目 录

[第一讲 1](#_Toc439837125)

[1.1 概述 1](#_Toc439837126)

[1.2 安装 2](#_Toc439837127)

[1.3 根分区下的目录 3](#_Toc439837128)

[1.4 常用命令（一） 4](#_Toc439837129)

[第二讲 5](#_Toc439837130)

[2.1 常用命令（二） 5](#_Toc439837131)

[2.2 操作 6](#_Toc439837132)

[2.3 用户、组（一） 7](#_Toc439837133)

[第三讲 9](#_Toc439837134)

[3.1 用户、组（二） 9](#_Toc439837135)

[3.2 权限 9](#_Toc439837136)

[第四讲 12](#_Toc439837137)

[4.1 链接文件 12](#_Toc439837138)

[4.2 重定向 12](#_Toc439837139)

[4.3 网络 12](#_Toc439837140)

[4.4 安装包 13](#_Toc439837141)

[第五讲 15](#_Toc439837142)

[5.1 软件安装 15](#_Toc439837143)

[第六讲 18](#_Toc439837144)

[6.1 软件安装 18](#_Toc439837145)

[6.2 进程 20](#_Toc439837146)

[第七讲 22](#_Toc439837147)

[7.1 Shell概述 22](#_Toc439837148)

[7.2 Shell变量的应用 24](#_Toc439837149)

[7.3 Shell脚本 26](#_Toc439837150)

[7.4 正则表达式 27](#_Toc439837151)

[7.5 test命令 28](#_Toc439837152)

[第八讲 30](#_Toc439837153)

[8.1 条件判定 30](#_Toc439837154)

[8.2 循环 31](#_Toc439837155)

[8.3 case分支语句 34](#_Toc439837156)

[8.4 shell函数 35](#_Toc439837157)

# 第一讲

## 1.1 概述

Linux是一款全球性的免费的开源的操作系统平台，其特点是实现了多任务多用户处理，主要是依赖内核kernel shell，且占用资源少（最小配置只要4Mb内存就能运行）。

1. 由来

类UNIX的，与UNIX都是服务器系统。

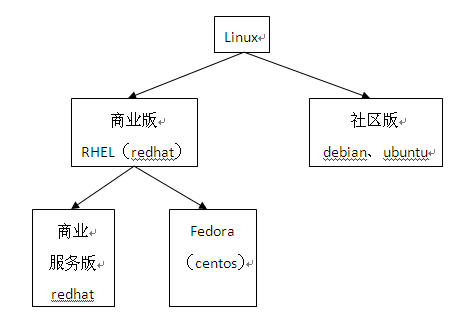
minix🡪Linux

1. 与UNIX的区别

Linux是免费开源的，UNIX是商业应用性软件操作系统，UNIX是与硬件进行捆绑销售的，UNIX是不开源的（部分开源，但是核心代码是不开源的）

1. 版本

主要分为两大类：商业版、社区版



1. red hat：红帽（全球唯一一家认证Linux能力的公司，占据了Linux市场的绝大部分）
2. Fedora：免费开源社区版本
3. centos：community enterprise （稳定性要优于fedora）
4. redhat enterprise：
5. debian：容错性极强的操作系统，但是系统分区过于复杂
6. ubuntu：由乌班图转义而来：仁和；南非（祖鲁或者豪萨）：人性

口号：我的存在是因为你们的存在。

倾向于桌面操作系统（图形界面），一般不用于服务器

1. suse：德国，很好地实现与各种数据库兼容，能够实现自动分区

## 1.2 安装

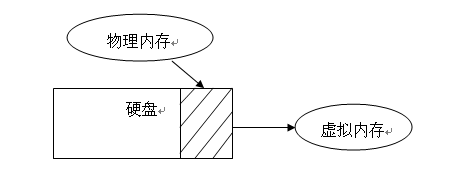
分为图形化界面版本和Linux 文本版本

1. Linux核心：
2. Linux中没有盘符的概念，所有的一切都称为文件（目录、文件夹、文件、操作系统、应用程序等）；
3. 只要能装到Linux系统上的软件，则都称为有用途软件
4. 分区（树状结构）
5. 根分区：使用“/”表示，必须要有

挂载：所有目录都是挂载在分区上面的

1. 引导分区：也叫做boot分区，用“/boot”表示，相当于windows中的BIOS引导盘。没有的话系统无法启动。必须存在，容量在50-100M，用于存储系统主核心shell程序
2. swap分区：又称为交换分区，当物理内存中的链接过多内存不够的时候，将多余的链接放到swap分区中，等物理内存中压力缓解之后再将swap中的链接放到物理内存中进行处理。用户无法进行访问，大小通常为物理内存的两倍
3. 文件类型：ext3、swap
4. 补充

* 安全模式：只加载系统启动最基本的设置
* 虚拟内存：指在物理内存不够用的时候使用设置的虚拟内存暂时存储后续的数据以缓解物理内存压力，不至于系统内存崩溃，范围在物理内存的1-1.5倍



1. 解析[root@localhost ~]#：root表示当前用户；localhost表示本机；~表示当前用户目录；#表示管理员登陆（$表示普通用户登陆）
2. 从text界面进入图形化界面：startx
3. 从图形化界面回到text界面：system🡪注销（logout）

