# 目录

[目录 1](#_Toc503383286)

[第1章 认识linux 10](#_Toc503383287)

[1.1 linux简介 10](#_Toc503383288)

[1.1.1 linux简述 10](#_Toc503383289)

[1.1.2 linux诞生历程 10](#_Toc503383290)

[1.1.3 GNU NPL 简介 10](#_Toc503383291)

[1.1.4 linux特点 11](#_Toc503383292)

[1.1.5 涉及的领域 11](#_Toc503383293)

[1.2 Centos与Red Hat的区别和联系 11](#_Toc503383294)

[1.3 Centos的安装 11](#_Toc503383295)

[1.3.1 配置虚拟机 11](#_Toc503383296)

[1.3.2 网卡模式 11](#_Toc503383297)

[1.3.3 nat 模式设置步骤 12](#_Toc503383298)

[1.3.4 桥接模式设置 14](#_Toc503383299)

[1.4 克隆机的网卡配置 15](#_Toc503383300)

[1.5 xshell的优化 16](#_Toc503383301)

[1.5.1 终端 16](#_Toc503383302)

[1.5.2 外观 16](#_Toc503383303)

[1.5.3 文件路径 17](#_Toc503383304)

[第2章 linux基础命令 17](#_Toc503383305)

[2.1帮助命令 17](#_Toc503383306)

[2.1.1 man 17](#_Toc503383307)

[2.1.2 help 18](#_Toc503383308)

[2.2文件和目录操作命令 18](#_Toc503383309)

[2.2.1 mv 18](#_Toc503383310)

[2.2.2 mkdir 18](#_Toc503383311)

[2.2.3 touch 18](#_Toc503383312)

[2.2.4 rmdir 19](#_Toc503383313)

[2.2.5 rm 19](#_Toc503383314)

[2.2.6 cd 19](#_Toc503383315)

[2.2.7 cp 19](#_Toc503383316)

[2.2.8 pwd 20](#_Toc503383317)

[2.2.9 ls 20](#_Toc503383318)

[2.2.10 tree 20](#_Toc503383319)

[2.2.11 find 21](#_Toc503383320)

[2.3文件查看和内容处理命令 21](#_Toc503383321)

[2.3.1 cat 21](#_Toc503383322)

[2.3.2 head 22](#_Toc503383323)

[2.3.3 tail 22](#_Toc503383324)

[2.3.4 grep 22](#_Toc503383325)

[2.3.5 egrep 23](#_Toc503383326)

[2.3.6 sed 23](#_Toc503383327)

[2.3.7 awk 23](#_Toc503383328)

[2.3.8 vi 23](#_Toc503383329)

[2.3.9 vim 23](#_Toc503383330)

[2.3.10 more 23](#_Toc503383331)

[2.3.11 less 24](#_Toc503383332)

[2.3.12 wc 24](#_Toc503383333)

[2.3.13 file 24](#_Toc503383334)

[2.3.14 stat 24](#_Toc503383335)

[2.3.15 dumpe2fs 24](#_Toc503383336)

[2.3.16 od 24](#_Toc503383337)

[2.3.17 sort 25](#_Toc503383338)

[2.3.18 scp 25](#_Toc503383339)

[2.3.19 split 26](#_Toc503383340)

[2.3.20 paste 27](#_Toc503383341)

[2.3.21 diff 28](#_Toc503383342)

[2.3.22 vimdiff 28](#_Toc503383343)

[2.3.23 uniq 28](#_Toc503383344)

[2.4其他命令 29](#_Toc503383345)

[2.4.1 alias 29](#_Toc503383346)

[2.4.2 unalias 29](#_Toc503383347)

[2.4.3 seq 29](#_Toc503383348)

[2.4.4 echo 29](#_Toc503383349)

[2.4.5 xargs 29](#_Toc503383350)

[2.4.6 env 29](#_Toc503383351)

[2.4.7 ln 30](#_Toc503383352)

[2.4.8 basename 30](#_Toc503383353)

[2.4.9 dirname 30](#_Toc503383354)

[2.4.10 rename 30](#_Toc503383355)

[2.4.11 which 31](#_Toc503383356)

[2.4.12 whereis 31](#_Toc503383357)

[2.4.13 locate 31](#_Toc503383358)

[2.4.14 readlink 31](#_Toc503383359)

[2.4.15 md5sum 32](#_Toc503383360)

[2.4.16 ssh 32](#_Toc503383361)

[2.4.17 tee 33](#_Toc503383362)

[2.4.18 tr 33](#_Toc503383363)

[2.4.19 dos2unix（需要事先进行安装） 34](#_Toc503383364)

[2.5 系统相关的命令 35](#_Toc503383365)

[2.5.1 uname 35](#_Toc503383366)

[2.5.2 runlevel 35](#_Toc503383367)

[2.5.3 init 35](#_Toc503383368)

[2.5.4 history 35](#_Toc503383369)

[2.5.5 shutdown 35](#_Toc503383370)

[2.5.6 halt 35](#_Toc503383371)

[2.5.7 chkconfig 35](#_Toc503383372)

[2.5.8 last 36](#_Toc503383373)

[2.5.9 lastb 36](#_Toc503383374)

[2.5.10 lastlog 36](#_Toc503383375)

[2.5.11 w 37](#_Toc503383376)

[2.5.12 du 37](#_Toc503383377)

[2.5.13 date 37](#_Toc503383378)

[2.5.14 netstat 39](#_Toc503383379)

[2.5.15 fdisk 39](#_Toc503383380)

[2.5.16 free 40](#_Toc503383381)

[2.5.17 partprobe 41](#_Toc503383382)

[2.5.18 mkfs 41](#_Toc503383383)

[2.5.19 mkswap 41](#_Toc503383384)

[2.5.20 swapon 41](#_Toc503383385)

[2.5.21 swapoff 42](#_Toc503383386)

[2.5.22 mount 42](#_Toc503383387)

[2.5.23 umount 42](#_Toc503383388)

[2.5.24 df 42](#_Toc503383389)

[2.5.25 fsck 43](#_Toc503383390)

[2.5.26 parted 43](#_Toc503383391)

[2.5.27 tune2fs 43](#_Toc503383392)

[2.5.28 resize2fs 44](#_Toc503383393)

[2.6 用户相关的命令 44](#_Toc503383394)

[2.6.1 useradd 44](#_Toc503383395)

[2.6.2 userdel 44](#_Toc503383396)

[2.6.3 usermod 45](#_Toc503383397)

[2.6.4 groupadd 45](#_Toc503383398)

[2.6.5 groupdel 46](#_Toc503383399)

[2.6.6 chown 46](#_Toc503383400)

[2.6.7 chmod 46](#_Toc503383401)

[2.6.8 umask 47](#_Toc503383402)

[2.6.9 chattr 47](#_Toc503383403)

[2.6.10 lsattr 47](#_Toc503383404)

[2.6.11 id 47](#_Toc503383405)

[2.6.12 chage 48](#_Toc503383406)

[2.6.13 su 48](#_Toc503383407)

[2.6.17 sudo 49](#_Toc503383408)

[2.6.18 visudo 49](#_Toc503383409)

[2.6.19 chpasswd 49](#_Toc503383410)

[2.6.20 passwd 50](#_Toc503383411)

[2.7 网络相关的命令 51](#_Toc503383412)

[2.7.1 ifconfig 51](#_Toc503383413)

[2.7.2 ip 52](#_Toc503383414)

[2.7.3 route 53](#_Toc503383415)

[2.7.4 dig 53](#_Toc503383416)

[2.7.5 nslookup 54](#_Toc503383417)

[2.7.6 tcpdump 54](#_Toc503383418)

[2.8 系统常用开关机重启以及注销常用的命令 55](#_Toc503383419)

[2.9 备注 56](#_Toc503383420)

[第3章 linux基础入门中部 56](#_Toc503383421)

[3.1 linux目录的基础知识 56](#_Toc503383422)

[3.1.1 linux目录的特点 56](#_Toc503383423)

[3.1.2 linux树状目录结构图 56](#_Toc503383424)

[3.1.3 linux常用目录详解 57](#_Toc503383425)

[3.2 linux文件属性 59](#_Toc503383426)

[3.2.1 linux文件的概述 59](#_Toc503383427)

[3.2.2 inode 索引节点 59](#_Toc503383428)

[3.2.3 block 60](#_Toc503383429)

[3.2.4 相关的面试题 61](#_Toc503383430)

[3.2.5 linux软硬连接 61](#_Toc503383431)

[3.2.6 软连接 62](#_Toc503383432)

[3.2.7 文件连接的总结 62](#_Toc503383433)

[3.2.8 目录连接文件的总结 63](#_Toc503383434)

[3.2.9 软硬连接的区别 63](#_Toc503383435)

[3.2.10 linux文件的删除原理 63](#_Toc503383436)

[3.3 linux文件的权限 64](#_Toc503383437)

[3.3.1 rwx详解 64](#_Toc503383438)

[3.3.2 权限位图解 65](#_Toc503383439)

[3.3.3 linux普通文件权限的读、写、执行的解读 65](#_Toc503383440)

[3.3.4 目录权限的读、写、执行的解读 65](#_Toc503383441)

[3.3.5 访问文件的图解说明 以/tmp/oldboyfile为例 66](#_Toc503383442)

[3.3.6 umak 66](#_Toc503383443)

[3.3.7 suid 67](#_Toc503383444)

[3.3.8 sgid 67](#_Toc503383445)

[3.4 用户 68](#_Toc503383446)

[3.4.1 用户的分类 68](#_Toc503383447)

[3.4.2 用户的详解 68](#_Toc503383448)

[3.4.3 用户相关的配置文件 69](#_Toc503383449)

[3.4.4 /etc/passwd 配置文件内容的详解 69](#_Toc503383450)

[3.4.5 删除用户的经验 69](#_Toc503383451)

[3.5 有关用户管理的三个重要目录和文件 70](#_Toc503383452)

[3.5.1 linux用户管理重要初始化目录 /etc/skel 70](#_Toc503383453)

[3.5.2 /etc/login.defs配置文件的详解（了解即可） 71](#_Toc503383454)

[3.5.3 /etc/default/useradd配置文件的详解 73](#_Toc503383455)

[3.6 sudo配置文件/etc/sudoers的讲解 74](#_Toc503383456)

[3.6.1 /etc/sudoer文件中各选项的作用 74](#_Toc503383457)

[3.6.2 配置sudo配置文件需要注意的事项 74](#_Toc503383458)

[3.6.3 用户的提权配置说明 74](#_Toc503383459)

[3.6.4 sudo执行过程图解 75](#_Toc503383460)

[3.6.5 sudo的审计功能 75](#_Toc503383461)

[3.6.6 sudo配置实例 76](#_Toc503383462)

[3.7 环境变量 77](#_Toc503383463)

[第四章 linux基础入门下 78](#_Toc503383464)

[4.1 linux通配符 78](#_Toc503383465)

[4.1.1 linux通配符与正则表达式的区别 78](#_Toc503383466)

[4.1.2 常用通配符 78](#_Toc503383467)

[4.1.3 举例 79](#_Toc503383468)

[4.2 linux 正则表达式 81](#_Toc503383469)

[4.2.1 linux基础正则符号 81](#_Toc503383470)

[4.2.2举例 81](#_Toc503383471)

[4.2.3 linux扩展的正则表达式 86](#_Toc503383472)

[4.2.4 举例 86](#_Toc503383473)

[4.2.5 元字符 87](#_Toc503383474)

[4.2.5 POSIX字符类 87](#_Toc503383475)

[4.3 linux 定时任务之crond 88](#_Toc503383476)

[4.3.1 什么是crond 88](#_Toc503383477)

[4.3.2 为什么要使用crond定时任务 88](#_Toc503383478)

[4.3.3 linux系统的定时任务 88](#_Toc503383479)

[4.3.4 crond的基本格式 88](#_Toc503383480)

[4.3.5 crond语法格式中时间段的含义表及取值范围 89](#_Toc503383481)

[4.3.6 crond语法格式中的特殊符号及其含义 89](#_Toc503383482)

[4.3.7 crontab 指令的说明 90](#_Toc503383483)

[4.3.8 使用者权限及定时任务文件 90](#_Toc503383484)

[4.3.9 crontab设置的规范 90](#_Toc503383485)

[4.3.10 crond /dev/null 2>&1的作用 91](#_Toc503383486)

[4.3.11 生产环境中调试定时任务的方法 91](#_Toc503383487)

[第5章 linux性能优化 92](#_Toc503383488)

[5.1 关闭selinux 92](#_Toc503383489)

[5.1.1 selinux的配置文件 92](#_Toc503383490)

[5.1.2 临时关闭 92](#_Toc503383491)

[5.1.3 永久关闭 92](#_Toc503383492)

[5.2 关闭防火墙 93](#_Toc503383493)

[5.2.1 临时关闭 93](#_Toc503383494)

[5.2.2 永久关闭 93](#_Toc503383495)

[5.2.3 查看iptables的状态 93](#_Toc503383496)

[5.3 linux中文显示设置 93](#_Toc503383497)

[5.3.1 配置文件 93](#_Toc503383498)

[5.3.2 设置中文字符 93](#_Toc503383499)

[5.3 隐藏linux版本信息 93](#_Toc503383500)

[5.3.1 版本信息的配置文件 93](#_Toc503383501)

[5.3.2 清空该配置文件中的内容 94](#_Toc503383502)

[5.4 精简开机自启动服务 94](#_Toc503383503)

[5.4.1 需要保留的重要开机自启动服务 94](#_Toc503383504)

[5.4.2 开启需要的服务 95](#_Toc503383505)

[5.4.3 chkconfig 原理 100](#_Toc503383506)

[5.5 忘记root密码的三种解决办法 101](#_Toc503383507)

[5.5.1 第一种方法 101](#_Toc503383508)

[5.5.2 第二种方法 104](#_Toc503383509)

[5.5.3 第三种方法 104](#_Toc503383510)

[5.6 修改主机名的规范步骤 105](#_Toc503383511)

[5.6.1 修改步骤 105](#_Toc503383512)

[5.6.2 实例 105](#_Toc503383513)

[第6章 磁盘 106](#_Toc503383514)

[6.1 linux磁盘设备的基础知识 106](#_Toc503383515)

[6.1.1 linux磁盘的历史简述 106](#_Toc503383516)

[6.1.2 磁盘的图形说明 106](#_Toc503383517)

[6.1.3 柱面的图形说明 107](#_Toc503383518)

[6.1.4 有关磁盘的重要名词的说明 107](#_Toc503383519)

[6.1.5 磁道、扇区、柱面的总结 107](#_Toc503383520)

[6.1.6 磁盘的工作原理 107](#_Toc503383521)

[6.1.7 磁盘大小的计算 108](#_Toc503383522)

[6.1.8 磁盘的接口类型 108](#_Toc503383523)

[6.1.9 生产环境中主流磁盘信息的对比 109](#_Toc503383524)

[6.1.10 磁盘选购的方法 109](#_Toc503383525)

[6.1.11 内存的查看方法 109](#_Toc503383526)

[6.1.12 磁盘的分类 110](#_Toc503383527)

[6.1.13 两者之间的比较 110](#_Toc503383528)

[6.2 RAID 110](#_Toc503383529)

[6.2.1 raid简介 110](#_Toc503383530)

[6.2.2 raid的级别 110](#_Toc503383531)

[6.2.3 raid0详解 111](#_Toc503383532)

[6.2.4 raid1详解 111](#_Toc503383533)

[6.2.5 raid5详解 112](#_Toc503383534)

[6.2.6 raid10详解 113](#_Toc503383535)

[6.2.7 常用RAID选择成本、性能、可靠性的对比 115](#_Toc503383536)

[6.2.8 常见raid的比较 115](#_Toc503383537)

[6.2.9 DELL R710使用6块盘的实际制作 115](#_Toc503383538)

[6.3 分区知识 116](#_Toc503383539)

[6.3.1 常用的分区工具 116](#_Toc503383540)

[6.3.2 swap分区的构建 120](#_Toc503383541)

[6.3.3 分区表的详解 121](#_Toc503383542)

[6.3.4 文件系统 121](#_Toc503383543)

[6.3.5 格式化与挂载 123](#_Toc503383544)

[6.3.6 常见的linux磁盘操作的总结 125](#_Toc503383545)

[6.3.7 安装xfs文件系统的操作实践 125](#_Toc503383546)

[6.3.8 企业案例 126](#_Toc503383547)

[第7章 linux网络基础知识 127](#_Toc503383548)

[7.1 硬件知识 127](#_Toc503383549)

[7.1.1 网卡 127](#_Toc503383550)

[7.1.2 网线 127](#_Toc503383551)

[7.1.3 HUB 127](#_Toc503383552)

[7.1.4 交换机 128](#_Toc503383553)

[7.1.5 路由器 128](#_Toc503383554)

[7.2 网络基础原理 129](#_Toc503383555)

[7.2.1 OSI七层模型及TCP/IP四层模型的简单总结 129](#_Toc503383556)

[7.2.2 linux常用端口及对照的服务 129](#_Toc503383557)

[7.2.3 数据包的封装与解封装的过程图解 130](#_Toc503383558)

[7.2.4 TCP/UDP的总结 131](#_Toc503383559)

[7.2.4.3 TCP与UDP的对比 131](#_Toc503383560)

[7.2.5 TCP的三次握手与四次断开详解 132](#_Toc503383561)

[7.2.6 访问网站的基本流程 134](#_Toc503383562)

[7.2.7 DNS解析原理 135](#_Toc503383563)

[7.3 相关命令总结 135](#_Toc503383564)

[7.4 tcpdump 138](#_Toc503383565)

[7.5 开启路由转发功能 140](#_Toc503383566)

# 第1章 认识linux

## 1.1 linux简介

### 1.1.1 linux简述

linux就像windows一样是一个操纵系统，但是多用于企业级的服务器，由于他的安全、稳定、免费、高效、可自由的更改源代码。使得他深受人们欢迎。现在常用的Linux == GNU的软件+linux内核（托瓦斯）

### 1.1.2 linux诞生历程

* **Unix**1969年诞生在贝尔实验室，在20世纪70年代末逐步盛行。但是**Unix有版权**
* 1984年**大胡子哥**发起自由软件运动，且成立了自由软件基金会（FSF）和GNU项目。最大的目的是解决版权问题
* 1984年谭邦宁教授开发出了用于教学的**类Unix**系统，并命名为**Minix**，但是只用在教学方面
* 1989年**Minix**可以运行在X86的计算机平台上（桌面）
* 1990年芬兰赫尔辛基大学的学生**脱袜子哥**接触**Minix**并学习研究其核心的设计理念，在1991年在**Minix**编写各种驱动和操作系统的组核心的设计理念，在1991年底公开**linux**内核源码（0.02）
* 1993年l**inux1.0**发行，并且转向GPL协议
* 1994年**linux**第一个商业版本（Slackware）出现

### 1.1.3 GNU NPL 简介

* GNU：全称是“GNU’s not unix”，类Unix操作系统是由一系列应用程序、系统库和开发工具构成的软件集合，并加上了用于资源分配和硬件管理的内核
* GNU GPL（GNU General Public License，通用公共许可证）最著名的开源许可协议。开源社区最著名的linux内核就是在GPL许可下发布的。GPL许可是自由软件基金会创建的喲

GPL的核心是保证任何人有共享和修改自由软件的自由，任何人有权取得、修改和重新发

布自由软件的源代码的权利，但是必须要同时给出具体更改的源代码

### 1.1.4 linux特点

* 兼容Unix，并具备Unix特性
* 开源、安全、免费、可随意修改内核源代码，没有版权问题
* 具有多用户、多任务、多线程、多CPU
* 有大量的并且免费的第三方应用程序

### 1.1.5 涉及的领域

* **IT服务器：**为中小企业提供开源的数据库、WEB等，没用版权问题
* **个人PC桌面：** 和windows一样 （主要是ubuntu）
* **嵌入式：**安卓手机、自动售货机、交换机、路由器、防火墙等

## 1.2 Centos与Red Hat的区别和联系

* Red Hat: 9.0的内核位2.4.20。在版本9.0之后，Red Hat 不再遵循GPL协议，成为收费产品（但仍开源）。
* Fedora：Red Hat的一个分支，遵循GPL协议。
* Centos：Red Hat的另一重要分支。遵循GPL协议。

