# Linux

## 常用命令

### 时间格式化

export TIME\_STYLE='+%Y-%m-%d %H:%M:%S'

### 查看以字符编码

echo $LANG

env | grep LANG

export | grep LANG

locale

### 设置字符编码

export LANG=zh\_ZN.UFT-8

设置当前会话字符编码 会话失效设置失效

Vim /etc/sysconfig/i18n

Source /etc/sysconfig/i18n 强制生效

### 根据端口号查看服务

netstat –anp | grep 8080

### 查看文件大小

du –h /file 查看明细以及总大小

du –sh /file 查看总大小

nginx软启动

./nginx -s reload

### 定时任务

crontab -e 添加要设置的定时任务

/etc/init.d/crond status 查看状态

Ps -ef|grep crond 查看进程

定时任务：cd /etc

**cron.d/ #系统定期需要做的任务**

cron.deny #用于控制是那些用户不能使用Crontab功能

crontab #设定定时任务执行文件

cron.hourly/ #每小时执行一次的job

cron.daily/ #每天执行一次的job

cron.weekly/ #每星期执行一次的job

cron.monthly/ #每月执行一次的job

30 10 \* \* 1,2,3,4,5 /weblogic/user\_projects/domains/sssl\_domain/clearAlllogs.sh

**注意：**

**脚本要写绝对路径**

**crond 服务是否启动**

**是否有权限执行**

**时间格式是否正确**

**service crond status**

**service crond start**

**查看开始是否启动crond**

**chkconfig|grep crond 2 3 4 5 为on则为开机启动**

**crond 0:off 1:off 2:on 3:on 4:on 5:on 6:off**

### Ntp

Check ntp on

1查看ntp服务状态

service ntpd status

2ntp启动

service ntpd start restart stop

3查看

watch ntpq -p

### Alias

用户家目录下 .bashrc

Ls /etcprofile.d

Unalias ls

### 通配符

Ls \*.txt 多个字符

Ls ?.txt 一个字符

Ls [123].txt

Ls [0-9].tx

Cat 1.txx>2.txt 覆盖

Cat 1.txt>>2.txt 追加

ls aaa2>a.txt 正确的结果

ls aaa 2>>a.txt错误的结果

ls aaa2&>a.txt 正确的结果

ls aaa 2&>>a.txt错误的结果

### 暂停

Ctrl +z 暂停

Bg

Fg

### Cut

cat /etc/passwd |head -2|cut -d ":" -f 1,2,3

### sort

sort /etc/passwd

### tee

sort ab.txt |uniq -c |tee -a 22.txt

### 正则表达式

**Grep ‘ro\*r’ passwd**

**零个或者多个前面的字符o**

**Grep ‘ro.r’ passwd**

**表示任意一个字符**

**Grep ‘ro?t’ passwd**

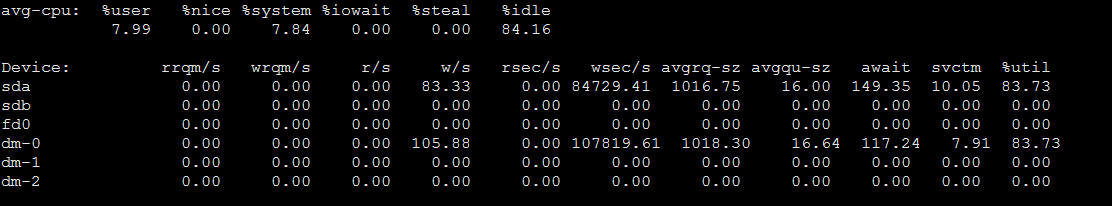
**零个或者一个o**

**Grep ‘ro+t’**

**1个或者多个前面的字符**

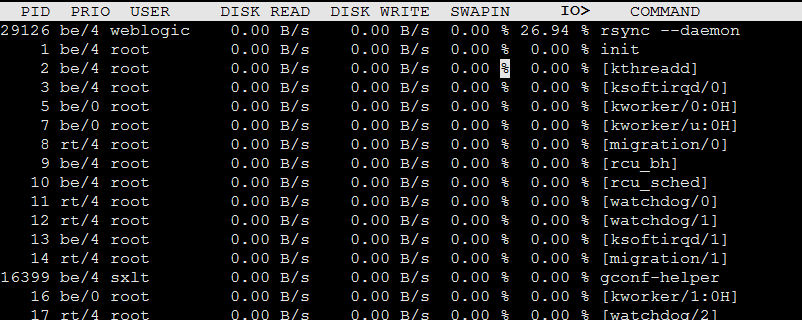
### Iostat

查看iostat -x 1 一秒刷新一次



### Iotop

Iotop -o 只显示有io操作的进程



### Wc

统计有多少个httpd请求

ps -ef|grep httpd |grep -v grep | wc -l

-l 统计行数

-c 统计字节数

-m 统计字符数

### Tcpdump

tcpdump tcp port 80 -i eth2 -w tcpdump2018.pcap

### 查询关键字并输出关键字所在行号

grep -n ERROR sssSvr5501.log

Head -100 (Tail -f 行号-100 )

### 统计文件包含某个字符

grep -o E022登录会话已经超时 wsbsSvr601.log |wc -l

Grep -o 只输出文件中匹配的部分

Wc -o 统计行数

### 测试系统io

iostat -d -x -k 1 100

### Scp

Scp weblogic@192.168.51.2:/home/weblogic/Apps/app2/clearlogs.sh /home/weblogic/Apps/app2/