## 1.3 根分区下的目录

1. bin：基本命令目录（绿色表示可执行）
2. dev：设备文件目录

* tty：终端串行设备

1. home：主目录，创建用户的时候会在该目录下自动创建一个同名的目录
2. lost+found：当出现系统数据丢失（比如断电）时会将数据保存在该目录下
3. net：网络文件目录
4. misc：空目录
5. proc：进程或内存使用信息（任务管理的相关信息）
6. sbin：高级命令目录
7. srv：空目录
8. tftpboot：ftp信息存放在该目录中（ftp根目录）
9. usr：软件安装时默认安装目录
10. boot：内核程序、引导分区程序
11. etc：配置文件目录（shift+pageup/pagedown 页面上翻/下翻）
12. lib：类库
13. media：媒体目录
14. mnt：光驱挂载点目录（将光驱中的资源挂载到该目录中）
15. opt：操作的文件信息
16. root：root用户/管理员目录
17. seLinux：Linux shell主配置文件目录
18. sys：系统目录
19. tmp：临时目录，用于存放垃圾文件
20. var：缓存、临时数据、临时变量存放在该目录中

## 1.4 常用命令（一）

1. 显示当前工作目录：pwd
2. 显示当前目录中的文件：ls 或dir
3. 切换目录：cd

* 切换到root目录：cd ~ 或cd /root
* 切换到根目录：cd /
* 切换到上级目录：cd ..
* 切换到当前目录：cd .

1. 创建文件夹：mkdir
2. 删除文件夹：rmdir（不能直接删除不为空的目录）
3. 创建文件：vi 或 vim 文件名.后缀

步骤：vi 文件名.后缀🡪按“i”（进入插入模式）🡪按“esc”（去除插入模式）🡪:wq（保存并退出）、:q（不保存退出）、:q！（强制退出）

1. 关机：shutdown/poweroff/halt
2. 重启：reboot
3. 注销：logout、exit

# 第二讲

## 2.1 常用命令（二）

1. 创建一个空白文件：touch 文件名.后缀（需要编辑的话使用vi 文件名.后缀）
2. 浏览文件：cat 文件名.后缀
3. 删除文件（包括文件、文件夹）：rm 文件名.后缀

* -f：不提醒直接删除
* -r：递归，从最里面一层一层往外面删除

rm –rf 文件夹名：直接删除文件夹且不提示

1. 查看命令帮助：help 命令 或 命令 --help 或 man 命令 或 info 命令

例子：halt help man halt info halt

当前目录下存在匹配文件的话按一下“tab”会自动补齐输入字符，按两下会将匹配的都显示出来

1. 复制文件：cp 原文件地址 目标文件地址

原文件地址可以是相对路径，也可以是绝对路径

相对路径：从当前目录开始（只能在当前目录及子目录中切换）

绝对路径：从根目录开始（可以切换到其他目录）

例子：假设当前目录为home目录，且在home目录中有文件夹zhangsan，zhangsan中有文件夹message，使用相对路径和绝对路径进入message

相对路径：cd zhangsan/message/

绝对路径：cd /home/zhangsan/message/

1. 剪切移动、文件重命名：mv 原文件地址 目标文件地址

剪切文件夹（非空文件夹）：mv –t 目标文件地址 原文件地址

例子：

1. 剪切移动：mv a.java /home/aa
2. 重命名：mv /home/a.java b.java
3. 将home目录下的aa夹文件移动到根目录下：mv -t / /home/aa
4. 查看当前用户：who am i 或者who 或者id
5. 改密码：passwd 新密码
6. 查找文件/文件夹：find -name 路径/文件名.后缀（返回文件所在路径）
7. 查找所配置环境变量的路径：which
8. 查找对应命令的路径：whereis 命令
9. 通过名字查找文件：locate 文件名.后缀 （实现的是模糊匹配，查找本地所有的类似名字文件）
10. 查找文件里面的内容：grep “查找的关键字” 文件名.后缀

查找文件内容对应行数：grep –n “查找的关键字” 文件名.后缀

1. 清屏：clear
2. 统计文件行数、字数、字节数：wc 参数 文件名.后缀

参数包括：-c 字节数 -m 字符数 -l 行数

1. 查询文件前几行：head –n（n表示行数） 文件名.后缀
2. 查询文件后几行：tail –n（n表示行数） 文件名.后缀
3. 剪切每行第几个信息（包括空格）：cut 参数 数字 文件名.后缀
4. 文件东西很多时可以使用more从前往后分页，less可以实现上下翻滚查看

例如：cut 文件名.后缀 | more

管道命令：将前面命令产生的结果交给后面的命令

1. 日历（当前月份）：cal

例如：cal 2015：显示2015年日历

1. 计算器：bc quit退出
2. 查看系统当前时间：date

设置系统时间：date -s “月/日/年 时:分:秒”

## 2.2 操作

1. 忘记密码：开机🡪图形界面按“e”（只有五秒时间）🡪再按“e”🡪选择第二个，再按“e”🡪“quiet”改为“1”，按“enter”🡪按“b”

“e”表示进入编辑

“1”表示单用户模式，相当于windows 的安全模式

1. 改运行级别：配置文件（etc）🡪vi inittab🡪id

运行级别，一共7个级别（0-6）

* 0：关机
* 1：单用户模式登陆（安全模式，可改用户密码）
* 2：多用户不带网络模式（单机模式）
* 3：多用户带网络模式（文本模式）
* 4：保留，保留给其他开发人员进行运行
* 5：图形化运行模式
* 6：重启（无限重启），用于查看系统与硬件资源是否匹配（兼容）