## 1.3 Centos的安装

### 1.3.1 配置虚拟机

详细步骤详见附录

### 1.3.2 网卡模式

#### 1.3.2.1 nat 模式

* 简介： 物理主机可以随便移动，虚拟机不需要更改网络配置

#### 1.3.2 桥接模式

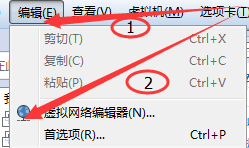
* 简介：物理主机移动时，虚拟机需要更改网络配置

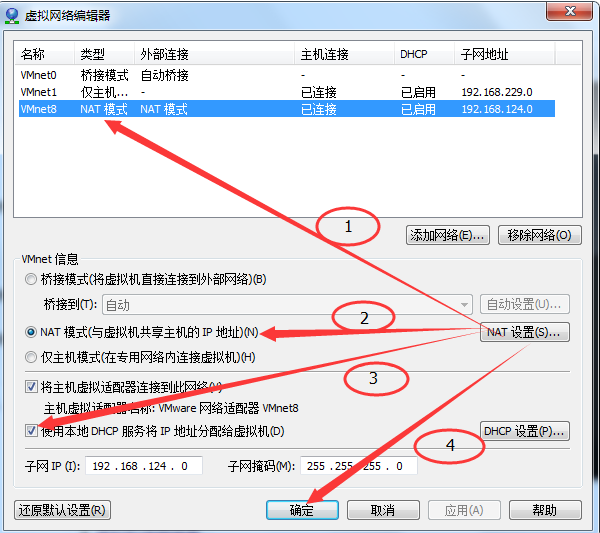
#### 1.3.3 Host-only 模式 仅主机模式

* 简介：虚拟机和物理主机可以互访，但是出不去

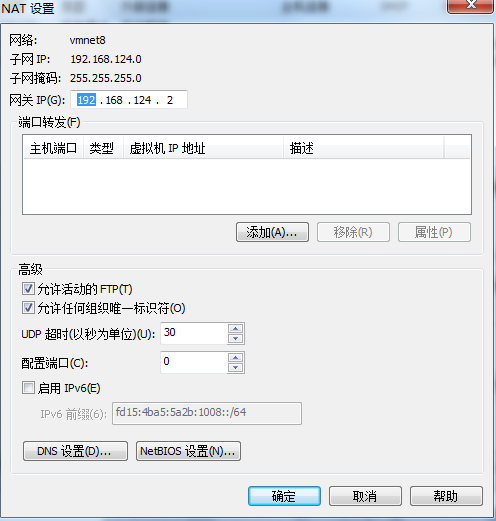
### 1.3.3 nat 模式设置步骤

#### 1.3.3.1 虚拟网络编辑器设置



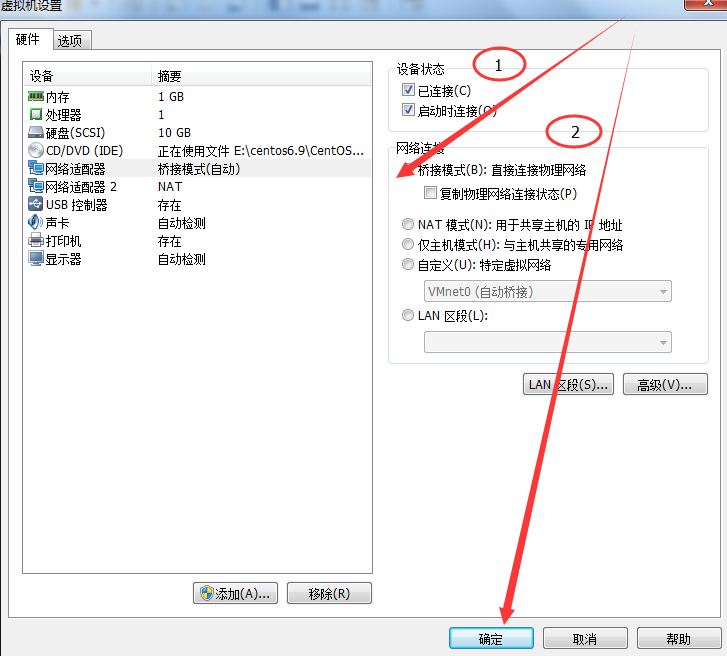


#### 1.3.3.2 nat设置

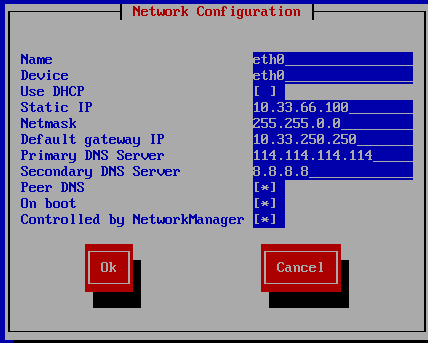


### 1.3.4 桥接模式设置

#### 1.3.4.1 选择网卡模式



#### 1.3.4.2 配置IP



## 1.4 克隆机的网卡配置

* 删除网卡配置文件中的UUID和HADDR

DEVICE=eth0

#HWADDR=00:0c:29:4d:7d:0a

TYPE=Ethernet

#UUID=c5e50e7b-2d23-4d2b-81ef-dd1d83eaaf81

* 清空/etc/udev/rules.d/70-persistent-net.rules

[root@zabbix-1 rules.d]# >70-persistent-net.rules

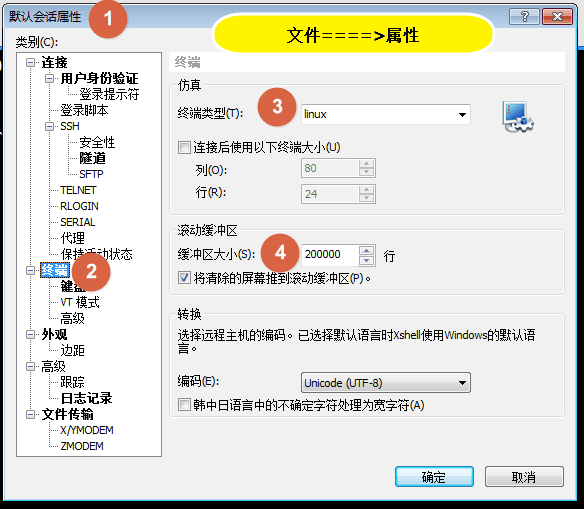
* 重新启动克隆机

[root@zabbix-1 ~]# reboot

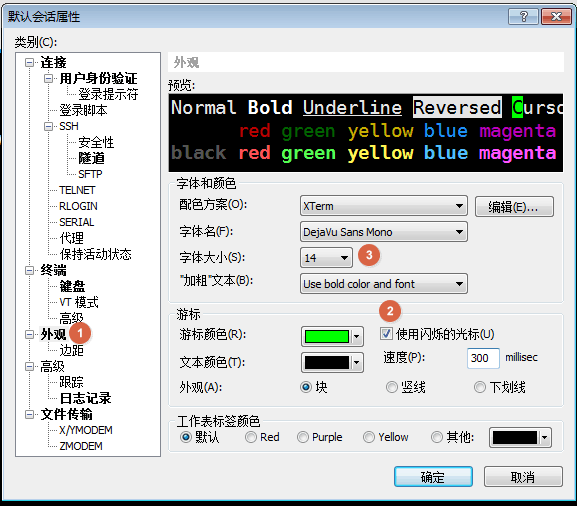
* 【说明】以上步骤可以简记为 两删一清空一重启

## 1.5 xshell的优化

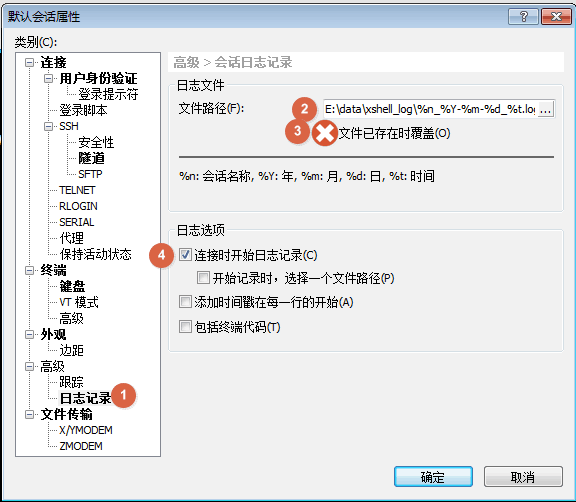
### 1.5.1 终端

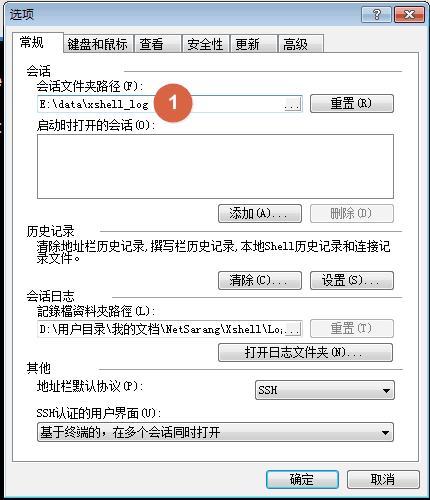


### 1.5.2 外观



### 1.5.3 文件路径





# 第2章 linux基础命令

## 2.1帮助命令

### 2.1.1 man

* 【功能】 外置命令时用man显示帮助信息

### 2.1.2 help

* 【作用】 内置命令时用help显示帮助信息

注意：一个命令是否为内置可以用enable+命令 进行查看

## 2.2文件和目录操作命令

### 2.2.1 mv

* 【功能】 移动目录或者文件，重命名目录或者文件
* 【常用参数】

-b：如果存在同名的文件，则在覆盖前进行备份

-i：如果存在同名的文件，则在覆盖前进行询问

-v：列出移动或者覆盖时的信息

-u：比较源文件与目标文件的修改时间，若目标文件新则不覆盖

* 【注意】

不能把一个文件移动到多个目录中去，就相当于不能把一个苹果放到多个框中一样

### 2.2.2 mkdir

* 【功能】 创建目录
* 【常用参数】

-p：递归创建目录

-v：显示创建的过程

-m：创建目录时设置改目录的权限

* 【举例】

[root@smile mnt]# mkdir -m 700 /test/

### 2.2.3 touch

* 【功能】 创建文件 若文件存在可以修改文件的时间戳
* 【常用参数】

-a：修改文件的访问（acess）时间 可以用stat +文件名进行查看

-m：修改文件修改（modify）时间

-c或--no-create：不创建不存在的文件

-d：在创建文件时使用指定的时间而非现在的时间

-t：修改已存在文件的时间（格式： 年月日时间）

-r：使用指定文件的时间戳更新文件的时间戳 -r后接文件

注意：access 表示最后一次访问文件的时间

modify 表示最后一次修改文件的时间

change 表示最后一次对文件属性（权限、大小等）的改变时间

### 2.2.4 rmdir

* 【功能】 删除空的目录，非空的目录不能删除呦
* 【常用参数】

-v：显示过程

### 2.2.5 rm

* 【功能】 删除文件或者目录，默认不能删除目录
* 【常用参数】

-r：删除目录

-f：强制删除

* 【注意】

正确的删除文件或目录的姿势：

* 用mv命令移动到/tmp（回收站）目录下，替代rm
* cd要删除文件的目录中去，用 find **.** -type f （文件）d（目录）-name “名字” |xargs rm -i
* 删除目录时，一定要切换到目的目录中在进行删除
* 删除文件时不要加 -r参数（脱裤子放屁），但是删除目录时一定要加-r参数

### 2.2.6 cd

* 【功能】 切换目录
* 【常用参数】

~：家目录 和什么也不加具有相同的效果

-：回到上一次的目录

.：表示当前目录

..：回到上一级目录

### 2.2.7 cp

* 【功能】 复制文件或目录，但是默认不能复制目录
* 【常用参数】

-r：递归复制 复制目录时用

-p：复制时保持文件或者目录的属性

-d：若是链接文件则复制链接而不是文件本身

-a：相当于-pdr

-b：在复制文件时 会先进行备份然后在覆盖

* 【注意】

在复制文件时若不想让其提醒，则用\cp或/bin/cp

### 2.2.8 pwd

* 【功能】 显示当前所在路径

### 2.2.9 ls

* 【功能】 列出目标目录中所有的子目录和文件
* 【常用参数】

-a：列出目录下的所有的文件，包括以 **.** 和 **..**开头的文件

-l：以长格式显示目录下的文件和目录信息

-d：只显示目录但是不现实目录下面的文件

-r：倒序显示目录或文件信息

-i：显示文件或目录的索引节点号

-t：按文件修改的时间进行排序 默认是按照升序进行排列的

-F：给不同的文件结尾加标识

-p：给目录结尾加斜线

--color=auto：显示颜色

* 【注意】

linux下面隐藏的文件都是以 . 开头的

* 【常用组合】

-ltr：按照倒序的时间显示目录或者文件 方便查看新建的目录或者文件

-lh：以人性话的格式显示文件和目录的大小

-lS：按照文件的大小从大到小进行显示

-ld：只显示子目录的信息

### 2.2.10 tree

* 【功能】 显示目录树
* 【常用参数】

-f：显示完整的绝对路径信息

-d：只显示目录

-r：以相反的方式进行排列

-a：显示所有文件和目录

-L<数字>：显示目录的深度

* 应用实例

快速区分某个目录下面的目录和文件的方法 可用 tree -d

### 2.2.11 find

* 【功能】 查找文件
* 【常用参数】

-type：后接文件的类型（f d 等）

-name：后接要查找的文件名字

-a：取交集 相当于and’

-o：是或者的意思 相当于or

！：取反的意思

-maxdepth 数字：查找的深度

-mtime +n/-n/n：按照文件修改的时间进行查找

-size：按照文件的大小来进行查找

* 【注意】

+n 表示n天以前 -n 表示n天以内 n 表示第n天

* 【应用实例】

在/data目录下找到oldboy.tx文件 同时查看文件的内容并将其删除

[root@oldboy ~]# find /data/ -type f -name 'oldboy.txt'

/data/oldboy.txt

[root@oldboy ~]# find /data/ -type f -name 'oldboy.txt' |xargs cat

oldboy linux

[root@oldboy ~]# find /data/ -type f -name 'oldboy.txt' |xargs rm -f

[root@oldboy ~]# ls /data/

oldgirl.txt test.txt

在/test目录下面找到名字为“file2.txt”的文件，并将其移动到/tnp目录下面

find . -type f -name "file2.txt" |xargs -i mv {} /tmp/

注意：-i参数

此题还有别的方法喲

注意：常和xargs和-exec 命令 {} \;结合起来使用，并且用的最多的就是用来删除文件代替rm

## 2.3文件查看和内容处理命令

### 2.3.1 cat

* 【功能】 查看文件的内容
* 【常用参数】

-n：由1开始对所有输出的行编号

-b：对输出的所有非空白行进行编号

* 【应用实例】

cat file1 file2 > file3 将两个文件合并为一个文件

cat > file创建文件

cat –n file1 > file2 将文件1的内容加上行号输入到file2中

### 2.3.2 head

* 【功能】 显示文件的前N行默认显示前10行
* 【常用参数】

-n：显示前#行 –n后接数字 但是此参数也可以不加只加数字

### 2.3.3 tail

* 【功能】 显示文件的后N行默认显示后10行
* 【常用参数】

-n：显示后#行 –n后接数字 此参数同样也可以不加

-f：具有实时监测的效果 常常用来分析日志

### 2.3.4 grep

* 【功能】 过滤需要的内容 **三剑客老三**
* 【常用参数】

-v：显示不包含过滤内容的所有内容

-c：将匹配到的行数进行计数然后输出

-n：对匹配的行在其首行显示行号

-s：不显示不存在或无匹配文本的错误信息

-A：除了显示匹配的行外，还显示匹配行的后#行

-B：除了显示匹配的行外，还显示匹配行的前#行

-C：除了显示匹配的行外，还显示匹配行前后各#行

--color=auto：对匹配的内容显示颜色

-i：不区分大小写（即忽大小写）

-a：在二进制文件中，以文本文件的方式搜索数据

-E：支持扩展正则

* 【应用实例】

显示/data/ 目录下面test.txt文件中的20到21行

[root@oldboy data]# grep 20 -A 1 test.txt

20

21

此题还有别的方法喲

### 2.3.5 egrep

* 【功能】 相当于grep -E 功能和grep一样
* 【应用实例】

就是可以同时过滤出多项

[root@oldboy test]# chkconfig --list |grep "3:on"|egrep "sshd|network|sysstat|rsylog|crond"

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

network 0:off 1:off 2:on 3:on 4:on 5:on 6:off

sshd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

sysstat 0:off 1:on 2:on 3:on 4:on 5:on 6:off

### 2.3.6 sed

【功能】 对文件增删改查 **三剑客老二**

* 【常用参数】

-n：取消默认输出

-i：直接修改文件的内容，不在屏幕上输出显示

* 【常用命令】

d：删除

a：新增 通常是在当前行的下一行出现

i：插入 通常是在当前行的上一行出现

c：取代 就是把符合条件的行给换成新的内容

s：取代 可直接进行取代 场合g搭配使用（s###g）

l：打印不可见字符

### 2.3.7 awk

* 【功能】 强大的文本分析工具 **三剑客中的老大**

### 2.3.8 vi

* 【功能】 文本编辑工具

### 2.3.9 vim

* 【功能】 比vi强大 常用来写代码

### 2.3.10 more

* 【功能】 分页显示输出的内容，回车一次向后显示一行，空格一次向后显示一屏，能进行回退（新版可以按b键回退 老版不行）

### 2.3.11 less

* 【功能】 分页显示输出的内容，回车一次先后显示一行，空格一次向后显示一屏，可以进行回退，按b键回退一页

### 2.3.12 wc

* 【功能】 用来计算数字，利用wc命令可以计算文件的Byte数、字数或是列数--
* 【常用参数】

-c或--bytes或--chars：只显示字节数

-l或--lines：只显示行数

-w或--words：只显示字数

-L：显示最大行的长度

### 2.3.13 file

* 【功能】 显示文件的类型
* 【常用参数】

-b：列出结果时，不显示文件名

-v：显示版本信息

### 2.3.14 stat

* 【功能】 显示文件的状态信息
* 【常用参数】

-t：以简洁的方式显示

--help：显示指令的帮助信息

--version：显示指令的版本信息

### 2.3.15 dumpe2fs

* 【功能】 显示ext2/ext3/ext4的文件系统的超级块和块组信息
* 【常用参数】

-b：打印文件系统中预留的块信息

-h：仅显示超级块的信息

-f：强制显示所有信息，用于不能识别的文件系统的标识

### 2.3.16 od

* 【功能】通常用于显示或查看文件中不能直接显示在终端的字符，常见的文件为文本文件和二进制文件
* 【常用参数】

-t：设置输出格式

--help：在线帮助

--version：显示版本信息

### 2.3.17 sort

* 【功能】依据不同的数据类型进行排序
* 【常用参数】

-k：选择以哪个分区进行排序

-b：忽略每行开始处的空白字符

-c：检查文件是否已经按照顺序进行排序

-f：排序时忽略大小写字母

-M：将前面3个字母依照月份的缩写进行排序

-n：依照数值大小进行排序

-r：以相反的顺序进行排序

-t<分隔符>：指定排序时所用的栏位分隔字符

-o<输出文件>：将排序后的结果输出到指定的文件中

### 2.3.18 scp

* 【功能】远程拷贝文件，和cp命令类似，但是cp只能在本机上进行复制，scp可以跨服务器，并且scp的传输是加密的，可能会稍微影响一下速度，当你的服务器变为只读 read only system时，用scp可以将文件移出来
* 【常用参数】

-1：使用ssh协议的版本为1

-2：使用ssh协议的版本为2

-4：使用IPV4寻址

-6：使用IPV6寻址

-B：使用批处理模式（在传输的过程中不在输出密码等）

-C：使用压缩的功能

-F：指定ssh的配置文件

-l：指定用户所使用的贷款 单位为Kbit/s

-o：指定ssh的参数

-P：指定远程主机的端口号

-q：不显示复制的进度条

-r：以递归的方式复制

* 【示例】

从远程主机复制文件到本地目录下

scp root@10.10.10.10:/opt/soft/nginx-0.5.38.[tar](http://man.linuxde.net/tar" \t "_blank" \o "tar命令).gz /opt/soft/

将远程主机的目录复制到本地

scp -r root@10.10.10.10:/opt/soft/mongodb /opt/soft/

从本地复制文件到远程主机上

scp /opt/soft/nginx-0.5.38.tar.gz [root@10.10.10.10:/opt/soft/scptest](mailto:root@10.10.10.10:/opt/soft/scptest)

将本地主机的目录复制到远程主机上

scp -r /opt/soft/mongodb root@10.10.10.10:/opt/soft/scptest

### 2.3.19 split

* 【功能】按照指定的行数或大小分割文件
* 【常用参数】

-b：指定分割的大小

-a：指定生成文件后缀的长度

-d：使用数字作为后缀

-l：指定分割的行数

* 【应用举例】

[root@oldboy 2017]# split -l 10 /etc/inittab new\_

[root@oldboy 2017]# ls new\_\*

new\_aa new\_ab new\_ac

[root@oldboy 2017]# head new\_\*

==> new\_aa <==

# inittab is only used by upstart for the default runlevel.

# ADDING OTHER CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.

# System initialization is started by /etc/init/rcS.conf

# Individual runlevels are started by /etc/init/rc.conf

# Ctrl-Alt-Delete is handled by /etc/init/control-alt-delete.conf

==> new\_ab <==

# Terminal gettys are handled by /etc/init/tty.conf and /etc/init/serial.conf,

# with configuration in /etc/sysconfig/init.

# For information on how to write upstart event handlers, or how

# upstart works, see init(5), init(8), and initctl(8).

# Default runlevel. The runlevels used are:

# 0 - halt (Do NOT set initdefault to this)

# 1 - Single user mode

# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)

==> new\_ac <==

# 3 - Full multiuser mode

# 4 - unused

# 5 - X11

# 6 - reboot (Do NOT set initdefault to this)

#

id:3:initdefault:

[root@oldboy 2017]# split -l 10 -a 4 /etc/inittab abc\_

[root@oldboy 2017]# ls abc\_\*

abc\_aaaa abc\_aaab abc\_aaac

### 2.3.20 paste

* 【功能】将多个文件进行合并
* 【常用参数】

-d：指定分隔符

-s：文件的内容合并其上方，而不是平行合并

* 【应用举例】

[root@oldboy 2017]# cat a

1

2

3

[root@oldboy 2017]# paste a newaa

1 # inittab is only used by upstart for the default runlevel.

