第七课 Linux变量

通配符：

\* 匹配任意多个字符，可以存在可以不存在，也可以存在多个

? 匹配任意一个字符，必须存在

ls -ld /etc/cron\* 列出/etc/目录所有以cron开头的文件或文件夹的详细属性

ls -ld /etc/cron?? 列出/etc目录下所有以cron开头的后面必须是两个字符的文件或文件夹详细属性

ls -ld /etc/cron?\* 列出/etc目录下以cron开头至少五个字符的文件或文件夹的详细属性

ls -ld /etc/cron\*.\* 列出/etc目录下以cron开头的并且后面必须带.的文件或文件夹的详细属性

ls -ld /etc/[acd]\* 列出/etc目录下第一个字符是a、c或d的文件或文件夹的详细属性

ls -ld /etc/\*[0-9]\* 列出/etc目录，名称中含有数字的文件或文件夹的详细属性

ls -ld /etc/\*[0-9] 列出/etc目录下，以数字结尾的

ls -ld /etc/[0-9]\* 列出/etc目录下，以数字开头的

grep 过滤关键字的命令

^ 开头

$ 结尾

grep ^cat /usr/share/dict/words 在/usr/share/dict/words文件中找出以cat开头的行

grep cat$ /usr/share/dict/words 在/usr/share/dict/words文件中列出以cat结尾的行

grep ^cat$ /usr/share/dict/words 在后面的文件中列出只有cat的行

grep ^cat /usr/share/dict/words | grep cat$ 列出开头是cat结尾也是cat的行

{}

mkdir -pv {a,b}{c,d} -p代表没有父目录先创建父目录子目录，适用于创建目录树（一串），-v代表显示附加信息，这里会在屏幕输出创建了哪些文件夹，可有可无，{}代表中的内容自己选，因此共创建了四个目录。

[root@localhost ~]# mkdir -pv {a,b}{c,d}

mkdir: created directory `ac'

mkdir: created directory `ad'

mkdir: created directory `bc'

mkdir: created directory `bd'

cp -a /var/named/chroot/var/named/named.{empty,bak} 建议大家在修改服务器的配置文件的时候提前做备份（cp一份），如果配置文件被改坏导致服务无法启动，将会复制/var/named/chroot/var/named中的named.empty复制一份还是在刚才目录中名称改为named.bak

上述命令等效于：

cp -a /var/named/chroot/var/named/named.empty /var/named/chroot/var/named/named.bak

mkdir -pv abc-{1..10} 批量创建目录名称为abc-1、abc-2、abc-3...abc-10一共10个目录（1..10表示1到10这个范围）

ls -a ~/.bash\* 可以看到几个以.bash开头的文件

.bashrc 用户每次打开终端自动执行里面的所有命令，用户自己设置的别名信息基本上都是写该文件的

.bash\_profile 用户每次登录系统自动执行所有的命令

.bashr\_logout 用户每次注销执行所有的命令

.bash\_history 用户注销后历史命令记录的文件

/etc/bashrc 所有用户都适用的，所有用户打开终端自动执行所有命令

/etc/profile 所有用户登录系统自动执行的命令

tr命令：转换字符

tr 'A-Z' 'a-z' < ~/.bashrc 将后面的文件中所有大写字符改为小写字符输出至屏幕，只改内存中的数据，真实文件内容不会修改

cat > a.txt 交互式写入文件，读取用户的输入写入到后面的文件中，用户输入完毕后，回车新建空白行使用Ctrl+d组合键保存进文档

cat > filea < fileb 将会把fileb文件的内容全部写入到filea中>输出重定向，<输入重定向，fileb做输入，filea做输出。**注意：命令格式要求先写输出，再写输入。**

cat > filea <<exit 读取用户输入到filea中，一旦用户输入exit并回车直接退出并保存文件，不需要Ctrl+d退出

head、tail、grep等基本上都是过滤行，显示一行的信息

cut是显示列的信息

cat /etc/passwd | wc -l wc是字数统计，-l显示行数，统计/etc/passwd一共多少行，就是系统有多少个用户

cut -d: -f1 /etc/passwd 以:为分隔符（delimiter），显示/etc/passwd的第一列所有内容(field 1)，就是显示系统中所有的用户名