1. 从文本界面切换到图形界面，并切换用户：init 5（init表示运行级别的切换）

注意：以不同用户从文本界面切换到图形界面时不能用log out（注销）重新切回到文本界面，此时要切换到文本界面则要用init 3（右键单击桌面🡪 open terminal（打开终端）🡪输入init 3）

## 2.3 用户、组（一）

1. 创建用户：useradd 用户名（useradd -help）

* -u：创建用户时指定用户id，没有指定的话会自动创建用户id
* -g：创建用户时指定用户的组id
* -G：创建用户时将用户添加到指定组，同时也创建用户自己的组

在创建用户的时候，会自动创建该用户的组（不但会自动创建用户id，还会自动创建目录，也会自动生成用户组id）。如果没有指定用户id和组id，会在最后一个组id基础上加1创建用户id和组id。

1. 设置密码（激活用户）：passwd 用户名
2. 切换用户（普通用户用$表示）：su 用户名

注意：管理员切换普通用户不需要密码，但是普通用户切换管理员需要密码

1. 查看用户信息：/etc🡪vi passwd

解析：xiaobai:x:503:503::/ home/xiaobai/:/bin/bash

* xiaobai：用户名
* x：表示加密
* 503:503：前一个是用户id，后一个是组id
* / home/xiaobai：用户所在目录
* /bin/bash环境所在路径

1. 查看组信息：/etc🡪vi group
2. 创建组：groupadd 组名

* -g：为新建的组指定组id

1. 删除组：groupdel 组名

先将组内用户全部删除之后才能删除组

1. 删除用户：userdel 用户名

* -r：删除用户的home目录
* -f：强制删除

任务：usermod、groupmod、gpasswd的man参数解释

# 第三讲

## 3.1 用户、组（二）

1. 更改用户账户：usermod

* -l：对用户进行重命名
* -L：锁账户密码（不能用密码登陆，但是可以通过root用户切换）
* -U：解锁

1. 更改组账户：groupmod

* -n：对组进行重命名

1. gpasswd

* -a：将用户添加到组
* -d：将用户从组中移除

1. 查看用户密码状态：cat /etc/shadow



* ！！表示没有密码
* ！表示账户密码被锁
* 乱序字符串表示有密码

## 3.2 权限

1. 详细显示所有目录信息：ls -l

文件权限表示：10位字符（1333分组）

文件类型 u g o

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| d | r | w | x | - | w | x | r | - | x |

1. 第一位表示文件类型

* d：表示文件夹
* -：表示普通文件
* l：表示链接文件
* c：表示字符设备文件（character）
* b：表示块设备文件（block）
* p：表示管道文件

1. u（二-四位）表示所属主权限（拥有者）

* r：表示可读，用4表示
* w：表示可写，用2表示
* x：表示可执行，用1表示
* -：表示没有任何权限

1. g（五-七位）表示所属组权限
2. o（八-十位）表示其他组权限

解析：-rw-r---w- 1 xiaohong root 7 Dec 24 17:06 a

-表示普通文件；rw-r---w-表示主用户具有可读可写的权限、组用户具有可读权限、其他组用户具有可写权限；1表示文件及其子目录下文件数量为1；xiaohong表示主用户为xiaohong；root表示组为root组；7表示内容字符数；Dec 24 17:06表示时间；a是文件名。

1. 改文件权限
2. 授权：chmod 参数+权限 文件

参数：u：拥有者 g：同组 o：其他组 a：所有

1. 撤权：chmod 参数-权限 文件

Linux中默认，权限是不继承的，但是可以用参数R实现递归继承

例如：chmod -R 权限 文件名

1. 权限设置：chmod [u][g][o] 对象

u、g、o都可用0-7进行表示，含义如下：

* 7：表示拥有者拥有所有权限
* 6：表示拥有可读可写权限
* 5：表示拥有可读可执行权限
* 4：表示拥有可读权限
* 3：表示拥有可写可执行权限
* 2：表示拥有可写权限
* 1：表示拥有可执行权限
* 0：没有任何权限

1. 颜色显示配置文件： /etc/DIR\_COLORS\_

* 绿色代表可执行文件
* 深蓝色代表文件夹
* 湛蓝色表示链接文件
* 黄色表示设备文件（加粗黄色）、管道文件
* 紫色表示socket套接字文件、图片文件
* 白色表示普通文件
* 红底白字表示无效链接文件、丢失文件
* 亮红表示归档文件或压缩文件

1. 改文件属性
2. 改变文件拥有者（拥有者是唯一的）：chown 新拥有者 文件
3. 更改文件所属组：chgrp 新组 文件

# 第四讲

## 4.1 链接文件

一种快捷方式，分为硬链接和软链接

1. 建立链接：ln
2. 硬链接：ln 目标 存放位置

例子：新建硬链接a指向a.txt文件：ln /home/a.txt /var/a

1. 软链接：ln -s 目标 存放位置

例子：新建软链接b指向a.txt文件：ln -s /home/a.txt /var/b

1. 硬链接、软链接区别
2. 硬链接相当于普通文件；软链接相当于链接文件，颜色为湛蓝色
3. 原文件内容发生变化的时候软、硬链接都会发生变化，反之亦然
4. 硬链接在创建的时候相当于复制文件，软链接才是真正的快捷方式（原文件删除时软链接为无效链接，硬链接还可以用）

