
link: null
title: 珠峰架构师成长计划
description: 分为内核版本和发型版本
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=256 sentences=332, words=1784

1. Linux

- Linux是一套免费使用和自由传播的类Unix操作系统
- 在服务器端领域和嵌入式领域有非常广泛的应用

2.版本

分为内核版本和发型版本

- [kernel \(https://www.kernel.org/\)](https://www.kernel.org/)
- 各个厂商会制作自己的发行版本
 - redhat
 - CentOS
 - debian
 - ubuntu
 - fedora

3. Linux与Windows的不同

- Linux严格区分大小写
- Linux中所有的内容以文件形式保存，包括硬件、用户和文件。
- Linux不靠扩展名区分文件类型，是靠权限来区分，但是有一些约定的扩展名，是给管理员看的
 - 压缩包 .gz .bz2 .tar.bz2 .tgz
 - 二进制文件 .rpm
 - 网页文件 .html .php
 - 脚本文件 .sh
 - 配置文件 .conf
- Windows下的程序不能直接在Linux中安装和运行
- Linux更多使用字符界面
 - 占用的系统资源更少
 - 减少了出错和被攻击的可能性，会让系统更稳定

4. Linux安装

4.1 什么是虚拟机

- 是一个虚拟PC的软件
- 可以在现有的操作系统上虚拟出一个新的硬件环境
- 相当于模拟出一台新的个人电脑
- 可以实现在一台机器上真正同时运行两个独立的操作系统
- [VMware \(http://www.vmware.com\)](http://www.vmware.com)
- [VMware8 \(VMware8\)](#)

4.2 虚拟机的主要特点

- 不需要分区或重新开机就能在同一台PC上使用两种以上的操作系统
- 本机系统可以与虚拟机系统网络通信
- 可以设定并且随时修改虚拟机操作系统的硬件环境
- 系统快照可以方便备份和回滚

4.3 建议VMWare配置

- CPU 建议主频1GHz以上
- 内存 建议2GB以上
- 硬盘 建议分区空闲空间8GB以上

4.4 虚拟机的安装

- [VMware8.0 \(http://www.vmware.com\)](http://www.vmware.com)

4.5 虚拟机使用

4.5.1 新建虚拟机

- 1. Create a New Virtual Machine开始新建虚拟机向导
- 1. 我以后再安装操作系统
- 1. Linux CentOS 32位
- 1. 20G硬盘

4.5.2 硬件管理

4.5.3 网络链接

- 虚拟机网卡-查看网络属性
 - VMware Virtual Ethernet Adapter For VMnet1
 - VMware Virtual Ethernet Adapter For VMnet8
- 网络连接
 - 桥接 直接连接到物理网络，会占用一个局域网内的真实IP地址，优先选择桥接
 - NAT 使用已共享的主机IP地址，可以上网，利用VMnet8网卡通信
 - Host-Only 与主机共享一个私有网络，只能本机使用，利用VMnet1进行通信，不能上网。

4.5.4 使用快照

- 可以使用快照
- 在合适的时间恢复快照

4.5.4 克隆 #

从当前的虚拟机克隆出一个虚拟机

- 可以克隆当前或者快照
- 克隆方式可以选择链接克隆或者完整克隆

4.5.5 分区 #



- 磁盘分区是使用分区编辑器在磁盘上划分几个逻辑部分
- 磁盘一旦划分成多个分区，不同类的目录与文件可以存储进不同的分区内
- 分区表的长度只有64个字节，里面又分成四项，每项16个字节。所以，一个硬盘最多只能分四个一级分区，又叫做"主分区"

4.5.6 扩展分区 #

- 为了突破4个分区的限制，就取出一个分区作为扩展分区
 - 扩展分区最多只能有1个
 - 主分区加扩展分区最多有4个
 - 不能写入数据，只能包含逻辑分区，逻辑分区最多是23个

4.5.7 格式化 #

- 格式化是指根据用户选定的文件系统(如FAT16(2G)、FAT32(4G)、NTFS、EXT2、EXT3、EXT4)对分区进行划分
- 目的是为了更好的写入和读取数据
- 主要是把整个分区切分成等大小的数据块，每个数据块是4KB，10K需要使用2个半的数据块。是存放文件的最小空间。
- 在分区中划出一片用于存放文件分配表，目录表等用户文件管理的磁盘空间。
 - ID
 - 修改时间
 - 权限
 - 数据块位置
- 格式化会清空数据

4.5.8 给分区分配盘符 #

- 分区
- 格式化
- 给分区分配设备文件名
- 分配盘符

4.5.9 硬件设备文件名 #

硬件 设备文件名 IDE硬盘 /dev/hd[a-d] SCSI/SATA/USB硬盘 /dev/sd[a-p] 光驱 /dev/cdrom或/dev/hdc 软盘 /dev/fd[0-1] 打印机(25针) /dev/lp[0-2] 打印机(USB) /dev/usb/lp[0-15] 鼠标 /dev/mouse