### Nmon

D

Io读写情况查看

### Jdk卸载安装

rpm -qa | grpe jdk

rpm -e java-1.7.0-openjdk-1.7.0.79-2.5.5.4.el6.x86\_64 --nodeps

vim /etc/profile

**#set java environment**

**JAVA\_HOME=/usr/java/sunjava6\_64**

**JRE\_HOME=$JAVA\_HOME/jre**

**CLASS\_PATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib**

**PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin**

**export JAVA\_HOME**

**export JRE\_HOME**

**export CLASS\_PATH**

**export PATH**

source /etc/profile

java -version

### Mount

mount -t nfs 192.168.50.2:/software/appstatic /software/appstatic

vim /etc/exports

**/software/appstatic \*(rw,no\_root\_squash)**

exportfs –a

service nfs restart 有肯能没有启动服务

挂载镜像文件

mount -t iso9660 -o,loop /home/sxlt/rhel-server-6.7-x86\_64-dvd.iso /mnt/

### 解锁

**pam\_tally2 --user=weblogic --reset**

### Yum源配置

**1上传ios文件到home/sxlt/下**

**2 mount -o loop /home/sxlt/rhel-server-6.7-x86\_64-dvd.iso /mnt/**

**3 vim vim /etc/yum.repos.d/local.repo**

**[local.repo]**

**name=local.repo**

**baseurl=file:///mnt**

**enabled=1**

**gpgcheck=0**

**yum -y install tree**

### 修改密码

**chattr -i /etc/passwd /etc/group /etc/shadow /etc/gshadow**

**echo 密码 | passwd --stdin weblogic**

**chattr +i /etc/passwd /etc/group /etc/shadow /etc/gshadow**

### ln 源端 目标端

ln -s /weblogic/user\_projects/domains/sssl\_domain/ ln\_weblogic

正确删除软连接 rm -rf ln\_weblogic

错误删除软连接 rm -rf ln\_weblogic/ 连文件都删除了 比较苦逼 小心

### Ls

**ls -ltr --time-style="+%y-%m-%d %H:%M:%S"**

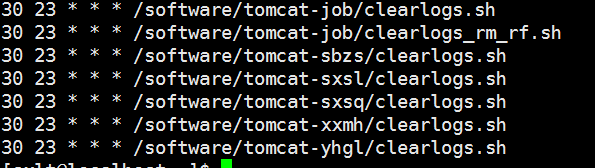
## 脚本

### 清空当前目录下所有日志

**清空当前目录下所有以log和out结尾的文件 结合定时任务**

**例如每天晚上十点钟清空日志**

**Crontab -l**



for i in `find . -name "\*.log" -o -name "\*.out" ` ;do cat /dev/null>$i;done;

**Summery:**

**定时任务查看命令 crontab -l**

**定时任务编辑命令 crontab -e**

**定时任务格式：**

**\* \* \* \* \* commond**

**分时日月周 命令**

### Tomcat启动脚本

#!/bin/bash

source ~/.bash\_profile

cd `dirname $0`

TOMCAT\_HOME=`pwd`

TOMCAT\_DIR='tomcat-sbzs'

echo "TOMCAT\_HOME:$TOMCAT\_HOME"

echo "TOMCAT\_DIR:$TOMCAT\_DIR"

# kill taxcp

TAXCP=`ps -ef|grep java|grep ${TOMCAT\_DIR}|grep ${TOMCAT\_HOME}|grep -v grep|awk '{print $2}'|xargs`

echo `ps -ef|grep java|grep ${TOMCAT\_DIR}|grep ${TOMCAT\_HOME}`

echo "[Kill Old's Pid:$TAXCP]"

if [[ "$TAXCP" ]] ;then

kill -9 $TAXCP

fi

cd $TOMCAT\_HOME

rm -rf logs/\*

rm -rf work/\*

rm -rf temp/\*

rm -rf \*.log

curl -s "http://192.168.51.32:6789/confCenter/conf?action=reload&confTable=conf\_default" > /dev/null

export TAXCP\_OPTS="-DsubGroup=sxds -DconfCenterAddr=http://192.168.51.32:6789/confCenter/conf -DconfTable=conf\_default -Ddubbo.registry.file=/tmp/.dubbo/dubbo-registry-sbzs.cache"

cd $TOMCAT\_HOME

/software/${TOMCAT\_DIR}/bin/catalina.sh start

tail -f ./logs/catalina.out

**Summery:**

**temp用户缓存目录**

**Work 是jsp转换成class的工作目录**

**重启删除temp和work防止缓存导致发布项目不生效**

**User.xml里面有tomca配置了tomcat的manger用户名和密码**

**Service.xml里面配置了端口号**

### Zookeeper启动脚本

port1=3881;

port2=3882;

port3=3883;

#根据端口号查询对应的pid

pid1=$(netstat -nlp | grep :$port1 | awk '{print $7}' | awk -F"/" '{ print $1 }');

pid2=$(netstat -nlp | grep :$port2 | awk '{print $7}' | awk -F"/" '{ print $1 }');

pid3=$(netstat -nlp | grep :$port3 | awk '{print $7}' | awk -F"/" '{ print $1 }');

#杀掉对应的进程，如果pid不存在，则不执行

if [ -n "$port1" ]; then

kill -9 $pid1;

fi

#杀掉对应的进程，如果pid不存在，则不执行

if [ -n "$port2" ]; then

kill -9 $pid2;

fi

#杀掉对应的进程，如果pid不存在，则不执行

if [ -n "$port3" ]; then

kill -9 $pid3;

Fi

**Summery：**

**netstat -lnp | grep 3881**

**netstat -nlp | grep 3881 | awk '{print $7}' | awk -F"/" '{ print $1 }'**

**根据端口号查询pid 然后根据pid kill 掉进程**

注意awk的使用

## Linux Kernel or Directory

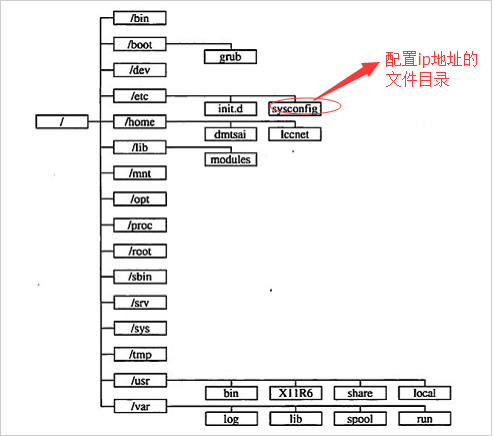
### Linux简介

Linux是一套免费使用和自由传播的类Unix操作系统，是一个基于POSIX和UNIX的多用户、多任务、支持多线程和多CPU的操作系统。它能运行主要的UNIX工具软件、应用程序和网络协议。它支持32位和64位硬件。Linux继承了Unix以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。Redhat的图形界面被称为X-Window 所以切换到图形界面命令为stratxsss

### Linux结构

**内核、用户界面、文件结构、实用工具**

#### 文件图





#### 文件结构

**/**

根目录,所有的目录、文件、设备都在/之下，/就是Linux文件系统的组织者，也是最上级的领导者。

**/bin**

bin就是二进制（binary）英文缩写。在一般的系统当中，你都可以在这个目录下找到linux常用的命令。系统所需要的那些命令位于此目录，比如ls、cp、mkdir等命令；功能和/usr/bin类似，这个目录中的文件都是可执行的、普通用户都可以使用的命令。作为基础系统所需要的最基础的命令就是放在这里。

**/boot**

Linux的内核及引导系统程序所需要的文件目录，比如vmlinuzinitrd.img文件都位于这个目录中。 在一般情 况下，GRUB或LILO系统引导管理器也位于这个目录。

**/cdrom**

这个目录在你刚刚安装系统的时候是空的。你可以将光驱文件系统挂在这个目录下。

例如：mount/dev/cdrom /cdrom

**/dev**

dev是设备(device)的英文缩写。这个目录对所有的用户都十分重要。因为在这个目录中包含了所有linux系统中 使用的外部设备。但是这里并不是放的外部设备的驱动程序。 这一点和我们常用的windows,dos操作系统不一样。 它实际上是一个访问这些外部设备的端口。我们可以非常方便地去访问这些外部设备，和访问一个文件，一个目录没有任 何区别。