## 4.2 重定向

1. 写入流（覆盖目标文件内容）：>

例子：将a.txt中的内容写入b.java中并覆盖b.java原有内容：cat a.txt > b.java

1. 追加：>>

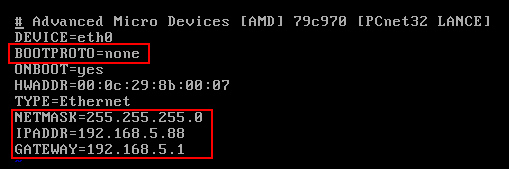
例子：将a.txt中的内容追加到b.java中：cat a.txt >> b.java

## 4.3 网络

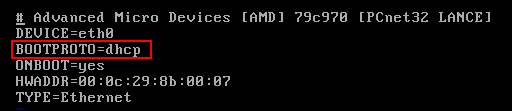
1. 查IP地址：ifconfig
2. ping 通：ping IP地址
3. 系统配置：setup
4. 改IP

* 方法一：ifconfig 网卡设备名 IP地址（设定为临时IP，重启之后将失效）
* 方法二：setup🡪Network Configuration（重启网卡服务：service network restart或./network start或重启生效）
* 方法三：修改配置文件：vi /etc/sysconfig/network-scripts/ifcfg-eth0

静态IP：



动态IP：



## 服务搭建

## 4.4 安装包

1. 类型
2. rpm包（redhat package manager）：红帽包管理软件

* 查询所有：-qa
* 安装（显示安装进度条）：-ivh
* 卸载：-e

1. bin包

* 安装：./包名（执行bin文件）

1. tar.gz包（源码包）

步骤：文件解压缩🡪修改与之对应的配置🡪编译🡪执行安装

tar包解压缩：tar -zxvf

1. deb包：ubuntu推出的，可以使用工具与rpm相互转换

dpkg，debian包管理器

启动目录：/etc/rc.d/init.d

查所有进程的端口：netstat

1. 挂载点
2. 加载挂载点：mount /dev/cdrom /mnt
3. 卸载挂载点：umount /mnt
4. 安装程序
5. 安装JDK

步骤：加载挂载点并将文件复制到一个可读写的目录中🡪./文件名🡪配置环境变量vi /etc/profile（ echo $PATH：查看PATH）🡪 rm -rf /usr/bin/java

配置环境变量方式：

* 分批导：

export JAVA\_HOME=/home/jdk1.6/bin

export CLASS\_PATH=/home/jdk1.6/bin

export PATH =./$PATH:/home/jdk1.6/bin

* 一次性导：

JAVA\_HOME=/home/jdk1.6/bin

CLASS\_PATH=/home/jdk1.6/bin

PATH =./$PATH:/home/jdk1.6/bin

在PATH语句最后添加JAVA\_HOME CLASS\_PATH

1. 安装ecplise

步骤：加载挂载点并将文件复制到一个可读写的目录中🡪unzip 文件名🡪tar -zxvf文件名.tar.gz🡪./文件名🡪vi eclipse.sh（编写启动器）🡪设置文件权限为可执行🡪运行：./ecplise.sh

启动器内容：

#!/bin/bash

export java\_home=/home/jdk1.6

export class\_path=/home/jdk1.6/bin

/home/eclipse/eclipse -vm /home/jdk1.6/bin/java -data /home/workspace&

# 第五讲

## 软件安装

1. 安装Apache

Apache（apache http server）阿帕奇，用于进行web网站发布的容器（使用率世界第一），跨平台的、安全性极高的纯web性的软件，只支持静态网页，常与mysql或tomcat匹配使用

html 超文本标记语言，相对xml语言更简单

Tomcat 可编辑动态网页

1. 安装包：rmp -ivh --nodeps 文件
2. 查询：rpm -qa | grep httpd
3. 启动服务：service httpd start（restart重启）
4. 修改配置文件：/etc/httpd/conf/httpd.conf

* 服务根目录：/etc/httpd
* 监听端口：80
* serverAdmin：root@localhost
* serverName：不能注释，要用对应的IP或用localhost
* DocumentRoot：“/var/www/html”

1. 关闭防火墙：setup

Apache相关包：httpd-2.2.3-43.el5.i386.rpm、apr-1.2.7-11.el5\_3.1.i386.rpm、apr-util-1.2.7-11.el5.i386.rpm、postfix-2.3.3-2.1.el5\_2.i386.rpm

安装apache步骤：安装包（httpd、apr、apr-util、postfix）🡪修改配置文件🡪启动httpd服务🡪添加html文件🡪重启httpd服务🡪关闭防火墙🡪外部访问：http://IP地址/文件名.html

1. 安装ftp

ftp（file transfer protocol），文件传输协议

1. 修改配置文件：/etc/vsftpd/vsftpd.conf

* 匿名访问：NO

1. 查看服务：/etc/rc.d/init.d
2. 启动服务：service vsftpd start（restart重启）
3. pub目录：/var/ftp/pub

ftp相关包：vsftpd-2.0.5-16.el5\_4.1.i386.rpm

安装ftp步骤：安装包（vsftpd）🡪修改配置文件🡪开启ftp服务🡪关闭内核防火墙🡪重启ftp服务🡪外部访问：ftp://IP地址

1. 安装samba服务器

samba服务器是一个文件共享服务器，实现了windows和linux文件跨平台共享

1. 启动samba服务：service smb start
2. 修改配置文件：/etc/samba/smb.conf

* 全局设置：global
* 日志信息
* Standalone Server Options

security=share

* Share Definitions

[public]

comment = public

path= /home/public

writeable = yes

public = yes

samba相关包：samba-3.0.33-3.28.el5.i386.rpm

步骤：安装包（samba主包）🡪修改配置文件🡪启动服务🡪添加信任站点（访问地址）🡪外部访问：\\IP地址\文件目录

1. 修改配置文件

* Standalone Server Options

security=user

1. 重启samba服务
2. cat /etc/passwd > smbpasswd
3. 添加用户：smbpasswd -a 用户名
4. 设置用户密码：smbpasswd 用户名
5. 删除会话：net session \\计算机名 /delete