4.5.10 分区设备文件名 #

- 设备文件名只要插入硬盘，Linux会自动检测和分配
- 1-4一定给主分区和扩展分区使用
- 逻辑分区一定从5开始

硬件 设备文件名 /dev/hda1 IDE硬盘接口(133M/秒) /dev/sda1 SCSI硬盘接口(200M/秒)、SATA硬盘接口(500M/秒)

4.5.10.1 IDE硬盘接口 #



4.5.10.2 SCSI硬盘接口

□

4.5.10.3 SATA硬盘接口

□

4.5.10 挂载点

- 在Linux中是以目录作为盘符，称为挂载点
- 把盘符(目录)和分区链接在一起的过程成为挂载
- /为根目录，必须挂载到一个分区上，默认所有子目录都会写入这个分区
- 同一级目录下面的所有子目录可以有自己独立存储空间
- 必须有的分区
 - / 根分区
 - swap分区(交换分区，虚拟内存，一般为内存的2倍，不要超过2G)
- 推荐分区
 - /boot (启动分区,200M) 单独分区，避免分区写满造成系统无法启动

□

4.6 linux系统安装

- 选择操作系统
- 选择光驱启动

CentOS 默认安装是最小安装。您现在可以选择一些另外的软件。

- ☐ Desktop
- ☐ Minimal Desktop
- ☐ Minimal
- ☒ Basic Server
- ☐ Database Server
- ☐ Web Server
- ☐ Virtual Host
- ☐ Software Development Workstation

请选择您的软件安装所需要的存储库。

☒ CentOS

⊕ (A) 添加额外的存储库

📄 修改库 (M)

或者.

- ☒ 以后自定义 (I)
- ☐ 现在自定义 (C)

您要进行哪种类型的安装？

OS

☐

使用所有空间
删除所选设备中的所有分区。其中包含其它操作系统创建的分区。
提示：这个选项将删除所选设备中的所有数据。确定您进行了备份。

OS

☐

替换现有 Linux 系统
只删除 Linux 分区（由之前的 Linux 安装创建的）。这样就不会删除您存储设备中的其它分区（比如 VFAT 或者 FAT32）。
提示：这个选项将删除您所选设备中的所有数据。确定您进行了备份。

OS

☐

缩小现有系统
缩小现有分区以便为默认布局生成剩余空间。

OS

☐

使用剩余空间
保留您的现有数据和分区且只使用所选设备中的未分区空间，假设您有足够的空间可用。

?

☒

创建自定义布局
使用分区工具手动在所选设备中创建自定义布局。

请选择源驱动器

设备	大小 (MB)	挂载点/ RAID/卷	类型	格式
▼ 硬盘驱动器				
▼ sda (/dev/sda)				
sda1	200	/boot	ext4	✓
sda2	2000	/home	ext4	✓
sda3	1000		swap	✓
▼ sda4	17279		扩展分区	
sda5	17278	/	ext4	✓

格式化警告

⚠

已选择要格式化以下已存在的设备，破坏所有数据。

/dev/sda partition table (MSDOS)

取消(C)

格式化(E)

存储设备警告

⚠

以下设备中可能包含数据。

VMware, VMware Virtual S

20480.0 MB pci-0000:00:10.0-scsi-0:0:0:0

没有在这个设备中探测到分区或文件系统。

这可能是由于该设备为空白、未分区或虚拟。如果不是，那么它可能有一些数据是您使用它进行安装后无法恢复的。可从这个安装中删除该设备以保护那些数据。

您确定这个设备中不包含有价值的信息吗？

☒ 在所有包含未探测分区或文件系统的设备中应用我的选择 (A)

是，忽略所有数据 (Y)

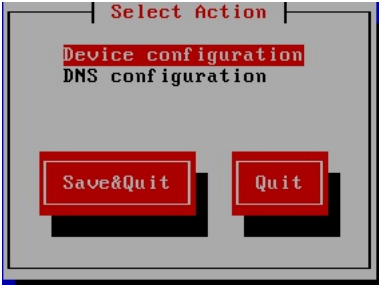
否，保留所有数据 (N)

Choose a Tool

Authentication configuration
Firewall configuration
Network configuration
System services

Run Tool

Quit



4.6.1 安装类型 <#>

- Install or upgrade an existing system 安装或者升级现有系统
- Install system with basic video driver 安装过程采用基本的显卡驱动
- Rescue installed system 进入系统修复模式
- Boot from local drive 退出安装从硬盘启动
- Memory test 存储介质检测

4.6.2 密码 <#>

- 复杂性
 - 八位字符以上、大小写字母、数字、符号
 - 不能是英文单词
 - 不能是和用户相关的内容
- 易记忆性
- 实效性

4.6.3 安装类型 <#>

- Desktop (桌面)
- Minimal Desktop (最小化桌面)
- Minimal (最小化)
- Basic Server (基本服务器,推荐)
- Database Server (数据库服务器)
- Web Server (网页服务器)
- Virtual Host (虚拟主机)
- software development workstation(软件开发工作站)

4.6.4 安装日志 <#>

- /root/install.log 存储了安装在系统中的软件包及其版本信息
- /root/install.log.syslog 存储了安装过程中留下的事件记录
- /root/anaconda-kernel.cfg 记录了安装过程中设置的选项信息，可以做为安装的模版文件