**/etc**

etc这个目录是linux系统中最重要的目录之一。在这个目录下存放了系统管理时要用到的各种配 置文件和子目 录。我们要用到的网络配置文件，文件系统，x系统配置文件， 设备配置信息，设置 用户信息等都在这个目录下。

一般发行版的/etc目录下通常有

rc0.d-rc6.d目录，里面对应包含了从init0-init6不同运行级所要启动的命令，

rc.d目录下有 8个目录和3个文件，其中目录为rc0.d-rc6.d，分别代表了各个run-level的脚本，

目录中文件名称均以S及K开头，后接两位 数字与该服务名称，这些文件都是link文件，

指向/etc/rc.d/init.d目录。在rc3.d目录中，以S开头的文件为执行该服务，以K开头的文件则是杀掉该服务的意思。

数字代表启动顺序。

**/home**

如果我们建立一个用户，用户名是"xx",那么在/home目录下就有一个对应的/home/xx路径，用来 存放用户的主 目录。

**/lib**

lib是库（library）英文缩写。这个目录是用来存放系统动态连接共享库的。几乎所有的应用程 序都会用到这 个目录下的共享库。因此，千万不要轻易对这个目录进行什么操 作，一旦发生问题， 你的系统就不能工作了。

**/lost+found**

在ext2或ext3文件系统中，当系统意外崩溃或机器意外关机，而产生一些文件碎片放在这里。 当系统启动的 过程中fsck工具会检查这里，并修复已经损坏的文件系统。有时系 统发生问题，有很 多的文件被移到这个目 录中，可能会用手工的方式来修复，或移到文件到原来的位置上。

**/mnt**

这个目录一般是用于存放挂载储存设备的挂载目录的，比如有cdrom等目录。可以参看/etc/fstab 的定义。有时我 们可以把让系统开机自动挂载文件系统，把挂载点放在这里也是可以的。主要看/etc/fstab 中怎么定义了；比如光驱可以 挂载到/mnt/cdrom。

**/media**

有些linux的发行版使用这个目录来挂载那些usb接口的移动硬盘(包括U盘)、CD/DVD

驱动器等等。

**/opt**

这里主要存放那些可选的程序。你想尝试最新的firefox测试版吗?那就装到/opt目录下吧，这样， 当你尝试 完，想删掉firefox的时候，你就可以直接删除它，而不影响系统 其他任何设置。安装到 /opt目录下的程序， 它所有的数据、库文件等等都是放在同个目录下面。

**/proc**

可以在这个目录下获取系统信息。这些信息是在内存中，由系统自己产生的。操作系统运行时， 进程信息 及内核信息（比如cpu、硬盘分区、内存信息等）存放在这里。/proc 目录伪装的文件系统 proc的挂载目录，proc 并不是真正的文件系统，它的定义可以参见/etc/fstab

**/root**

Linux超级权限用户root的家目录。

**/sbin**

这个目录是用来存放系统管理员的系统管理程序。大多是涉及系统管理的命令的存放，是超级权 限用户 root 的可执行命令存放地，普通用户无权限执行这个目录下的命令，这个目录和 /usr/sbin;/usr/X11R6/sbin或 /usr/local/sbin 目录是相似的；我们记住就行了，凡是目录sbin中包含 的都是root权限才能执行的。

**/selinux**

好像是对SElinux的一些配置文件目录，SElinux可以让你的linux更加安全。

**/srv**

服务启动后，所需访问的数据目录，举个例子来说，www服务启动读取的网页数据就可以

放在/srv/www中

**/sys**

Linux内核中设计较新的一种虚拟的基于内存的文件系统，它的作用与proc有些类似，但除了与 proc相同的具 有查看和设定内核参数功能之外，还有为Linux统一设备模型作为 管理之用。

**/tmp**

临时文件目录，用来存放不同程序执行时产生的临时文件。有时用户运行程序的时候，会产生临 时文件。 /tmp就用来存放临时文件的。/var/tmp目录和这个目录相似。

**/usr**

这是linux系统中占用硬盘空间最大的目录。用户的很多应用程序和文件都存放在这个目录下。 在这个目录下，

你可以找到那些不适合放在/bin或/etc目录下的额外的工具。比如像游戏阿，一些打印 工具拉等等。/usr目录包含了 许多子目录：/usr/bin目录用于存放程序;/usr/share用于存放一些共享的 数据，比如音乐文件或者图标等等;

**/usr/lib**

目录用于存放那些不能直接运行的，但却是许多程序运行所必需的一些函数库文件。你的软件包 管理器(应 该是“新立得”吧)会自动帮你管理好/usr目录的。

**/usr/local**

这里主要存放那些手动安装的软件，即不是通过“新立得”或apt-get安装的软件。它和/usr目 录具有相类似 的目录结构。让软件包管理器来管理/usr目录，

而把自定义的脚 本(scripts)放到/usr/local目录下面，我想这应该是个不错的主意。

**/usr/share**

系统共用的东西存放地，比如/usr/share/fonts是字体目录，

/usr/share/doc和/usr/share/man帮助文件。

**/var**

这个目录的内容是经常变动的，看名字就知道，我们可以理解为vary的缩写，/var下有/var/log 这是用来存放 系统日志的目录。

/var/www目录是定义Apache服务器站点存放目录；

**/var/lib**

用来存放一些库文件，比如MySQL的，以及MySQL数据库的的存放地。

配置系统  .bash\_profile

环境变量  配置用户的环境变量

环境变量：配置.bashrc文件可以指定某些程序在用户登录的时候就自动启动

 　　　　系统环境变量设置：在root目录下  Vi .bash\_profile

修改PATH=$PATH:$HOME/bin :/安装目录/bin

需要退出（quit），重新登陆

当希望临时加入某个环境变量：用export PATH =$PATH : $HOME /bin:/root/test/t1

已定义好的环境变量：

SHELL:默认的shell

PATH:路径

USER：当前登录用户的用户名

显示变量内容：

echo $PATH

echo $USER

echo $SHELL

也可以使用env命令显示环境变量

通配符：

\* 代表多个字母或数字

？  一个

#### 重启

shutdown - h now 立刻关机

shutdown - r now 重启

reboot 重启

#### 设备硬盘操作

挂载设备

查看设备：使用命令“fdisk –l”可以查看系统的存储设备

挂载设备 ：首先使用mkdir命令建立挂载点目录，然后再使用mount命令挂载相关设备

mkfs [选项][-t <文件系统类型>] [设备名称] [区块数]

说明：把指定的设备格式为指定的文件系统。

查看磁盘使用情况

df [-参数] 比如， df-l

查看某个目录是在哪个分区

df [目录全路径]

查看Linux系统分区具体情况

fdish -l

#### 网络配置

配置网络接口可以使用三种不同的工具来完成：

使用网络接口配置程序netconfig

使用图形配置工具

使用终端命令ifconfig

监控网络状态信息:

显示网络统计信息的命令netstat

此命令用来显示整个系统目前的网络情况。例如目前的连接、数据包传递数据 、或路由表的内容。

如 netstat -an? netstat -anp|more (查看网络端口的使用的情况)

显示数据包经过历程命令：traceroute

route 查看路由表

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

1.追踪路由 ：tracert 目标ip/域名

2.测试两个ip是否畅通：ping 目标IP

3.window下查看IP情况：ipconfig

4.Linux/unix下查看IP情况：ifconfig

5.linux 网络环境配置：

第一种：

(1) 用root身份登陆，运行setup命令进入到text mode setup utility

对网络进行配置，这里可以进行ip、子网掩码、默认网管、dns的设置

(2) 这时网卡的配置没有立即生效，运行

/etc/rc.d/init.d/network restart命令我们刚才作的配置生效。

第二种：

(1)ifconfig eth0 x.x.x.x对网卡进行设置

(2)infconfig eth0 network x.x.x.x对子网掩码设置

对广播地址和dns使用默认的

注意：这样配置网络会立即生效，但是是临时生效

?

第三种：

(1) 修改/etc/sysconfig/network-scripts/ifcfg-eth0

这个文件的各个属性可以修改，包括ip，子网掩码、广播地址、默认网关

(2)这是网卡的配置没有生效

运行/etc/rc.d/init.d/network restart 命令我们刚才做的设置才生效

Linux中的所有设备都是文件，这种方法是最底层的方法，永久性修改

VMware下Linux和window ping

(1) windows下ipconfig查看VM网卡IP

(2) 配置Linux的IP使两者处于同一个子网的 即可以ping通

ifconfig <设备名> <IP地址> netmask <掩码>

例如：

ifconfig eth0 192.168.6.11 netmask 255.255.255.0

在网络配置界面中，通过“激活”或者“解除”按钮可以启动或者禁用网络接口，

网络控制程序network

/etc/rc.d/init.d/network? start|stop|restart

命令ifconfig :

ifconfig <设备名> [up|down]

命令ifup/ifdown

ifup eth0

ifdown eth0

配置文件名

功能

/etc/sysconfig/network

最基本的网络信息，系统启动时读取该文件

/etc/sysconfig/network-scripts/\*

此目录下的文件是系统启动时用来初始化网络的一些信息，例如：第一块以太网卡对应的文件为ifcfg-eth0

/etc/host.conf

域名解析的控制文件

/etc/hosts

域名或主机名与IP地址的映射文件

/etc/resolv.conf

域名服务器设置文件

/etc/protocols

定义使用的网络互联协议及协议号

/etc/services

设定主机的不同端口的网络服务

可以在 vim /etc/rc.d/rc.local 中加入 ifconfig eth0 192.168.35.0

每次启动的时候都会重新将ip设置为 192.168.35.0

#### 安装软件

获取最新版本的Apache源代码

将源代码解压缩

./configure –- prefix=/home/myapache

执行编译命令：make

执行安装命令：make install?

账号和用户管理

用户和组的配置信息保存在以下三个文件中：

/etc/passwd

/etc/shadow

/etc/group

2.10 更改文件所有者命令chown

chown [选项] user[:group] <file>...

更改文件访问权限命令chmod

chmod [选项] <mode> <file>...

mode：[ugoa][[+-=][rwxX]...][,…]，

mode也可以用数字来表示权限：

chmod abc file

a,b,c各为一个数字，分别表示User、Group、及Other的权限。权限是关于可读（r）、可写(w)、可执行(r)三个属性设置 值的和，其中r=4，w=2，x=1，

例如：

若要rwx属性，则4+2+1=7；

若要rw-属性，则4+2=6；

若要r-x属性，则4+1=7

#### 进程管理

Linux操作系统包括三种不同类型的进程，每种进程都有自己的特点和属性：

交互进程:由shell启动的进程。

批处理进程:这种进程和终端没有联系，是一个进程序列。

守护进程:在后台持续运行的进程。

前台启动:一般地，用户键入一个命令，就已经启动了一个前台的进程。

后台启动 :对于非常耗时进程，可以然进程在后台运行。从后台启动进程其实就是在命令结尾加上一个“&”号

每个进程，都会对应一个父进程，而这个父进程可以复制多个子进程

每个进程都可能以两种方式存在：后台或前台

进程就是正在执行的程序

显示系统执行的进程：

ps命令，可以不加任何参数

1. ps -a :显示当前终端的所有进程信息

2：ps -u :以用户的格式显示进程信息

3. ps -x :显示后台进程运行的参数

ps -aux

终止进程：kill/killall

终止某个进程：kill 进程号? 如：kill 6251

kill -q 5222 ：因为某些进程会捕捉某些信息，如果直接不能结束进程，可以使用“? -q”传送信息

动态监控进程：top

top和ps类似，top在执行一段时间可以更新正在进行的进程

1.监视特定用户

top：输入此命令,按回车键，查看执行的进程

u：然后输入“u” 回车，在输入用户名即可

2.终止指定的进程

top：

k: 输入k回车，再输入要结束的进程ID

3.top -d 10 :指定系统更新进程的时间为10秒

按小q退出

2.16：软件安装

1.linux JDK的安装

a.把\*\*.iso 文件挂载在虚拟机上做好配置

mount /mnt/cdrom

unmount /mnt/cdrom

b.把安装文件拷贝到 /home

cp 文件 /home

c.cd /home

d.安装

./\*\*.bin

e.看看文件 /etc/profile[环境配置文件]

f.配置刚才自己安装的JDK

2.eclipse

a.安装文件拷贝到 /home

b.安装

? tar -zxvf \*\*.tar.gz

c.启动eclipse[进入图形界面]

startx ./eclipse

./eclipse & 后台方式运行

#### RPM管理

介绍：一种用于互联网下载包的打包及安装工具，它包含在某些Linux分发版中。它生成 具有.RPM扩展名的文件。

RPM是RedHat Package Manager（RedHat软件包管理工具）的缩写

RPM包的名称格式：apache-1.3.23-11.i386.rpm

\*\*RPM 常用命令

rpm -qa:查询所安装的所有rpm软件包

rpm -qa | more

rpm -pa | grep x

rpm -q 软件包名 （查询软件包是否安装）

rpm -q foo

rpm -qi 软件包名：查询软件包信息

rpm -ql file

rpm -ql 软件包名 ：查询软件包中的文件

rpm -ql file

rpm -qf文件全路径名：查询文件所属的软件包

rpm -qf /etc/passwd

rpm -qp包文件名：查询包的信息、对这个软件包的介绍

rpm -qp jdk-1-5\_linux -i586.rpm

安装rpm包：

rpm -i RPM包路径名称：安装包到当前系统

i = install

rpm -ivh RPM包全路径名称：安装包到当前系统有提示信息

参数说明：i = install 安装

v = verbase 提示

h = hash 进度条

删除rpm包：rpm -e jdk

如果其它软件包依赖于您要卸载的软件包，卸载时则会产生错误的信息

如果忽略这个错误信息继续卸载，使用 --nodeps命令行选项

升级rpm包：

rpm -u RPM包全路径包

作为命令语言互

#### Xshell连接Liunx

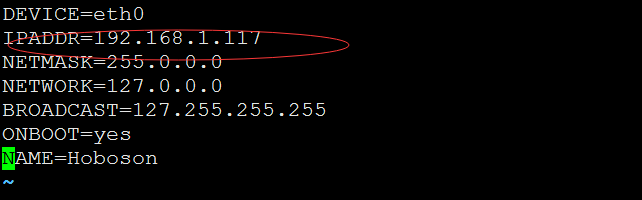
刚刚安装好liunx操作系统 ifconfig时候可能没有配置eth0，无法远程连接，