或 net use \* /delete

步骤：改配置文件🡪 cat /etc/passwd > smbpasswd🡪添加用户并设置密码🡪重启samba服务🡪外部访问：\\IP地址\文件目录

1. 安装tomcat

Tomcat是用于实现java应用发布服务的中间件，是一款容器，进行的是动态页面的发布，开源的、免费的

1. 查进程：ps -ef | grep tomcat
2. 修改配置文件：/ect/skel/.bash\_profile

export JAVA\_HOME=/home/jdk1.6

export CLASS\_PATH=$CLASS\_PATH:$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib

export PATH=$PATH:$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin:$HOMR/bin

export CATLINA\_BASE=/usr/local/tomcat

export CATLINA\_HOME=/usr/local/tomcat

1. 应用配置文件：source .bash\_profile
2. 启动服务：./startup.sh

安装步骤：安装包🡪重命名为tomcat🡪修改配置文件🡪source /ect/skel/.bash\_profile🡪在tomcat目录的bin目录下启动服务：./startup.sh🡪外部访问：http://IP地址：8080

# 第六讲

## 6.1 软件安装

1. 安装mysql
2. 安装包：rpm -ivh --nodeps 包
3. 查看状态：service mysqld status
4. 启动服务：service mysqld start
5. 服务启动文件：/etc/my.cnf
6. 进入mysql：mysql
7. 退出mysql：quit
8. 登陆有密码的用户：mysql -u 用户名 -p
9. 登陆用户：mysql -u 用户名
10. 添加密码：mysqladmin password 新密码
11. 显示所有数据库：show databases；
12. 切换数据库：use 数据库名；
13. 显示表：show tables；
14. 改root的密码：update mysql.user set password=password(“新密码”) where user=’root’;
15. 更新表：flush privileges;

相关包：mysql-5.0.77-4.el5\_4.2.i386、mysql-devel-5.0.77-4.el5\_4.2.i386、mysql-server-5.0.77-4.el5\_4.2.i386

步骤：安装包（devel、server、主包）🡪启动服务🡪验证是否成功（mysql、mysqladmin --help）

1. 安装dhcp

dynamic host configuration protocol，动态主机配置协议

1. 启动dhcp服务：service dhcpd start（重启：restart）
2. 配置文件：/etc/dhcpd.conf
3. 样本文件：/usr/share/doc/dhcp-3.0.5/dhcpd.conf.sample
4. 重启网络服务：service network restart
5. 客户端验证

* 释放IP地址：ipconfig /release
* 获取IP地址：ipconfig /renew

在linux中，IP地址是从大到小进行分配的

相关包：dhcp-3.0.5-23.el5.i386.rpm、dhcpv6-1.0.10-18.el5.i386.rpm、dhcp-devel-3.0.5-23.el5.i386.rpm

1. 安装步骤：安装包（3.0、6-1、devel）🡪修改配置文件🡪改主机IP为静态IP🡪重启网络服务🡪启动DHCP服务🡪客户端验证
2. 安装DNS

域名解析系统

1. 启动服务：service named start
2. 配置文件：/usr/share/doc/bind-9.3.6/sample/var/named/

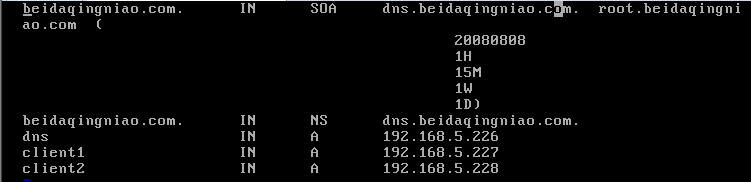
./my.external.zone.db



./my.internal.zone.db



1. 在/var/named/新建beidaqingniao.com（正向区域文件）



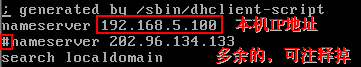
1. 在/var/named/新建5.168.192（方向区域文件）



1. 在etc下新建：vi named.conf（主文件）



1. 改本机地址：/etc/resolv.conf



1. 检测：host或nslookup

相关包：bind-9.3.6-4.P1.el5\_4.2.i386、bind-devel-9.3.6-4.P1.el5\_4.2.i386

步骤：安装包🡪将5.168.192和beidaqingniao.com 拷到/var/named/目录下，将named.conf拷到/ect/目录下🡪修改/etc/resolv.conf

## 6.2 进程

1. 查看所有进程：ps -ef

PDI表示进程号，1号为所有进程的根，其他进程依赖于它

1. 查询samba进程：ps -ef | grep smb
2. 杀进程：kill 进程号
3. 强制杀进程：kill -9 进程号

杀掉-bash进程之后会退出系统

1. 查看动态进程：top

僵尸进程：zombie，子进程被关闭时，子进程会通知父进程进行关闭，如果父进程由于某些原因没有关闭，则此时父进程就成为了僵尸进程，僵尸进程越多则软件模块间的独立性较差，耦合性较强，僵尸进程有害无益（浪费资源浪费空间）