2 #

3 # ADDING OTHER CONFIGURATION HERE WILL

[root@oldboy 2017]# paste -d: a newaa

1:# inittab is only used by upstart for the default runlevel.

2:#

3:# ADDING OTHER CONFIGURATION HERE WILL

[root@oldboy 2017]# paste -s a newaa

1 2 3

# inittab is only used by upstart for the default runlevel. # # ADDING OTHER CONFIGURATION HERE WILL

[root@oldboy 2017]#

### 2.3.21 diff

【功能】比较两个文件之间的不同（最多只能比较两个文件）

【常用参数】

-a：相对于源文件增加了哪些内容

-d：相对于源文件删除了哪些内容

-c：相对于源文件改变了哪些内容

【应用举例】

[root@oldboy 2017]# diff -a svn2 svn.txt

5d4

< stu10315

8a8

> sjfk23d2

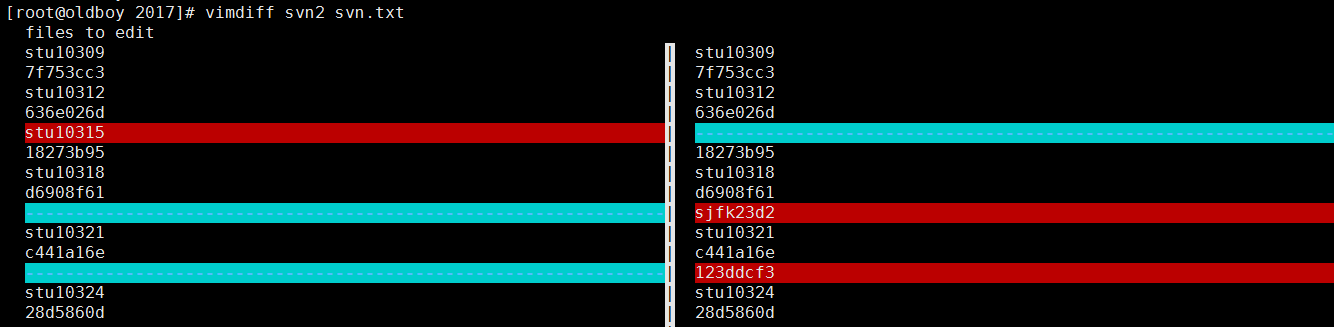
10a11

> 123ddcf3

注意：d前面表示的是 第一个文件中的行数 后面表示的是第二个文件中的行数

### 2.3.22 vimdiff

【功能】可视化的文件比较工具（最多可以支持4个文件）

【应用举例】

### 2.3.23 uniq

* 【功能】用于去除文件中重复的行，常于sort结合起来使用
* 【常用参数】

-c：在每行行首加上该行重复的次数

-i：不区分大小写

-d：仅显示重复的行

-u：仅显示唯一的行

## 2.4其他命令

### 2.4.1 alias

* 【功能】 查看别名 设置别名
* 【注意】

要想使设置的别名永久生效，只需把其放到~/.bashrc 并使用source命令让其生效即可

### 2.4.2 unalias

* 【功能】 取消别名 后根命令名

### 2.4.3 seq

* 【功能】 打印序列 默认是竖着打印
* 【常用参数】

-s：横着打印 指明分隔符 默认时\n

-w：在列前面加0 使得列的宽度相同

-f：指定格式显示

* 【常用格式】

seq 【选项】尾数

seq 【选项】首数 尾数

seq 【选项】首数 增量 尾数

### 2.4.4 echo

* 【功能】 打印

### 2.4.5 xargs

* 【功能】 从标准输入读入创建执行命令，给其他命令传递参数的一个过滤器，也是组合多个命令的一个工具
* 【常用参数】

-n：n后面跟数字 表示分组

-i：find . -type f -name "file2.txt" |xargs -i mv {} /tmp/

-t：先打印命令然后在执行

* 【应用举例】

[root@oldboy tmp]# find /tmp/ -type f -name "a"|xargs mv -t /tmp/ceshi

[root@oldboy tmp]# ls /tmp/ceshi

a

### 2.4.6 env

* 【功能】 显示系统所有的环境变量

### 2.4.7 ln

* 【功能】 创建软硬连接
* 【常用参数】

-s：创建软连接

什么也不加创建的是硬链接

* 注意：

ln 源文件 目标文件 （且目标文件事先不能存在）

### 2.4.8 basename

* 【功能】从长路径中取文件名
* 【常用举例】

[root@oldboy test]# basename /test/stu\_102999\_1.JPG

stu\_102999\_1.JPG

[root@oldboy test]# basename /test/stu\_102999\_1.JPG 1.JPG

stu\_102999\_

[root@oldboy test]# basename /test/stu\_102999\_1.JPG .JPG

stu\_102999\_1

### 2.4.9 dirname

* 【功能】从长路径中取出目录
* 【常用举例】

[root@oldboy test]# dirname /test/stu\_102999\_1.JPG

/test

### 2.4.10 rename

* 【功能】重命名 格式：rename from to file

其中from代表的是需要替换或者要处理的字符，可以文件的一部分也可以是问文件的扩展名，to代表的是替换后的内容（即是要替换为啥）

* 【常用举例】

[root@oldboy test]# touch stu\_102999\_{1..5}\_finished.jpg

[root@oldboy test]# ls

a.sh c.sh.bak stu\_102999\_2\_finished.jpg

a.txt d.txt stu\_102999\_3\_finished.jpg

bbb.sh e.txt stu\_102999\_4\_finished.jpg

b.sh f.txt stu\_102999\_5\_finished.jpg

c.sh stu\_102999\_1\_finished.jpg test.txt

[root@oldboy test]# rename "\_finished" "" \*.jpg

[root@oldboy test]# ls

a.sh b.sh d.txt stu\_102999\_1.jpg stu\_102999\_4.jpg

a.txt c.sh e.txt stu\_102999\_2.jpg stu\_102999\_5.jpg

bbb.sh c.sh.bak f.txt stu\_102999\_3.jpg test.txt

### 2.4.11 which

【功能】查找并显示给定文件的绝对路径 从环境变量$PATH设置的目录里查找符合条件的文件

### 2.4.12 whereis

* 【功能】查找二进制程序、源代码文件、man手册页等相关文件的路径
* 【常用参数】

-b：只查找二进制文件

-m：只查找说明文件

-u：查找不包含指定类型的文件

-s：只查找源代码文件

* 【注意】

which命令只能用于程序名的搜素，而且只搜索二进制文件（-b）、man说明文件（-m）、源代码文件（-s），如果省略参数则返回所有信息

### 2.4.13 locate

* 【功能】通过名字查找文件
* 【常用参数】

-u：更新slocate数据库

--version：显示版本信息

* 【注意】

locate命令是find -name 的另一种写法，但是前者要比后者快的多，原因是因为他搜索的不是具体目录，而是搜索一个数据库/var/lib/locatedb,这个数据库含有本地所有文件的信息，linux系统会自动创建这个数据库，并且每天自动更新一次，所以用该命令查找不到最新变动的文件，要想避免这种情况在查找之前先使用updatedb命令手动更新数据库。

### 2.4.14 readlink

* 【功能】查找符号连接（即软连接）的源文件

### 2.4.15 md5sum

* 【功能】计算和检验文件的校验和
* 【常用参数】

-w：当检验不正确时给出警告信息

--status：验证成功时不输出任何信息

-c：从指定文件中读取MD5校验和，并进行校验

* 【常用举例】

[root@oldboy ~]# md5sum test.txt

ba2f0f3909a4fb539a5dd22e7b0075e8 test.txt

[root@oldboy ~]# md5sum test.soft

ba2f0f3909a4fb539a5dd22e7b0075e8 test.soft

[root@oldboy ~]# md5sum test.hard

ba2f0f3909a4fb539a5dd22e7b0075e8 test.hard

[root@oldboy ~]# md5sum oldboy.log

dec2b3d503edf5ff761edffe59d4cefa oldboy.log

[root@oldboy ~]# md5sum test.txt > md5.log

[root@oldboy ~]# cat md5.log

ba2f0f3909a4fb539a5dd22e7b0075e8 test.txt

[root@oldboy ~]# md5sum -c md5.log

test.txt: OK

[root@oldboy ~]# echo "11" >> test.txt

root@oldboy ~]# md5sum -c md5.log

test.txt: FAILED

md5sum: WARNING: 1 of 1 computed checksum did NOT match

* 【注意】

链接文件的MD5值是相同的

### 2.4.16 ssh

* 【功能】远程连接工具 可以实现安全的登录到服务器上
* 【常用参数】

-1：使用ssh协议的版本为1

-2：使用ssh协议的版本为2

-4：使用IPV4寻址

-6：使用IPV6寻址

-A：开启认证代理转发功能

-a：关闭认证代理转发功能

-b：使用本机指定地址作为对应连接的源ip地址

-C：请求压缩所有数据

-F：指定ssh的配置文件

-f：在后台执行ssh命令

-g：允许远程主机连接主机的转发端口

-i：指定身份文件

-p：指定远程服务器的端口

-q：静默模式

-l：指定连接远程服务器登录用户名

### 2.4.17 tee

* 【功能】输出到文件，并且输出到标准输出（也可以理解为既把内容输出到文件中，还把内容输出到屏幕上）
* 【常用参数】

-a：向文件重定向时，使用追加模式（即不会覆盖原有的数据）

-i：忽略中断信号

* 【举例说明】

[root@oldboy ~]# echo stu{01..10}|tee a

stu01 stu02 stu03 stu04 stu05 stu06 stu07 stu08 stu09 stu10

[root@oldboy ~]# cat a

stu01 stu02 stu03 stu04 stu05 stu06 stu07 stu08 stu09 stu10

### 2.4.18 tr

* 【功能】转换或者删除字符串，注意，tr后不能直接跟文件，应该‘<’文件
* 【常用参数】

-d：删除

* 【应用举例】

[root@oldboy ~]# cat person.txt

101,oldboy,CEO

102,zhangyao,CTO

103,Alex,COO

104,yy,CFO

105,feixue,CIO

[root@oldboy ~]# tr '[a-z]' '[A-Z]' <person.txt 转换字符串

101,OLDBOY,CEO

102,ZHANGYAO,CTO

103,ALEX,COO

104,YY,CFO

105,FEIXUE,CIO

[root@oldboy ~]# tr -d '0' <person.txt 删除0

11,oldboy,CEO

12,zhangyao,CTO

13,Alex,COO

14,yy,CFO

15,feixue,CIO

[root@oldboy ~]# tr -d '\t' <person.txt 删除制表符

101,oldboy,CEO

102,zhangyao,CTO

103,Alex,COO

104,yy,CFO

105,feixue,CIO

[root@oldboy ~]# tr -d '\n' <person.txt 删除换行符

101,oldboy,CEO102,zhangyao,CTO103,Alex,COO104,yy,CFO105,feixue,CIO[root@oldboy ~]#

### 2.4.19 dos2unix（需要事先进行安装）

* 【功能】将windows下的文件格式转换为linux下的文件格式
* 【常用参数】

-k：保持输出文件的日期不变

-q：不提示任何警告信息（俗称安静模式）

-c：转换模式 默认时ASCII

-o：写入到源文件

-n：写入到新文件

-V：显示版本信息

* 【应用举例】

dos2unix -k file

dos2unix -k file1 file2 file3

dos2unix -k -o file1 file2 file3

dos2unix -k -n oldfile newfile

## 2.5 系统相关的命令

### 2.5.1 uname

* 【功能】 打印系统信息
* 【常用参数】

-a：显示系统的详细信息

-m：显示系统是32位的还是64位的

-n：显示主机名

-r：显示内核版本

-v：显示操纵系统的版本

### 2.5.2 runlevel

* 【功能】 显示当前用户的运行级别

### 2.5.3 init

* 【功能】 切换用户的运行级别
* 【常用格式】 init 数字 切换运行级别

### 2.5.4 history

* 【功能】 显示历史命令
* 【常用参数】

-c：清空所有历史命令

-d：清空某条具体的命令信息

### 2.5.5 shutdown

* 【功能】 关机
* 【常用格式】

shutdown -h 0/now shutdown -r 0/now

### 2.5.6 halt

* 【功能】 关机 相当于init 0

### 2.5.7 chkconfig

* 【功能】 检查、设置系统的各种服务
* 【常用参数】

--add：增加所指定的系统服务

--del：删除所指定的系统服务

--level<级别号>：指定服务需要在那个级别开启或者关闭

--list：查看自启动服务，默认显示所有

--list<服务名>：显示指定的服务

### 2.5.8 last

* 【功能】显示用户最近登录的信息，他会读取/var/log/wtmp的文件，并把该文件记录登入用户的名单全部显示出来
* 【常用参数】

-a：将从何处登入系统的主机名称或IP，显示在最后一行

-d：将ip转化为主机名称

-f<记录文件>：指定记录文件

-n：设置列出名单的行数

-R：不显示登录系统的主机名或者IP地址

-x：显示系统关机，重新开机，以及执行等级改变的信息

### 2.5.9 lastb

* 【功能】显示用户错误的登录列表，该命令可以发现系统的登录异常，会读取/var/log/btmp的文件，并把文件内记录的登录失败的用户名单全部列出来
* 【常用参数】

-a：把从何处登录的主机名或IP显示在最后一行

-d：将IP地址转化为主机名称

-n：设置列出名单的行数

-R：不显示登录系统的主机名或IP地址

-x：显示系统关机，重新开机，以及执行等级改变的信息

-f<记录文件>：指定记录文件

### 2.5.10 lastlog

* 【功能】显示所有用户最近一次的登录信息，读取的文件是/var/log/lastlog
* 【常用参数】

-b<天数>：显示指定天数前的登录信息

-h：显示帮助信息

-t<天数>：显示指定天数以来的登录信息

-u<用户>：显示指定用户的最近登录信息

### 2.5.11 w

* 【功能】显示已经登录用户的列表，并显示用户正在执行的指令，单独执行w命令时会显示所有用户，后加用户时，显示某个用户的
* 【常用参数】

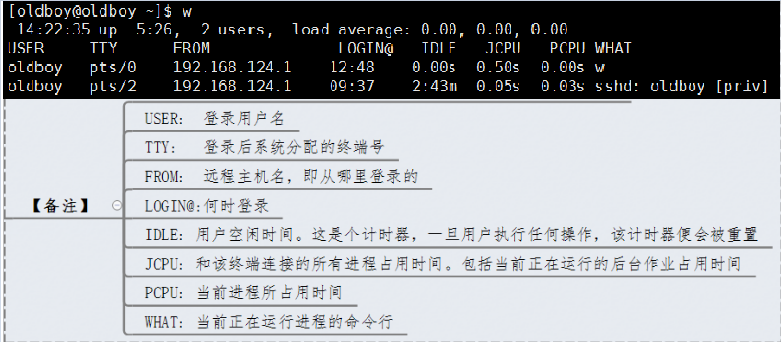
-h：不显示标题信息

-u：列出当前进程和CPU时忽略用户名，主要用于执行su命令后的情况

-s：简洁模式，不显示用户登录时间、终端阶段作业、和程序所消耗的CPU时间

-V：显示版本信息

* 【图解】



### 2.5.12 du

* 【功能】查看文件和目录的大小
* 【常用参数】

-h：以人性话的格式的显示，K\M\G为单位

-s：显示总计，只列出最后相加的总值

### 2.5.13 date

* 【功能】显示系统的时间
* 【常用参数】

-s：设置当前时间，只有root用户才能设置

-d：显示字符串描述的时间，而不是现在的时间

* 【常用标记】

%y：年 %m：月 %d：日

%H：时 %M：分 %S：秒

%w：周

%X：时分秒 格式为：07:45:02 PM

%x：年月日 格式为：12/12/2017

%F：年月日 格式为：2017-12-12

%D：直接显示日期 格式为：12/12/17

* 【注意】

date -s 设置时间时若

后面只加日期（即年月日）时，会把时间（即时分秒）设置为00:00:00

后面只加时间时，则日期不会变，变得只是具体的时间

* 【常用举例】

[root@oldboy ~]# date

Mon Dec 11 08:40:35 CST 2017

[root@oldboy ~]# date +%y-%m-%d

17-12-11

[root@oldboy ~]# date +%F

2017-12-11

[root@oldboy ~]# date +%w

1

[root@oldboy ~]# date +%F

2017-12-11

[root@oldboy ~]# date

Mon Dec 11 20:52:47 CST 2017

[root@oldboy ~]# date +%y-%m-%d

17-12-11

[root@oldboy ~]# date +%x

12/11/2017

[root@oldboy ~]# date +%X

08:57:17 PM

[root@oldboy ~]# date +%H:%M:%S

20:57:44

[root@oldboy ~]# tar zcvf oldboy\_$(date +%F).tar.gz ./oldboy

[root@oldboy ~]# tar zcvf test\_`date +%F`.tar.gz test.txt

显示前后几天

[root@oldboy ~]# date +%F -d "-1day"

2017-12-10

[root@oldboy ~]# date +%F -d "-2day"

2017-12-09

[root@oldboy ~]# date +%F -d "+2day"

2017-12-13

### 2.5.14 netstat

* 【功能】显示linux系统的网络状态信息
* 【常用参数】

-a：显示所有选项（显示所有连线中的Socket）

-c：持续列出网络状态

-t：仅显示tcp相关的选项

-u：仅显示udp相关的选项

-n：拒绝显示别名 能显示数字的全部转化为数字

-l：仅列出又在监听服务状态的

-p：显示建立相关连接的程序名

-r：显示路由信息 路由表

-e：显示扩展信息

-s：按各个协议进行统计

### 2.5.15 fdisk

* 【功能】观察硬盘的实际使用情况，也可对硬盘进行分区
* 【常用参数】

-l：列出磁盘的分区表状况

-s<分区编号>：将指定分区的大小输出到屏幕上

-v：显示版本信息

* 【实例】

[root@oldboy ~]# fdisk /dev/sdb

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): m

Command action

a toggle a bootable flag

b edit bsd disklabel 删除选择的分区

c toggle the dos compatibility flag

d delete a partition

l list known partition types

m print this menu

n add a new partition 建立新的磁盘分区

o create a new empty DOS partition table

p print the partition table 列出磁盘目前分区的情况

q quit without saving changes 退出但是不保存 此时只是在内存中进行的操作

s create a new empty Sun disklabel

t change a partition's system id

u change display/entry units

v verify the partition table

w write table to disk and exit 对分区信息进行保存

x extra functionality (experts only)

[root@oldboy ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): n

Command action

e extended 创建扩展分区

p primary partition (1-4) 创建主分区

[root@oldboy ~]# fdisk /dev/sdb

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): n

Command action

l logical (5 or over) 建立逻辑分区

p primary partition (1-4)

