

useradd, userdel, usermod, passwd, chsh, chfn id

组管理：

groupadd, groupdel, groupmod, gpasswd

权限管理：

chown, chgrp, chmod, umask

useradd [options] USERNAME

-u UID
-g GID（基本组）
-G GID,...（附加组）
-c "COMMENT"
-d /path/to/directory
-s SHELL
-m -k
-M
-r: 添加系统用户

userdel:

userdel [option] USERNAME

-r: 同时删除用户的家目录

id：查看用户的帐号属性信息

-u

-g

-G

-n

finger: 查看用户帐号信息

finger USERNAME

修改用户帐号属性：

usermod

-u UID

-g GID

-a -G GID：不使用-a选项，会覆盖此前的附加组；

-c

-d -m：

-s

-l

-L：锁定帐号

-U：解锁帐号

chsh: 修改用户的默认shell
chfn：修改注释信息
密码管理：

passwd [USERNAME]
    --stdin
    -l
    -u
    -d: 删除用户密码
pwck：检查用户帐号完整性

组管理：

创建组:groupadd

groupadd

-g GID

-r：添加为系统组

groupmod

-g GID

-n GRPNAME

groupdel

gpasswd：为组设定密码

文件权限管理:

chown: 改变文件属主(只有管理员可以使用此命令)

chown USERNAME file,...

-R: 修改目录及其内部文件的属主

--reference=/path/to/somefile file,...

chown USERNAME:GRPNAME file,...

chown USERNAME.GRPNAME file,...

chgrp GRPNAME file,...

-R

--reference=/path/to/somefile file,...

chmod: 修改文件的权限

修改三类用户的权限:

chmod MODE file,...

-R

--reference=/path/to/somefile file,...

练习：手动创建用户linuxer UID 999 GID 999

Linux 重定向

本质是改变其stdout,stdin,stderr

>: 覆盖输出

>>: 追加输出

2>: 重定向错误输出

2>>: 追加方式

&>: 重定向标准输出或错误输出至同一个文件

<: 输入重定向

<<: Here Document

shell编程，到此为止才真正开始：

bash变量的类型：

- 本地变量(局部变量)
- 环境变量
- 位置变量:
  - $1, $2, ...
  - shift
- 特殊变量:
  - $?
  - $#：参数的个数
  - $*: 参数列表
  - $@：参数列表

shell中如何进行算术运算：

A=3

B=6

1、let 算术运算表达式

```
let C=$A+$B
```

2、$[算术运算表达式]

```
C=$[$A+$B]
```

3、$((算术运算表达式))

```
C=$(($A+$B))
```

4、expr 算术运算表达式，表达式中各操作数及运算符之间要有空格，而且要使用命令引用

```
C=`expr $A + $B`
```

bash支持的引号：

``：命令替换

""：弱引用，可以实现变量替换

''：强引用，不完成变量替换

if 语句
条件测试

字符串比较，数字比较，条件组合，文件和目录权限和存在性测试

带颜色输出:
格式如下:
\033[字背景颜色;文字颜色m字符串\033[0m"
文本颜色：30黑色，31红色，32绿色，33黄色，34蓝色，35洋红，36青色，37白色;
字背景颜色：40黑色，41红色，42绿色，43黄色，44蓝色，45洋红，46青色，47白色;

变量进阶:
${parameter#*word} 变量中第一次出现word的时候右边的全部字符(从左往右)
${parameter##*word} 变量中最后一次出现word的时候右边的全部字符(从左往右)
example"
FILE=/usr/local/src
${FILE#*/}: usr/local/src
${FILE##*/}: src
${parameter%word*} 变量中第一次出现word的时候左边的全部字符(从右向左)
${parameter%%word*} 变量中最后一次出现word的时候左边的全部字符(从右往左)
example:
${FILE%/*}: /usr/local
${FILE%%/*}:

for循环

方式一:

```
for VAR in con1 con2 con3……;do
    statement1
    statement2
    .....
done
```

方式二:

```
for((初始值;限制值;步长));do
    statement1
    statement2
    ....
 done
```

1.while循环

```
while [condition];do (条件成立进入循环)
        statement1
        statement2
        .....
done
```

2.until循环

```
until [condition];do   (条件不成立进入循环)
      statement1
      statement2
      .....
done
```

find

实时
精确
支持众多查找标准
遍历指定目录中的所有文件完成查找，速度慢；

locate

命令格式:
find 查找路径 查找标准 查找到以后的处理运作
查找路径：默认为当前目录
查找标准：默认为指定路径下的所有文件
处理运作：默认为显示

匹配标准：

-name 'FILENAME'：对文件名作精确匹配

文件名通配：

*：任意长度的任意字符

?