1. 快捷键
2. u：显示用户
3. q：退出
4. h：表示帮助
5. Z：表示换颜色
6. B：表示加粗
7. o：表示改变显示列表
8. iptables表示防火墙
9. 查看防火墙状态：service iptables status
10. 开启：service iptables start
11. 关闭：service iptables stop

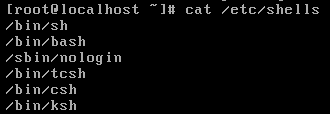
# 第七讲

## 7.1 Shell概述

Shell是一个特殊的应用程序，是一种环境

1. 作用：命令解释器，介于操作系统内核与用户之间
2. 分类
3. Bsh：70年代中期产生
4. Csh：C语言
5. Ksh：兼容Bsh和Csh
6. Zsh：兼具各种Shell优点
7. bash：Bsh的升级版

不同的Shell命令操作是有区别的，Linux默认的是bash。



1. Shell环境的切换
2. 修改/etc/passwd中对应用户的登录Shell
3. usermod -s Shell路径 用户名（root用户可操作）
4. 直接输入其他Shell类型（或完整的路径）
5. chsh -s Shell类型的路径
6. 查找Shell（/bin目录下执行）
7. which \*sh
8. whereis \*sh
9. find \*sh
10. 命令历史
11. 存放位置：~/.bash\_history
12. 查看历史命令：history
13. 清除历史命令：history -c
14. 调用历史命令

* ！n（执行第n条命令）
* ！str（执行以“str”开头的命令，“str”可改为执行过的任意命令开始几个字符）

1. 修改命令记录条数：histsize 参数（默认1000条）（/etc/profile中）
2. 命令别名

为使用频率较高的复杂命令行设置简短的调用名称

1. 存放位置：~/.bashrc
2. 查看命令别名：alias 别名
3. 设置命令别名：alias 别名=‘实际执行的命令’
4. 取消已设置的命令别名：unalias 别名
5. 交互式设备文件
6. 标准输入：从该设备接收用户输入的数据，默认设备为键盘，设备文件/dev/stdin，文件编号为0
7. 标准输出：通过该设备向用户输出数据，默认设备为显示器，设备文件/dev/stdout，文件编号为1
8. 标准错误：通过该设备报告执行出错的信息，默认设备为显示器，设备文件/dev/stderr，文件编号为2
9. 重定向
10. 重定向标准输入：<，将命令中接收输入的途径由默认的键盘更改为指定的文件
11. 重定向标准输出（覆盖）：>，将命令的执行结果输出到指定的文件中，而不是直接显示在屏幕上
12. 重定向标准输出（追加）：>>，将命令执行的结果追加输出到指定文件
13. 重定向标准错误：2>，清空指定文件的内容，并将标准错误信息保存到该文件中
14. 重定向标准错误：2>>，将标准错误信息追加输出到指定的文件中
15. 重定向标准输出和标准错误：&>，将标准输出、标准错误的内容全部保存到指定的文件中，而不是直接显示在屏幕上

## 7.2 Shell变量的应用

1. 变量定义

变量名=变量值

1. 变量名：使用固定的名称，由系统预设或用户定义
2. 变量值：能够根据用户设置、系统环境变化而变化
3. 变量种类
4. 用户自定义变量：又称为本地变量，由用户自己定义、修改和使用，默认只能在当前的Shell环境中使用，也称为局部变量
5. 环境变量：由系统维护，用于设置用户的Shell工作环境，只有极少数的变量用户可以修改
6. 预定义变量：Bash预定义的特殊变量，不能直接修改
7. 位置变量：通过命令行给程序传递执行参数
8. 定义新的变量

变量名要以英文字母或下划线开头，区分大小写

1. 格式：变量名=变量值
2. 查看变量的值：echo $变量名
3. 从键盘输入内容为变量的值：read -p “信息” 变量名

read命令-p表示从控制台获取一个数

read 变量名1 变量名2：使用“变量值1 变量值2”进行赋值（使用空格隔开）

1. 结合三种符号给变量赋值

* 双引号“”：允许通过$符号引用其他变量值
* 单引号‘’：禁止引用其他变量值，$视为普通字符，为引用变量名
* 反撇号``：表示对命令的引用，将命令执行的结果输出给变量

对于字符串，单引号、双引号输出的内容是一样的；对于变量，双引号输出变量的值，单引号输出变量名

例子：查看终端登录用户数

定义变量并赋值：num=`who | wc -l`

引用：echo $num

1. 设置变量作用范围：export 变量名（设置输出的是全局变量）
2. 清除自定义变量：unset 变量名
3. 表达式计算

* expr 变量1 操作符 变量2（中间要用空格隔开）
* 变量名=`expr变量1 操作符 变量2`（乘号需用转义符（\）之后才能使用）
* 变量名=$[变量1 操作符 变量2]（乘号不需要转义符可直接使用）