free -m | head -2 | tail -1 | awk '{print $4}' 列出当前系统可用的内存数量 (awk '{print $4}'表示**取空白符分割**的第4列)

cut -c2-5 /usr/share/dict/words 列出/usr/share/dict/words文件中第2--5个字符

作业：使用cut或awk等显示本主机的IP地址，只要IP地址信息，其他任何信息都不希望显示

ifconfig | grep "inet addr:" | awk '{print $2}'|cut -d: -f2

变量是在shell脚本中使用非常多的，变量分为两部分：

变量名：固定数值表示，用户定义或者系统预设值

变量值：会发生变化，这个才是变量的由来

变量的分类：

用户自定义变量：变量名和变量值都是用户自行定义，用于脚本中

环境变量：系统会根据环境来生成变量，用户名、UID、PWD，有些变量可以修改

预定义变量：系统预先定义好为了完成某些特性功能，这类变量不能修改

位置变量：用户执行命令的时候使用的选项和对象表示方式

用户自定义变量：

ABC=123 定义变量，变量名是ABC，变量值是123，变量名可以有数字、字母还有特殊符号，必须以字母开头，严格区分大小写

echo $ABC 查看ABC变量对应内容

第二种赋值方式是交互式赋值

read -p "Please input your name:" NM 定义变量名是NM，由用户输入变量值 (p for prompt)

echo $NM 查看变量对应的数值

系统中的引号

\反斜杠：保护一个字符(转义，将命令中的后续字符（如空格）视为字符串的一部分)

'单引号：保护''中间所有字符，绝对保护，**里面任何字符都是普通字符**而已

"双引号：保护""中间所有字符，相对保护，**只保护基本字符空格等**，其他字符`$等都有他们特殊含义

`反撇号：中间使用的命令，将命令的结果执行完成后再给外面的命令使用

mkdir -pv a\ b 创建一个文件夹名为a b

mkdir -pv 'a b'

rm -rf "a b"

date +%Y-%m-%d 定义date命令输出格式

mkdir -pv test-`date +%Y%m%d` 创建一个文件夹名称是test-后面是当前年月日

mkdir -pv "`date`" 创建一个目录名为date命令执行的结果

mkdir -pv ‘`date`‘ 创建一个目录名为`date`

expr是数值运算

+

-

\\*

/ 除，默认只能显示整数

% 显示余数

expr 1 + 1 计算1+1，注意+前后都要空格

环境变量是和用户使用环境相关，会随着用户不同发生改变

常见的环境变量

UID

USER

HOME

PWD

PATH 用户执行命令的时候默认会从该变量的记载中进行查找，一旦找到直接执行，如果都查找完毕还是无法发现直接提示命令找不到

PS1 是命令的一级提示符，就是命令提示符

PS2 是二级命令提示符

/etc/init.d/network start 使用绝对路径可以执行

network start相对路径提示命令找不到

PATH=/etc/init.d/:$PATH 临时有效，重启自动消失，可以写入到.bashrc或.bash\_profile或bashrc或profile

位置变量：命令的选项或者参数以空格为分隔，第一个选项或参数是$1第二个是$2第三个是$3以此类推，最多是$9

预定义变量：系统预先定义好的，每个都有特殊含义，不能修改

$? 上一条命令的返回值，如果是0的代表是正确，任何非0数值都是错误，对于系统而言必须有一个量化的数值来代表命令的成功还是错误，$?就可以来帮助用户做判断

$0 命令字，执行的命令

$# 代表位置变量的数量

$\* 列出所有的位置变量

$$ 列出进程号

$! 后台运行的最后一个进程号

ls

echo $? 如果返回0代表上一条命令执行成功，如果是非0数值代表命令执行失败了。