[]

-iname 'FILENAME': 文件名匹配时不区分大小写

-regex PATTERN：基于正则表达式进行文件名匹配

-user USERNAME: 根据属主查找

-group GROUPNAME: 根据属组查找

-uid UID: 根据UID查找

-gid GID: 根据GID查找

匹配标准：

-nouser：查找没有属主的文件

-nogroup：查找没有属组的文件

-type

f：普通文件d,c,b,l,p,s

-size [+|-]

#k

#M

#G

匹配标准：

-mtime

-ctime

-atime

[+|-]#

-mmin

-cmin

-amin

[+|-]#

匹配标准：

-perm MODE：精确匹配

-MODE: 文件权限能完全包含此MODE时才符合条件

-644

644: rw-r--r--

755: rwxr-xr-x

750: rwxr-x---

运作：

-print: 显示
-ls：类似ls -l的形式显示每一个文件的详细
-ok COMMAND {} \; 每一次操作都需要用户确认
-exec COMMAND {} \;

练习：

1、新建一个没有家目录的用户openstack；
2、复制/etc/skel为/home/openstack；
3、改变/home/openstack及其内部文件的属主属组均为openstack；
4、/home/openstack及其内部的文件，属组和其它用户没有任何访问权限

练习：

1、创建一个用户mandriva，其ID号为2002，基本组为distro（组ID为3003），附加组为linux；

2、创建一个用户fedora，其全名为Fedora Community，默认shell为tcsh；

3、修改mandriva的ID号为4004，基本组为linux，附加组为distro和fedora；

4、将mandriva的默认shell改为/bin/bash;

5、添加系统用户hbase，且不允许其登录系统；

练习：写一脚本
能接受一个参数(文件路径)
判定：此参数如果是一个存在的文件，就显示“OK.”；否则就显示"No such file."
练习：写一个脚本
给脚本传递两个参数(整数)；
显示此两者之和，之乘积；
练习：写一个脚本，完成以下任务
1、使用一个变量保存一个用户名；
2、删除此变量中的用户，且一并删除其家目录；
3、显示“用户删除完成”类的信息；
练习：
传递一个用户名参数给脚本，判断此用户的用户名跟其基本组的组名是否一致，并将结果显示出来。
练习：写一个脚本
给定一个文件：
如果是一个普通文件，就显示之；
如果是一个目录，亦显示之；
否则，此为无法识别之文件；

练习：写一个脚本，完成以下任务
1、添加5个用户，user1,..., user5
2、每个用户的密码同用户名，而且要求，添加密码完成后不显示passwd命令的执行结果信息；
3、每个用户添加完成后，都要显示用户某某已经成功添加；

练习，写一个脚本，完成以下要求：
1、添加3个用户user1, user2, user3；但要先判断用户是否存在，不存在而后再添加；
2、添加完成后，显示一共添加了几个用户；当然，不能包括因为事先存在而没有添加的；
3、最后显示当前系统上共有多少个用户；

练习：写一个脚本，完成以下任务
1、使用一个变量保存一个用户名；
2、删除此变量中的用户，且一并删除其家目录；
3、显示“用户删除完成”类的信息；

练习，写一个脚本，完成以下要求：
给定一个用户：
1、如果其UID为0，就显示此为管理员；
2、否则，就显示其为普通用户；

练习：写一个脚本
判断当前系统上是否有用户的默认shell为bash；
　如果有，就显示有多少个这类用户；否则，就显示没有这类用户；

练习：写一个脚本
给定一个用户，判断其UID与GID是否一样
如果一样，就显示此用户为“good guy”；否则，就显示此用户为“bad guy”。

练习：写一个脚本
给定一个文件，比如/etc/inittab
判断这个文件中是否有空白行；
如果有，则显示其空白行数；否则，显示没有空白行。

练习：
传递一个用户名参数给脚本，判断此用户的用户名跟其基本组的组名是否一致，并将结果显示出来。

项目实战:

mkscript

1.脚本自动添加注释

2.脚本自动语法检测

终极小项目:

1. 自动发邮件报警
2.内存。磁盘，CPU阀值监控

Q&A

Next:

sed & awk & grep

好玩的iptables

从头构建自己的linux发行版 裁剪