## 登录tty的快捷建

Ctrl +Alt+F1:回到图形界面

Ctrl+Alt+F2 :terminal 1

Ctrl+Alt+F3:terminal 2

Ctrl+Alt+F4 :terminal 3

Ctrl+Alt+F5:terminal 4

Ctrl+Alt+F6 :terminal 5

## 打开图形模式终端的方式

调出系统搜索框，在搜索框中输入“terminal”

## 命令行提示符

[xiongzhuang@localhost ~]$

Xiongzhuang当前用户的名字（username） linux是多用户的操作系统

@前面是用户名，后面是所在的域

Localhost 电脑主机的名字

~当前所在目录的名称，会随着用户进入不同目录而改变

~表示当前用户的家目录（home directory）

$指示你所具有的权限 $表示普通用户 #表示root用户

## 切换至root用户

使用sudo su命令

（

若不能切换：使用vi /etc/sudoers

找到root ALL=（ALL） ALL

在下面插入 userName ALL=(ALL) ALL

）

退出root身份，可以用ctrl+d的组合键 或者用exit命令

## 查看用户名和主机名

whoami 查看用户名

Hostname 查看主机名

## 时间

date 获取当前时间

## 命令的参数

1参数是写在命令之后的一些补充选项。命令和参数之间有空格隔开

1. 格式：command parameters
2. command和parameter分别是英语“命令”和“参数”的意思
3. 参数里可以包含多个参数，由空格隔开
4. 参数也可以包含数字，字母等等
5. 参数没有固定的格式，但是一般来说还是遵循一定的规范

短参数

1 最长用的参数形式就是一个短横线后接一个字母：command -p

2 一次加好几个短参数，可以用空格隔开：command -p -a -T -c

3多个短参数也可以合并在一起 command -paTc

4字母的大小写代表的意义不同

长参数

1长参数没有短参数那么常用，但也是很有用的

2短参数是以一个短横线开始，而长参数是以两个短横线开始的

3格式：command --parameter

4多个长参数是不能像多个短参数那样合并写

5而是只能以空格隔开：command --parameter1 --parameter2

6可以组合使用短参数和长参数

7 同一个意义的参数有短参数和长参数的两种形式

参数的值

1有一些参数需要赋值。短参数和长参数的赋值方式不一样

2短参数复制通常是 command -p 10

3长参数复制通常是 command --parameter=10

## 找到命令

输入一个字母 按两次tab键可以补全命令，文件目录 文件名称

## 命令的历史记录

上下键来查找最近的命令

history 查找命令的历史 加！加命令的编号就会运行该命令

Ctrl+L用于清理终端的内容,就是清屏的作用，同clear命令

Ctrl+D给终端传递EOF（End Of File,文件结束符）

Shift+PgUp用于向上滚屏，与鼠标的滚轮向上滚屏是一个效果

Shift+PgDown 用于向下滚屏，与鼠标的滚动向下是一个效果

Ctrl+A光标跳到一行命令的开头，Home键有相同的效果

Ctrl+E光标跳到一行命令的结尾 End键有相同的效果

Ctrl+U删除光标左侧所有字符

Ctrl+K删除光标右侧所有字符

Ctrl+W删除光标左侧的一个单词

Ctrl+Y粘贴用Ctrl+U,Ctrl+K或者Ctrl+W删除的字符串

## Linux的根目录下的直属子目录

bin :是英语binary的缩写，表示“二进制文件”，bin目录包含了会被所有用户锁使用的可执行程序

boot : boot表示启动 boot目录包含于linux启动密切相关的文件

dev：device的缩写，表示设备，包含外设，里面的所有子目录都对应一个外设

etc : et cetera的缩写，翻译成英语就是and so on ,表示“..等等”，etc目录包含系统的配置文件,

home: 放置私人的文件 私人目录 /home/xiongzhuang

lib : library表示库，lib目录包含被程序锁调用的库文件，例如.so结尾的文件，windows下的库文件则是以.dll结尾

media : 表示媒体 USB ，SD卡，DVD,光盘插入电脑时，linux可以让我们通过media的子目录来访问这些外设中的内容

mnt : mount的缩写 表示挂载临时挂载，临时挂载一些装置

opt : optional application software package 的缩写 表示可选的应用软件包 安装第三方软件

root : 超级用户的目录

sbin : sytsem binary 表示系统二进制文件

srv: service表示服务，表示网络服务启动之后所需要取用的数据

tmp： temporary 表示临时的 普通用户和程序存放临时文件的地方

usr : Unix Software Resource的缩写 Unix操作系统软件资源（类似etc ）

var : variable的缩写表示动态的，可变的，包含log日志文件

## 显示当前目录的路径

pwd:打印出当前目录的路径，Print Working Directory的缩写

which :显示一个命令的对应的可执行程序的位置，每个命令对应的是一个可执行程序，而which则能使我们看到命令所对应的程序在linux中的位置，linux中的可执行程序没有后缀，而window则是以.exe结尾