1设置ip:

cd /etc/sysconfig/network-scripts

cat ifcfg-lo>ifcfg-eht0

vim eth0



*临时设置IP*

*ifconfig eth0 192.168.1.155 netmask 255.255.255.0*

*2 在客户端ping所设置的ip ping 192.168.1.115*

*3 在服务器端telnet 192.168.1.115 22*

*4 关闭防火墙*

*1) 重启后生效*

*开启： chkconfig iptables on*

*关闭： chkconfig iptables off*

*2) 即时生效，重启后失效*

*开启： service iptables start*

*关闭： service iptables stop*

#### 内核

内核是操作系统的核心，具有很多最基本功能，它负责管理系统的进程、内存、设备驱动程序、

文件和网络系统，决定着系统的性能和稳定性。

Linux 内核由如下几部分组成：内存管理、进程管理、设备驱动程序、文件系统和网络管理等

#### 安装

打开vt否则无法安装

安装完毕后 设置ip 关闭防火墙

#### 启动过程

**启动第一步：加载bios(Basic Input Output System)**

当你打开计算机电源，计算机会首先加载BIOS信息，BIOS信息是如此的重要，以至于计算机必须在最开始就找到它。这是因为BIOS中包含了CPU的相关信息、设备启动顺序信息、硬盘信息、内存信息、时钟信息、PnP特性等等。在此之后，计算机心里就有谱了，知道应该去读取哪个硬件设备了

**启动第二部：读取MBR(Master Boot Record)**

众所周知，硬盘上第0磁道第一个扇区被称为MBR，也就是Master Boot Record，即主引导记录，它的大小是512字节，别看地方不大，可里面却存放了预启动信息、分区表信息。

系统找到BIOS所指定的硬盘的MBR后，就会将其复制到0×7c00地址所在的物理内存中。其实被复制到物理内存的内容就是Boot Loader，而具体到你的电脑，那就是lilo或者grub了

**启动第三部：Boot Loader**

Boot Loader 就是在操作系统内核运行之前运行的一段小程序。通过这段小程序，我们可以初始化硬件设备、建立内存空间的映射图，从而将系统的软硬件环境带到一个合适的状态，以便为最终调用操作系统内核做好一切准备。 Boot Loader有若干种，其中Grub、Lilo和spfdisk是常见的Loader。 我们以Grub为例来讲解吧，毕竟用lilo和spfdisk的人并不多。 系统读取内存中的grub配置信息（一般为menu.lst或grub.lst），并依照此配置信息来启动不同的操作系统。

**启动第四步－－加载内核**根据grub设定的内核映像所在路径，系统读取内存映像，并进行解压缩操作。此时，屏幕一般会输出“Uncompressing Linux”的提示。当解压缩内核完成后，屏幕输出“OK, booting the kernel”。 系统将解压后的内核放置在内存之中，并调用start\_kernel()函数来启动一系列的初始化函数并初始化各种设备，完成Linux核心环境的建立。至此，Linux内核已经建立起来了，基于Linux的程序应该可以正常运行了**。 启动第五步－－用户层init依据inittab文件来设定运**行等级 内核被加载后，第一个运行的程序便是/sbin/init，该文件会读取/etc/inittab文件，并依据此文件来进行初始化工作。 其实/etc/inittab文件最主要的作用就是设定Linux的运行等级，其设定形式是“：id:5:initdefault:”，这就表明Linux需要运行在等级5上。Linux的运行等级设定如下： 0：关机 1：单用户模式 2：无网络支持的多用户模式 3：有网络支持的多用户模式 4：保留，未使用 5：有网络支持有X-Window支持的多用户模式 6：重新引导系统，即重启 关于/etc/inittab**文件的学问，其实还有很多 启动第六步－－init**进程执行rc.sysinit 在设定了运行等级后，Linux系统执行的第一个用户层文件就是/etc/rc.d/rc.sysinit脚本程序，它做的工作非常多，包括设定PATH、设定网络配置（/etc/sysconfig/network）、启动swap分区、设定/proc等等。如果你有兴趣，可以到/etc/rc.d中查看一下rc.sysinit**文件，里面的脚本够你看几天**的 启动第七步－－启动内核模块 具体是依据/etc/modules.conf文件或/etc/modules**.d目录下的文件来装载内核模块。 启动**第八步－－执行不同运行级别的脚本程序 根据运行级别的不同，系统会运行rc0.d到rc6.d中的相应的脚本程序，来完**成相应的初始化工作和启动相应的服务。 启动第九步－－**执行/etc/rc.d/rc.local 你如果打开了此文件，里面有一句话，读过之后，你就会对此命令的作用一目了然： # This script will be executed \*after\* all the other init scripts. # You can put your own initialization stuff in here if you don’t # want to do the full Sys V style init stuff. rc.local就是在一切初始化工作后，Linux留给用户进行个性**化的地方。你可以把你想设置和启动的东西放到这里。 启动**第十步－－执行/bin/login程序，进入登录状态 此时，系统已经进入到了等待用户输入username和password的时候了，你已经可以用自己的帐号登入系统了。:) 漫长的启动过程结束了，一切都清静了… 其实在这背后，还有着更加复杂的底层函数调用，等待着你去研究…本文就算抛砖引玉了:) 本文参考了如下文章，精炼荟萃而成： http://bbs.chinaunix.net/thread-835918-1-1.html http://hi.baidu.com/fembed/blog/item/b9f0881f51145866f624e4be.html http://baike.baidu.com/view/9485.htm

*Linux系统有7个运行级别(runlevel)：*

1. *运行级别0：系统停机状态，系统默认运行级别不能设为0，否则不能正常启动*
2. *运行级别1：单用户工作状态，root权限，用于系统维护，禁止远程登陆*
3. *运行级别2：多用户状态(没有NFS)*
4. *运行级别3：完全的多用户状态(有NFS)，登陆后进入控制台命令行模式*
5. *运行级别4：系统未使用，保留*
6. *运行级别5：X11控制台，登陆后进入图形GUI模式*
7. *运行级别6：系统正常关闭并重启，默认运行级别不能设为6，否则不能正常启动*