4.6.5 配置IP地址 <#>

4.6.5.1 setup <#>

- 防火墙配置
- 网络配置
- 系统服务
- 验证配置

4.6.5.2 ifcfg-eth0 <#>

```
cat /etc/sysconfig/network-scripts/ifcfg-eth0
```

参数 含义 TYPE=Ethernet #网卡类型 DEVICE=eth0 #网卡接口名称 ONBOOT=yes #系统启动时是否自动加载 BOOTPROTO=static #启用地址协议 --static:静态协议 --bootp协议 --dhcp协议 IPADDR=192.168.1.11 #网卡IP地址 NETMASK=255.255.255.0 #网卡网络地址 GATEWAY=192.168.1.1 #网卡网关地址 DNS1=10.203.104.41 #网卡DNS地址 HWADDR=00:0C:29:13:5D:74 #网卡设备MAC地址 BROADCAST=192.168.1.255 #网卡广播地址

4.6.5.2 网卡接口关闭与激活 <#>

```
ifdown eth0 #关闭网络
ifup eth0 #启动网络
```

4.6.5.3 网络服务启动与关闭 <#>

```
service network restart #重启网络服务
```

5. linux常用命令 <#>

5.1 常见目录 <#>

- / 根目录
- /boot 启动目录，启动相关文件
- /dev 设备文件
- /etc 配置文件
- /home 普通用户的家目录,可以操作
- /lib 系统库保存目录
- /mnt 移动设备挂载目录

- **/media** 光盘挂载目录
- **/misc** 磁带机挂载目录
- **/root** 超级用户的家目录,可以操作
- **/tmp** 临时目录,可以操作
- **/proc** 不能直接操作,保存的是内存的挂载点
- **/sys** 不能直接操作,保存的是内存的挂载点
- **/var** 变量
- **/bin** 普通命令
- **/sbin** 命令保存目录,级用户才可以执行的命令
- **/usr/bin** 系统软件资源目录 面向普通用户的系统命令
- **/usr/sbin** 系统软件资源目录 面向超级用户的系统命令

5.2 命令基本格式 #

5.2.1 命令提示符 #

```
[root@zhangrenyang ~]
```

- **root** 当前登录用户
- **localhost** 主机名
- **~** 当前工作目录,默认是当前用户的家目录, **root**就是**/root**,普通用户是 **/home/用户名**
- 提示符 超级用户是 **#**,普通用户是 **\$**

5.2.2 命令格式 #

- 命令 [选项] [参数]
- 当有多个选项时,可以写在一起
- 一般参数有简化和完整写法两种 **-a** 与 **--all**等效

5.2.3 ls #

- 查询目录中的内容
- **ls** [选项] [文件或者目录]
- 选项
 - **-a** 显示所有文件,包括隐藏文件
 - **-l** 显示详细信息
 - **-d** 查看目录本身的属性而非子文件 **ls /etc/**
 - **-h** 人性化的方式显示文件大小
 - **-i** 显示**inode**,也就是i节点,每个节点都有ID号
- 默认当前目录下的文件列表

5.2.3.1 -l #

显示详细信息

```
drwxr-xr-x . 1 root root 800 Sep 16 00:19 logs
```

drwxr-xr-x . 1 root root 800 Sep 16 00:19 logs 文件类型和权限 ACL权限 硬链接引用计数 所有者 所属组 文件大小 最后修改时间 文件名

5.2.3.2 文件类型和权限 #

```
-rwxr-xr-x
```

- 文件类型 - 文件、d 目录、l 软链接文件
- **u**(所有者)、**g**(所属组)、**o**(其他人)
- **r**(read) 读取、**w**(write) 写入、**x**(execute) 执行

5.3 文件处理命令 #

5.3.1 mkdir #

- 建立目录 **make directory**
- **mkdir -p** [目录名]
 - **-p** 递归创建

5.3.2 cd #

- 切换所在目录 **change directory**
- **cd** [目录]
 - **~** 家目录
 - 家目录
 - **~** 上次目录
 - **.** 当前目录
 - **..** 上级目录
- 相对路径是参照当前所在目录
- 绝对路径是从根目录开始
- 按**TAB**键可以补全命令和目录

5.3.3 pwd #

- 显示当前目录 **pwd**

5.3.4 rmdir #

- 删除目录 **remove empty directory**
- **rmdir** [目录名]

5.3.5 rm #

- 删除文件或者目录 **remove**
- **rm** [文件或者目录]
 - **-r** 删除目录
 - **-f** 强制删除
- **rm -rf** 文件或者目录] 递归强制删除所有目录

5.3.6 cp #

- **copy** 复制命令
- **copy** [源文件或者目录] [目标文件]

- o -r 复制目录,默认是复制文件
- o -p 连带文件属性复制
- o -d 若源文件是链接文件,则复制连接属性
- o -a 相当于 -rpd

5.3.7 mv

- 移动文件或者改名 move
- mv [源文件或者目录] [目标文件]