1. 常用运算符：+、-、\\*、/、%
2. 查看所有变量：set
3. 常见的环境变量（用冒号隔开）

* 当前用户的登录名称：$USER、$LOGNAME
* 当前用户的UID号：$UID
* 用户使用登陆的Shell：$SHELL
* 用户的宿主目录：$HOME
* 用户的当前工作目录：$PWD
* 当前用户的命令搜索路径：$PATH
* $CLASS\_PATH、$JAVA\_HOME…

1. 位置变量：表示为$n（n为1~9），表示第n个位置参数

例子：./exam01 one two three

$1，第1个位置参数one

1. 预定义变量
2. $#：命令行中位置参数的个数
3. $\*：所有位置参数
4. $?：上一条命令执行后返回的状态（0表示成功，非0表示出现异常）
5. $$：当前所在进程的进程号
6. $!：后台运行的最后一个进程号
7. $0：当前执行的进程/程序名

需用echo进行输出

1. 查进程号：pgrep 服务名

## 7.3 Shell脚本

1. 用途：完成特定的、较复杂的系统管理任务
2. 格式：集中保存多条linux命令，普通文本文件
3. 执行方式：按照预设的顺序依次解释执行
4. 创建脚本（文件名.sh）
5. 内容

* 运行环境设置：#！/bin/bash（首句），用于指定使用哪一种Shell程序进行解释
* 注释信息：以#开始的说明性文字
* 可执行的Linux命令

1. 为脚本文件添加可执行权限
2. 执行

* 利用环境直接执行具有“x” 权限的脚本文件：./文件名.sh
* 使用指定解释器：Shell类型 文件名.sh
* source命令（或.）读取脚本内容执行：source 文件名.sh 或 .文件名.sh

例子：

vi /opt/dbbak.sh

DAY=`date+%Y%m%d`

SIZE=`du -sh /var/lib/mysql`

echo “Date:$DAY” >> /tmp/dbinfo.txt

echo “Date Size:$Size”>> /tmp/dbinfo.txt

cd /opt/dbbak

tar zcvf mysqlbak-${DAY}.tar.gz /var/lib/mysql /tmp/dbinfo.txt

rm -rf /tmp/dbinfo.txt

调用：crontab -e

统计：du

## 7.4 正则表达式

正则表达式是用在字符串处理上的一项“表达式”，它不是一个工具，而是一个字符串处理的标准，支持的工具有vi、grep、sed、awk。

1. ^：行首定位符
2. $：行尾定位符
3. \<：词首定位符
4. \>：词尾定位符
5. \*：匹配0个或多个前一个字符
6. .：匹配任意一个字符
7. []：匹配一组字符里的任意字符
8. [x-y]：匹配指定范围的字符
9. [^]：匹配不在指定范围的字符
10. \：转义字符，用来屏蔽一个元字符的特殊含义
11. 根据字符x出现的次数匹配

x\{m\}：m次

x\{m，\}：大于等于m次

x\{m，n\}：大于等于m次且小于等于n次

## 7.5 test命令

测试特定的表达式是否成立，当条件成历史，命令执行后的返回值为0，否则为其他数值

1. 格式：test 条件表达式

[ 条件表达式 ]

1. 常见的测试类型
2. 测试文件状态
3. 字符串的比较
4. 整数值比较
5. 逻辑测试
6. 测试文件状态
7. 格式：[ 操作符 文件或目录 ]
8. 常用的测试操作符

* -d：测试是否为目录（Directory）
* -e：测试目录或文件是否存在（Exist）
* -f：测试是否为文件（File）
* -r：测试当前用户是否有权限读取（Read）
* -w：测试当前用户是否有权限写入（Write）
* -x：测试当前用户是否可执行（Excute）该文件
* -L：测试是否为符号连接（Link）文件

例子：[ -d /etc/vsftpd ] 必须用空格隔开

1. &&表示连接输出

例子：[ -f a.java ] && echo “yes”：执行成功则返回yes

1. 整数值比较
2. 格式：[整数1 操作符 整数2]
3. 常用的操作符

* -eq：等于（Equal）
* -ne：不等于（Not Equal）
* -gt：大于（Greater Than）
* -lt：小于（Lesser Than）
* -le：小于或等于（Lesser or Equal）
* -ge：大于或等于（Greater or Equal）

1. 字符串的比较
2. 格式

* [ 字符串1 = 字符串2 ]
* [ 字符串1 ！= 字符串2 ]
* 是否为空：[ -z 字符串]

1. 常用的操作符

* =：字符串内容相同
* !=：字符串内容不同，! 号表示相反的意思
* -z：字符串内容为空

1. 逻辑测试
2. 格式：[ 表达式1 ] 操作符 [ 表达式2 ]
3. 常用操作符

* 逻辑与：-a 或&&
* 逻辑或：-o或||
* 逻辑否：！

1. 补充
2. 查看磁盘使用率：df
3. free：查看系统空间
4. awk：用于以空格或制表符作为分隔，输出指定区域的字段数据
5. mysql默认端口：3306
6. netstat -ano