## Linux Common Commands

### 命令的操作

more more [选项]? <file>…分屏显示命令

less less [选项] <filename> 按页显示命令

clear 清除屏幕命令

history 查看历史命令记录

### 目录相关命令

cd .. 可进入上一层目录

cd - 进入上一个进入的目录

cd ~ ?可进入用户的home目录

pwd? 显示当前在哪个路径

ls 列出文件和目录

ls -a? 显示隐藏文件

ls -l 显示常列表格式

mkdir 建立目录

rmdir? 删除空目录

touch? 建立空文件

### 文件用户权限

2.4.1文件操作

cp 复制命令

将档案 aaa 复制(已存在),并命名为 bbb:

cp aaa bbb

将所有的C语言程式拷贝至 Finished 子目录中:

cp \*.c Finished

cp -r dir1 dir2? 递归复制（复制子目录信息）

mv ?移动文件和改文件 将档案 aaa 更名为 bbb:

mv aaa bbb

将所有的C语言程式移至 Finished 子目录中:　　mv -i \*.c

rm 删除文件和目录

***删除目录时，必须具有对其父目录的写权限。***

rm -rf \* 删除所有内容（包括目录和文件）

如何修改文件的访问权限

chmod 777 along

范例 :将档案 file1.txt 设为所有人皆可读取:

chmod ugo+r file1.txt

将档案 file1.txt 设为所有人皆可读取:

chmod a+r file1.txt

将档案 file1.txt 与 file2.txt 设为该档案拥有者,与其所属同一个群体者可写入,但其他以外的人则不可写入:

chmod ug+w,o-w file1.txt file2.txt

将 ex1.py 设定为只有该档案拥有者可以执行:

chmod u+x ex1.py

将目前目录下的所有档案与子目录皆设为任何人可读取:

chmod -R a+r \*

### 用户组操作

?su-? 切换成系统管理员

su 用户之间的切换

logout 用户注销

useradd? xiaoming? 添加用户（root下）

passwd xiaoming? 设置xiaoming的密码

userdel? xiaoming? 删除用户

userdel -r xiaoming删除用户及其主目录

每个用户必须属于某一个组，不能独立于组外。

每个文件有所有者、所在组、其它组的概念

\*1，所有者：一般指创建者

用ls -ahl 命令可以查看文件的所有者

用chown 用户名 文件名 来修改文件的所有者

\*2，文件所在组 ----用户所在的组

ls -ahl? 可以看见文件的所有组

chgrp 组名 文件名 修改文件所在组

\*3，其它组---除开文件的所有者和所在组的用户外，

? 系统其他用户都是文件的其他组

添加组 ---groupadd policeman

查看Linux中所有组 ： vi /etc/group 查看该文件或cat /etc/group | more

创建用户并指定将该用户分配到哪个组

useradd -g 组名 用户名

查看Linux中所有用户信息：vi /etc/passwd 或者cat /etc/passwd

文件权限：-----分为三种：r-可读，用四表示

?| | | w-可写，用2表示

-|rw-|r--|r--? x-可执行，用1表示

?| | |

上图中由右向左：

第1部分：其它组的用户对该文件的权限

第2部分：文件所在组对该文件的权限

第3部分：文件的所有者对该文件的权限

第4部分：文件类型 ---普通文件(-),目录(d),链接(l)

-d <dirName>：指定用户主目录，默认情况下，将会在/home目录下新建一个与用户名相同的用户主目录

删除用户的命令为userdel，该命令的格式为：userdel <用户名>

修改用户属性

usermod –g<主组名> -G <组名> -d <用户主目录> -s <用户shell>

在添加用户时，可以指定将该用户添加到哪个组中，同样的用root的管理权限可以改变

某个用户所在的组：usermod -g 组名 用户名

可以用 usermod -d 目录名 用户名 改变该用户登录的初始目录?

增加用户组

groupadd <新组名>

删除用户组

? groupdel <组名>

修改组成员：直接编辑/etc/group文件，将用户名写到对应的组名的后面

whoami命令的功能在于显示用户自身的用户名。

who [选项]：该命令主要用于查看当前在线的用户情况

w命令 ：用于显示登录到系统的用户情况

finger命令可用于查找和显示用户信息，并且在查找后显示指定账号的相关信息

chfn命令能够改变系统存储的用户信息

?切换用户身份：su [用户名]

### 帮助命令

man? <command> info <command> help [command]

**alias**

命令：alias显示系统当前定义的所有alias

alias cp ='cp-i'

alias ll='ls -l --color = tty'

### 文件查看连接命令

cat cat [选项] <file1> …

more? ?显示文件内容，带分页

less 显示文件内容，带分页

grep 在文本中查询指定内容

|管道命令[把上一个命令的结果给|后的命令处理]

grep "shunping" aaa.java

grep -n ......? 显示行数?

grep -n "shunping" aaa.java > kkk.bak 重定向命令

ls -l > a.txt? 列表的内容写入文件 a.txt（覆盖写）

ls -l >> a.txt 追加写到文件的末尾

find的使用：

在特定目录下搜索并显示指定名称的文件和目录，搜索一段时间内被存 取/变? 更的文件或目录。

find /home -amin -10 十分钟内存取的文件和目录

find /home -atime -10 十小时

find /home -cmin -10 十分钟内更改过的

find /home -size +10k 大小为10k的

将目前目录及其子目录下所有延伸档名是 c 的档案列出来。

# find . -name "\*.c"

# find. -name \\*.c

将目前目录其其下子目录中所有一般档案列出

# find . -type f

将目前目录及其子目录下所有最近 20 分钟内更新过的档案列出

# find . -ctime -20

### Vi编辑命令

vi有三种基本工作模式，分别是：命令模式（command mode）、插入模式（insert mode）和底行模式（last line mode）

进入vi：

命令“vi 文件名”

命令“vi”，在退出vi时再指定文件名

选项“+n”，表示希望在进入vi之后，光标处于文件中第n行上，

选项“+”表示希望在进入vi之后光标处于文件最末行。

**i o a I OＡ　分别为：插入在光标左边　插入在光标下一行　插入在光标右侧　　插入在该行行头　插入在该行上一行　插入在该行行尾**

保存文件:

1.在命令模式下,连按两次大写字母<Z>。

2.在末行模式下:

?:w vi保存当前编辑的文件，但并不退出vi，而是继续等待用户输入命令。 :w <newfile>

?:w! <newfile>? 把当前文件的内容保存到指定的文件newfile中，如果newfile已经存在，则覆盖原有内容。

在末行模式下，有四种方法可以退出vi返回到shell：

?:q 系统退出vi返回到shell。在用此命令时，若编辑的文件没有被保存，则vi在窗口的最末行给出提示信息。

?:q!? vi放弃所作修改而直接退到shell下。

?:wq 先保存文件，然后再退出vi返回到shell。

?:x? 该命令的功能与命令模式下的ZZ命令功能相同

0（数字0） 移到当前行的行首

$? 移到当前行的行尾

|  |  |  |
| --- | --- | --- |
| h或向左方向键 | 光标向左移动一个字符 | j或者向下方向键 |
| j或者向下方向键 | 光标向下移动一个字符 | K或者向上方向键 |
| K或者向上方向键 | 光标向上移动一个字符 | l或者向右方向键 |
| l或者向右方向键 | 光标向右移动一个字符 | Ctrl + f 或者pageUP键 |
| Ctrl + f 或者pageUP键 | 屏幕向前移动一页 | Ctrl + b 或者pageDOWN键 |
| Ctrl + b 或者pageDOWN键 | 屏幕向后移动一页 | Shift+6或者0（数字0） |
| Shift+6或者0（数字0） | 移动到本行行首 | Shift+4 |
| Shift+4 | 即’$’移动到本行行尾 | G |
| G | 光标移动到文本的最末行 | nG（n是数字） |
| nG（n是数字） | 移动到该文本的第n行 | gg |
| gg | 移动带该文本的首行 | **一般模式下查找与替换** |
| **一般模式下查找与替换** |  |  |

|  |  |  |
| --- | --- | --- |
| /word | 向光标之后寻找一个字符串名为word的字符串，当找到第一个word后，按”n”继续搜后一个 | ?word |
| ?word | 想光标之前寻找一个字符串名为word的字符串，当找到第一个word后，按”n”继续搜前一个 | :n1,n2s/word1/word2/g |
| :n1,n2s/word1/word2/g | 在n1和n2行间查找word1这个字符串并替换为word2，你也可以把”/”换成”#” | :1,$s/word1/word2/g |
| :1,$s/word1/word2/g | 从第一行到最末行，查找word1并替换成word2 | **一般模式下删除、复制粘贴** |
| **一般模式下删除、复制粘贴** |  |  |

|  |  |
| --- | --- |
| x,X | x为向后删除一个字符，X为向前删除一个字符 |
| nx（n为数字） | 向后删除n个字符 |
| dd | 删除光标所在的那一行 |
| ndd（n为数字） | 删除光标所在的向下n行 |
| d1G | 删除光标所在行到第一行的所有数据 |
| dG | 删除光标所在行到末行的所有数据 |
| yy | 复制光标所在的那行 |
| nyy | 复制从光标所在行起向下n行 |
| p,P | p复制的数据从光标下一行粘贴，P则从光标上一行粘贴 |
| y1G | 复制光标所在行到第一行的所有数据 |
| yG | 复制光标所在行到末行的所有数据 |
| J | 讲光标所在行与下一行的数据结合成同一行 |
| u | 还原过去的操作 |

|  |
| --- |
| **进入编辑模式** |

|  |  |
| --- | --- |
| i | 在当前字符前插入字符 |
| I | 在当前行行首插入字符 |
| a | 在当前字符后插入字符 |
| A | 在当前行行末插入字符 |
| o | 在当前行下插入新的一行 |
| O | 在当前行上插入新的一行 |
| r | 替换光标所在的字符，只替换一次 |
| R | 一直替换光标所在的字符，一直到按下ESC |

|  |
| --- |
| **命令模式** |

|  |  |
| --- | --- |
| :w | 将编辑过的文本保存 |
| :w! | 若文本属性为只读时，强制保存 |
| :q | 退出vim |
| :q! | 不管编辑或未编辑都不保存退出 |
| :wq | 保存，退出 |
| :e! | 将文档还原成最原始状态 |
| ZZ | 若文档没有改动，则不储存离开，若文档改动过，则储存后离开，等同于:wq |
| :w [filename] | 编辑后的文档另存为filename |
| :r [filename] | 在当前光标所在行的下面读入filename文档的内容 |
| :set nu | 在每行的行首显示行号 |
| :set nonu | 取消行号 |
| n1,n2 w [filename] | 将n1到n2的内容另存为filename这个文档 |
| :! command | 暂时离开vim运行某个linux命令，例如 :! ls /home 暂时列出/home目录下的文件，然后会提示按回车回到vim |

### More less

more命令功能：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命令功能：less命令的用法与more命令类似，也可以用来浏览超过一页的文件。所不同的是less命

令**除了可以按空格键向下显示文件外，还可以利用上下键来卷动文件。当要结束浏览时，只要在less命**

**令的提示符“: ”下按Q键即可。**

### 打包 压缩 解压

*打包： tar -cvf Hoboson.tar Hoboson*

*压缩： tar -zcvf Hoboson.tar.gz Hoboson*

*压缩： tar -jcvf Hoboson.tar.gz2 Hoboson*

*在选项f之后的文件档名是自己取的，我们习惯上都用 .tar 来作为辨识。*

*如果加z选项，则以.tar.gz或.tgz来代表gzip 压缩过的tar包；如果加j选项，则以.tar.bz2来作为tar包名*

*查看tar.gz有哪些内容*

*Tar -zxvf Hoboson.tar.gz*

*由于我们使用 gzip 压缩的log.tar.gz，所以要查阅log.tar.gz包内的文件时，就得要加上z这个选项了。*

### 防火墙

检查是否安装防火墙：rpm -qa|grep iptables

service iptables status

1）立即关闭，并非永久关闭 service iptables stop 2）永久关闭 iptables -F 关闭防火墙功能 chkconfig iptables off 禁止防火墙启动 另外，运行 setup 在界面，选择Firewall configuration，进入下一界面，选择 Security Level为Disabled，保存

### Ftp或者Sftp链接

*A: 文件传送协议FTP(File Transfer Protocol)是TCP/IP协议簇中的一个成员，也是现在因特网上最广泛的文件传送协议。FTP协议包括两个部分，一个是FTP客户端，另一个是FTP服务器。当然，FTP服务器是用来存储文件资源的，FTP客户端通过访问FTP服务器来获得资源的。*

*B: 安全文件传送协议SFTP(Secure File Transfer Protocol)可以为文件传送提供安全的加密/解密技术。基本语法和FTP差不多。SFTP是SSH的一部分，在SSH软件包中，已经包含了一个SFTP(Secure File Transfer Protocol)的安全文件传输子系统，SFTP本身没有单独的守护进程，它必须使用sshd守护进程（端口号默认是22）来完成相应的连接操作。由于这种传输方式使用了加密/解密技术，文件传送相对来说是很安全的，但是是有代价的，它的传输效率比FTP要低得多*

test

## hell Summary

### Shell

动式地解释和执行用户输入的命令只是shell功能的一个方面

另外shell还可以进行程序设计，他提供了定义变量和参数的手段以及丰富的程序控制结构。使用shell类似于DOS中的 批处理文件称为shell script

shell有很多，常用的有3种：/bin/sh? /bin/csh /bin/ksh

查看shell种类 ：ls -l /bin/\*sh

1.查看目前使用的是哪种shell

env [该命令可以显示当前操作系统的环境变量]