Linux以点开头的文件都是隐藏文件,不包括. 和..两个文件 .表示当前目录 ..表示上级目录

## 目录相关的命令

man ls 查看ls命令的帮助手册 man为查看命令手册

Centos的终端默认是有颜色标注的一般来说，

蓝色->目录

绿色->可执行文件

红色->压缩文件

浅蓝色->链接文件

灰色->其他文件

ls 是list的缩写。List是列举列出的意思

ls命令用于列出当前目录下的文件和目录

ls --color=auto来开启颜色标注：ls --color=auto

ls --color=none来关闭颜色标注：ls --color=none

ls -a 列出当前目录下的所有文件（包括隐藏文件）

-A不列出.和..两个目录

ls -i 显示文件的inode

ls -ll 详细列表

ls -lh以Ko,Mo,Go的形式显示文件的大小

ls -ls按照最近的修改时间排序

ls中的参数使用效果是叠加的 比如ls lath

cd 是英文change directory的缩写 表示的是切换目录

cd .. 回到上级目录

回到家目录的三种方法

cd /home/xiongzhuang

cd ~

cd

du命令：disk usage的缩写 表示“磁盘占用” 显示目录包含的文件的大小

du 可以让我们知道文件和目录所占的空间的大小，相比于ls -l命令，du命令统计的才是真正的文件大小，du命令会深入遍历每个目录的子目录，统计所有文件的大小

du -h 以ko,mo,go的形式显示文件大小

du -a 显示目录和文件的大小

du -s 只显示当前目录的总计大小

## 操纵文件

Ctrl+c能够结束大部分终端的命令（类似alt+f4）

cat和less命令：显示文件内容

cat是concatenate的缩写，表示“连接/串联”，cat命令能够一次性在终端中显示文件的所有内容

less能够显示一个终端屏幕所能展示的文件内容，

在less中使用以下快捷键

空格键可以查询下一个终端屏幕的内容，

回车键能够查看下一行的内容，

y键能够后退一行（也可以通过上下键）

d键能够前进半页，

b键能够后退一页，与键盘的PageUp效果一样，

u键能够后退半页，

q键退出文件

=键显示你在文件的什么位置，会显示当前页面的内容是文件中的第几行到第几行，显示当前页占有整个文件的百分之多少以及字节百分比

h键显示帮助文档

/键可以搜索字符在文件中的匹配位置 比如/you 则会查找文件中包含you字符的位置，按n键可以跳向下一个搜索结果，N则会跳向上一个搜索结果

head用于显示文件的开头几行，默认显示前10行

head -n {num} 显示文件开头的前num行

tail命令用于显示文件结尾的几行，默认显示后10行

tail -n {num} 显示文件结尾的后num行

tail -f 追踪文件的更新每过一秒会检查一下文件是否有新内容也可以通过 -s来制定间隔检查的秒数 例如tail -f -s 4 xxxx

touch命令：创建一个空白的文件 “触摸，触碰”的意思

touch 可以创建一个或多个文件

mkdir 创建一个目录 make directory

mkdir也支持创建一个或多个文件

mkdir -p 可以递归的创建文件夹

（创建目录和文件的时候如果有空格可以用””来创建 但是在linux中为了避免不必要的麻烦尽量不要用特殊字符来命名文件或目录）

## 文件的复制和移动操作

cp : copy的缩写，拷贝的意思，可以拷贝单个或多个文件和目录

-r或者-R 拷贝目录 recursive递归

\* 通配符

cp \*.txt folder 把当前目录下的所有txt文件拷贝到folder目录中下

cp ha\* folder 把当前目录下的所有ha开头的文件都拷贝到folder目录下

mv move的缩写标识移动 mv有两个功能 移动文件（目录） 重命名文件（目录）

rm命令 删除文件和目录 可以删除一个或多个文件目录 甚至可以删除linux系统

-i 删除时终端会询问是否删除

-f 终端不会询问用户是否删除（强制删除）

-r 删除目录 并递归删除之下的目录

rmdir 只能删除空目录

rm -rf /\* 或者rm -rf/ 删除系统根目录 极度危险

ln命令：创建连接

Linux下有两种连接类型：

Physical link :物理连接或硬连接

Symbolic link:符号连接或软连接

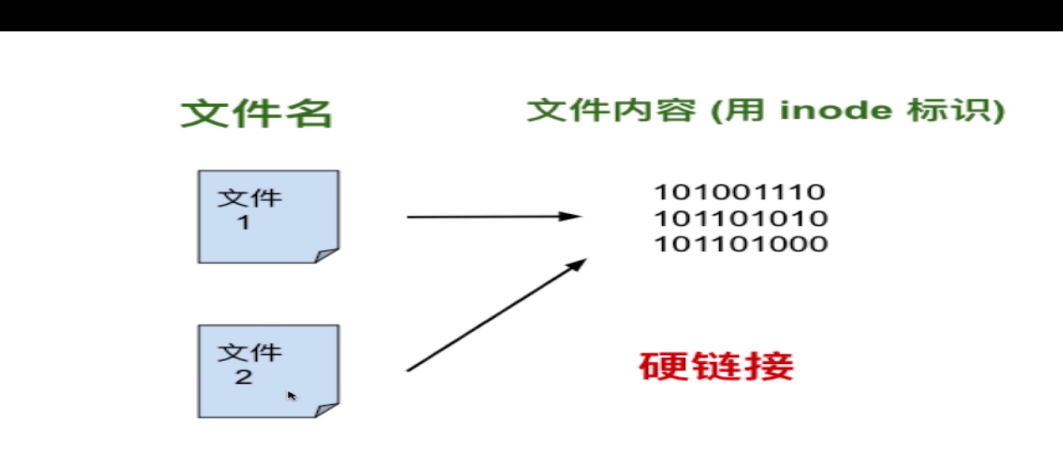
Linux文件存储：文件名和文件内容是分开存放的，方便linux管理

文件有三个部分：文件名，权限和文件内容

每个文件的文件内容被分配至一个表示号码，就是inode,因此每个文件名都绑定到它的文件内容（用inode标识）

ln 创建硬链接 硬链接绑定的是同一个文件 修改硬链接的文件同时也会修改源文件 只是用两个文件名修改同一个文件

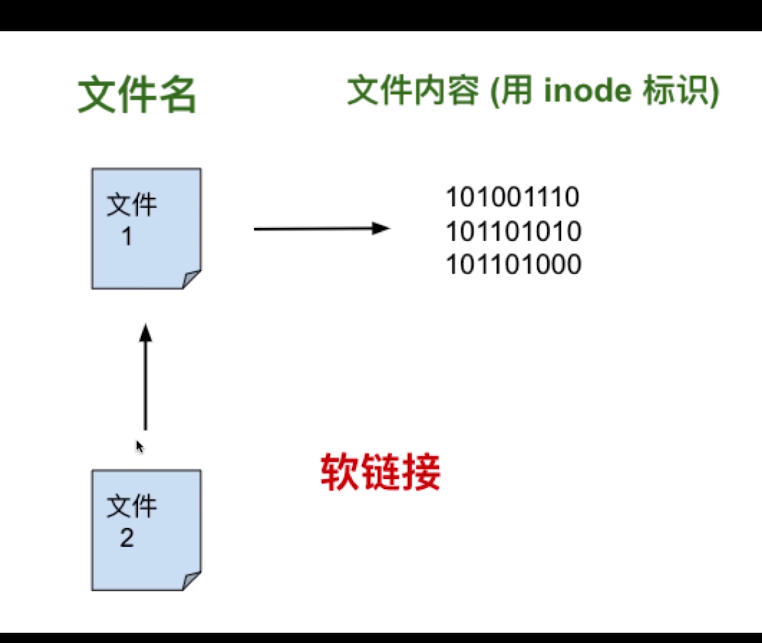
缺陷： 只能创建指向文件的硬链接，不能创建指向目录的，（通过一些参数的修改也可以但是比较复杂 所以创建指向目录的一般都是用软连接）



ln file1 file2 创建一个file1的硬链接名为file2

ln -s file1 file2 创建一个file2的软连接 名为file2

(file2是file1的一个快捷方式，他指向的是file1所以显示的是file1的内容，file2的inode和file1不一样，也就是文件内容不一样)



软连接的特点：

如果删除file1那么file2就会变成死链接，因为只想的文件不见了（硬链接删除其中一个另一个不会受到影响）

软连接可以只想目录，硬链接不行

## 用户与权限

sudo 暂时代替root用户执行root用户权限下的命令

sudo su 切换到root用户（只在当前终端，关闭终端后会重置当前用户）

exit 当在root用户下 执行该命令则会退回到当前用户

su 切换到root用户

su - 切换到root用户并打印登录时间去到root用户的家目录

添加用户

usaradd 添加用户（只能用root用户添加） useradd thomos

passwd 修改用户密码 passwd 密码

userdel 删除用户 userdel thomas

userdel -r或者--remove删除用户并删除家目录 userdel -r thomas

## 用户群组

groupadd 创建一个群组

groupdel 删除一个群组

usermod 修改用户账户

usermod -l 对用户重命名

usermod -g 修改用户群组 usermod -g friends thomas(将thomas放入friends群组)

usermod -G 将用户放入多个群组 usermod -G friends,happy,funny thomas

usermod -aG 在不退出当前群组的同时加入其它群组

groups 获取用户属于哪个群组

chown 改变文件或群组的所有者

chown thomas file.txt 修改file.txt文件的所有者为thomas （群组不变）

chgrp 修改文件群组

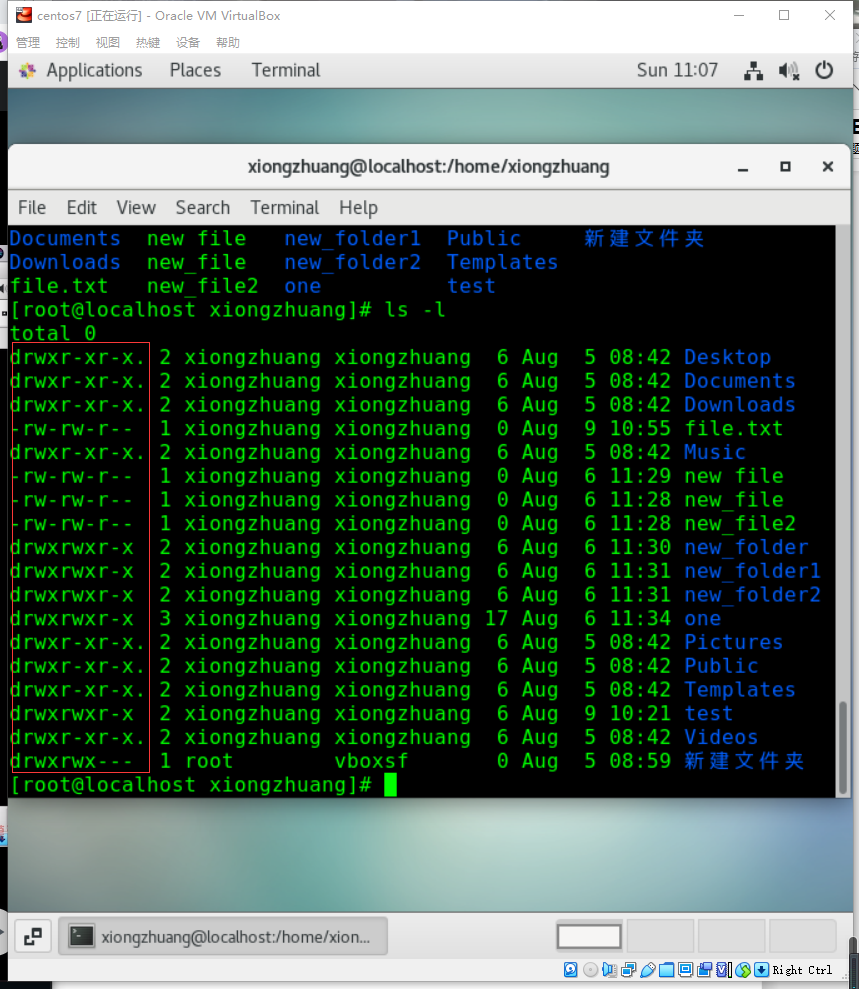
chgrp thomas file.txt 修改file.txt文件的群组为thomas