5.3.8 ln

- 链接命令,生成链接文件 link

5.3.8.1 硬链接特征

- 拥有相同的i节点和存储block块,可以看作是同一个文件
- 可以通过i节点访问
- 不能跨分区
- 不能针对目录使用
- 一般不使用

5.3.8.2 软链接特征

- ln -s [源文件] [目标文件]
 - o -s 创建软链接
- 类似Windows快捷方式
- 软链接拥有自己的节点和Block块,但是数据块中只保存源文件的文件名和i节点号,并没有实际的文件数据
- lwxrwxrwx ! 软链接 软链接的文件权限都是 777
- 修改任意一个文件,另一个都会改变
- 删除源文件,软链接不能使用
- 软链接源文件必须写绝对路径

5.4 文件搜索命令

5.4.1 locate

- 在后台数据库中按文件名搜索,速度比较快
- 数据保存在 /var/lib/mlocate 后台数据库,每天更新一次
- 可以 updatedb 命令立刻更新数据库
- 只能搜索文件名
- /etc/updatedb.conf 建立索引的配置文件
 - o PRUNE_BIND_MOUNTS = "yes" 全部生效,开启搜索限制
 - o PRUNEFSS 不搜索的文件系统
 - o PRUNENAMES 忽略的文件类型
 - o PRUNEPATHS 忽略的路径 /tmp

5.4.2 whereis

- 搜索命令所在路径以及帮助文档所在位置
- whereis 命令名 whereis ls
 - o -b 只查找可执行文件
 - o -m 只查找帮助文件

5.4.3 which

- 可以看到别名 which ls
- 能看到的都是外部安装的命令
- 无法查看Shell自带的命令,如 which cd

5.4.4 环境变量

```
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
```

- 定义的是系统搜索命令的路径
- echo \$PATH

5.4.5 find

- 文件搜索命令
- find [搜索范围] [搜索条件]

5.4.5.1 按名称搜索

- 避免大范围的搜索,会非常消耗系统资源

```
find / -name aaa.log
```

5.4.5.2 通配符

- find是在系统当中搜索符合条件的文件名,如果需要匹配,使用通配符匹配,通配符是完全匹配
- 通配符
 - o * 匹配任意内容
 - o ? 匹配任意一个字符
 - o [] 匹配任意一个中括号内的字符

```
find . -name "ab[cd ef]"
```

5.4.5.3 -i

不区分大小写

```
find / -iname A.log
```

5.4.5.4 -user

按所有者进行搜索

```
find /root -user root
find /root -nouser
```

5.4.5.5 按时间搜索

```
find /nginx/access.log -mtime +5
```

参数 含义 **atime** 文件访问时间 **ctime** 改变文件属性 **mtime** 修改文件内容 参数 含义 **-5** 5天内修改的文件 **5** 5天前当前修改的文件 **+5** 5天前修改的文件

5.4.5.6 按大小搜索

- k小写,M大写

```
find . -size 100k
```

参数 含义 **-8k** 小于8K **8k** 等于8K **+8k** 大于8K **+8M** 小于8M

5.4.5.7 按i节点搜索

```
find . -inum 123456
```

5.4.5.8 综合应用

```
find /tmp -size +10k -a -size -20k
```

- 查找/etc目录下, 大于10KB并且小于20KB的文件
- **-a** 逻辑与, 两个条件都满足
- **-o** 逻辑或, 两个条件满足一个就可以

```
find /tmp -size +10k -a -size -20k -exec ls -lh {} \;
```

- **exec** 对上个命令的结果进行操作

5.4.5.9 grep

- 在文件当中匹配符合条件的字符串
- **grep "10" access.log**
 - **-i** 忽略大小写
 - **-v** 排除指定字符串
- **find**命令, 在系统当中搜索符合条件的文件名, 如果需要匹配, 使用通配符匹配, 通配符是完全匹配
- **grep**命令 在文件当中搜索符合条件的字符串, 如果需要匹配, 使用正则表达式进行匹配, 正则表达式时包含匹配

5.5 帮助命令

5.5.1 基本用法

- **man** 命令 获取指定命令的帮助
- **man ls** 查看ls的帮助

5.5.2 man的级别

- 1 查看命令的帮助
- 2 查看可被内核调用的函数的帮助
- 3 查看函数和函数库的帮助
- 4 查看特殊文件的帮助
- 5 查看配置文件的帮助
- 6 查看游戏的帮助
- 7 查看其它的帮助
- 8 查看系统管理员可用命令的帮助
- 9 查看和内核相关文件的帮助

5.5.3 查看命令级别

- 查看命令级别
- **1p: POSIX utilities**
- **POSIX**表示可移植操作系统接口 (Portable Operating System Interface of UNIX, 缩写为 **POSIX**), **POSIX**标准定义了操作系统应该为应用程序提供的接口标准。

```
man -f ls
whatis ls
man 1 ls
man 1p ls
```

5.5.4 关键字搜索

```
- man -k passwd
```

5.5.5 shell 内部帮助

- **whereis**找到就是外部, 找不到就是内部

```
help cd
```

5.6 压缩与解压缩命令

```
.zip .gz .bz2 .tar.gz .tar.bz2
```

5.6.1 zip格式

- 压缩文件 **zip** 压缩文件名 源文件
- 压缩目录 **zip -r** 压缩文件名 源目录
- 解压 **unzip** 压缩文件名

```
mkdir book
touch book/1.txt
touch book/2.txt
zip -r book.zip book
unzip book.zip
```