### 2.5.16 free

* 【功能】查看内存的情况
* 【常用参数】

-m：以MB为单位显示内存的情况

-b：以字节为单位显示内存的情况

-k：以KB为单位显示内存的情况

-o：不显示缓冲区调节列

-s<间隔秒数>：持续显示内存的情况

-t：会多显示一行内存的总和

-V：显示版本信息

### 2.5.17 partprobe

* 【功能】用于重读分区表（也可以理解为让分区表的变化通知内核）
* 【常用参数】

-d：不更新内核

-s：显示摘要和分区信息

-h：显示帮助信息

-v：显示版本信息

### 2.5.18 mkfs

* 【功能】创建linux文件系统，mkfs本身并不执行建立linux文件系统的工作，而是调用相关的程序来执行
* 【常用参数】

-t<文件系统>：指定建立文件系统类型

-c：在制作文件系统前，检查partition是否有坏轨

### 2.5.19 mkswap

* 【功能】创建swap交换分区
* 【常用参数】

-c：建立交换分区前，先检查是否有坏的区块

-f：在SPARC电脑上建立交换分区时，需要加上该参数

-v1：建立新式交换分区

-v0：建立旧式交换分区，此选项为预设值

### 2.5.20 swapon

* 【功能】启用swap分区
* 【常用参数】

-a：将/etc/fstab文件中所有设置为swap的设备，启用为交换分区

-p<优先顺序>：指定交换分区的优先顺序

-s：显示交换分区的信息

-V：显示版本信息

-h：显示帮助信息

### 2.5.21 swapoff

* 【功能】关闭swap分区
* 【常用参数】

-a：将/etc/fstab文件中所有设置为swap的设备，全部关闭

-h：显示帮助信息

-V：显示版本信息

### 2.5.22 mount

* 【功能】挂载磁盘
* 【常用参数】

-l：显示已加载的文件系统列表 实际列出的就是/etc/mtab文件中的内容

-a：加载/etc/fstab文件中描述的所有文件系统

-r：将文件系统加载为只读模式

-n：加载没有写入/etc/mtab中的文件系统

-v：显示执行的详细过程

-h：像是帮助信息

-V：显示版本信息

### 2.5.23 umount

* 【功能】卸载挂载的磁盘
* 【常用参数】

-a：卸载/etc/fstab文件中描述的所有文件系统

-n：卸载时不要将信息写入到/etc/mtab文件中

-t<文件系统类型>：仅删除指定的文件系统

-r：若无法成功卸载，则尝试已只读的方式重新挂入文件系统

-v：显示执行的详细过程

-V：显示版本

-h：显示帮助信息

### 2.5.24 df

* 【功能】显示磁盘空间的信息
* 【常用参数】

-a：显示全部的文件系统

-h：以人类可读的方式信息磁盘空间信息

-H：等于-h，但是是以1000为单位进行计算的

-i：显示inode信息

-l：仅显示本地的文件系统

-t<文件系统类型>：仅显示指定文件系统类型的磁盘信息

-x<文件系统类型>：不要显示指定文件系统类型的磁盘信息

-T：显示文件系统的类型

--no-sync：在取得磁盘使用信息前，不要执行sync指令 ，此选项为预设值

--sync：在取得磁盘使用信息前，先执行sync指定

--help：显示帮助信息

--version：显示版本信息

### 2.5.25 fsck

* 【功能】检查并尝试修复文件系统中的错误
* 【常用参数】

-a：自动修复文件系统不询问任何问题

-A：检查/etc/fstab文件中的全部文件系统

-N：不执行，仅列出会实际执行会执行的动作

-P：当搭配-A时，会同时检查所有的文件系统

-s：按照顺序执行检查，而不是同时进行检查

-r：在检查的时候采用交互式模式

-R：当搭配-A是，会略过/下的文件系统不检查

-t<文件系统类型>：指定要检查的文件系统类型

-T：再检查的时候，不显示标题信息

-V：显示详细的执行过程

注意：在磁盘未损坏之前不要轻易的去检查磁盘

### 2.5.26 parted

* 【功能】对磁盘进行分区（通常磁盘大于2T时，采用parted分区）
* 【常用参数】

-i：交互模式 默认就是

-h：显示帮助信息

-s：脚本模式。不提示用户

-v：显示版本信息

### 2.5.27 tune2fs

* 【功能】修改文件系统的信息
* 【常用参数】

-c：调整最大的加载次数

-i：设置相邻两次文件系统检查间隔的时间

-l：显示超级块信息

-f：强制执行修改，即使发生错误

-o：设置或清除文件系统加载的特性或选项

-r：设置文件系统保留块的大小

-j：为ext2文件系统添加日志功能

-u：设置可以使用文件系统保留块的用户

-e：设置内核代码监测到错误的行为

### 2.5.28 resize2fs

* 【功能】调整文件系统的大小（一般用于;VM、drdb的扩容），如果文件系统在挂载的情况下，他只能进行扩容，不能减小
* 【常用参数】

-d：打开调试特性

-f：强制执行，忽略安全检查

-F：在调整文件系统的大小前，刷新文件系统设备的缓冲区

## 2.6 用户相关的命令

### 2.6.1 useradd

* 【功能】 创建linux中新的系统用户
* 【常用参数】

-u<uid>：指定用户的uid

-g<群组>：指定用户所属的群组

-G<群组>：指定用户所属的附加群组

-c<备注>：加上备注，备注文字会保存在/etc/passwd配置文件中的备注栏位上

-d<目录>：指定用户登入时的起始目录

-s<shell>：指定用户登入后所使用的shell

-r：建立系统账号

-n：取消建立以用户名称为名的群组

-m：自动检录用户的登入目录，没的话进行创建 常和-d参数一块使用

-M：不要自动建立用户的登入目录

### 2.6.2 userdel

* 【功能】删除指定用户，以及用户相关的文件，默认不加选项只删除用户，而不删除相关文件
* 【常用参数】

-f：强制删除用户，即使当期用户已登录

-r：删除用户的同时，删除所有与用户相关的文件

* 【注意】

删除用户时，在创建相同名字的账户时，如果出现下面的错误，只需删除/var/spool/mail与账户同名的文件即可

[root@oldboy ~]# useradd hello -g incahome

Creating mailbox file: File exists

[root@oldboy ~]# cd /var/spool/

[root@oldboy spool]# ls

abrt anacron cron mail postfix

abrt-upload at lpd plymouth

[root@oldboy spool]# cd mail/

[root@oldboy mail]# ls

hello oldboy oldgirl root test testt

[root@oldboy mail]# \rm hello

### 2.6.3 usermod

* 【功能】修改用户的基本信息，但是不允许修改正在线上使用的用户
* 【常用参数】

-g<群组>：修改用户所属的群组

-G<群组>：修改用户所属的附加组

-c<备注>：修改用户账户的备注文字

-d<目录>：修改用户登录时的目录

-s<shell>：修改用户登录后的shell

-u<uid>：修改用户的uid

-L：锁定用户密码，使用户的密码无效

-U：解除密码的锁定

### 2.6.4 groupadd

* 【功能】创建一个新的工作组，该工作组的信息将被添加到系统文件中
* 【常用参数】

-g：指定新建组的组id

-r：创建系统工作组，系统工作组的gid小于500

-o：允许添加组id不唯一的工作组

-k：覆盖配置文件“/etc/login.defs”

### 2.6.5 groupdel

* 【功能】删除指定的工作组，若有用户在该群组中，则必须要先删除该用户才能删除该组

### 2.6.6 chown

* 【功能】改变某个用户的所有者和所属组，只有文件的属主和超级用户才能使用该命令
* 【常用参数】

-R：递归处理，将指定目录下的所有文件及子目录一并处理

-f：不显示错误信息

-v：显示执行命令的过程

--version：显示版本信息

--help：显示在线帮助信息

--reference=<参考文件或目录>：把指定文件或目录的拥有者与所属组全部设成和参考文件的相同

-h：只对符号连接的文件进行修改，而不更改其他任何相关的文件

### 2.6.7 chmod

* 【功能】更改文件或者目录的权限，但是符号连接的属性无法进行修改，要想修改符号连接，只需修改其被指向的源文件即可
* 【常用参数】

-R：递归处理，将目录下面所有文件及其子目录一并处理

-v：显示执行的过程

--reference=<参考文件或目录>：把指定文件或目录的拥有者和所属组全部设成和参考文件的相同

-f：不显示错误信息

【权限范围的表示】

u：表示属主

g：表示属组

o：表示其它

a：表示所有

r：可读权限 4

w：可写权限 2

x：可执行权限 1

-：无权限 0

s：特殊权限说明

### 2.6.8 umask

* 【功能】设置限制新文件权限的掩码 系统默认的是0022
* 【常用参数】

-S：以符号方式输出权限掩码

-p：输出的权限掩码可直接作为指令来执行

### 2.6.9 chattr

* 【功能】改变文件的属性
* 【常用参数】

-R：递归处理

-V：显示指令执行过程

-v：显示版本信息

+：开启文件或目录的该项属性

-：关闭文件或目录的该项属性

=：指定文件或目录的该项属性

* 【常用属性】

i：不得更改文件或目录

### 2.6.10 lsattr

* 【功能】显示文件的特殊属性
* 【示例】

[root@oldboy ~]# lsattr test.txt

-------------e- test.txt 代表普通文件

### 2.6.11 id

* 【功能】显示用户的信息
* 【常用参数】

-g：显示用户所属组id

-G：显示用户的附加组id

-u：显示用户的uid

--help：显示帮助信

--version：显示版本信息

### 2.6.12 chage

* 【功能】修改密码和帐号的有效期限
* 【常用参数】

-m：密码可更改的最小天数，为0时代表任何时间都可以更改密码

-M：密码保持有效的最大天数

-W：密码到期前，提前收到警告信息的天数

-E：账号到期的日期，过了这天，账号将不可以在使用

-d：上一次更改的日期

-l：列出账户信息的设置

-I：停滞时期，如果一个密码已过期这些天，那么此帐号将不可以在使用

* 【举例】

要求oldboy用户7天内不能修改密码，60天以后必须修改密码，过期前10天通知oldboy

用户，过期30天后禁止用户登录

[root@oldboy ~]# chage -m 7 -M 60 -W 10 -I 30 oldboy

[root@oldboy ~]# chage -l oldboy

Last password change : Nov 21, 2017 #11月21密码修改

Password expires : Jan 20, 2018 #1月20密码过期

Password inactive : Feb 19, 2018 #2月19密码被禁用

Account expires : never

Minimum number of days between password change : 7 #7天之内禁止修改密码

Maximum number of days between password change : 60 #60天以后必须修改密码

Number of days of warning before password expires : 10 #密码过期前10天开始提醒

### 2.6.13 su

* 【功能】用于用户身份的切换
* 【常用参数】

-：切换用户身份时 同时加载环境变量（如果不加-的话，则不能加载环境变量）

-c<指令>：执行完该指令后恢复原来的用户身份

* 【举例】

[root@oldboy ~]# su - oldboy

[oldboy@oldboy ~]$ pwd

/home/oldboy #加-时 带有环境变量

[oldboy@oldboy ~]$ su -

Password:

[root@oldboy ~]# su oldboy

[oldboy@oldboy root]$ pwd

/root #不加-时 不带有环境变量

[root@oldboy ~]# su - oldboy

[oldboy@oldboy ~]$ su -c ls root

Password:

????.bash\_history oldboy

### 2.6.17 sudo

* 【功能】用来以其他身份来执行命令 默认是以root身份来执行
* 【常用参数】

-l：列出目前用户可以执行的命令

-h：显示帮助信息

-b：在后台执行命令

-k：结束密码的有效期限，即下次在执行是需要输入密码（默认有效期限是 5分钟）

-V：显示帮助信息

### 2.6.18 visudo

* 【功能】编辑sudo的配置文件 具有检查语法的功能

sudo的配置文件是 /etc/sudoers

### 2.6.19 chpasswd

* 【功能】批量更改用户的工具
* 【常用参数】

-e：输入后的密码是加密后的密文

-h：显示帮助信息

-m：当被支持的密码未被加密时，使用MD5加密代替DES加密

* 【用户密码文件的格式】

username:passwd

注意：前提是用户必须是事先存在的，并且不能有空行，然后执行chpasswd < 用户密码文件

【实例】

写一个脚本批量创建20个用户，并未其添加随机的密码

################################第1种方法#####################################

for n in `seq 3 4`

do

useradd user$n

echo "user$n:user$n`echo user$n|md5sum|cut -c5-9`"|chpasswd

done

################################第2种方法#####################################

path=/tmp/guocheng

for n in `seq 5 6`

do

useradd user$n

done

for n in `seq 5 6`

do

echo "user$n:user$n`echo user$n|md5sum|cut -c5-9`">>$path

done

/usr/sbin/chpasswd < $path

### 2.6.20 passwd

* 【功能】用来设置用户的认证信息，包括用户密码、密码过期时间等（主要是用来设置用户密码的），普通用户只能修改自己的密码，且在修改密码时需要提供老密码，超级用户就不需要
* 【常用参数】

-d：删除用户密码，只用系统管理员才能使用该选项

-f：强制执行

-l：锁住密码

-u：解开已经上锁的用户

-S：列出密码的相关信息 只有系统管理员才能使用该选项

-k：设置只有在密码过期失效后，才能进行更新密码

【举例说明】

[root@oldboy ~]# passwd -d user3

Removing password for user user3.

passwd: Success

[root@oldboy ~]# passwd -l user4

Locking password for user user4.

passwd: Success

[root@oldboy ~]# passwd -S user5

user5 PS 2018-10-14 0 99999 7 -1 (Password set, SHA512 crypt.)

[root@oldboy ~]# passwd -u user4

Unlocking password for user user4.

passwd: Success

[root@oldboy ~]# passwd -k user4

Changing password for user user4.

Changing password for user4.

(current) UNIX password:

## 2.7 网络相关的命令

### 2.7.1 ifconfig

* 【功能】查看网络接口信息及修改网卡的相关配置
* 【用法】ifconfig 【设备名】 【选项参数】
* 【常用参数】

up：启动指定网络设备/网卡

down：关闭指定网络设备/网卡

-a：无论是否激活，显示所有网络接口

add：给指定网卡配置地址

del：删除指定网卡的地址

arp：打开ARP协议

arp：关闭ARP协议

mtu<字节数>：设置网卡的最大传输单元

netmask<子网掩码>：设置网卡的子网掩码

broadcast：设置广播地址

* 【应用举例】

##############################启用或者关闭网卡#################################

[root@oldboy ~]# ifconfig 设备名 up

[root@oldboy ~]# ifconfig 设备名 down

############################修改网卡的MAC地址###############################

[root@oldboy ~]# ifconfig 设备名 hw ether 00:AA:BB:CC:dd:EE

##################################配置IP地址##################################

[root@oldboy ~]# ifconfig eth1 192.168.124.130

[root@oldboy ~]# ifconfig eth1 192.168.124.130 netmask 255.255.255.0

[root@oldboy ~]# ifconfig eth1 192.168.124.130 netmask 255.255.255.0 broadcast 192.168.124.255

###############################启用或关闭arp协议##############################

[root@oldboy ~]# ifconfig eth1 arp

[root@oldboy ~]# ifconfig eth1 -arp

###############################设置最大的传输单元##############################

[root@oldboy ~]# ifconfig eth1 mtu 1500

################################设置网卡别名###################################

[root@oldboy ~]# ifconfig eth1:0 10.0.0.1/8 up

[root@oldboy ~]# ifconfig eth1:0 10.0.0.1/8 down

注意：ifconfig设置的东西只要网卡一重启就没了

### 2.7.2 ip

* 【功能】显示或操作linux下主机路由、网络设备等，是linux下强大的网络工具
* 【常用参数】

-s：显示更详细的信息

-f：强制使用指定的协议簇

-0：输出信息每条记录输出一行，即使内容较多，也不换行显示

-r：显示主机时，不使用ip地址，而使用主机域名

-4：强制使用ipv4协议

-6：强制使用ipv6协议

* 【应用举例】

############################设置或删除IP及查看IP##############################

[root@oldboy ~]# ip addr add 10.0.0.1/8 dev eth1

[root@oldboy ~]# ip addr del 10.0.0.1/8 dev eth1

[root@oldboy ~]# ip addr show eth1

#####################################显示路由信息##############################

[root@oldboy ~]# ip route show

[root@oldboy ~]# ip r s

[root@oldboy ~]# ip route

[root@oldboy ~]# ip route list

################################显示网络的运行状态#############################

[root@oldboy ~]# ip link list

[root@oldboy ~]# ip -s link list 显示的更加详细

##################################显示邻居表信息###############################

[root@oldboy ~]# ip neigh list

### 2.7.3 route

* 【功能】设置并显示路由表
* 【用法】route [选项] [参数]
* 【常用选项】

-C：显示路由缓存

-n：不使用主机名称，而直接使用IP

-ee：显示更详细的信息

-e：netstat格式显示路由表

-net：到一个网络的路由

-host：到一个主机的路由

* 【常用参数】

add：增加指定的路由记录

del：删除指定的路由记录

target：目的网络或目的主机

netmask：设置掩码

gw：设置网关

metric：设置路由跳数

### 2.7.4 dig

* 【功能】用来查询域名
* 【常用参数】

@<服务器名>：指定DNS（像8.8.8.8）

-t<类型>：指定要查询的DNS数据类型

-x：执行逆向查询（即由IP查）

-b<ip地址>：当主机具有多个ip时，指定使用本机的那个ip进行查询

* 【应用举例】

[root@oldboy ~]# dig @8.8.8.8 baidu.com +trace

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.62.rc1.el6 <<>> @8.8.8.8 baidu.com +trace

; (1 server found)

;; global options: +cmd

. 3843 IN NS c.root-servers.net.

. 3843 IN NS k.root-servers.net.

. 3843 IN NS l.root-servers.net.

. 3843 IN NS j.root-servers.net.

. 3843 IN NS i.root-servers.net.

. 3843 IN NS b.root-servers.net.

### 2.7.5 nslookup

* 【功能】用来查询域名
* 【常用选项】

-sil：不显示任何警告信息

* 【应用举例】

[root@oldboy ~]# nslookup -sli www.baidu.com

\*\*\* Invalid option: sli

Server: 192.168.124.2

Address: 192.168.124.2#53

Non-authoritative answer:

www.baidu.com canonical name = www.a.shifen.com.