chown thomas:friends file.txt 修改文件的所有者为thomas，并修改群组为friends

-R 递归设置子目录和子文件

chown -R xiongzhuang:xiongzhuang /home/thomas 将thomas家目录的所有者和群组都改为xiongzhuang

## 文件权限管理



运行ls -l 第一列就是文件访问权限符

d 英语directory的缩写 表示目录 就是说这是一个目录

l 英语link的缩写 表示连接 就是说这是一个连接

r 英语read的缩写 表示读 就是说可以读这个文件

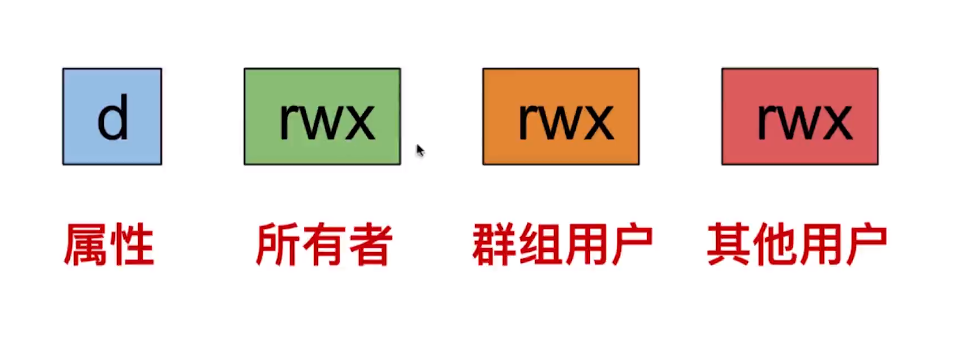
w 英语write的缩写 表示写 就是说可以修改这个文件

x 英语execute的缩写 表示执行 就是说可以运行这个文件

如果x权限在一个目录上的话表示这个目录可以被读，如果同时又r权限就是可以打开此目录来查看其子目录和子文件

如果相应位置有字母表示的是有相应权限 如果是短横- 表示没有 相应的权限

第十一位上的点 表示selinux的安全标签 是一种访问控制体系



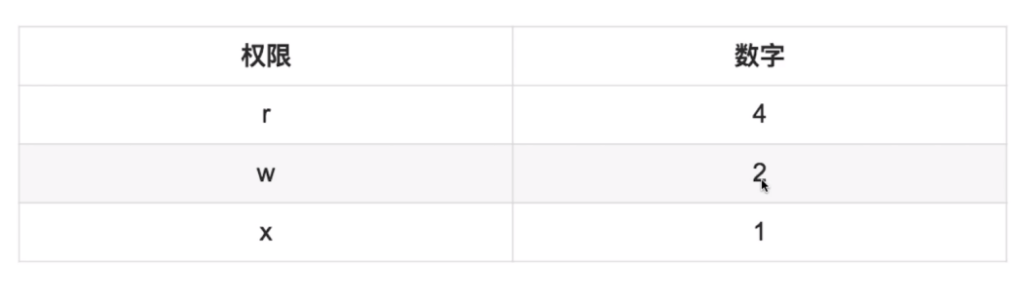
第一组rwx表示文件的所有者对于此文件的访问权限

第二组rwx表示文件所属群组的其他用户对于此文件的访问权限

第三组rwx表示文件除前两组之外的其他用户对于此文件的访问权限

chmod 修改文件的访问权限 change mode

权限所对应的数字



所有可能的权限数字组合



chomd 600 file.txt 设置file.txt文件对于所有者为可读写，对于所在群组的其他用户没有任何权限，对于其他用户没有任何权限

用字母来分配权限

u user的缩写，是英语用户的意思，表示所有者

g group的缩写，是英语群组的意思，表示群组用户

o other的缩写，是英语其他的意思，表示其他用户

a all的缩写，是所有的意思，表示所有用户

与之配合的还有

+ 加号，表示添加的权限

- 减号，表示去除权限

= 等号，表示分配权限

chmod u+rx file 文件file的所有者增加读和运行的权限

chmod g+r file 文件file的群组其他用户增加读的权限

chmod o-r file 文件file的其他用户移除读的权限

chmod g+r, o-r file 文件file的群组其他用户增加读的权限，其他用户移除读的权限

chmod go-r file 文件file的群组其他用户和其他用户均移除读的权限

chmod +x file 文件file的所有用户增加运行的权限

chmod u=rwx,g=r,o=- file 文件的所有者具有读写执行的权限，群组中其他用户具有读的权限，其他用户没有权限

chmod 配合-R 的参数可以递归的修改目录与文件访问权限

## Nano文本编辑器

Ctrl + x 退出文本编辑器

Ctrl + o 保存文件

ctrl + r 打开其他文件

ctrl + y 跳到上一个屏幕

ctrl + v 跳到下一个屏幕

ctrl + k 剪切当前行

ctrl + w 查找文本

ctrl + u 粘贴刚剪切的内容

ctrl + g 打开帮助

ctrl + / 替换

ctrl + F 向前移动一格光标

ctrl + B 向后移动一格光标

ctrl + p 向上移动一行

ctrl + n 向下移动一行

按下esc 再按下x键去除下边的帮助文档

nano file 用nano打开文件file

nano -m file 激活鼠标

nano -i 激活自动缩进的功能

nano -A 激活智能home 键

可以通过在用户家目录中创建.nanorc文件来配置nano

set mouse 激活鼠标

set autoindent 激活自动缩进的功能

set smarthome 激活智能home键

nanorc配置文件 全局nano配置的nanorc文件在etc下

bashrc配置文件 终端的配置文件在etc下名字是bashrc，可以通过bashrc文件来修改终端的展示形式

profile配置文件 配置通过F2到F6切换的需要登录非图形界面的终端的文件

profile文件会调用.bashrc，修改了.bashrc也就是间接的修改了profile文件，profile文件会用profile本身的配置加上.bashrc文件的配置，