5.6.2 gzip

命令 示例 含义 **gzip** 源文件 **gzip a.txt** 压缩为.gz格式的压缩文件, 源文件会消失 **gzip -c** 源文件 > 压缩文件 **gzip -c yum.txt > yum.txt.gz** 压缩为.gz格式的压缩文件, 源文件不会消失 **gzip -r** 目录 **gzip -r xx** 压缩目录下的所有子文件, 但是不压缩目录 **gzip -d** 压缩文件名 **gzip -d yum.txt.gz** 解压缩文件,不保留压缩包 **gunzip** 压缩文件 **gunzip yum.txt.gz** 解压缩文件,不保留压缩包

- 压缩是压缩目录下的文件

5.6.3 .bz2格式压缩

命令 示例 含义 **bzip2** 源文件 **bzip2 1.txt** 压缩为.bz2格式的文件, 不保留源文件 **bzip2 -k** 源文件 **zip2 -k 1.txt** 压缩为.bz2格式的文件, 保留源文件 **bzip2 -d** 压缩文件名 **bzip2 -d 1.txt.bz2** 解压压缩包 **bunzip2** 压缩文件名 **bunzip2 1.txt.bz2** 解压压缩包

- **bzip2** 不能压缩目录

5.6.4 tar #

- 打包命令
- tar -cvf 打包文件名 源文件
 - -c 打包
 - -v 显示过程
 - -f 指定打包后的文件名

```
tar -cvf book.tar book
gzip book.tar
bzip2 book.tar
```

- x 解打包

```
tar -xvf book.tar
```

5.6.4 tar.gz压缩格式 #

可以先打包为 .tar格式，再压缩为 .gz格式 -z 压缩为.tar.gz格式 -x 解压缩.tar.gz格式 |命令|示例|含义| |---| |tar -zcvf 压缩包名 .tar.gz源文件|tar -zcvf book.tar.gz book|可以先打包为 .tar格式，再压缩为 .gz格式| |tar -zxvf 压缩包名.tar.gz|tar -zxvf book.tar.gz|解压.tar.gz压缩包| |tar -jcvf 压缩包名 .tar.bz2源文件|tar -zcvf .tar.bz2 book|可以先打包为 .tar格式，再压缩为 .bz2格式| |tar -jxvf 压缩包名.tar.bz2|tar -zxvf book.tar.bz2|解压.tar.bz2压缩包|

5.7 关机和重启命令 #

5.7.1 shutdown #

shutdown 关机命令

- -c 取消前一个关机命令
- -h 关机
- -r 重启

```
shutdown -r 06:00
shutdown -c
```

5.7.2 init #

关机

```
init 0
```

重启

```
init 6
```

系统的运行级别

- 0 关机
- 1 单用户
- 2 不 完全多用户， 不包含NFS服务
- 3 完全多用户
- 4 未分配
- 5 图形界面
- 6 重启

5.7.3 logout #

退出登录

```
logout
```

5.8 挂载 #

5.8.1 挂载命令 #

- mount 查看系统中已经挂载的设备

```
mount
mount -a
```

5.8.2 挂载命令格式#

- mount [-t 文件系统] [-o 特殊选项] 设备文件名 挂载点
- 选项
 - -t 文件系统 ext4 iso9660
 - -o 特殊选项

5.8.3 挂载光驱 #

```
mkdir /mnt/cd
mount -t iso9660 /dev/sr0 /mnt/cdrom
```

5.8.4 卸载光驱 #

```
umount /dev/sr0
umount /mnt/cdrom
```

5.8.4 挂载U盘 #

```
fdisk -l
mount -t vfat /dev/sdb1 /mnt/usb
```

- linux默认不支持NTFS格式

** 5.9 查看登录用户信息 #**

5.9.1 w #

查看登录用户信息

- USER 登录的用户名
- TTY 登录的终端 tty1 本地终端 pts/0远程终端
- FROM 登录的IP
- LOGIN 登录时间
- IDLE 用户闲置时间
- JCPU 该终端所有进程占用的时间

- PCPU 当前进程所占用的时间
- WHAT 正在执行的命令

5.9.2 who

查看登录用户信息

- USER 登录的用户名
- TTY 登录的终端 tty1 本地终端 pts/0 远程终端
- LOGIN 登录时间（登录的IP）

5.9.3 last

查看当前登录和过去登录的用户信息 默认读取 /var/log/wtmp 文件

- 用户名
- 登录终端
- 登录IP
- 登录时间
- 退出时间(在线时间)

5.9.4 lastlog

查看所有用户的最后一次登录时间

- 用户名
- 登录终端
- 登录IP
- 最后一次登录时间

6. shell

- shell是一个命令行解释器，它为用户提供了一个向Linux内核发送请求以便运行程序的界面系统级程序
- 用户可以用Shell来启动、挂起、停止或者编写一些程序
- Shell还是一个功能相当强大的编程语言，易编写，易调试，灵活性较强。
- Shell是解释执行的脚本语言，在Shell中可以直接调用Linux系统命令。