# 第八讲

## 8.1 条件判定

1. 语法
2. if 条件测试命令

then 命令序列 (如果then不换行的话，在条件测试命令后加;)

fi

1. if 条件测试命令

then 命令序列1

else 命令序列2

fi

1. if 条件测试命令1 ; then

命令序列1

elif 条件测试命令 ;then

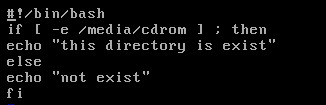
命令序列2

else

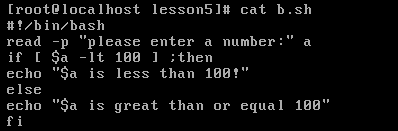
命令序列3

注意：if-elif-else与条件测试命令之间都需用空格隔开

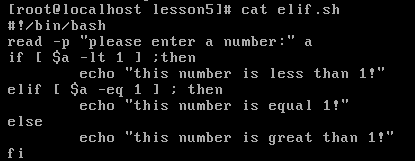
例1：判断是否存在路径/media/cdrom



例2：输入一个数判断是否小于100



例3：输入一个数判断其与1的关系



## 8.2 循环

1. for循环

语法一：

for 变量名 in取值列表

#取值列表以逗号或空格隔开（以逗号是看做一个整体）

do

命令序列

done

语法二：

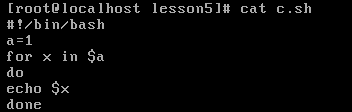
for((初始化;bool表达式;步进)) #初始化不需要进行数据类型的声明

do

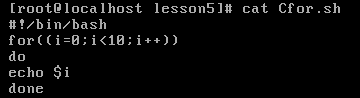
命令序列

done

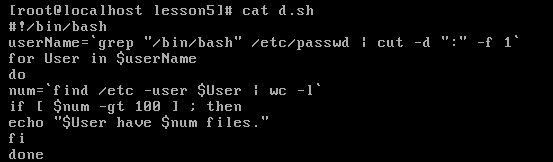
例1：for循环



例2：for循环输出0-9



例3：查找使用/bin/bash作为解释器的用户，输出/etc目录中拥有文件数大于100的用户，并指出文件数



1. while循环

语法：

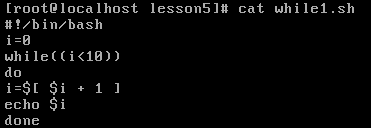
while 命令或表达式

do

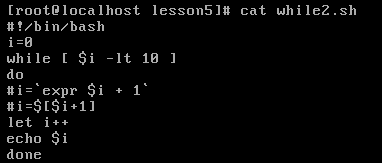
命令序列

done

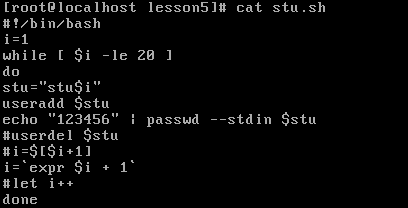
例1：while循环输出1-10



等效于：



例2：新建20个用户，用户名格式为stu1、stu2…默认密码为123456



1. until循环

语法：

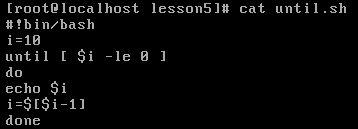
until 条件测试命令

do

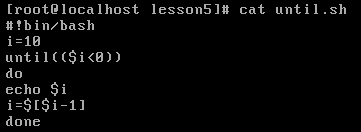
命令序列

done

例1：输出10-1



等效于：



1. break：在循环语句中，表示跳出当前所有循环体，执行循环体后的语句
2. continue：在循环语句中，表示跳出当前循环，继续下一次循环

## 8.3 case分支语句

语法：

case 变量值 in

模式1)

命令序列1

;; #使用双分号结束

模式2)

命令序列1

;;

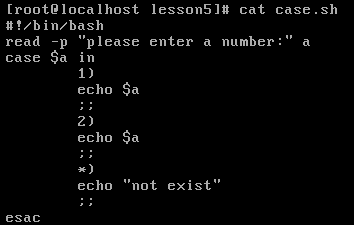
\*) #默认模式使用\*

默认执行的命令序列

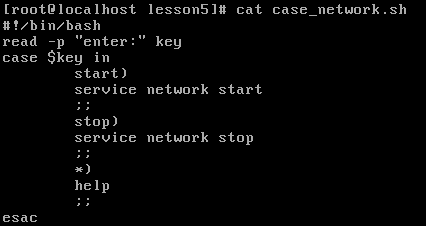
;;

esac

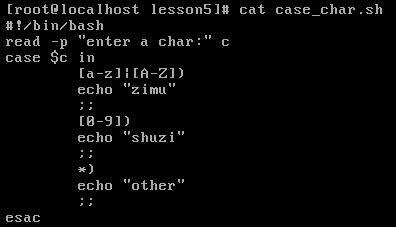
例1：



例2：通过输入信息控制网络服务的启动与停止，默认显示帮助信息



例3：判断输入字符的性质



## 8.4 shell函数

在编写Shell脚本程序时，将一些需要重复使用的命令操作，定义为公共使用的语句块，即可称为函数。合理使用Shell函数，可以使脚本内容更加简洁，增强程序的易读性，提高执行效率。

1. 定义

语法一：

function 函数名{

命令序列

}

语法二：

函数名(){

命令序列

}

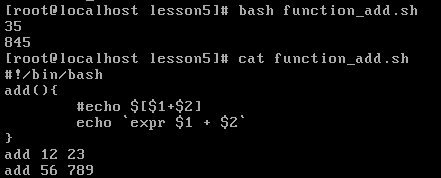
1. 调用

语法：函数名

1. 向函数内传参

语法：函数名 参数1 参数2…

例1：定义加法函数，用于计算2个整数的和，并调用计算12+23、56+789



补充：

SSH：使用安全内核进行连接，可实现远程登录，相关工具有putty、SecureCRTSecureFX