修改配置文件时需要重启linux服务器才能生效，如果想要立即生效需要用到source命令重新加载配置文件

RC结尾的是配置文件，通常在系统启动的时候被调用

rc是runcomm的缩写，即运行命令

## 正则表达式和数据操作

grep 筛选数据

grep text file text标识要搜索的文本 file代表要搜索的文件（如果搜索文本有空格就需要用双引号）

grep -i 忽略大小写

grep -n 显示行号

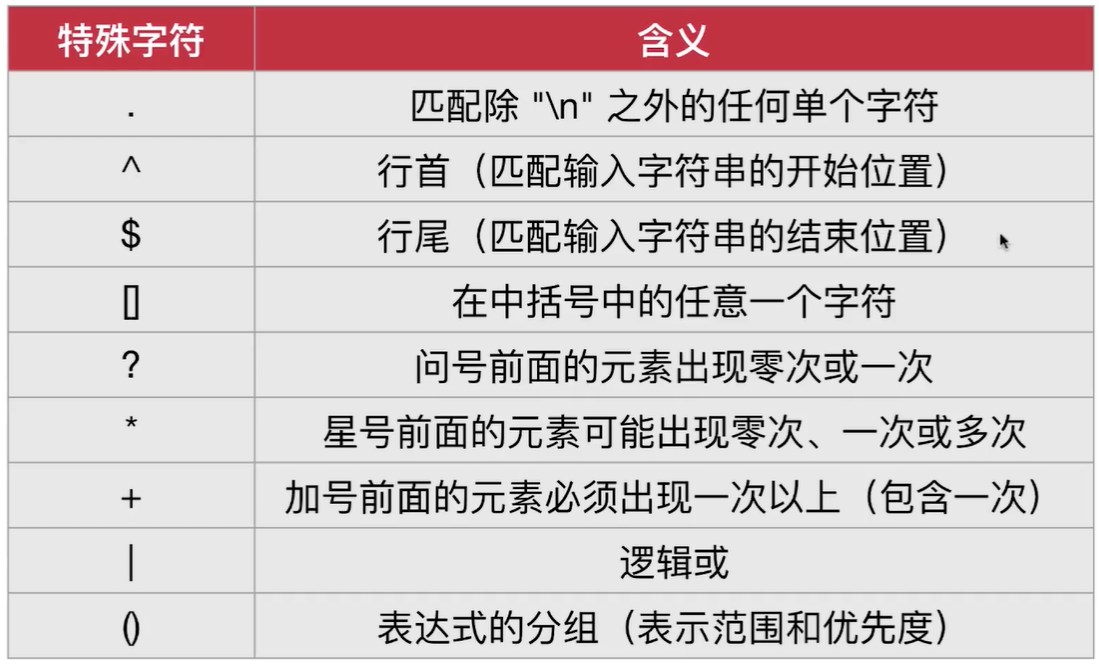
grep -v 只展示没有搜索内容的行

grep -r 在所有子目录与当前目录的文件中搜索

rgrep = grep -r

grep -E 使用正则表达式

egrep = grep -E



sort 为文件排序

sort file 为文件中的文字排序

sort -o target source 将源文件中的文字排序输入到另一个文件

sort -r 倒序排列

sort -R 随机排序

sort -n 对数字排序按照数值的大小



第一列表示行数

第二列表示单词数

第三列标识字节数

wc 文件的统计

wc -l 只对行数统计

wc -w 只对单词统计

wc -c 对字节统计

wc -m 对字符统计

uniq 删除文件的重复的内容 只能删除连续的重复行的内容

uniq source target 删除重复的内容并输出到另一个文件

uniq -c统计重复的行数

uniq -d 只显示重复行的值

cut -c 2-4 file 剪切文件的前2到前4个字符

cut -d 使用分隔符来剪切

cut -f 剪切下用分隔符分割的哪一块或哪几块区域  
cut -d , -f 1,3 notes.csv 剪切逗号隔开的第一部分和第三部分的数据

cut -d, -f 2- notes.csv 剪切逗号隔开第二部分以及之后的部分

## 流，管道，重定向

管道：把两个命令连起来使用，一个命令的输出作为另一个命令的输入

流： 计算机科学中，流是时间上可用的一系列数据元素，我们可以把流比喻成传送带上的物件，每个时间点传输一个，而不是多个打包传输

重定向： 重定向到文件或者另一个命令的输入（通过在命令间插入特定的符号，可以称为重定向流符号）

>重定向到一个文件（如果文件不存在，则会新建一个文件，如果已经存在那么就会把文件内容覆盖掉）

cut -d / -f 2 notes.csv > student 将命令的结果重定向到一个student的文件

/dev/null 黑洞文件

>>重定向到文件的末尾 追加的作用

1. ,2>>,2>&1 重定向错误输出到文件

cat file 2> error 重定向错误输出到文件error

cat file 2>> error 追加错误输出

cat file >error 2>&1标准输出和错误输出都重定向文件error

cat file >> error 2>&1 标准输出和错误输出都追加在error文件末尾

stdin 从键盘向终端输入数据，这是标准输入 标准输入流

stdout 指终端输出的信息（不包括错误的信息）标准输出

stderr 指终端输出的错误信息 标准错误输出 标准错误输出流

输入重定向和管道符号

< 符号用于指定命令的输入

cat < notes.csv = cat notes.csv效果一样 原理不同

cat notes.csv 指的是cat命令接受的输入时note.csv这个文件名，首先要打开notes.csv文件，然后打印内容

cat < notes.csv 指的是cat命令直接接受的是notes.csv这个文件的内容，cat只负责将其内容打印，打开文件并将文件内容传递给cat命令的工作则交给终端完成

<< 从键盘读取

<<符号的作用是将键盘的输入重定向为某个命令的输入

sort -n << END 输入数字并以END为结束 sort -n会将之排序

wc -m << END 输入文本并以END为结束 wc -m 命令会计算输入字符的字符数

sort -n << haha > numbersort 键盘输入数字进行排序并输出到numbersort文件中

|管道符 建立命令管道

命令1|命令2 命令1输出的结果作为命令2的输入

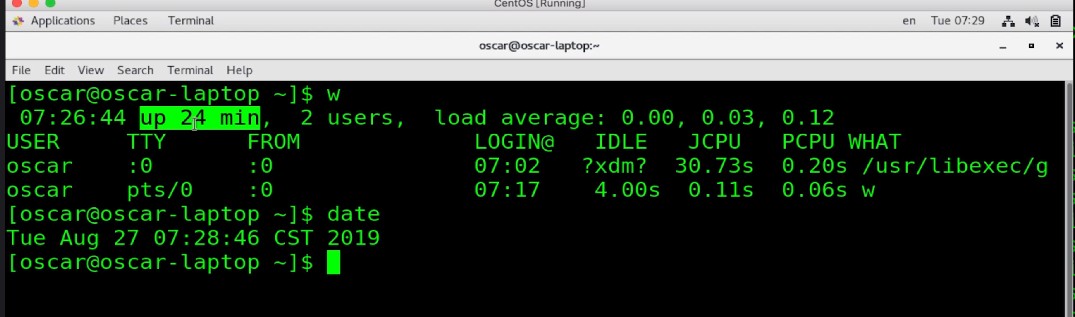
cut -d , -f 1 notes.csv | sort 剪切notes.csv 文件并将剪切后的字符进行排序

du | sort -nr |head 统计当前目录下的目录大小并进行倒序排列 并展示前10行

sudo grep log -Ir /var/log | cut -d : -f 1 | sort | uniq

遍历/var/log这个目录及其子目录，列出所有包含log这个关键字的行|然后剪切出文件名那一列（由冒号分隔的第一个区域）| 然后根据文件名排序| 然后去除重复的文件名

## 进程和系统监测

w 

第一行 当前时间 用户登录时间 登录用户数 系统负载 1分钟 5分钟 15分钟

下面的是

第一列：USER 用户名

第二列：TTY 登录的终端名称 :0的意思是指本地就是目前所在的图形终端 pts 表示伪终端从属 表示另外的终端 tty表示ctrl打开的纯命令行终端

第三列 from 登录用户的ip地址

第三列 login@ 显示登录时间

第四列 IDLE 空闲时间 表示多久没有活跃了

第五列 JCPU 该终端所有相关进程使用的cpu的时间

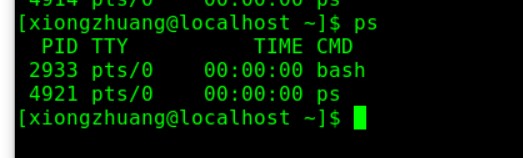
第六列 PCPU 执行当前进程的使用时间

第七列 WHAT 当前用户正运行的程序

who 显示当前有多少用户登录着

ps命令和top命令 列出运行的进程

ps命令显示的进程列表是静态的不会随着进程的增加而动态增加，只会显示当前终端运行的进程



PID 表示进程号

TTY 进程运行所在的终端

TIME 表示进程运行多久

CMD 产生这个进程的程序名

PPID 进程的父进程

ps -ef 列出所有进程

ps -efH 以乔木装列出所有进程

ps -u xxx xxx用户运行的进程

ps -aux 通过cpu和内存使用来过滤进程

ps -aux --sort -pcpu |less 通过cpu使用率来降序排列

ps -aux --sort -pmem |less 通过内存使用率来降序排序

ps -aux --sort -pcpu,+pmem|head 将cpu和内存参数合并一块，并通过管道显示前10个结果

pstree 以树形结构显示进程 与 ps -axjf命令相似

top 进程的动态列表

top会根据cpu使用率进行排序

f /F 改变进程列表排序所参照的列

u root 获取root用户的进程

k pid 结束某个进程

glances 系统监控软件

yum install epel\* -y

yum install python-pip python-devel -y

yum install glances -y

ctrl + c停止终端中正在运行的进程

kill 杀死进程（可以同时结束多个进程）

kill -9 强制结束进程

killall +程序名 结束所有进程  
halt 关闭系统

reboot 重启系统

poweroff 关机

## 管理前后台进程

linux中的进程状态

R 运行(正在运行或在运行队列中等待)