** 6.1 查看支持的shell #**

- /etc/shells

** 6.2 echo #**

- 输出命令
- -e 支持反斜线控制的字符转换

控制字符 作用 \a 输出警告音 \b 退格键，也就是向左删除键 \n 换行符 \r 回车键 \t 制表符，也就是Tab键 \v 垂直制表符 \onnn 按照八进制ASCII码表输出字符，其中0为数字零，nnn是三位八进制数 \xhh 按照十六进制ASCII码表输出字符，其中hh是两位十六进制数

- 30m-47m
- echo -e "\e[1;31m warning \e[0m"

[符号]颜色| [#30m]黑色| [#31]红色| [#32]绿色| [#33]黄色| [#34]蓝色| [#35]洋红| [#36]青蓝| [#37]白色|

** 6.3 编写执行shell #**

```
echo -e "\e[1;34m hello world \e[0m"
```

赋予执行权限，直接运行

```
chmod 755 hello.sh
./hello.sh
```

通过Bash调用执行脚本

```
bash hello.sh
```

** 6.4 别名 #**

- 命令别名 == 小名
- 临时生效
- alias
- alias rm="rm -f"
- 写入环境变量配置文件 vi ~/.bashrc
- source ~/.bashrc
- unalias 别名 删除别名

** 6.5 命令的生效顺序 #**

- 绝对路径或者相对路径
- 别名
- bash内部命令
- 按照\$PATH环境变量定义的目录查找顺序找到的第一个命令

** 6.6 命令快捷键 #**

命令 含义 ctrl+c 强制终止当前命令 ctrl+l 清屏 ctrl+a 光标移动到命令行首 ctrl+e 光标移动到命令行尾 ctrl+u 从光标所在的位置删除到行首 ctrl+z 把命令放入后台 ctrl+r 在历史命令中搜索

** 6.7 历史命令 #**

- history [选项] [历史命令保存文件]
- 选项
 - -c 清空历史命令
 - -w 把缓存中的历史命令写入历史命令保存文件 ~/.bash_history
- 默认保存1000条 /etc/profile HISTSIZE=10000

** 6.8 调用 #**

- 使用上下箭头调用以前的历史命令
- 使用 !n 重复执行第n条历史命令
- 使用 !! 重复执行上一条命令
- 使用 !字符 重复执行最后一条以该字符串开头的命令

** 6.9 输出重定向 #**

6.9.1 标准输入输出

设备 设备文件名 文件描述符 类型 键盘 /dev/stdin 0 标准输入 显示器 /dev/stdout 1 标准输出 显示器 /dev/stderr 2 标准错误输出 类型 符号 作用 标准输出重定向 命令 > 文件 以覆盖的方式，把命令的正确输出输出到指定的文件或设备当中 标准输出重定向 命令 >> 文件 以追加的方式，把命令的正确输出输出到指定的文件或设备当中 错误输出重定向 命令 > 文件 以覆盖的方式，把命令的错误输出输出到指定的文件或设备当中 错误输出重定向 命令 >> 文件 以追加的方式，把命令的错误输出输出到指定的文件或设备当中 正确输出和错误输出同时保存 命令 >> 文件 2>&1 以覆盖的方式，把正确输出和错误输出都保存到同一个文件当中 正确输出和错误输出

同时保存 命令>文件 2>>1 以追加的方式,把正确输出和错误输出都保存到同一个文件当中 正确输出和错误输出同时保存 命令&>文件 以覆盖的方式,把正确输出和错误输出都保存到同一个文件当中 正确输出和错误输出同时保存 命令<&>文件 以追加的方式,把正确输出和错误输出都保存到同一个文件当中 正确输出和错误输出同时保存 命令>>文件1 2>文件2 以覆盖的方式,正确的输出追加到文件1中,把错误输出追加到文件2中

6.9.2 输入重定向

- 命令 < 文件把文件做为命令的输入
- 命令 << 标识符 标识符把标识符之间内容作为命令的输入
- wc < access.log
- wc <>

** 6.10 管道符号

6.10.1 多命令顺序执行

|多命令执行符|格式|作用| |:命令1;命令2|多个命令执行,命令之间没有任何逻辑联系| |:&&|命令1&&命令2|逻辑与 当命令1正确执行,则命令2才会执行 当命令1执行不正确,则命令2不会执行| |:||命令1|| 命令2|逻辑或 当命令1执行不正确,则命令2才会执行 当命令1正确执行,则命令2不会执行|

```
- date;ls;date;ls
- ls && echo yes || echo no
```

6.10.2 管道符号

- 命令1的正确输出会作为命令2的操作对象
- 命令1|命令2

```
ls /etc/ | more
netstat -an | grep ESTABLISHED | wc -l
```

6.10.3 通配符

匹配文件名和目录名 |通配符|作用| |:|:|:| |:|匹配一个任意字符| |:|匹配0个或任意字符,也就是可以匹配任意内容| |:|匹配中括号中任意一个字符| |:|匹配中括号中任意一个字符,代表范围| |:|匹配不是中括号中的一个字符|