Name: www.a.shifen.com

Address: 61.135.169.125

Name: www.a.shifen.com

Address: 61.135.169.121

[root@oldboy ~]# nslookup

> baidu.com

Server: 192.168.124.2

Address: 192.168.124.2#53

Non-authoritative answer:

Name: baidu.com

Address: 220.181.57.217

Name: baidu.com

Address: 111.13.101.208

Name: baidu.com

Address: 123.125.114.144

### 2.7.6 tcpdump

【功能】一款强大的抓包工具

【常用命令】

#抓取所有经过 eth0，目的或源地址是 192.168.29.162 的网络数据

命令：tcpdump -n -i eth0 host 192.168.29.162

# 源地址

命令：tcpdump -i eth1 src host 192.168.29.162

# 目的地址

命令：tcpdump -i eth1 dst host 192.168.29.162

#抓取当前服务器eth0网卡端口8080的网络数据

命令：tcpdump -n -i eth0 port 8080

#抓取mysql执行的sql语句

命令：tcpdump -i eth1 -s 0 -l -w - dst port 3306 | strings

#抓取mysql通讯的网络包(cap用wireshark打开)

命令tcpdump -n -nn -tttt -i eth0 -s 65535 'port 3306' -w 20160505mysql.cap

#抓取SMTP 数据

命令：tcpdump -i eth1 'tcp[tcpflags] & tcp-syn != 0 and tcp[tcpflags] & tcp-ack != 0'

#抓取HTTP GET数据，"GET "的十六进制是 47455420

命令：tcpdump -i eth1 'tcp[(tcp[12]>>2):4] = 0x47455420'

#抓取SSH返回，"SSH-"的十六进制是 0x5353482D

命令：tcpdump -i eth1 'tcp[(tcp[12]>>2):4] = 0x5353482D'

#实时抓取端口号8080的GET包，然后写入GET.log

命令：tcpdump -i eth0 '((port 8080) and (tcp[(tcp[12]>>2):4]=0x47455420))' -nnAl -w /tmp/GET.log

#抓取指定SYN个数，-c 参数指定抓多少个包。

命令：time tcpdump -nn -i eth0 'tcp[tcpflags] = tcp-syn' -c 10

## 2.8 系统常用开关机重启以及注销常用的命令

【重启命令】

reboot 生产中常用

【关机命令】

shutdown -r now/0 关机 生产中常用

halt 立即停止系统 生产中常用

【注销命令】

logout

exit

ctrl +d 生产中常用

## 2.9 备注

vi，vim，三剑客，find 命令详解见别处

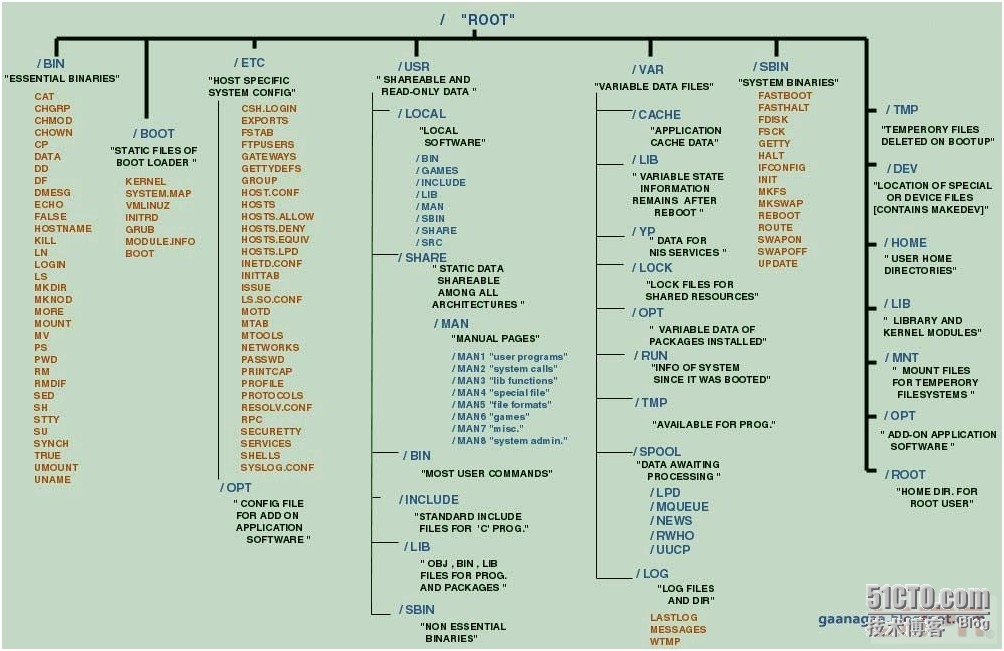
# 第3章 linux基础入门中部

## 3.1 linux目录的基础知识

### 3.1.1 linux目录的特点

* / 是所有目录的顶点 （linux一切皆文件）
* 每个目录下面都可以挂载设备or磁盘
* 目录结构像是一个倒挂的树
* 目录和磁盘分区是没有关联的
* / 下不同的目录可能会对应不同的分区或者磁盘
* 所有的目录都是按照一定的类别有规律的组织和命名的
* linux下的设备不挂载，无法使用（不挂载的设备or磁盘就像是个无出口的监狱）

### 3.1.2 linux树状目录结构图



### 3.1.3 linux常用目录详解

#### 3.1.3.1 / 目录

|  |  |
| --- | --- |
| 目录 | 功能描述 |
| / | 所有目录的顶点 |
| /bin/ | 常用二进制命令所在的目录，可用which命令进行查看 |
| /boot/ | linux内核和系统引导程序所需的文件目录 |
| **/dev/** | 设备文件目录，例如声卡、网卡等等 |
| **/etc/** | 二进制软件安装包的配置文件的默认路径（配置文件的默认安装路径） |
| /etc/opt/ | /opt/的配置文件 |
| /etc/X11/ | X\_Window系统的配置文件 |
| **/home** | 普通用户的家目录，包含保存文件的文件、个人设置等，一般为单独的分区 |
| /lib/ | 库文件存放目录 |
| /mnt/ | 一般是用于挂载临时储存设备的挂载目录 |
| /sbin/ | root用的命令存放的目录 |
| /tmp/ | 临时文件存放目录，在**系统重启时该目录下文件不会被保留** |
| /var/ | 可变化的目录，一般用于存放系统产生的日志 |
| /var/log/messges | 记录系统产生的日志 |
| /var/log/secure | 显示系统的登录信息 |
| /media/ | 可移除媒体的挂载点 |
| /opt/ | 可选应用软件包 |
| /lost+found/ | 在ext3文件系统中，当系统意外崩溃或意外关机时，会产生一些文件碎片在这里，当系统在开机启动的过程中fsck工具会检查这里，并修复需要用手工的方式来修复，或移到文件原来的位置上 |
| /proc/ | 虚拟文件系统，将内核与进程状态归档为文本文件 |
| /var/log/clientmqueue | 邮件临时目录 |

#### 3.1.3.2 /etc/ 目录

|  |  |
| --- | --- |
| 目录 | 功能描述 |
| **/etc/rc.d /etc/rc\*.d** | **启动或改变运行级时运行的scripts目录** |
| **/etc/hosts** | **本地域名解析文件** |
| **/etc/sysconfig/network/** | **网络的配置文件** |
| **/etc/resolv.conf** | **linux本地客户端的DNS配置文件** |
| **/etc/fstab** | **开机自动挂载系统文件** |
| **/etc/inittab** | **设定系统启动时init进程把系统设置成什么级别的runlevel及加载相关的启动配置文件** |
| **/etc/exports** | **设置NFS系统用的配置文件路径** |
| **/etc/init.d** | **该目录用来存放系统启动脚本** |
| **/etc/profile** | **系统全局环境变量，永久生效的配置文件，定义别名以及PATH变量等** |
| **/etc/issue** | **认证前的输出信息，默认输出版本内核信息** |
| /etc/motd | 设置认证后的输出信息 |
| /etc/securetty | 确认安全终端，即那个终端允许root登录，一般只列出虚拟控制台，这样就不可能通过modem或网络闯入系统并得到超级用户特权 |
| **/etc/skel/** | **默认创建该用户时，把该目录拷贝到家目录下** |
| **/etc/group** | **类似/etc/passwd文件，但是不是用户的而是用户组的** |
| **/etc/passwd** | **用户数据库,里面给了用户名、真实姓名、家目录、加密的口令和用户的其他信息** |
| **/etc/shadow** | **是在安装了影子口令软件的系统上的影子口令文件，影子口令文件将/etc/passwd文件中加密的口令移动到/etc/shadow中，而/etc/shadow只对root用户可读，这使得破译密码变得更难** |
| **/etc/sudoers** | **可以sudo命令的配置文件** |
| /etc/opt | opt的配置文件 |

#### 3.1.3.3 /usr/ 目录

|  |  |
| --- | --- |
| 目录 | 功能描述 |
| /usr/games | 存放这XteamLinux自带的小游戏 |
| /usr/doc | linux技术文档 |
| /usr/include | 用来存放linux开发和编译应用程序所需要的头文件 |
| /usr/lib | 存放一些常用的动态连接共享库和静态档案库 |
| /usr/sbin | 也是可执行程序的目录，但大多存放涉及系统管理的命令，相似的目录是/sbin或/usr/local/sbin等 |
| **/usr/local** | 该目录一般是用来存放用户自编译安装软件的存放目录，一般是通过源码包安装的软件，若没有指定特定目录的话，就默认安装在该目录下面 |
| /usr/local/bin | 用户安装的小的应用程序，和一些在/usr/local目录下大应用程序的符号链接 |
| /usr/local/sbin | 系统全局环境目录，可放置一些不需要加路径执行的脚本等 |
| /usr/share | 系统共用的东西存放地， |
| /usr/src | 内核源码目录的存放路径 |

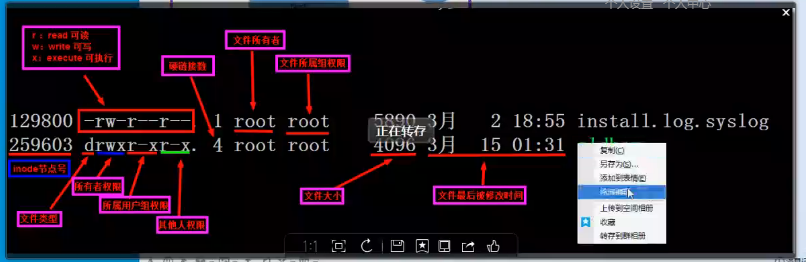
#### 3.1.3.4 /proc 目录

|  |  |
| --- | --- |
| 目录 | 功能描述 |
| /proc/inturrupts | 系统的中断文件 |

## 3.2 linux文件属性

### 3.2.1 linux文件的概述

* linux系统中文件和目录的属性主要包括：索引节点号（inode）、文件类型及权限、硬链接的数量、文件或目录所属的用户（属主）、文件或目录所属的组（属组）、文件的大小、文件的修改时间、实际的文件名或目录名。
* ls -lhi 执行结果的分析



### 3.2.2 inode 索引节点

#### 3.2.2.1 基本特点

* 磁盘（存储设备）被分区格式化为ext4系统后，会生成两部分内容：inode和block（两者都有很多个）。
* inode是用来存放文件的属性的以及作为文件的索引（指向文件的实体），所以inode本身是有大小的，在Centos5系列中inode默认大小是128字节，Centos6系列中inode默认大小是256字节。
* inode除了存放文件的属性信息外，还会存放执行数据块的指针
* block是用来存放实际数据的，常见的一个block的大小为4K，其引导分区为1K,其它的普通分区为4K。
* inode中不包含文件名，因为文件名不属于文件的属性。
* 索引节点号表现形式是一串数字，像每个人的身份证号一样，是唯一的。
* 系统在读取文件的时候，首先通过文件名找到inode号，然后才能读取到文件的内容。
* inode节点号相同的文件，互为硬链接，可以理解为同一个超市的不同入口。
* ext3/ext4 文件系统下，一个文件占用且只能占用一个inode。
* ext3/ext4文件系统下，一个文件被创建后，至少要占用一个inode和一个block。
* 若一个文件很大（如高清片），则它需要占用多个block，若一个文件很小（如只有0.01K）那么它将占用一个block，且剩余的空间就浪费了，即不能在存储其他东西了。

#### 3.2.2.2 查看inode和block的大小

[root@oldboy test]# dumpe2fs /dev/sda3 |egrep -i "inode size|block size"

dumpe2fs 1.41.12 (17-May-2010)

Block size: 4096

Inode size: 256

### 3.2.3 block

* 磁盘读取数据按block为单位进行读取的
* 一个文件可能会占用多个block，但是每读取一个block就会消耗一个磁盘IO
* 若要提升磁盘IO性能，那么就需要一次读取的数据尽可能的多
* 一个block只能存放一个文件，无论内容有多大，假设一个block的大小为4K，那么存放一个1K的文件，剩下的空间就不能在存储数据了，只能浪费掉，
* block并不是越大越好，太大时对于存放小文件会浪费磁盘空间，一般根据业务的文件大小进行选择，一般Centos6默认就是4K
* 可以在格式化的时候改变inode和block的大小
* 格式化的命令：mkfs.ext4 -b 2048 -I 256 /dev/sdb 其中 -b是指定block的大小，-I是指定inode的大小

### 3.2.4 相关的面试题

* 【题目】

磁盘空间满了，写不进去东西，有3种情况。

第一种：block满了，可以用df -h 查看。看Use%，删除一些文件即可

第二种：inode满了，可以用df -i查看，看IUse%，删除一些小的（不用的）文件即可

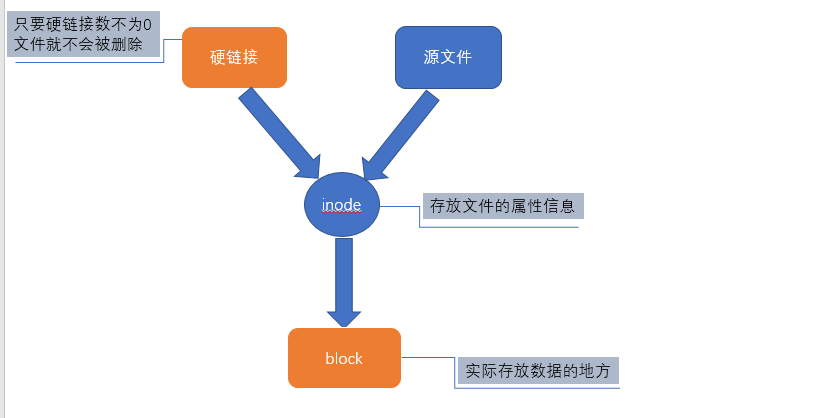
第三种：硬链接数为0，进程调用数不为0

### 3.2.5 linux软硬连接

#### 3.2.5.1 硬链接的创建过程

* ln 源文件 目标文件

#### 3.2.5.2 硬链接的原理图



#### 3.2.5.3 硬连接的特点

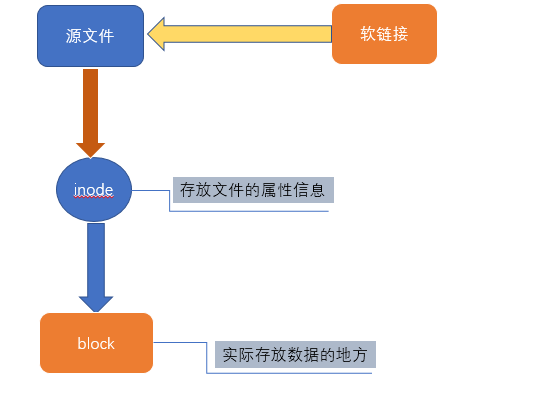
* 多个文件共享同一个索引节点，不占用实际空间
* 硬链接是一个文件的多个入口（可以理解为一个超市具有多个入口一样）
* 只要连接数不为0，文件就存在，所以硬链接可以防止文件误删除
* 删除源文件或硬链接的任意一个，则文件实体并不会被删除，只有同时删除源文件和其所有硬链接，文件实体才会被删除
* 硬链接文件是普通文件，可以用rm命令删除
* 对于静态文件（即没有被进程调用的文件），当i\_link数为0时，文件就被删除了（i\_link的查看方法ls –l 的第三列）

### 3.2.6 软连接

#### 3.2.6.1 软连接的创建过程

* ln -s 源文件 目标文件

#### 3.2.6.2 软连接的原理图

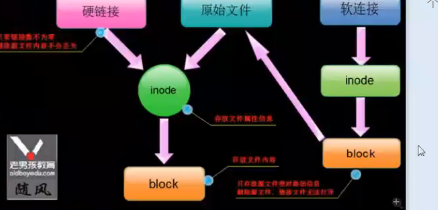


#### 3.2.6.3 软连接的特点

* 软连接类似与windows下的快捷方式（可以通过readlink查看其指向）
* 删除源文件时，软连接仍然存在，但是无法访问指向源文件的内容了
* 软连接失效时一般是白字红底闪烁提示
* 软连接和源文件是不同类型的文件，文件也不同，inode节点号也不同
* 软连接的类型是l，可以用rm命令删除

### 3.2.7 文件连接的总结

* 删除软连接文件对源文件和硬链接文件无任何影响
* 删除硬链接文件对源文件和软连接文件无任何影响
* 删除源文件对硬链接文件无影响，但是会导致软连接文件失效，有白字红底闪烁提示
* 硬链接和源文件具有相同的inode号，可以认为是同一个文件，也可以理解为一个文件具有多个入口
* 软连接和源文件具有不同的inode号，是不同的文件，软连接是源文件的快捷方式，具有源文件的位置指向
* 软硬连接的图形理解



### 3.2.8 目录连接文件的总结

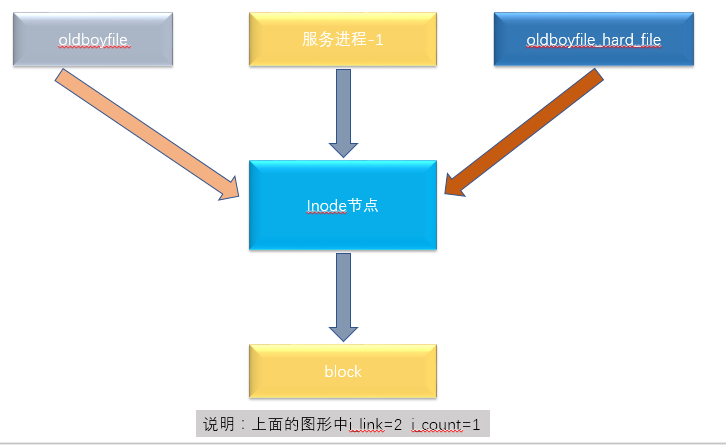
* 目录不可以创建硬链接，但是可以创建软连接
* 目录的硬链接不能跨越文件系统（从硬链接的原理进行理解）
* 每个目录下面都有一个硬链接 **.** 号，和对应上级目录的硬链接 **..**
* 在父目录下创建一个子目录，则父目录的硬链接数加1（因为在子目录里面都有一个 **..** 指向父目录），但是在下面创建文件，父目录的硬链接数不会增加

### 3.2.9 软硬连接的区别

* 硬链接文件和源文件共享一个inode号，软连接文件和源文件具有不同的inode号
* 创建硬链接和软连接的过程不同（创建硬链接 ln 源文件 目标文件 创建软连接 ln -s 源文件 目标文件）
* 删除源文件，硬链接可以访问，软连接不能访问
* 目录可以创建软连接，不能创建硬链接
* 软连接不能修改源文件的名字，硬链接可以修改
* 软连接可以跨系统，硬链接不可以
* 删除软硬连接不影响源文件，删除硬链接，连接数减1

### 3.2.10 linux文件的删除原理

* linux文件的删除由两个东西控制：i\_link（硬链接数）和i\_count（进程的调用数），只有当两者的数值都为0时文件才被删除
* 用rm来删除 使得i\_link为0 重启使用这个文件的进程 使得i\_count为0
* linux删除原理的图解



注意：只有让i\_link和i\_count的值都为0 文件才会被删除

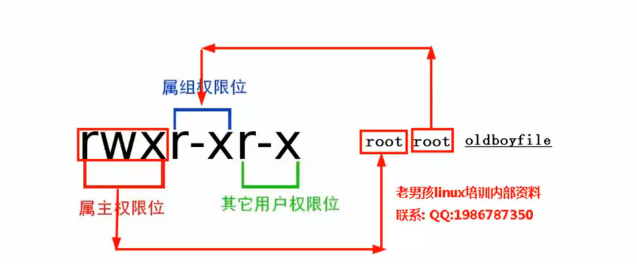
**删除文件的权限是受父目录的控制和文件本身的权限无关**

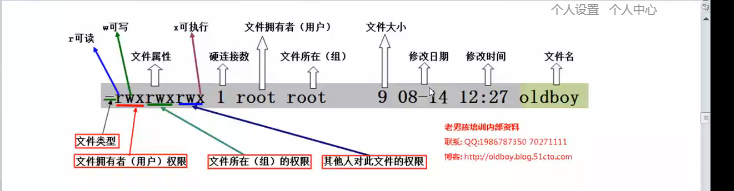
## 3.3 linux文件的权限

### 3.3.1 rwx详解

* r read 4 可读权限
* w write 2 可写权限 即修改文件的内容
* x execute 1 可执行权限 执行命令 运行脚本等
* - 0 没有权限

### 3.3.2 权限位图解





【说明】

* 权限位共有9位
* 前三位是属主（用户）权限位，相当于文件的主人
* 中间三位属组（用户组）权限位，相当于自己的朋友圈
* 后三位是其他权限位，相当于其他人

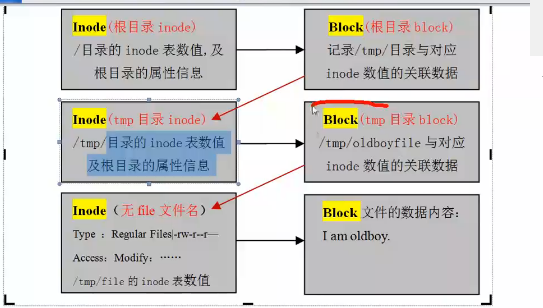
### 3.3.3 linux普通文件权限的读、写、执行的解读

* 可读r：表示具有读取、阅读文件内容的权限
* 可写w：表示具有新增、修改文件容的权限。但是如果没有r权限的配合，在用vi/vim进行编辑的时候，会提示无法编辑，但是可以强制编辑，也可以用echo重定向（或追加）
* 可执行x：表示具有执行文件的权限，但是必须文件要先有执行权限才可以，对于普通用户来说，必须要有r权限的配合文件才可以被执行，而root用户只需x即可