S 中断（休眠中，受阻）

D 不可中断

Z 僵死

T 停止

前台进程

例如ls 直接显示在前台

后台进程（不必等命令结束就可以输入其他命令）

&在后台运行进程（缺点：后台进程与终端相关联，一旦终端关闭或者用户登出，进程就会自动结束）

find / -name “\*.log” & 在后台运行find命令

nohup命令 使进程与终端分离（不受到终端关闭的影响）

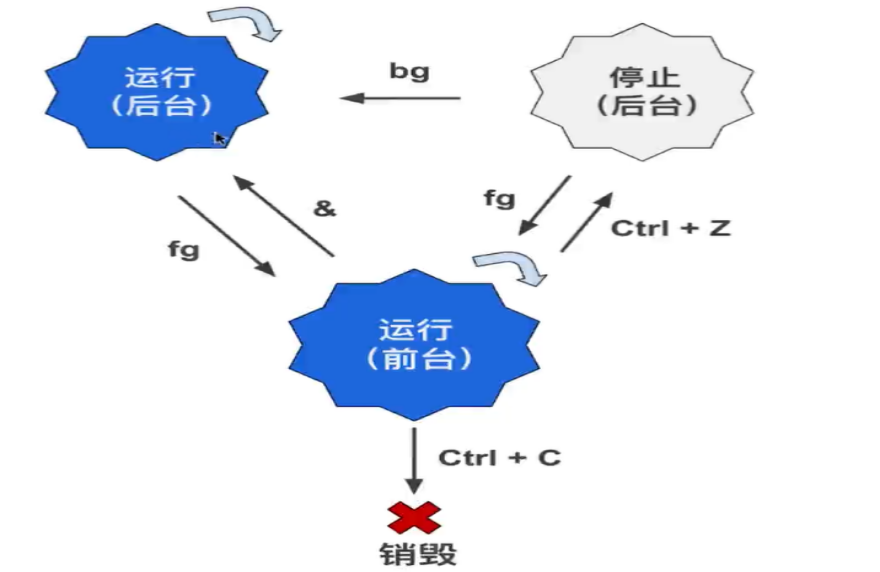
ctrl +z 使进程转到后台并停止运行

bg 是进程转到后台 bg不加参数会默认的去作用于最近的一个后台进程，如果加上%1 %2

或者1 2 就会结束对应的进程标号的进程

jobs 显示后台进程状态

fg 使进程转为前台运行



## 任务的定时和延期

date “+%H” 显示当前的小时

date "+%H:%M:%S" 显示时：分：秒

at命令 延时执行一个程序

at 22:10 在晚上10点10分执行某个命令 ctrl+d退出输入

at 22:10 12/10/19 在2019年12月10日的22点10分执行

at now +10 minutes 在10分钟之后执行程序

minutes表示分钟

hours表示小时

days表示天

weeks表示星期

months表示月

years表示年

atp 列出正在等待执行的at任务

atrm 删除正在等待执行的at任务 后接任务编号

sleep 休息一会儿

touch file; sleep 15 m; rm file 创建文件等待15分钟后删除文件

&& &&号前的命令执行成功才会执行后面的命令

|| ||号前的命令执行失败才会执行后面的命令

分号 不论分号前的命令执行成功与否，都执行分号后的命令

crontab 定时执行程序

安装crontab

yum install vixie-cron crontabs 安装crontabs

chkconfig crond on 设置为开机自启动

service crond start 启动

contab 的三个主要参数

-l 显示crontab文件

-e 修改crontab文件

-r 删除crontab文件

书写crontab文件的格式

m h dom mon dow command

m 表示分钟

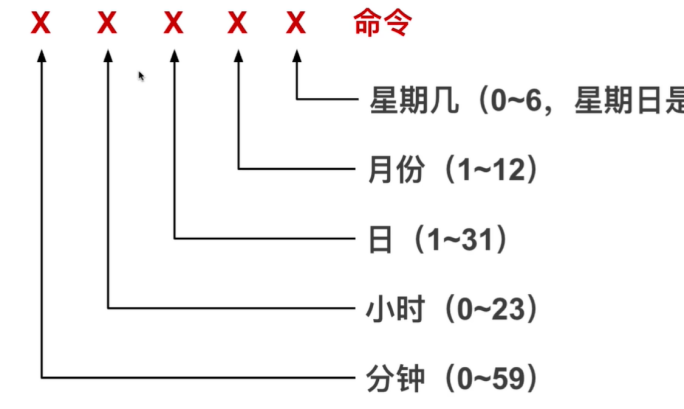
h 表示小时

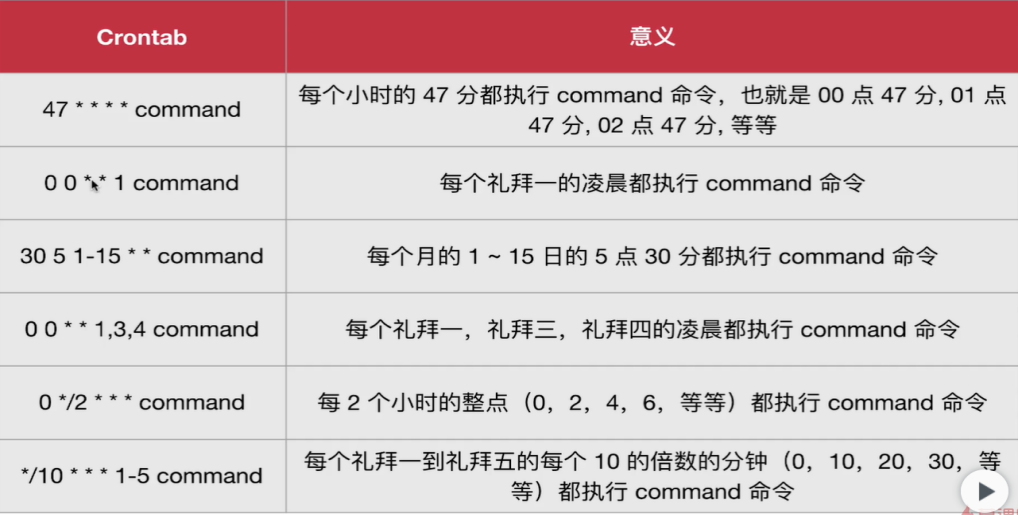
dom 表示一个月的那一天

mon 表示月份

dow 表示星期几

command 表示需要定时执行的命名





例子：

10 22 \* \* \* ls 在每天的22点10分执行ls命令

语法正确会显示crontab: installing new crontab 意思是安装新的crontab文件

## 文件的解压和压缩

tar 命令对于文件进行归档

-c 表示创建

-v 表示冗余会显示操作的细节

-f 表示文件制定归档文件

-tf 显示归档里的内容，并不解开归档（查看tar包里的内容）

-rvf 追加一个文件到tar归档里

-xvf 将生成的tar 解开

-zcvf 将文件通过gzip来压缩归档

-zxvf 将gz文件解压并解开tar

-jcvf 将文件通过bz2来压缩归档

-jxvf 将bz2文件解压并解开tar

gzip压缩tar包 直接跟tar包名可以对tar进行压缩，压缩后的文件结尾为gz

gunzip 解压gz结尾的文件

bzip2 压缩tar包 直接跟tar包名可以对tar包进行压缩,压缩后的文件结尾为bz2

bunzip2 解压bz2结尾的文件

zcat zmore zless 显示gz文件的内容

bzcat bzmore bzless 显示bz2文件的内容

zip file.zip file 压缩file为zip结尾的压缩文件

unzip file.zip 解压结尾为zip的文件

## 编译安装文件

alien 可以将deb安装包与rpm安装包互相转换

安装alien需要先执行yum install -y epel-release 安装epel

sudo alien -r xxx.deb 会将deb包转化为rpm包

sudo rpm -i xxx.rpm 安装rpm安装包（最好直接下载rpm包 alien转换的包并不能直接安装）

## 远程连接与SSH

查看IP ifconfig ip addr  
http协议 web最常用的传输网页的协议  
https协议 安全的超文本传输协议  
ftp协议 文件传输协议  
telnet 简单传输文件信息 没有加密

对称加密算法：用一个秘钥来加密信息 加密和解密用一个superkey来传输数据

非对称加密：使用公钥对数据进行加密，使用私钥对数据进行解密

对称加密的缺陷是superkey传输不安全，非对称加密的缺陷是太消耗系统资源

SSH协议是指首先通过非对称加密交换对称加密的密钥，再通过对称加密的方式来对数据进行加密解密

服务器先用明文的方式将公钥提供给客户机，然后客户机再通过公钥加密superkey的信息给服务器，然后服务器通过私钥解密之后，然后服务器再通过superke加密数据传递给客户机，客户家再通过superkey进行解密 获取想要的数据