6.10.4 其它符号

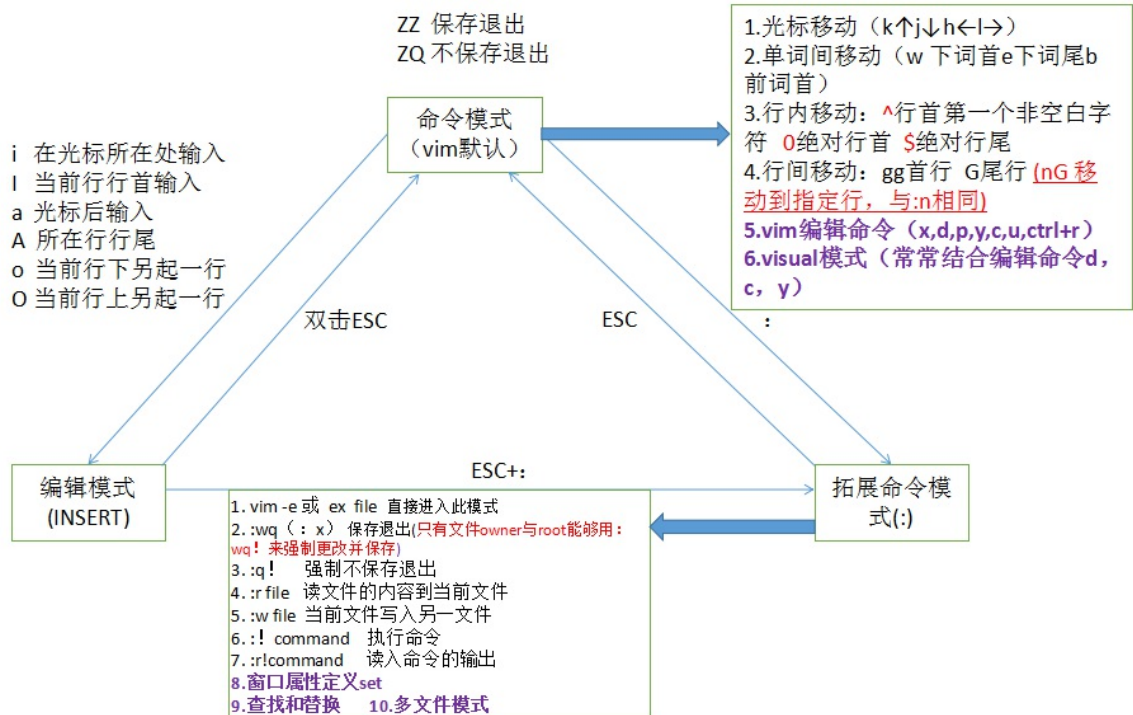
符号 作用 " 单引号。在单引号中所有的特殊符号,如\$和都没有特殊含义 "" 双引号,在双引号里特殊符号都没有特殊含义,但是 \\$ 例外,拥有调用变量值,引用命令和转义的含义

反引号,扩起来的是系统命令 \$() 和反引号一样 # 在shell脚本中, #开头的行代表注释 \$ 用于调用变量的值 \ 转义符号

```
- a='ls'
- b=$(ls)
```

7. vi编辑器

- VI visual interface
- 可视化接口
- 类似与windows中的记事本
- vim支持多级撤销
- 跨平台
- 语法高亮
- 支持图形界面



** 7.1 操作模式

模式名 含义 Command Mode 命令模式 等待输入的模式 Insert Mode 输入模式 编辑模式, 用于输入文本 Last Line Mode 底行模式(尾行、末行) 可以输入指令, 搜索, 保存

- vim + abc
- vim +3 abc
- vim +5 abc
- vim +/xxx abc
- vim a b c :n

** 7.2 常用指令

7.2.1 底行模式

- :w 保存

- :q 退出
- :! 强制保存
- :ls 列出所有的文件
- :n 下一个
- :N 上一个
- :15 跳转到指定行
- /xxx 从光标位置开始向后搜索 xxx 字符串
- ?xxx 从光标位置开始向前搜索

7.2.2 命令模式 #

- h 光标左移
- j 光标下移
- k 光标上移
- l 光标右移
- ctrl+f 向下翻页(front)
- ctrl+b 向上翻页
- ctrl+d 向下翻半页
- ctrl+u 向上翻半页
- dd 删除光标所在行
- o 在光标所在行的下方插入一行并切换到输入模式
- yy 复制光标所在的行
- p 在光标所在行的下方粘贴
- P 在光标所在行的上方粘贴

8. 用户和用户组 #

- 使用操作系统的人都是用户
- 用户组是具有相同系统权限的一组用户

** 8.1 用户组 #**

8.1.1 /etc/group #

- /etc/group 存储当前系统中所有用户组信息
- group:x:123:abc,def
- 组名称:组密码占位符:组编号:组中用户名列表
- root 组编号为0
- 1-499系统预留的编号 预留给安装的软件和服务的
- 用户手动创建的用户组从500开始
- 组密码占位符都是x

8.1.2 /etc/gshadow #

- 存放当前系统中用户组的密码信息
- 和group中的记录一一对应
- Group: *:abc
- 组名称 组密码 组管理者 组中用户名

8.1.3 /etc/passwd #

- 存储当前系统中所有用户的信息
- user:x:123:456:xxxx:/home/user:/bin/bash
- 用户名:密码占位符:用户编号: 用户注释信息:用户主目录:shell类型