2.shell的修改:

直接输入：chsh -s 输入新的shell 如：/bin/csh

### At

在shell提示符下输入”at 时间”，然后按回车键。这时在下一行shell会等待用户继续输入要执行的命令。每一行输入一 个命令，所有命令都输入完毕后按Ctrl+d键结束。

将各个命令写入shell脚本中，然后使用下面格式设置在指定时间执行shell脚本中的命令：

at 时间 –f脚本文件。

### batch命令

crone命令在系统启动时由一个shell脚本自动启动，进入后台。

cron启动后搜索/var/spool/cron目录，寻找以/etc/passwd文件中的用户名命名的crontab文件，被找到的这种文件将载入内 存。

如果没有crontab文件，就转入“休眠”状态，释放系统资源。

cron每分钟“醒”过来一次，查看当前是否有需要运行的命令。

如果发现某个用户设置了crontab文件，它将以该用户的身份去运行文件中指定的命令。命令执行结束后，任何输出都将 作为邮件发送给crontab的所有者，或者/etc/crontab文件中MAILTO环境变量中指定的用户。

内存查看命令free

磁盘空间用量查看命令df

### 设置系统日期

1.date命令。可以直接输入date 来查看系统时间

2.利用date命令来更改系统时间

date MMDDHHMCCYY.SS:指定月月日日时时分分年年年年.秒秒

3.查看日历：cal 3 2002? ：查看2001年3月的日历

4.查看年历：cal 2008

### 任务调度命令

任务调度指系统在某个时间执行的特定的命令和程序

任务调度分类：

1.系统工作：有些重要的工作必须周而复始的执行，如病毒扫描.....

2.个别用户工作：个别用户可能希望执行某些程序

置任务调度文件：/etc/crontab

设置个人任务调度：执行crontab -e命令

接着输入任务到调度文件

如：5\*\*\*\* ls -l /etc/ > /tmp/to.txt

意思是说每小时的第五分钟执行ls -l /etc/ > /tmp/to.txt

任务调度的使用：

1.设置任务 crontab -e

2.每隔一定时间去执行 data > /home/mydata1

希望每天凌晨2：00执行 date >> /home/mydate

02\*\*\* date >> /home/mydate

调度文件的规则：

字段名称 ?说明? 范围

分钟 每小时第几分钟 0~59

小时 每日的第几个小时 0~23

日期 每月的第几天 1~31

日历 ?每年的第几个月 1~12

星期 每周的第几天 0~6

3.怎么样调度多个任务？

a.在crontab -e 中直接写

b.可以把所有的任务写入一个可执行文件（shell编程）

例子： vi mytask.sh

date >> /home/mydate

cp /home/mydate /root

chmod 744 mytask.sh

crontab -e

\*\*\*\*\* /root/mytask.sh

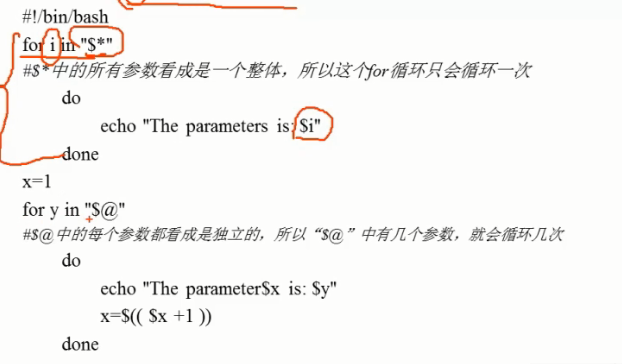
终止任务调度： crontab -r ：终止任务调度

? crontab -l ：列出当前有哪些任务调度

### 多命令执行



### $\*和$@区别



**Test**

## linux系统日常管理

### 命令w

查看当前负载情况

*第一行从左面开始显示的信息依次为：时间，系统运行时间，登录用户数，平均负载*

*第二行开始以及下面所有的行，告诉我们的信息是，当前登录的都有哪些用户，以及他们是从哪里登录的等等*

### 修改密码

Xshell登陆linux后，再次输入passwd 则出现：

Changing password for user root.

New password: root..

Retype new password: root..

passwd: all authentication tokens updated successfully.

密码修改成功

### 破解root密码

redhat忘记root密码的解决方法

redhat默认安装方式下用的是grub引导的，所以在出现启动菜单时（就是选择引导菜单的时候），

就可以按“e”键进入编辑状态，具体操作如下：

1.**在出现grub画面时，用上下键选中你平时启动Linux的那一项，然后按e键**

2. **再次用上下键选中你平时启动linux的那一项(类似于kernel /boot/vmlinuz-2.4.18-14 ro　root=LABEL=/)，**

**然后按e键**

3. **修改你现在见到的命令行，加入空格single，结果如下：kernel /boot/vmlinuz-2.4.18-14 roroot=LABEL=/ single**

**4. 回车返回，然后按b键启动，即可直接进入linux命令行**

**5. 在#提示符下直接输入命令：passwd root 然后写入你的新密码**

**6. #提示符下输入reboot 重启**

## Linux杂谈

### Unix特点

*可靠新强*

*伸缩性好*

*开放性好*

*植根与网络*

*面型数据库使用*

### 运行级别

***Linux系统有7个运行级别(runlevel)****运行级别0：系统停机状态，系统默认运行级别不能设为0，否则不能正常启动 运行级别1：单用户工作状态，root权限，用于系统维护，禁止远程登陆 运行级别2：多用户状态(没有NFS) 运行级别3：完全的多用户状态(有NFS)，登陆后进入控制台命令行模式 运行级别4：系统未使用，保留 运行级别5：X11控制台，登陆后进入图形GUI模式 运行级别6：系统正常关闭并重启，默认运行级别不能设为6，否*则不能正常启动

### 用户账号的添加、删除与修改

增加用户

Useradd hoboson

Passwd hoboson

修改用户名

Passwd 回车修改自己的密码

Passwd hoboson 回车修改hoboson用户的密码

usermod -l **hoboson0316** hoboson

删除用户

userdel hoboson0316

查看用户

cat /etc/passwd

显示内容如下

**〔用户名〕：〔密码〕：〔UID〕：〔GID〕：〔身份描述〕：〔主目录〕：〔登陆Shell〕**

**root:x:0:0:root:/root:/bin/bash**

**bin:x:1:1:bin:/bin:/sbin/nologin**

**daemon:x:2:2:daemon:/sbin:/sbin/nologin**

**adm:x:3:4:adm:/var/adm:/sbin/nologin**

**lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin**

**sync:x:5:0:sync:/sbin:/bin/sync**

**shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown**

**halt:x:7:0:halt:/sbin:/sbin/halt**

**mail:x:8:12:mail:/var/spool/mail:/sbin/nologin**

**news:x:9:13:news:/etc/news:**

**uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin**

**operator:x:11:0:operator:/root:/sbin/nologin**