如果只是想用ssh远程连接到别的机器就只需要安装openssh-client

yum install openssh-clients

如果想要作为服务器开放ssh服务，则需要安装openssh-server

yum install openssh-server

启动sshd systemctl start sshd

关闭sshd systemctl stop sshd

重启sshd systemctl restart sshd

查看sshd systemctl status sshd

设置sshd为开机自启动 systemctl enable sshd

使用linux连接远程服务器首先安装openssh-clients，然后使用命令ssh root@ip

全局config文件

ssh客户端的配置文件： /etc/ssh/ssh\_config

ssh服务端的配置 ： /etc/ssh/sshd\_config





修改config文件的话需要重启sshd服务来使之生效

## SSH免密码登录

ssh两种安全验证的方法

1 基于口令 用户名密码

2基于秘钥 客户机生成秘钥对，把公钥上传到服务器，并与服务器的公钥进行比较，这用验证登录的方法更加安全，也被成为“公钥验证登录”  
1 ssh-keygen 生成秘钥对（公钥和私钥）默认使用rsa算法的秘钥

运行命令后会生成两个文件 id\_rsa.pub公钥 id\_rsa 私钥

2 ssh-copy-id root@ip地址 把客户机的公钥传送到服务器 服务器会将秘钥文件追加到服务器的~/.ssh/authorized\_keys文件中

如果取消公钥免密码登录的话就需要使用

ssh -o PreferredAuthentications=password -o PubkeyAuthentication=no user@ip地址

## Vim文本编辑器

vim的多种模式

交互模式(默认模式不可输入文字)

h,j,k,l 上下左右移动

0 移动到行首

$ 移动到行末 = shirt +4

w 一个单词一个单词的移动

x 删除字符（先按数字的话就会删除光标后的多少个字符）

dd 行（先按数字的话就会删除光标下的多少行）

dw 删除单词（删除一个单词，如果光标在一个字符上的话只会删除到下一个空格的字符）

d3w 删除3个单词（或者通过先按数字再按dw）

d0 删除当前光标到行首的所有字符

d$ 删除当前光标到行末的所有字符

yy 复制当前行到内存中

y$ 复制光标到行末的所有字符

y0 复制光标到行首的所有字符

p 粘贴会粘贴到下一行（p前加数字就会粘贴多少次）

r 替换字符（按下r再输入想要替换的字符）

R 切换到替换模式

u 撤销最近的修改（输入数字就会撤销多少次的操作）

ctrl +r 重做之前的动作

g 跳转到指定行（按下数字并按下G会跳转到指定行号相当于连续按gg）

G 跳到最后一行

gg 跳到第一行

/ 向下查找（按下/输入文本 就会高亮文件中有该文本的地方）

? 向上查找

n 移入下一个匹配字符处

shrit+n 移动到上一个匹配字符处

s 删除当前字符并进入插入模式

shirt+s 删除当前行并进入插入模式

插入模式（使用i来进入插入模式）

I 进入一行的行首来进行增加

i 在当前位置进行编辑

a 在光标的下一个位置进行输入

A 在行尾进行输入

o 在下面另起一行输入

O 在上面另起一行输入

命令模式

按esc键进入命令模式

:w 保存文件

:q 退出vim

:q! 强制退出并不保存最近一次修改

:wq 保存并退出

:x 保存并瑞出

:set nu 显示行号

:s /see/look 将光标所在行see的首个匹配项变为look

:s /see/look/g 将光标所在行的所有see的匹配项变为look

:#,# s/旧字符串/新字符串/g 将第#行到第#行的旧字符串替换为新字符串

:7,9 s/see/look/g 将第7行到第9行的see的匹配项变为look

:%s/旧字符串/新字符串/g 会替换所有的匹配字符串

:r合并文件(:r 文件路径)

:sp 横向分屏

:sp 文件名 就会将另一个文件与当前文件分屏查看

:vsp 垂直分屏

分屏下的快捷建

ctrl+w切换到另一个分屏



ctrl + w + 扩大该分屏

ctrl + w - 减少改分屏

ctrl + w = 重新均匀分配所有分屏

ctrl + w r 调换分屏的位置

ctrl + w q 退出该分屏

ctrl + w o 只留下当前分屏

:! 运行外部命令

可视模式

v 字符可视模式

V 行可视模式

ctrl +v 块可视模式

可视模式下输入如下命令

d 标识删除

I 插入

## Vim的配置

将etc下的vimrc文件复制到家目录下命名为.vimrc

syntax 配置语法高亮

set number 显示行号

set showcmd 显示命令

set ignorecase 在查找的时候忽略大小写

set mouse=a 激活鼠标（可视模式下U可以将文本转为大写，u可以将文本转为小写 d或x键可以删除选中文本）

可以参考网上的vim配置（github）

## 安全的文件传输，轻松同步

wget 可以从网络获取想要的文件

scp 网间拷贝

scp file.txt [root@192.168.1.5:/root](mailto:root@192.168.1.5:/root) 将当前目录下的file.txt文件拷贝到远程电脑ip为192.168.1.5的root目录下

-P 指定端口

ftp -p ftp.fr.debian.org 连入ftp服务器

sftp username@ip

rsync 同步备份

-a 保留文件的所有信息

-r 递归调用

-v 冗余模式

rsync -arc Images/ bak/ 将images目录下的所有文件全部备份在bak目录下

## IP地址

host 主机名 可以获取到该主机名的IP地址

# Shell

Sh:Bourne Shell的缩写 shell的祖先

Bash : Bourne Again Shell的缩写 是shell的进阶版本

Ksh: Korn Shell的缩写 一般在收费的Unix版本上比较多见

Csh: C Shell 的缩写 比较类似C语言

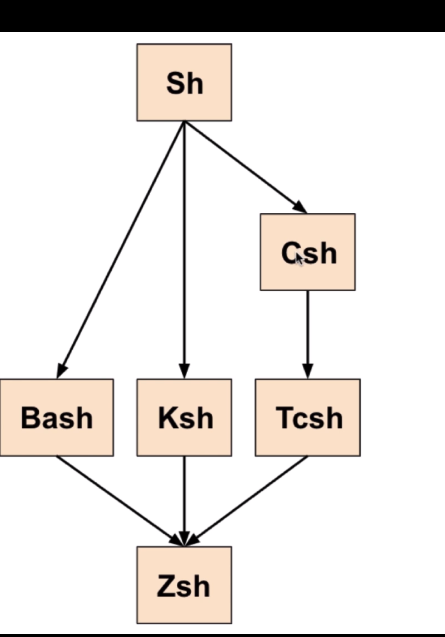
Tcsh: Tenex C Shell的缩写 Csh的优化版本

Zsh: Z shell 集Bash Ksh和Tcsh各家之大成

安装shell的命令 yum install ksh (安装ksh)

chsh 改变shell

bash是默认的shell



编写一个shell

#! 被称为Sha-bang 或者Shebang

#!/bin/bash指定该脚本文件的shell为bash

shell的注释以#号开头

./test.sh 运行shell脚本文件

bash -x test.sh 以调试模式运行shell文件

echo #PATH查看path环境变量 如果把shell脚本放入列出的文件夹中的话那么在所有的目录中都可以执行该shell

## shell的变量

message=’Hello World’ 定义一个message变量 等号左右不要加空格！！！！

echo 在英语中是回声的意思 他的作用是在终端中显示传入的信息

echo -e 使转义符生效 例如 echo -e “first line\nsecond line”