8.1.4 /etc/shadow #

- 存放当前系统中所有用户的密码信息
- user:xxx:.....
- 用户名:密码:

8. 用户操作 #

添加组

```
groupadd student
```

修改组名称

```
groupmod -n stu student
```

修改组编号

```
groupmod -g 111 stu
```

添加分组并指定编号

```
groupadd -g 222 teacher
```

删除分组

```
groupdel 222
```

添加分组

```
groupadd teacher
```

为用户指定所属组

```
useradd -g teacher zhangsan
```

为用户指定所属组

```
useradd -g teacher lisi
```

为用户指定工作目录

```
useradd -d /home/zhangsan zhangsan
```

指定注释

```
usermod -c iamateacher zhangsan
```

修改用户名

```
usermod -l zhangsan zhangsan2
```

指定文件夹

```
usermod -d /home/zhangsan2 zhangsan2

修改用户所属组

usermod -g stu zhangsan2

删除用户

userdel zhangsan2

删除所属文件夹

userdel -r lisi
```

9. 用户命令 #

```
显示登录的用户名

whoami

显示指定用户信息，包括用户编号，用户名 主要组的编号及名称，附属组列表

id zhangsan

显示zhangsan用户所在的所有组

groups zhangsan

显示用户详细资料

finger zhangsan
```

10. 附录 #

“ 10.1 系统启动 #”

10.1.1 BIOS #

- 计算机通电后，第一件事就是读取刷入ROM芯片的开机程序，这个程序叫做(Basic Input/Output System)

10.1.2 硬件自检 #

- BIOS程序首先检查，计算机硬件能否满足运行的基本条件，这叫做“硬件自检”（Power-On Self-Test）
- 如果硬件出现问题，主板会发出不同含义的蜂鸣，启动中止。如果没有问题，屏幕就会显示出CPU、内存、硬盘等信息。

10.1.3 启动顺序 #

- 硬件自检完成后，BIOS把控制权转交给下一阶段的启动程序。
- 这时，BIOS需要知道，“下一阶段的启动程序”具体存放在哪一个设备
- BIOS需要有一个外部储存设备的排序，排在前面的设备就是优先转交控制权的设备。这种排序叫做“启动顺序”（Boot Sequence）
- BIOS按照“启动顺序”，把控制权转交给排在第一位的储存设备。
- 这时，计算机读取该设备的第一个扇区，也就是读取最前面的512个字节。如果这512个字节的最后两个字节是0x55和0xAA，表明这个设备可以用于启动；如果不是，表明设备不能用于启动，控制权于是被转交给“启动顺序”中的下一个设备。
- 这最前面的512个字节，就叫做“主引导记录”（Master boot record，缩写为MBR）

10.1.4 主引导记录的结构 #

- “主引导记录”只有512个字节，放不了太多东西。它的主要作用是，告诉计算机到硬盘的哪一个位置去找操作系统。
 - （1）第1-446字节：是用来记录系统的启动信息的,调用操作系统的机器码
 - （2）第447-510字节(64个字节): 分区表（Partition table），分区表的作用，是将硬盘分成若干个区
 - （3）第511-512字节：主引导记录签名（0x55和0xAA）

10.1.5 分区表 #

- 磁盘分区是使用分区编辑器在磁盘上划分几个逻辑部分
- 磁盘一旦划分成多个分区，不同类的目录与文件可以存储进不同的分区内
- “主引导记录”因此必须知道将控制权转交给哪个区
- 分区表的长度只有64个字节，里面又分成四项，每项16个字节。所以，一个硬盘最多只能分四个一级分区，又叫做“主分区”
 - （1）第1个字节：如果为0x80，就表示该主分区是激活分区，控制权要转交给这个分区。四个主分区里面只能有一个是激活的。
 - （2）第2-4个字节：主分区第一个扇区的物理位置（柱面、磁头、扇区号等等）。
 - （3）第5个字节：主分区类型，比如FAT32、NTFS等。
 - （4）第6-8个字节：主分区最后一个扇区的物理位置。
 - （5）第9-12字节：该主分区第一个扇区的逻辑地址。
 - （6）第13-16字节：主分区的扇区总数。

10.1.6 硬盘启动 #

- 计算机的控制权就要转交给硬盘的某个分区了
- 四个主分区里面，只有一个是激活的。计算机会读取激活分区的第一个扇区，叫做“卷引导记录”（Volume boot record，缩写为VBR）

10.1.7 操作系统 #

- 控制权转交给操作系统后，操作系统的内核首先被载入内存。
- 以Linux系统为例，先载入 /boot目录下面的 kernel。内核加载成功后，第一个运行的程序是 /sbin/init。它根据配置文件（Debian系统是/etc/initab）产生init进程。这是Linux启动后的第一个进程，pid 进程编号为1，其他进程都是它的后代
- 然后，init线程加载系统的各个模块，比如窗口程序和网络程序，直至执行 /bin/login程序，跳出登录界面，等待用户输入用户名和密码。