### 3.3.4 目录权限的读、写、执行的解读

* 可读r：表示具有浏览目录下面文件及子目录的权限，但是如果没有x权限，则不能进入到目录中（即不能cd dir），若果没有x权限，用ls进行查看时，显示无权限访问，但是可以看到目录线下面的所有文件名，如果没有x权限，用ls -l进行查看时，所有的属性都是？，并且也会提示无权访问，但是也可以看到目录下面的所有文件名
* 可写w：表示具有在目录下面增加、删除、修改文件名的权限（但是需要x权限的配合）
* 可执行x：表示可以进入到目录的权限（例如cd dir），但是如果没有r权限的配合，则不能列出目录下面的东西，没有w权限的配合，则无法新建和删除文件

### 3.3.5 访问文件的图解说明 以/tmp/oldboyfile为例



### 3.3.6 umak

#### 3.3.6.1 umask的作用

* umask是用来设置限制新文件权限的掩码 系统的默认值是0022

#### 3.3.6.2 文件和目录默认权限的计算

* 文件的默认权限是644 （因为创建文件的最大权限是666）
* 目录的默认权限是755 （因为创建目录的最大权限是777）
* 对于文件来说，用文件的最大权限减去umask值，当umask中有偶数时，直接相减即可，当有奇数时，将相减的结果中的奇数加1即可
* 对于目录来说，用目录的最大权限直接相减即可

#### 3.3.6.3 目录和文件安全的权限

* 目录较安全的权限就是755 文件较安全的权限就是644
* linux默认的权限是安全权限的临界点，在工作中尽量给这个权限，或者低于这这个临界点，但是不要高于

### 3.3.7 suid

#### 3.3.7.1 作用范围

* suid是针对命令和二进制程序的

#### 3.3.7.2 总结

* 用户前三位权限中x的位上如果有s就表示具有suid权限
* 当用户前三位权限中没x执行权限时随，suid权限显示的就是S（大），有的话显示的是s（小）
* suid权限的作用就是让普通用户可以以root（或其他）用户角色运行只有root（或其他）账号才能运行的程序或命令

### 3.3.8 sgid

#### 3.3.8.1 与suid的区别

* sgid即可以针对文件也可以针对目录进行设置

#### 3.3.8.2 总结

* 【对于文件】
* sgid仅对二进制命令程序有效
* 二进制命令或程序需要有可执行权限x
* 执行命令的任意用户可以获得该命令程序执行期间所属组的权限
* 【举例】

普通用户可以使用locate命令

[user1@oldboy ~]$ which locate

/usr/bin/locate

[user1@oldboy ~]$ ll /usr/bin/locate

-rwx--s--x. 1 root slocate 38464 Mar 12 2015 /usr/bin/locate

发现用户组权限中应该出现x的位置上显示s，表示这个命令具有SGID权限只有x权限却没有rw权限

locate这个命令实际上会去访问/var/lib/mlocate/mlocate.db文件

[root@oldboy ~]# ll /var/lib/mlocate/mlocate.db

-rw-r----- 1 root slocate 1328934 Dec 26 09:40 /var/lib/mlocate/mlocate.db

通过上面可知道普通用户对其并没有访问权限，所以理论上普通用户并不能执行这个命令。但是locate这个命令具有SGID权限，所以在运行locate这个命令时，这个命令或得到slocate用户组的权限 相当于把普通用户加入到了这个用户组中 所以普通用户也可以执行该命令

* 【对于目录】
* linux中默认情况下所有用户创建文件，默认用户和组都是自身
* sgid可以让用户在此目录下创建的文件和目录，具有和此目录相同的用户组设置

【举例】

[root@oldboy ~]# mkdir /home/admins -p

[root@oldboy ~]# groupadd adminuser

groupadd: group 'adminuser' already exists

[root@oldboy ~]# ll -ld /home/admins/

drwxr-xr-x 2 root root 4096 Dec 26 10:06 /home/admins/

[root@oldboy ~]# chown .adminuser /home/admins/

[root@oldboy ~]# ll -d /home/admins/

drwxr-xr-x 2 root adminuser 4096 Dec 26 10:06 /home/admins/

[root@oldboy ~]# touch /home/admins/test

[root@oldboy ~]# ll /home/admins/test

-rw-r--r-- 1 root root 0 Dec 26 10:08 /home/admins/test

[root@oldboy ~]# chmod g+s /home/admins/

[root@oldboy ~]# touch /home/admins/oldboy.txt

[root@oldboy ~]# ll /home/admins/oldboy.txt

-rw-r--r-- 1 root adminuser 0 Dec 26 10:09 /home/admins/oldboy.txt

## 3.4 用户

### 3.4.1 用户的分类

* 超级用户 root UID为0
* 普通用户 UID的取值范围为500-65535
* 虚拟用户 UID的取值范围为1-499

### 3.4.2 用户的详解

* 超级用户就像是皇帝一样，可以为所欲为
* 普通用户就像是老百姓，拥有的权利极低，但是可以被root授权（可以被封官），并且可以推翻皇帝自己当家做主（只需将UID改为0即可）
* 虚拟用户就像是傀儡一样，他的作用就是满足系统进程对文件属主的要求
* UID就像是我们的身份证号一样，是全国唯一的，用户名像是我们的姓名一样

### 3.4.3 用户相关的配置文件

* /etc/passwd 用户信息文件
* /etc/shadow 用户密码文件
* /etc/gpasswd 组信息文件
* /etc/gshadoe 组密码文件

### 3.4.4 /etc/passwd 配置文件内容的详解

#### 3.4.4.1 各字段的含义

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| root | :x | :0 | :0 | :root | :/root | :/bin/bash |
| 账号名称 | 账号密码 | 账号UID | 账号GID | 用户说明 | 用户家目录 | shell解释器 |

3.4.4.2 各自段的说明

|  |  |
| --- | --- |
| 字段名称 | 注释说明 |
| 账号名称 | 和UID对应，在系统中唯一 |
| 账号密码 | 早期密码是在这里面存放的，后来为了安全，将其移到了/etc/shadow中，只保留了一个x |
| 账户UID | 账户的UID一般由一串整数控制，取值范围位0-65535，在系统中是唯一的 |
| 账户GID | 账户的GID一般由一串整数控制，取值范围位0-65535 |
| 用户说明 | 该字段是对账号的描述信息 |
| 用户家目录 | 用户登陆后首先进入的目录，一般为/home/用户名这样的目录 |
| shell解释器 | 用户登录后使用shell，Centos默认的shell位bash，若不希望用户具有登录功能，只需修改passwd配置文件，将该项改为/sbin/nologin |

### 3.4.5 删除用户的经验

* 一般不能确认用户相关的目录是否有重要数据，删除时就不能用 -r参数
* 编辑/etc/passwd文件，然后注释掉用户，观察一段时间，这样出问题后可以还原，相当于操作前备份
* 把登录shell修改为/sbin/nologin
* openldap（类似活动目录）账号统一管理的，ldap库中干掉用户，所有服务器中就没该账户的信息了

注意！只要涉及到删除和修改就要谨慎且不要手残

## 3.5 有关用户管理的三个重要目录和文件

### 3.5.1 linux用户管理重要初始化目录 /etc/skel

* 【说明】

/etc/skel/使用来存放新用户环境变量的目录，当我们添加新用户时，这个目录下的所有文件会自动复制到新添加用户的家目录下，默认情况下/etc/skel/目录下的所有文件都是隐藏文件（以**.**开头的文件），通过添加、修改、删除/etc/skel目录下的文件，可以为新创建的用户提供统一的、标准的初始化用户环境

* 【企业环境的作用】
* 可以把通知的内容放到该目录下，让登陆的用户去看
* 统一初始化用户的环境变量
* 【企业面试题】

#####################出现-bash-4.1$ 问题的原因及其解决的方法####################

出现这种情况的原因是：该用户家目录下的环境变量文件丢失

解决方法：将/etc/skel目录下的文件拷贝一份到该用户的家目录下面即可

##################################模拟环境#####################################

[root@oldboy ~]# su - oldboy

[oldboy@oldboy ~]$ ls

????.bash\_history oldboy

[oldboy@oldboy ~]$ ls -a

. .. .bash\_history ????.bash\_history .bash\_logout .bash\_profile .bashrc oldboy .viminfo

[oldboy@oldboy ~]$ pwd

/home/oldboy

[oldboy@oldboy ~]$ \rm -f .bash\*

[oldboy@oldboy ~]$ ls -a

. .. ????.bash\_history oldboy .viminfo

[oldboy@oldboy ~]$ logout

[root@oldboy ~]# su - oldboy

-bash-4.1$

##################################解决方法#####################################

-bash-4.1$ cp -a /etc/skel/.bash\* ./

-bash-4.1$ ls -a

. .. .bash\_history ????.bash\_history .bash\_logout .bash\_profile .bashrc oldboy readme.txt skel .viminfo

-bash-4.1$ logout

[root@oldboy ~]# su - oldboy

[oldboy@oldboy ~]$ ls-a

-bash: ls-a: command not found

[oldboy@oldboy ~]$ ls -a

. .. .bash\_history ????.bash\_history .bash\_logout .bash\_profile .bashrc oldboy readme.txt skel .viminfo

* 【举例】

在/etc/skel/.bash\_profile文件中添加别名alias grep='grep --color=auto'，并在/etc/skel目录下

面创建一个名为Readme.txt的文件，则在创建新用户后，在执行grep命令时 默认过滤出来的内容就会显示颜色，并且在新用户的家目录下还有一个名为Readme.txt的文件。

[root@oldboy skel]# echo "alias grep='grep --color=auto'">>/etc/skel/.bash\_profile

[root@oldboy skel]# tail -1 /etc/skel/.bash\_profile

alias grep='grep --color=auto'

[root@oldboy skel]# touch Readme.txt

[root@oldboy skel]# ls -a

. .. .bash\_logout .bash\_profile .bashrc Readme.txt

[root@oldboy skel]# useradd ceshi

[root@oldboy skel]# su - ceshi

[ceshi@oldboy ~]$ ls -a

. .. .bash\_logout .bash\_profile .bashrc Readme.txt

[ceshi@oldboy ~]$ grep "3306" /etc/services

mysql 3306/tcp # MySQL

mysql 3306/udp # MySQL

### 3.5.2 /etc/login.defs配置文件的详解（了解即可）

* 【说明】

/etc/login.degs是用来定义创建用户时需要的一些用户配置信息。如创建用户时是否需要家目录，UID和GID的范围，用户及密码的有效期限等等

* 【配置文件内容的解释】

[root@oldboy ~]# cat /etc/login.defs

# Please note that the parameters in this configuration file control the

# behavior of the tools from the shadow-utils component. None of these

# tools uses the PAM mechanism, and the utilities that use PAM (such as the

# passwd command) should therefore be configured elsewhere. Refer to

# /etc/pam.d/system-auth for more information.

# \*REQUIRED\*

# Directory where mailboxes reside, \_or\_ name of file, relative to the

# home directory. If you \_do\_ define both, MAIL\_DIR takes precedence.

# QMAIL\_DIR is for Qmail

#QMAIL\_DIR Maildir

MAIL\_DIR /var/spool/mail

#MAIL\_FILE .mail

################################默认密码的有效期###############################

# Password aging controls:

# PASS\_MAX\_DAYS Maximum number of days a password may be used.

# PASS\_MIN\_DAYS Minimum number of days allowed between password changes.

# PASS\_MIN\_LEN Minimum acceptable password length.

# PASS\_WARN\_AGE Number of days warning given before a password expires.

PASS\_MAX\_DAYS 99999

PASS\_MIN\_DAYS 0

PASS\_MIN\_LEN 5

PASS\_WARN\_AGE 7

############################UID、GID的取值范围###############################

# Min/max values for automatic uid selection in useradd

UID\_MIN 500

UID\_MAX 60000

# Min/max values for automatic gid selection in groupadd

GID\_MIN 500

GID\_MAX 60000

# If defined, this command is run when removing a user.

# It should remove any at/cron/print jobs etc. owned by

# the user to be removed (passed as the first argument).

#USERDEL\_CMD /usr/sbin/userdel\_local

# If useradd should create home directories for users by default

# On RH systems, we do. This option is overridden with the -m flag on

# useradd command line.

CREATE\_HOME yes #是否创建家目录

# The permission mask is initialized to this value. If not specified,

# the permission mask will be initialized to 022.

UMASK 077 #创建家目录的UMASK 所以家目录的权限时700

# This enables userdel to remove user groups if no members exist.

USERGROUPS\_ENAB yes

#################################密码加密的方式################################

# Use SHA512 to encrypt password.

ENCRYPT\_METHOD SHA512

### 3.5.3 /etc/default/useradd配置文件的详解

* 【说明】

/etc/default/useradd文件是在使用useradd添加用户时一个需要调用的一个默认的配置文件，可以使用useradd -D或者vi/vim进行编辑。

* 【文件内容的说明】

[root@oldboy ~]# cat /etc/default/useradd

# useradd defaults file

GROUP=100 #依赖于/etc/login.defs USERGROUPS\_ENAB,如果为no则此处控制

HOME=/home #把用户的家目录放在/home目录下

INACTIVE=-1 #是否启用账号过期停权 -1表示不启用

EXPIRE= #账号终止日期 不设置表示不启用

SHELL=/bin/bash #新用户默认使用的shell类型

SKEL=/etc/skel #配置新用户家目录默认文件的存放路径，/etc/skel目录就是在这里配置生效的，即在我们创建新用户时，用户家目录下面的文件，都是从这里配置的目录中复制过去的

CREATE\_MAIL\_SPOOL=yes #表示创建mail文件

* 【举例】

将新用户默认的shell类型改为/bin/tcsh

[root@oldboy ~]# cat /etc/default/useradd

# useradd defaults file

GROUP=100

HOME=/home

INACTIVE=-1

EXPIRE=

SHELL=/bin/bash

#SHELL=/bin/tcsh

SKEL=/etc/skel

CREATE\_MAIL\_SPOOL=yes

[root@oldboy ~]# sed -n "s#SHELL=/bin/bash#SHELL=/bin/tcsh#gp" /etc/default/useradd

SHELL=/bin/tcsh

[root@oldboy ~]# sed -i "s#SHELL=/bin/bash#SHELL=/bin/tcsh#gp" /etc/default/useradd

[root@oldboy ~]# useradd ceshi2

[root@oldboy ~]# tail -1 /etc/passwd

ceshi2:x:1008:1008::/home/ceshi2:/bin/tcsh

## 3.6 sudo配置文件/etc/sudoers的讲解

### 3.6.1 /etc/sudoer文件中各选项的作用

|  |  |  |
| --- | --- | --- |
| 选项 | 作用 | 格式 |
| Host Aliases | 定义主机别名 | Host\_Alias 别名 = 主机 |
| User Aliases | 定义用户别名 | User\_Alias ADMINS = 用户 |
| Command Aliases | 定义命令别名 | Cmnd\_Alias 命令别名 = 命令 |
| Runas\_Alias | 定义运行身份别名 | Runas\_Alias OP = 账户 |
| 在=的右边有多个时中间用逗号分割 | | |

### 3.6.2 配置sudo配置文件需要注意的事项

* Host Aliases和User Aliases在配置时并不是必须的
* User Aliases可以定义用户组但是用户组前面必须要加%
* 定义的用户和用户组必须事先存在
* 命令必须使用绝对路径
* 命令超过一行时，用\换行
* 在定义时，可以使用正则
* ！表示非，即禁止执行该命令

### 3.6.3 用户的提权配置说明

* 固定格式：

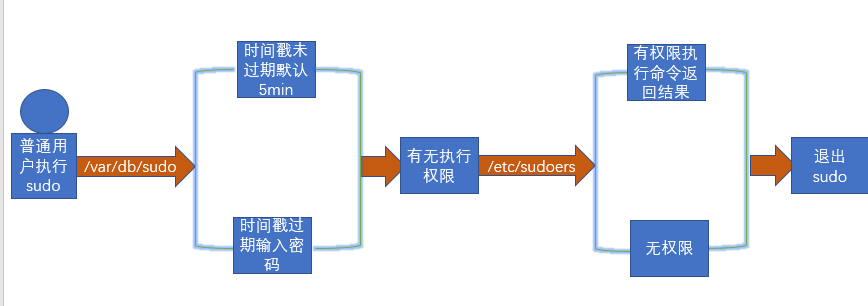
用户 机器 角色 命令

* 举例：

root ALL=(ALL) ALL

OLD\_ADMINS ALL=(OP1) USERCMD, DISKCMD, NETMAGCMD, CTRLCMD

### 3.6.4 sudo执行过程图解



### 3.6.5 sudo的审计功能

* 【说明】

sudo的审计功能只记录sudo命令的用户操作，并不记录其他的

* 【实现过程】

###################################实现代码####################################

[root@oldboy ~]# rpm -qa |egrep "sudo|rsyslog"

rsyslog-5.8.10-10.el6\_6.x86\_64

sudo-1.8.6p3-27.el6.x86\_64

[root@oldboy ~]# echo "Defaults logfile=/var/log/sudo.log">>/etc/sudoers

[root@oldboy ~]# tail -1 /etc/sudoers

Defaults logfile=/var/log/sudo.log

###################################验证过程####################################

[oldboy@oldboy ~]$ su -

Password:

[root@oldboy ~]# tail -1 /etc/sudoers

Defaults logfile=/var/log/sudo.log

[root@oldboy ~]# su - oldboy

[oldboy@oldboy ~]$ sudo -l

[sudo] password for oldboy:

Matching Defaults entries for oldboy on this host:

!visiblepw, always\_set\_home, env\_reset, env\_keep="COLORS DISPLAY HOSTNAME HISTSIZE INPUTRC KDEDIR LS\_COLORS", env\_keep+="MAIL

PS1 PS2 QTDIR USERNAME LANG LC\_ADDRESS LC\_CTYPE", env\_keep+="LC\_COLLATE LC\_IDENTIFICATION LC\_MEASUREMENT LC\_MESSAGES",

env\_keep+="LC\_MONETARY LC\_NAME LC\_NUMERIC LC\_PAPER LC\_TELEPHONE", env\_keep+="LC\_TIME LC\_ALL LANGUAGE LINGUAS \_XKB\_CHARSET

XAUTHORITY", secure\_path=/sbin\:/bin\:/usr/sbin\:/usr/bin, logfile=/var/log/sudo.log

User oldboy may run the following commands on this host:

(root, oldboy) /usr/sbin/useradd, /usr/sbin/userdel, /usr/bin/passwd [A-Za-z]\*, /bin/chmod, /bin/chown, (root, oldboy)

/sbin/fdisk, /sbin/parted, (root, oldboy) /sbin/ifconfig, /etc/init.d/network, (root, oldboy) /usr/sbin/reboot, /usr/sbin/halt

[oldboy@oldboy ~]$ su -

Password:

[root@oldboy ~]# tail /var/log/sudo.log

Dec 23 12:52:14 : oldboy : TTY=pts/0 ; PWD=/home/oldboy ; USER=root ;

COMMAND=list

Dec 23 12:52:23 : oldboy : TTY=pts/0 ; PWD=/home/oldboy ; USER=root ;

COMMAND=/usr/sbin/useradd dddd

Dec 23 12:53:14 : oldboy : command not allowed ; TTY=pts/0 ; PWD=/home/oldboy ;

USER=root ; COMMAND=/bin/rm -f /etc/hosts

Dec 24 10:34:56 : oldboy : TTY=pts/1 ; PWD=/home/oldboy ; USER=root ;

COMMAND=list

### 3.6.6 sudo配置实例

剧本：

工作需求:

开发人员：User\_Alias KAIFA\_ADMINS = kaifa01, kaifa02

命令权限：KAIFACMD

身份权限：OP1

运维人员：User\_Alias OLD\_ADMINS = oldboy, oldgirl, %sa

命令权限: USERCMD DISKCMD NETMAGCMD CTRLCMD

身份权限：OP1

网络工程师：User\_Alias OLD\_NETADMINS = leo, maya

命令权限：NETMAGCMD

身份权限：OP2

演员：

#用户别名分类：

User\_Alias KAIFA\_ADMINS = kaifa01, kaifa02

User\_Alias OLD\_ADMINS = oldboy, oldgirl, %sa

User\_Alias OLD\_NETADMINS = leo, maya

#岗位命令分类别名：

Cmnd\_Alias USERCMD = /usr/sbin/useradd, /usr/sbin/userdel, \

/usr/bin/passwd [A-Za-z]\*, /bin/chmod, /bin/chown

Cmnd\_Alias DISKCMD = /sbin/fdisk, /sbin/parted

Cmnd\_Alias NETMAGCMD = /sbin/ifconfig, /etc/init.d/network

Cmnd\_Alias CTRLCMD = /usr/sbin/reboot, /usr/sbin/halt

Cmnd\_Alias KAIFACMD = /bin/grep, /usr/bin/tail, /bin/cat

#角色

Runas\_Alias OP1 = root, oldboy

Runas\_Alias OP2 = root, oldgirl

#拍片3

#root ALL=(ALL) ALL

#用户 机器 角色 命令

KAIFA\_ADMINS ALL=(OP1) KAIFACMD

OLD\_ADMINS ALL=(OP1) USERCMD, DISKCMD, NETMAGCMD, CTRLCMD

OLD\_NETADMINS ALL=(OP2) NETMAGCMD

## 3.7 环境变量

PATH：所有变量的路径所在地 用冒号分割

LANG：字符集变量

export：设置环境变量

注意：命令要想执行，必须放到echo $PATH 输出的路径中

* 【应用举例】

[root@oldboy ~]# echo $PATH

/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin

[root@oldboy ~]# cat oldgirl

echo oldboy

[root@oldboy ~]# PATH="$PATH:/root"

[root@oldboy ~]# echo $PATH

/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/root

[root@oldboy ~]# oldgirl

oldboy

# 第4章 linux基础入门下

## 4.1 linux通配符

### 4.1.1 linux通配符与正则表达式的区别

* 通配符常用于命令行，用来查找文件，而正则表达式常用于linux三剑客，个别可以用于命令行，正则主要用来找文本（文件的内容）

### 4.1.2 常用通配符

|  |  |
| --- | --- |
| 符号 | 作用 |
| \* | 代表所有 |
| ？ | 代表一个字符 |
| ; | 两个命令的分隔符 |
| # | 配置文件的注释 |
| | | 管道 |
| ~ | 当前用户的家目录 |
| - | 上一次的路径 |
| $ | 变量前需加的符号 |
| / | 路径分隔符，也是根的意思 |
| > | 输出重定向，会覆盖原数据 |
| >> | 追加输出重定向，内容追加到文件的末尾 |
| < | 输入重定向，（xargs ，tr） |
| << | 追加输入重定向，（cat） |
| ‘ ’ | 单引号，不具有变量置换功能， 即所见即所得 |
| “ ” | 双引号，具有变量置换功能，解析变量后输出，不加引号相当于双引号，常加双引号 |
| ` ` | 反引号 Tab上面的键 中间的为命令，会先执行，相当于$() |
| { } | 中间为命令区 块组合或内容序列 |
| ! | 逻辑运算中的 非 |
| && | 相当于and 且 当第一个命令执行成功后，执行下一个命令 |
| || | 相当于or 或 当第一个命令执行失败后，执行下一个命令 |
| . | 代表当前目录 |
| .. | 代表上一级目录 |

### 4.1.3 举例

* \*

[root@oldboy ~]# mkdir /test

[root@oldboy test]# touch {a..c}.sh

[root@oldboy test]# ls \*.sh

a.sh b.sh c.sh

[root@oldboy test]# ls \*

a.sh b.sh c.sh test.txt

* ?

[root@oldboy test]# touch bbb.sh

[root@oldboy test]# ls

a.sh bbb.sh b.sh c.sh test.txt

[root@oldboy test]# ls -lh ?.sh

-rw-r--r-- 1 root root 0 Mar 20 06:00 a.sh

-rw-r--r-- 1 root root 0 Mar 20 06:00 b.sh

-rw-r--r-- 1 root root 0 Mar 20 06:00 c.sh

[root@oldboy test]# ls -lh ???.sh

-rw-r--r-- 1 root root 0 Mar 20 06:05 bbb.sh

* ;

[root@oldboy test]# pwd;pwd

/test

/test

* |

[root@oldboy test]# ls

a.sh bbb.sh b.sh c.sh test.txt

[root@oldboy test]# echo "I like oldnoy linux" >a.sh

[root@oldboy test]# find /test/ -type f -name "a.sh" |xargs cat

I like oldnoy linux

* $

[root@oldboy ~]# echo $LANG

en\_US.UTF-8

* <

[root@oldboy test]# seq 10 >b.sh

[root@oldboy test]# xargs <b.sh

1 2 3 4 5 6 7 8 9 10

[root@oldboy test]# xargs -n 2 <b.sh

1 2

3 4

5 6

7 8

9 10

* <<

[root@oldboy test]# ls

a.sh a.txt bbb.sh b.sh c.sh test.txt

[root@oldboy test]# cat >>c.sh <<EOF

> I like linux.

> I very like linux.

> EOF

[root@oldboy test]# cat c.sh

I like linux.

I very like linux.

* ‘’ 和 “”

[root@oldboy test]# echo '`date`'

`date`

[root@oldboy test]# echo "`date`"

Sun Mar 20 17:38:33 CST 2016

* ``

[root@oldboy test]# cat `find . -type f -name "c.sh"`

I like linux.

I very like linux.

* {}

[root@oldboy test]# touch {d..f}.txt

[root@oldboy test]# ls

a.sh a.txt bbb.sh b.sh c.sh d.txt e.txt f.txt test.txt

[root@oldboy test]# cp c.sh{,.bak}

[root@oldboy test]# ls

a.sh a.txt bbb.sh b.sh c.sh c.sh.bak d.txt e.txt f.txt test.txt

## 4.2 linux 正则表达式

### 4.2.1 linux基础正则符号

|  |  |
| --- | --- |
| 正则表达式 | 作用 |
| ^ | 匹配以什么什么开头的内容，在vi/vim中代表一行的开头 |
| $ | 匹配以什么什么结尾的内容，在vi/vim中代表一行的结尾 |
| ^$ | 表示空白行 |
| . | 代表且只能代表任意一个字符 |
| \ | 转义字符，\.就代表一个点本身，让所有有意思的特殊字符脱掉马甲，还原其本性 |
| \* | 重复0个或多个前面的字符，例如o\* 就表示0个o 1个o或者多个o |
| .\* | 表示所有 |
| ^.\* | 表示以任意多个字符开头 |
| .\*$ | 表示以任意多个字符结尾 |
| [] | 匹配字符集内的任意一个字符，例[abc]，匹配abc中的任意一个 [a-zA-Z] [0-9] |
| [^] | 匹配不包含字符集内的任意一个字符，例[^abc] 匹配abc意外的任意一个字符 |
| a\{n,m\} | 重复前面的字符n到m次，其中\表示转义的意思，如果用sed -r或者egrep可以去掉\ |
| a\{n,\} | 重复前面的字符至少n次，其中\表示转义的意思，如果用sed -r或者egrep可以去掉\ |
| a{,m\} | 重复前面的字符至多m次，Centos5不能用，Centos6可以用，其中\表示转义的意思，如果用sed -r 或者egrep 可以去掉\ |
| a\{n\} | 重复前面的数字n次，其中\表示转义的意思，如果用sed -r 或者egrep可以去掉\ |

### 4.2.2举例

* ^和$以及^$

[root@oldboy ~]# grep "^m" oldboy.log

my blog is http://oldboy.blog.51cto.com

my qq num is 49000448

my god,i am not oldbey,but OLDBOY!

[root@oldboy ~]# grep "m$" oldboy.log

my blog is <http://oldboy.blog.51cto.com>

[root@oldboy ~]# grep -n "^$" oldboy.log

3:

8:

[root@oldboy ~]# grep -nv "^$" oldboy.log

1:I am oldboy teacher!

2:I teach linux.

4:I like badminton ball,billiard ball and chinese chess!

5:my blog is http://oldboy.blog.51cto.com

6:our site is http://www/etiantian.org

7:my qq num is 49000448

9:not 4900000448.

10:my god,i am not oldbey,but OLDBOY!

11:gd

12:good

13:goood

14:glad

15:oldboy1

* . 和 \* 以及.\*

[root@oldboy ~]# grep -n "." oldboy.log

1:I am oldboy teacher!

2:I teach linux.

4:I like badminton ball,billiard ball and chinese chess!

5:my blog is http://oldboy.blog.51cto.com

6:our site is http://www/etiantian.org

7:my qq num is 49000448

9:not 4900000448.

10:my god,i am not oldbey,but OLDBOY!

11:gd

12:good

13:goood

14:glad

15:oldboy1

[root@oldboy ~]# grep -n "0\*" oldboy.log

1:I am oldboy teacher!

2:I teach linux.

3:

4:I like badminton ball,billiard ball and chinese chess!

5:my blog is http://oldboy.blog.51cto.com

6:our site is http://www/etiantian.org

7:my qq num is 49000448

8:

9:not 4900000448.

10:my god,i am not oldbey,but OLDBOY!

11:gd

12:good

13:goood

14:glad

15:oldboy1

[root@oldboy ~]# grep -n ".\*" oldboy.log

1:I am oldboy teacher!

2:I teach linux.

3:

4:I like badminton ball,billiard ball and chinese chess!

5:my blog is http://oldboy.blog.51cto.com

6:our site is http://www/etiantian.org

7:my qq num is 49000448

8:

9:not 4900000448.

10:my god,i am not oldbey,but OLDBOY!

11:gd

12:good

13:goood

14:glad

15:oldboy1

* ^.\* 和.\*$

[root@oldboy ~]# grep -n ".\*$" oldboy.log

1:I am oldboy teacher!

2:I teach linux.

3:

4:I like badminton ball,billiard ball and chinese chess!

5:my blog is http://oldboy.blog.51cto.com

6:our site is http://www/etiantian.org

7:my qq num is 49000448

8:

9:not 4900000448.

10:my god,i am not oldbey,but OLDBOY!

11:gd

12:good

13:goood

14:glad

15:oldboy1

[root@oldboy ~]# grep -n "^.\*" oldboy.log

1:I am oldboy teacher!

2:I teach linux.

3:

4:I like badminton ball,billiard ball and chinese chess!

5:my blog is http://oldboy.blog.51cto.com

6:our site is http://www/etiantian.org

7:my qq num is 49000448

8:

9:not 4900000448.

10:my god,i am not oldbey,but OLDBOY!

11:gd

12:good

13:goood

14:glad

15:oldboy1

* []和[^]

[root@oldboy ~]# grep -n "[a-z]" oldboy.log

1:I am oldboy teacher!

2:I teach linux.

4:I like badminton ball,billiard ball and chinese chess!

5:my blog is http://oldboy.blog.51cto.com

6:our site is http://www/etiantian.org

7:my qq num is 49000448

9:not 4900000448.

10:my god,i am not oldbey,but OLDBOY!

11:gd

12:good

13:goood

14:glad

15:oldboy1

[root@oldboy ~]# grep -n "[^a-z]" oldboy.log

1:I am oldboy teacher!

2:I teach linux.

4:I like badminton ball,billiard ball and chinese chess!

5:my blog is http://oldboy.blog.51cto.com

6:our site is http://www/etiantian.org

7:my qq num is 49000448

9:not 4900000448.

10:my god,i am not oldbey,but OLDBOY!

15:oldboy1

* a\{n,m\} a\{n\} a\{n,\} a{,m\}

[root@oldboy ~]# grep -no "0\{2\}" oldboy.log

7:00

9:00

9:00

[root@oldboy ~]# grep -n "0\{2,3\}" oldboy.log

7:my qq num is 49000448

9:not 4900000448.

[root@oldboy ~]# grep -no "0\{2,3\}" oldboy.log

7:000

9:000

9:00

[root@oldboy ~]# grep -no "0\{2,\}" oldboy.log

7:000

9:00000

[root@oldboy ~]# grep -no "0\{,3\}" oldboy.log

7:000

9:000

9:00

### 4.2.3 linux扩展的正则表达式

|  |  |
| --- | --- |
| 正则表达式 | 作用 |
| + | 表示重复1个或一个以上前面的字符 |
| ？ | 表示重复0个或一个前面的字符 |
| | | 表示同时过滤多个字符串 |
| （） | 分组过滤 后向引用 |

### 4.2.4 举例

[root@oldboy ~]# cat oldboy.log

I am oldboy teacher!

I teach linux.

I like badminton ball,billiard ball and chinese chess!

my blog is http://oldboy.blog.51cto.com

our site is http://www/etiantian.org

my qq num is 49000448

not 4900000448.

my god,i am not oldbey,but OLDBOY!

gd

good

goood

glad

oldboy1

[root@oldboy ~]# grep -E "go?d" oldboy.log

my god,i am not oldbey,but OLDBOY!

gd

[root@oldboy ~]# grep -E "go.d" oldboy.log

good

[root@oldboy ~]# grep -E "go+d" oldboy.log

my god,i am not oldbey,but OLDBOY!

good

goood

### 4.2.5 元字符

#### 4.2.5.1 元字符的定义

* 元字符是一种perl风格的正则表达式，只有一部分文本处理工具支持他，并不是所有的文本处理工具都支持他

#### 4.2.5.2 元字符的符号

|  |  |  |
| --- | --- | --- |
| 正则表达式 | 作用 | 备注 |
| \b | 单词的边界 | 重点掌握 |
| \B | 非单词的边界 | 了解即可 |
| \d | 单个数字字符 | 了解即可 |
| \D | 单个非数字字符 | 了解即可 |
| \w | 单个单词字符(字母、数字与\_) | 了解即可 |
| \W | 单个非单词字符 | 了解即可 |
| \n | 换行符 | 了解即可 |
| \s | 单个空白字符 | 了解即可 |
| \S | 单个非空白字符 | 了解即可 |
| \r | 回车 | 了解即可 |
| \t | 横向制表符 | 了解即可 |
| \v | 垂直制表符 | 了解即可 |
| \f | 换页符 | 了解即可 |

### 4.2.5 POSIX字符类

|  |  |
| --- | --- |
| 正则表达式 | 作用 |
| [:alnum:] | 匹配任何一个数字或字母字符 |
| [:alpha:] | 匹配任意一个字母字符（包括大小写） |
| [:blank:] | 匹配空格和制表符（包括横向和纵向） |
| [:digit:] | 匹配任意一个数字字符 |
| [:lower:] | 匹配任意一个小写字母 |
| [:upper:] | 匹配任意一个大写字母 |
| [:space:] | 匹配一个包括换行符、回车等在内的所有空白字符 |
| [:print:] | 匹配任何一个可以打印的字符 |
| [:graph:] | 匹配任何一个可以看得见且能够打印的字符 |
| [:xdigit:] | 匹配任何一个十六进制数（0-9、a-f、A-F） |
| [:cntrl:] | 匹配任何一个控制字符（ASCII字符集中的前32个字符） |

* 【注意】

元字符和POSIX字符只需了解，用到时回来查看即可

## 4.3 linux 定时任务之crond

### 4.3.1 什么是crond

* crond是linux系统中用来定期执行命令或者指定程序任务的一种服务或软件。在默认情况下，只要我们安装万Centos后，系统就会自动执行crond任务调度服务

### 4.3.2 为什么要使用crond定时任务

* linux定时任务的crond相当于我们平时生活中的闹钟的功能，能够满足周期性执行任务的需求

### 4.3.3 linux系统的定时任务

* linux系统的定时任务可以分为两种
* linux系统自身定期执行任务的工作：系统周期性执行的任务工作，如轮训系统日志、备份系统数据、清理系统缓存等

【实例】

[root@oldboy ~]# ll /var/log/messages\*

-rw------- 1 root root 783486 Dec 18 17:27 /var/log/messages

-rw------- 1 root root 408031 Dec 9 15:22 /var/log/messages-20171209

-rw------- 1 root root 202218 Dec 11 09:05 /var/log/messages-20171211

-rw------- 1 root root 397963 Dec 13 21:43 /var/log/messages-20171213

-rw------- 1 root root 807362 Dec 17 08:28 /var/log/messages-20171217

* 用户执行的任务工作：某个用户或系统管理员定期要执行的任务

【实例】

========================每5分钟自动更新一下时间========================

[root@oldboy ~]# crontab -l

#update time

\*/5 \* \* \* \* /usr/bin/ntpstat time.nist.gov >/dev/null 2>&1

### 4.3.4 crond的基本格式

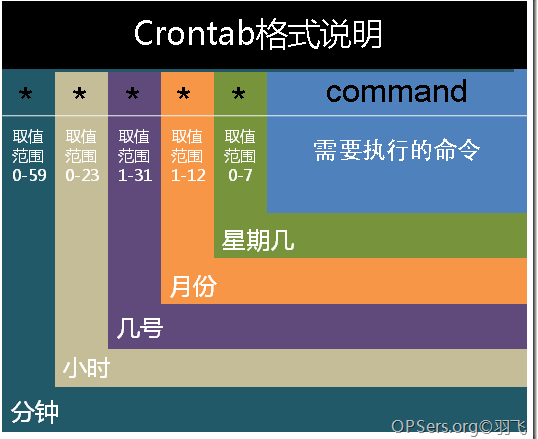
\* \* \* \* \* cmd

* 【注意事项】
* cmd为要执行的命令或者脚本
* 每个段之间必须要有空格

### 4.3.5 crond语法格式中时间段的含义表及取值范围

|  |  |  |
| --- | --- | --- |
| 段 | 含义 | 取值范围 |
| 第一段 | 代表分钟 | 00-59 |
| 第二段 | 代表小时 | 00-23 |
| 第三段 | 代表天 | 01-31 |
| 第四段 | 代表月 | 01-12 |
| 第五段 | 代表周 | 0-7（0和7代表的都是周日） |

#### 4.3.5.1 crontab语法的图解说明



### 4.3.6 crond语法格式中的特殊符号及其含义

|  |  |  |
| --- | --- | --- |
| 符号 | 含义 | 例子 |
| \*(星号) | 任意时间都 即每的意思 | 00 01 \* \* \* cmd 表示每天的凌晨一点执行cmd |
| -(减号) | 表示分割符 即表示一个时间段 | 00 9-11 \* \* \* cmd 表示每天的9，10，11点的整点分别执行cmd |
| ,(逗号) | 表示分割时间段 | 00 9,11 \* \* \* cmd 表示每天的9点和11点的整点分别执行cmd |
| /n(斜线) | n表示数字 即每隔n单位时间 | 00 \*/6 \* \* \* cmd 表示每隔6小时执行一下cmd |

### 4.3.7 crontab 指令的说明

|  |  |  |
| --- | --- | --- |
| 参数 | 含义 | 例子 |
| -l（字母） | 查看crontab中的内容 | crontab -l |
| -e | 编辑crontab | crontab -e |
| -r | 删除crontab中的内容 | crontab -r |
| -i | 删除crontab中的内容前会提示 | crontab -i |
| -u | 设定某个用户的crontab服务 | crontab -u oldboy -l |
| 注意：crontab -l/-e 编辑的就是/var/spool/cron/当前用户的文件 | | |

### 4.3.8 使用者权限及定时任务文件

|  |  |
| --- | --- |
| 文件 | 说明 |
| /etc/cron.deny | 该文件中所列的用户不能使用crontab命令 |
| /etc/cron.allow | 该文件中所列用户允许使用crontab命令，优先于/etc/cron.deny |
| /var/spool/cron | 所有用户crontab的配置文件默认都放在此目录，文件名以用户名命名 |

### 4.3.9 crontab设置的规范

* 为定时任务添加必要的注释说明
* 定时任务命令或程序最好写到脚本中执行
* 时间变量问题用\(反斜线)进行转义，最好放在脚本中执行
* 定时任务中的命令要使用绝对路径（即全路径）
* 执行shell脚本任务时前要加/bin/sh或/bin/bash
* 避免不必要的程序及命令的输出
* 定时任务结尾加 >/dev/null 2>&1(如果不加他的话在Centos6中，会在 /var/spool/postfix/maildrop目录中产生大量的小文件，在Centos5中会在/var/spool/clientmqueue目录中产生大量的小文件，造成的结果是使磁盘的inode填满，然后磁盘会报错，不能在往磁盘中写数据，此时可以开启postfix服务，当不开启时，只需在结尾加上>/dev/null 2>&1，然后把这些小文件删除即可)

#### 4.3.9.1 crond规范操作的过程

* 首先要在命令行执行成功，然后在复制执行成功的命令到脚本中，以便减少出错的机会
* 然后测试脚本，测试成功后，复制脚本的规范路径到定时任务的规范路径里，切记不要手敲，防止不必要的错误发生
* 先在测试环境下进行测试，然后在正式部署到生产环境中