在脚本中使用echo $message 显示message变量的值（变量前需要加$符号）

单引号：变量被包含在单引号中变量不会被解析会直接被认为是字符串显示

双引号: 不用与单引号忽略所有字符，双引号忽略大多数特殊字符但不包括美元符号（$）

反引号（`） 反斜杠\

反引号: 反引号要求shell执行被它括起来的内容

read 请求输入 read命令读取到的文本会立即被储存在一个变量里（执行脚本后会让用户输入一个值）可以输入输入多个变量值

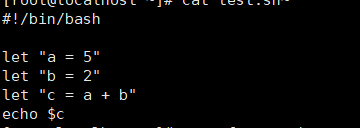
-p显示提示信息 read -p ‘输入显示的提示信息’ 变量1 变量2

-n 显示字符数目 read -p ‘输入显示的提示信息’ -n 5 变量1 变量2

-t 限制输入时间(以秒为单位) read -p ‘输入显示的提示信息’ -t 5 变量1

-s 隐藏输入内容 read -p ‘输入显示的提示信息’ -s 变量1

## 数学运算

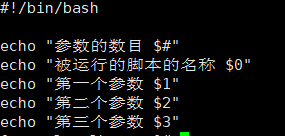


使用let命令进行数学运算

## 环境变量

env 显示系统的环境变量

export 输出 添加环境变量



输出脚本运行的参数

shift 可以挪移参数指向 常被用于循环中使得参数一个一个的被处理

array={‘value1’ ’value2’ ’value3’} 定义数组 数组中的元素中间需要加空格

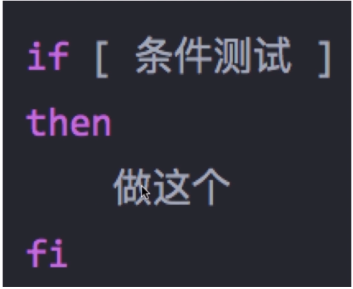
${array[2]} 输出数组下标为2的元素

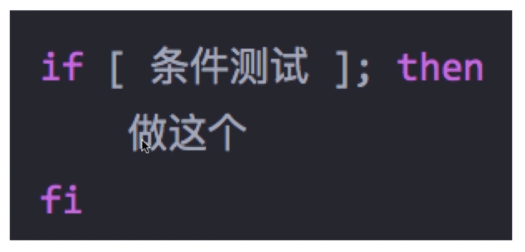
array[3]=’value4’ 给数组赋值

$array[\*] 输出数组所有元素

不少shell语言的数组下表是以1开始的 例如Csh Tcsh Zsh

## shell条件

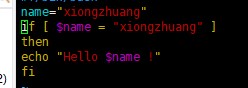




fi表示结束

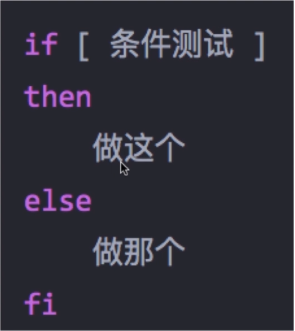
then表示那么

中括号的条件测试两边需要加空格

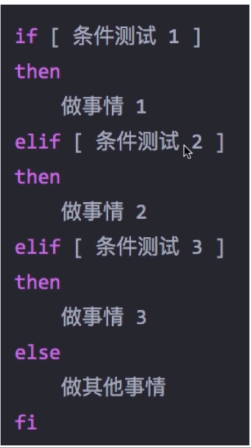


shell语言中的等于是用一个等号表示的

else 否则

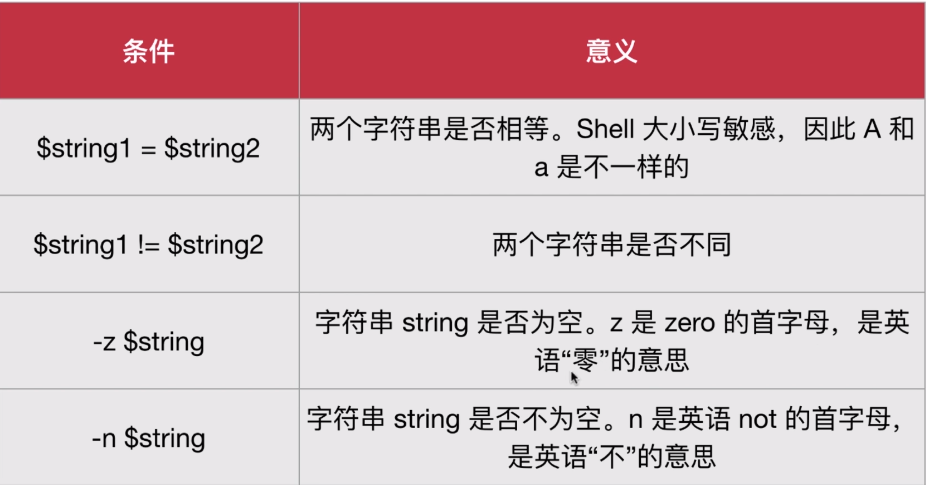


elif else if

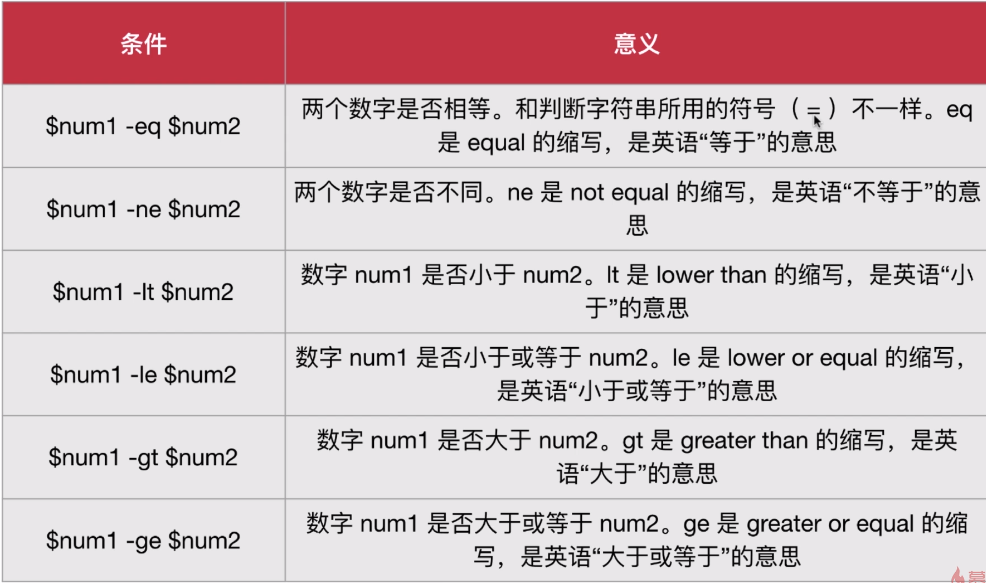


### 条件测试

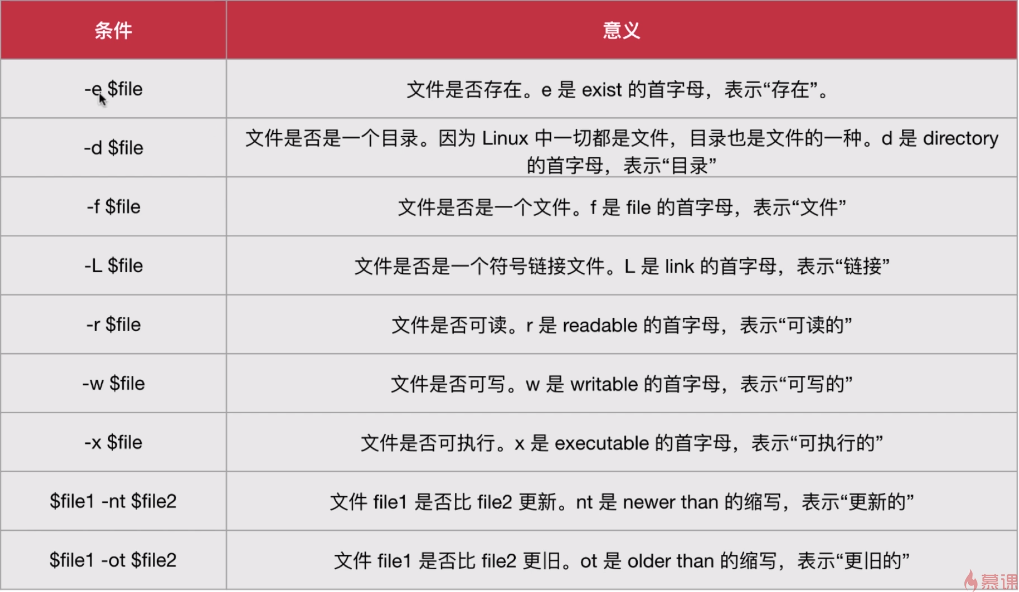
测试字符串



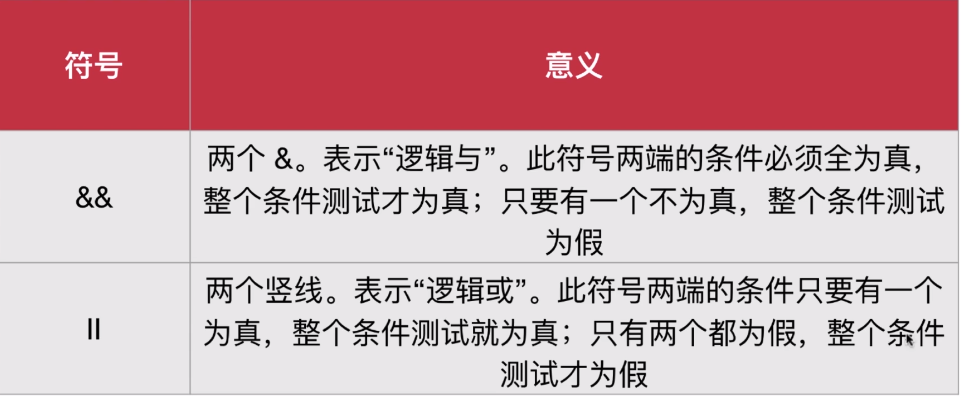
测试数字



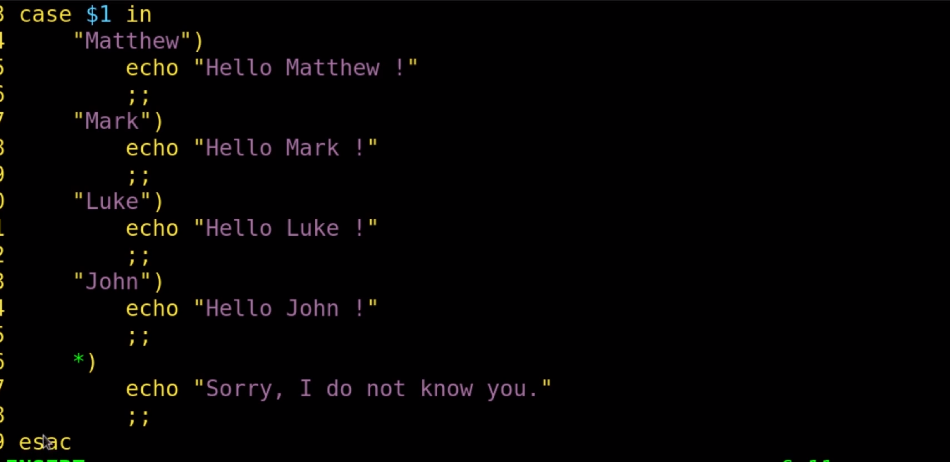
测试文件



测试逻辑



反转测试 ！叹号表示非



case 对于同一个变量进行条件判断

in 后加条件判断

两个分号表示break;

\*) 表示的是else

esac 结束

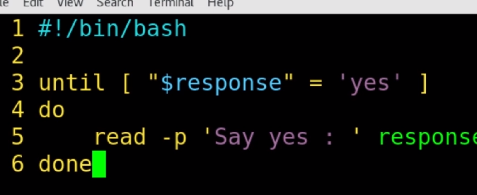
case中的或和并且 用单个&或|来表示

## shell的循环语句

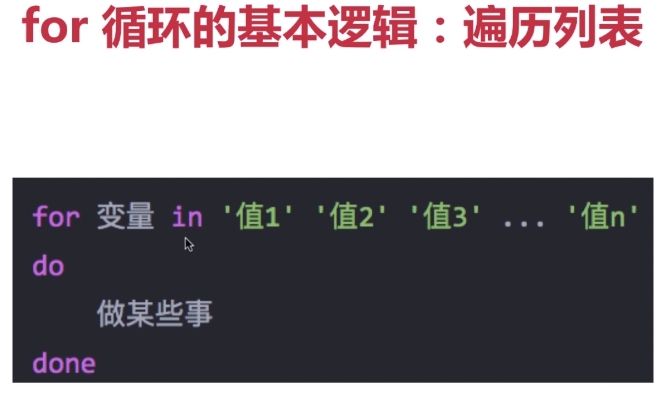
while



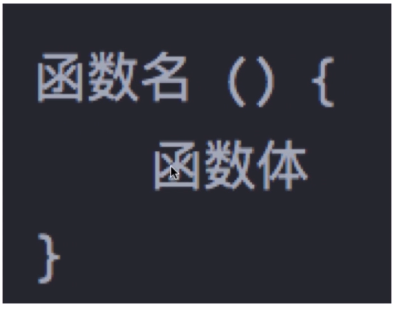
until



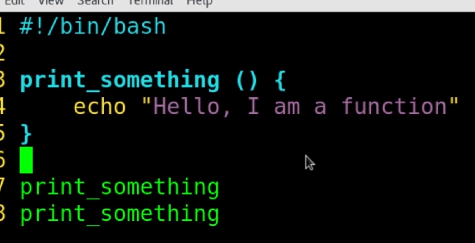
for



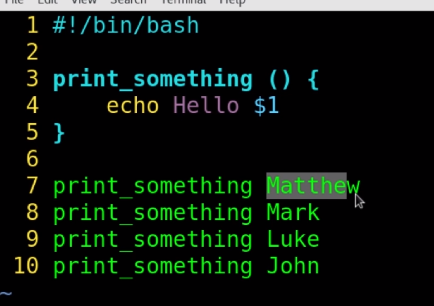
## shell 函数

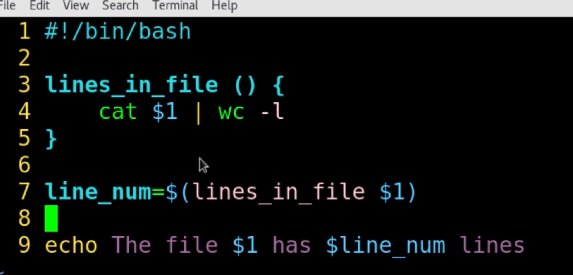


函数名后面跟着的圆括号里不加任何参数

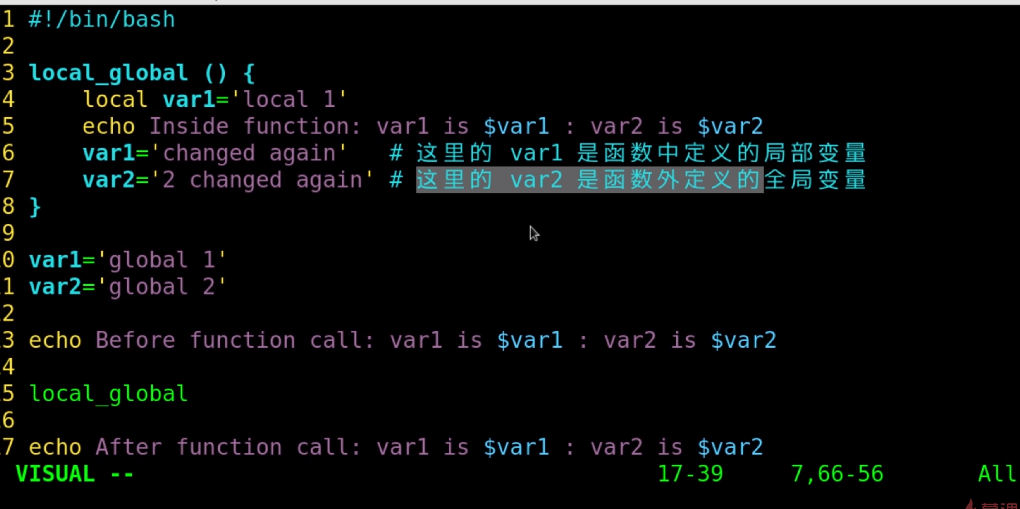


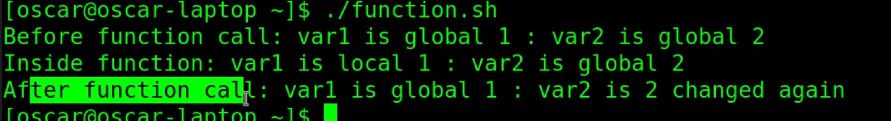
加参数的方式是在调用代码函数的后面放置参数





统计文件行数并打印

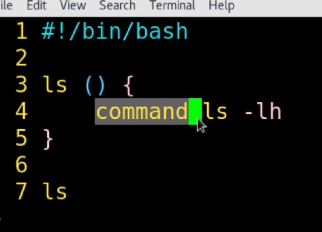




函数作用域与java相同

重载命令

command 重载命令



使ls 执行ls -lh的效果