### 4.3.10 crond /dev/null 2>&1的作用

* 如果在定时任务的结尾不加 /dev/null 2>&1时，会造成磁盘阵列的inode空间被占满，这些垃圾文件在Centos5系列中会存放在/var/spool/clientmqueue目录下（Centos5系列默认装的是sendmail服务），在Centos6系列中会存放在/var/spool/postfix/maildrop目录下（Centos6系列默认装的是postfix服务）

### 4.3.11 生产环境中调试定时任务的方法

* 规范的公司开发和运维的操作流程：个人的开发环境配置—》办公室环境的测试—》idc机房的测试环境测试—》idc机房的正式环境

#### 4.3.11.1 增加执行频率的调试任务

* 在测试时，把任务执行频率调快一点，看能不能正常执行，若可以正常执行，则在改成需要的任务时间（但是有些任务并不允许频繁执行的，例如定时向数据库里插入数据，这样的任务必须要在测试环境上测试好，这样可以减少线上出错的机会）

#### 4.3.11.2 调整系统时间任务

* 用需要的任务时间设置完后，可以修改系统的当前时间，改成任务执行时间的前几分钟来测试（或者重启定时任务）

#### 4.3.11.3 通过日志输出调试定时任务

* 在脚本中加入日志输出，然后把输出打印到指定的文件中，然后观察日志内容的结果，看是否正确执行

#### 4.3.11.4 注意一些定时任务命令带来的问题

* \* \* \* \* echo “==” >>/tmp/err.log > /dev/null 2 >&1 这将不能正确的执行任务配置，因为前面前面多了 >>或者将后面的 /dev/null 2>&1去掉

#### 4.3.11.5 注意环境变量导致的定时任务故障

* 特别注意在调试java程序任务的时候，注意环境变量，将环境变量的定义加到脚本中

#### 4.3.11.6 通过定时任务日志调试正式任务

[root@oldboy ~]# tail -f /var/log/cron

Dec 19 09:01:01 oldboy anacron[1636]: Jobs will be executed sequentially

Dec 19 09:05:01 oldboy CROND[1660]: (root) CMD (/usr/bin/ntpstat time.nist.gov >/dev/null 2>&1 )

Dec 19 09:10:01 oldboy CROND[1664]: (root) CMD (/usr/bin/ntpstat time.nist.gov >/dev/null 2>&1 )

Dec 19 09:10:01 oldboy CROND[1665]: (root) CMD (/usr/lib64/sa/sa1 1 1)

Dec 19 09:15:01 oldboy CROND[1670]: (root) CMD (/usr/bin/ntpstat time.nist.gov >/dev/null 2>&1 )

Dec 19 09:20:01 oldboy CROND[1694]: (root) CMD (/usr/bin/ntpstat time.nist.gov >/dev/null 2>&1 )

Dec 19 09:20:01 oldboy CROND[1695]: (root) CMD (/usr/lib64/sa/sa1 1 1)

# 第5章 linux性能优化

## 5.1 关闭selinux

### 5.1.1 selinux的配置文件

**/etc/sysconfig/selinux**

### 5.1.2 临时关闭

[root@oldboy test]# getenforce 查看selinux的状态

[root@oldboy test]# setenforce 0 关闭selinux

### 5.1.3 永久关闭

sed -n "s#SELINUX=enable#SELINUX=disabled#gp" /etc/sysconfig/selinux 首先看修改的内容

是否正确

sed -i "s#SELINUX=enable#SELINUX=disabled#" /etc/sysconfig/selinux 执行修改操作

* 【注意】

通常在关闭selinux时临时关闭和永久关闭会结合起来使用，因为永久关闭只有在关机重启后才能生效

## 5.2 关闭防火墙

### 5.2.1 临时关闭

[root@oldboy test]# /etc/init.d/iptables stop

[root@oldboy test]# service iptables stop

### 5.2.2 永久关闭

[root@oldboy test]# chkconfig iptables off 设置不让其开机自启动

[root@oldboy test]# chkconfig --list iptables 进行查看

### 5.2.3 查看iptables的状态

[root@oldboy test]# /etc/init.d/iptables status

[root@oldboy test]# service iptables status

* 【注意】

通常临时关闭和永久关闭也是结合起来使用

## 5.3 linux中文显示设置

### 5.3.1 配置文件

**cat /etc/sysconfig/i18n**

### 5.3.2 设置中文字符

[root@oldboy test]# cat /etc/sysconfig/i18n

LANG="en\_US.UTF-8" 只需将此项修改为 LANG=”zh\_CN.UTF-8” 即可

SYSFONT="latarcyrheb-sun16"

[root@oldboy test]# echo "LANG="zh\_CN.UTF-8"" > /etc/sysconfig/i18n

[root@oldboy test]# source /etc/sysconfig/i18n 执行此命令 让其生效即可

## 5.3 隐藏linux版本信息

### 5.3.1 版本信息的配置文件

**/etc/issue或/etc/issue.net**

### 5.3.2 清空该配置文件中的内容

[root@oldboy test]# >/etc/issue 清空

[root@oldboy test]# cat /etc/issue 查看是否清空

或

[root@oldboy test]# >/etc/issue.net 清空

[root@oldboy test]# cat /etc/issue.net 查看是否清空

注意：上面的两个配置文件只需清空一个就行

* 【注意】

可以在/etc/mtod 文件中设置系统的登录信息，例如：

[root@oldboy test]# cat /etc/motd

welcome to oldboy training!

## 5.4 精简开机自启动服务

### 5.4.1 需要保留的重要开机自启动服务

* crond

该服务用于周期性的执行系统用户配置的任务计划，有要执行的周期性任务时就要开启，此服务几乎是生产场景必须要用的一个软件

* rsyslog

日志相关软件，这是操作系统提供的一种机制，系统的守护程序通常会使用rsyslog程序将各种信息写到各个系统的日志文件中，Centos6以前此服务的名字为syslog

* network

系统启动时若想激活或关闭各个网络接口，则必须考虑开启

* sshd

远程连接linux服务器时需要用到这个服务程序，所以必须要开启，若不开启linux服务器就无法进行远程连接

* sysstat

sysstat是一个软件包，包含监测系统性能及效率的一组工具，这些工具对于我们手机系统性能很有帮助，比如CPU使用率，硬盘和网络吞吐数据等，对这些数据的收集和分析，有利于判断系统运行是否正常，所以他是提高系统运行效率、安全运行服务器的得力助手

* sysstat软件包集成的主要工具为：
* iostat 工具提供CPU使用率及硬盘吞吐效率的数据
* mpstat 工具提供与单个或多个处理器相关的数据
* sar 工具负责收集，报告并存储系统的活跃信息

### 5.4.2 开启需要的服务

#### 5.4.2.1 第一种方法：关掉不需要的，因为默认是开启的

* 实现过程

[root@oldboy ~]# chkconfig --list |grep "3:on" 找出所有开机自启动的服务

abrt-ccpp 0:off 1:off 2:on 3:on 4:on 5:on 6:off

abrtd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

acpid 0:off 1:off 2:on 3:on 4:on 5:on 6:off

atd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

auditd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

blk-availability 0:off 1:on 2:on 3:on 4:on 5:on 6:off

cpuspeed 0:off 1:on 2:on 3:on 4:on 5:on 6:off

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

haldaemon 0:off 1:off 2:on 3:on 4:on 5:on 6:off

ip6tables 0:off 1:off 2:on 3:on 4:on 5:on 6:off

iptables 0:off 1:off 2:on 3:on 4:on 5:on 6:off

irqbalance 0:off 1:off 2:on 3:on 4:on 5:on 6:off

kdump 0:off 1:off 2:on 3:on 4:on 5:on 6:off

lvm2-monitor 0:off 1:on 2:on 3:on 4:on 5:on 6:off

mdmonitor 0:off 1:off 2:on 3:on 4:on 5:on 6:off

messagebus 0:off 1:off 2:on 3:on 4:on 5:on 6:off

netconsole 0:off 1:off 2:on 3:on 4:on 5:on 6:off

netfs 0:off 1:off 2:on 3:on 4:on 5:on 6:off

network 0:off 1:off 2:on 3:on 4:on 5:on 6:off

nfs-rdma 0:off 1:off 2:on 3:on 4:on 5:on 6:off

ntpd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

ntpdate 0:off 1:off 2:on 3:on 4:on 5:on 6:off

oldboyd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

postfix 0:off 1:off 2:on 3:on 4:on 5:on 6:off

psacct 0:off 1:off 2:on 3:on 4:on 5:on 6:off

quota\_nld 0:off 1:off 2:on 3:on 4:on 5:on 6:off

rdisc 0:off 1:off 2:on 3:on 4:on 5:on 6:off

rdma 0:off 1:on 2:on 3:on 4:on 5:on 6:off

restorecond 0:off 1:off 2:on 3:on 4:on 5:on 6:off

rngd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

rsyslog 0:off 1:off 2:on 3:on 4:on 5:on 6:off

saslauthd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

smartd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

sshd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

svnserve 0:off 1:off 2:on 3:on 4:on 5:on 6:off

sysstat 0:off 1:on 2:on 3:on 4:on 5:on 6:off

udev-post 0:off 1:on 2:on 3:on 4:on 5:on 6:off

xinetd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

[root@oldboy ~]# chkconfig --list |grep "3:on" |egrep "sshd|network|sysstat|rsyslog|crond" 过滤出需要开机自启动的服务

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

network 0:off 1:off 2:on 3:on 4:on 5:on 6:off

rsyslog 0:off 1:off 2:on 3:on 4:on 5:on 6:off

sshd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

sysstat 0:off 1:on 2:on 3:on 4:on 5:on 6:off

[root@oldboy ~]# chkconfig --list |grep "3:on" |egrep -v "sshd|network|sysstat|rsyslog|crond"

排除掉需要开机自启动的服务

abrt-ccpp 0:off 1:off 2:on 3:on 4:on 5:on 6:off

abrtd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

acpid 0:off 1:off 2:on 3:on 4:on 5:on 6:off

atd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

auditd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

blk-availability 0:off 1:on 2:on 3:on 4:on 5:on 6:off

cpuspeed 0:off 1:on 2:on 3:on 4:on 5:on 6:off

haldaemon 0:off 1:off 2:on 3:on 4:on 5:on 6:off

ip6tables 0:off 1:off 2:on 3:on 4:on 5:on 6:off

iptables 0:off 1:off 2:on 3:on 4:on 5:on 6:off

irqbalance 0:off 1:off 2:on 3:on 4:on 5:on 6:off

kdump 0:off 1:off 2:on 3:on 4:on 5:on 6:off

lvm2-monitor 0:off 1:on 2:on 3:on 4:on 5:on 6:off

mdmonitor 0:off 1:off 2:on 3:on 4:on 5:on 6:off

messagebus 0:off 1:off 2:on 3:on 4:on 5:on 6:off

netconsole 0:off 1:off 2:on 3:on 4:on 5:on 6:off

netfs 0:off 1:off 2:on 3:on 4:on 5:on 6:off

nfs-rdma 0:off 1:off 2:on 3:on 4:on 5:on 6:off

ntpd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

ntpdate 0:off 1:off 2:on 3:on 4:on 5:on 6:off

oldboyd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

postfix 0:off 1:off 2:on 3:on 4:on 5:on 6:off

psacct 0:off 1:off 2:on 3:on 4:on 5:on 6:off

quota\_nld 0:off 1:off 2:on 3:on 4:on 5:on 6:off

rdisc 0:off 1:off 2:on 3:on 4:on 5:on 6:off

rdma 0:off 1:on 2:on 3:on 4:on 5:on 6:off

restorecond 0:off 1:off 2:on 3:on 4:on 5:on 6:off

rngd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

saslauthd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

smartd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

svnserve 0:off 1:off 2:on 3:on 4:on 5:on 6:off

udev-post 0:off 1:on 2:on 3:on 4:on 5:on 6:off

xinetd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

[root@oldboy ~]# chkconfig --list |grep "3:on" |egrep -v "sshd|network|sysstat|rsyslog|crond" |awk '{print "chkconfig",$1,"off"}' 将不需要开机自启动的服务打印出来

chkconfig abrt-ccpp off

chkconfig abrtd off

chkconfig acpid off

chkconfig atd off

chkconfig auditd off

chkconfig blk-availability off

chkconfig cpuspeed off

chkconfig haldaemon off

chkconfig ip6tables off

chkconfig iptables off

chkconfig irqbalance off

chkconfig kdump off

chkconfig lvm2-monitor off

chkconfig mdmonitor off

chkconfig messagebus off

chkconfig netconsole off

chkconfig netfs off

chkconfig nfs-rdma off

chkconfig ntpd off

chkconfig ntpdate off

chkconfig oldboyd off

chkconfig postfix off

chkconfig psacct off

chkconfig quota\_nld off

chkconfig rdisc off

chkconfig rdma off

chkconfig restorecond off

chkconfig rngd off

chkconfig saslauthd off

chkconfig smartd off

chkconfig svnserve off

chkconfig udev-post off

chkconfig xinetd off

[root@oldboy ~]# chkconfig --list |grep "3:on" |egrep -v "sshd|network|sysstat|rsyslog|crond" |awk '{print "chkconfig",$1,"off"}'|bash 执行上面的命令

[root@oldboy ~]# chkconfig --list |grep "3:on" 查看是否只剩下需要开机自启动的服务

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

network 0:off 1:off 2:on 3:on 4:on 5:on 6:off

rsyslog 0:off 1:off 2:on 3:on 4:on 5:on 6:off

sshd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

sysstat 0:off 1:on 2:on 3:on 4:on 5:on 6:off

#### 5.4.2.2 第二种方法：关掉所有开机自启动的服务，开启我们需要的

* 实现过程

[root@oldboy ~]# chkconfig --list |grep "3:on" |awk '{print "chkconfig",$1,"off"}'|bash 关闭所有

的开机自启动服务

[root@oldboy ~]# chkconfig --list |grep "3:off" 将所有不是开机自启动的服务找出来

abrt-ccpp 0:off 1:off 2:off 3:off 4:off 5:off 6:off

abrtd 0:off 1:off 2:off 3:off 4:off 5:off 6:off

acpid 0:off 1:off 2:off 3:off 4:off 5:off 6:off

atd 0:off 1:off 2:off 3:off 4:off 5:off 6:off

auditd 0:off 1:off 2:off 3:off 4:off 5:off 6:off

blk-availability 0:off 1:on 2:off 3:off 4:off 5:off 6:off

cpuspeed 0:off 1:on 2:off 3:off 4:off 5:off 6:off

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

haldaemon 0:off 1:off 2:off 3:off 4:off 5:off 6:off

ip6tables 0:off 1:off 2:off 3:off 4:off 5:off 6:off

iptables 0:off 1:off 2:off 3:off 4:off 5:off 6:off

irqbalance 0:off 1:off 2:off 3:off 4:off 5:off 6:off

kdump 0:off 1:off 2:off 3:off 4:off 5:off 6:off

lvm2-monitor 0:off 1:on 2:off 3:off 4:off 5:off 6:off

mdmonitor 0:off 1:off 2:off 3:off 4:off 5:off 6:off

messagebus 0:off 1:off 2:off 3:off 4:off 5:off 6:off

netconsole 0:off 1:off 2:off 3:off 4:off 5:off 6:off

netfs 0:off 1:off 2:off 3:off 4:off 5:off 6:off

network 0:off 1:off 2:off 3:off 4:off 5:off 6:off

nfs-rdma 0:off 1:off 2:off 3:off 4:off 5:off 6:off

ntpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off

ntpdate 0:off 1:off 2:off 3:off 4:off 5:off 6:off

oldboyd 0:off 1:off 2:off 3:off 4:off 5:off 6:off

postfix 0:off 1:off 2:off 3:off 4:off 5:off 6:off

psacct 0:off 1:off 2:off 3:off 4:off 5:off 6:off

quota\_nld 0:off 1:off 2:off 3:off 4:off 5:off 6:off

rdisc 0:off 1:off 2:off 3:off 4:off 5:off 6:off

rdma 0:off 1:on 2:off 3:off 4:off 5:off 6:off

restorecond 0:off 1:off 2:off 3:off 4:off 5:off 6:off

rngd 0:off 1:off 2:off 3:off 4:off 5:off 6:off

rsyslog 0:off 1:off 2:off 3:off 4:off 5:off 6:off

saslauthd 0:off 1:off 2:off 3:off 4:off 5:off 6:off

smartd 0:off 1:off 2:off 3:off 4:off 5:off 6:off

sshd 0:off 1:off 2:off 3:off 4:off 5:off 6:off

svnserve 0:off 1:off 2:off 3:off 4:off 5:off 6:off

sysstat 0:off 1:on 2:off 3:off 4:off 5:off 6:off

udev-post 0:off 1:on 2:off 3:off 4:off 5:off 6:off

xinetd 0:off 1:off 2:off 3:off 4:off 5:off 6:off

[root@oldboy ~]# chkconfig --list |grep "3:off" |egrep "sshd|crond|network|rsyslog|sysstat"

将需要开机自启动的服务过滤出来

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

network 0:off 1:off 2:off 3:off 4:off 5:off 6:off

rsyslog 0:off 1:off 2:off 3:off 4:off 5:off 6:off

sshd 0:off 1:off 2:off 3:off 4:off 5:off 6:off

sysstat 0:off 1:on 2:off 3:off 4:off 5:off 6:off

[root@oldboy ~]# chkconfig --list |grep "3:off" |egrep "sshd|crond|network|rsyslog|sysstat" |awk '{print "chkconfig",$1,"on"}' 打印要执行的命令

chkconfig crond on

chkconfig network on

chkconfig rsyslog on

chkconfig sshd on

chkconfig sysstat on

[root@oldboy ~]# chkconfig --list |grep "3:off" |egrep "sshd|crond|network|rsyslog|sysstat" |awk '{print "chkconfig",$1,"on"}'|bash 执行该命令

[root@oldboy ~]# chkconfig --list |grep "3:on" 查看是否只剩下需要开机自启动服务

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

network 0:off 1:off 2:on 3:on 4:on 5:on 6:off

rsyslog 0:off 1:off 2:on 3:on 4:on 5:on 6:off

sshd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

sysstat 0:off 1:on 2:on 3:on 4:on 5:on 6:off

### 5.4.3 chkconfig 原理

#### 5.4.3.1 添加一个自定义服务到开机自启动中步骤

* 首先将服务的脚本写到/etc/init.d/目录下面
* 然后赋予其可执行权限
* 在服务脚本中添加下面两行内容

# chkconfig: 2345 88 66 其中2345代表的是chkcongfig中对应的级别 88代表的是K88

开启 66代表的是S66 停止

# description:It is 90 hou description 自定义的描述信息

* 将服务添加至chkcongfigf去 chkconfig --add 服务名

#### 5.4.3.2 示例

[root@oldboy rc3.d]# vim /etc/init.d/laoniaod

# chkconfig: 2345 98 99

# description: This is a 90 hou description

echo I like meizi

[root@oldboy rc3.d]# chmod +x /etc/init.d/laoniaod

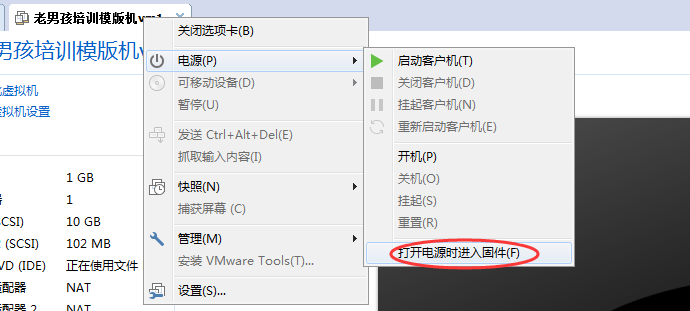
[root@oldboy rc3.d]# chkconfig --add laoniaod

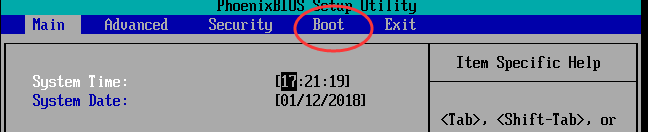
[root@oldboy rc3.d]# chkconfig --list laoniaod

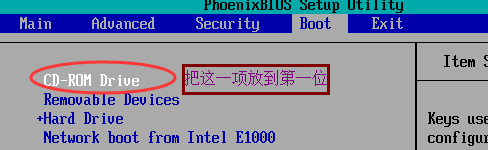
laoniaod 0:off 1:off 2:on 3:on 4:on 5:on 6:off

## 5.5 忘记root密码的三种解决办法

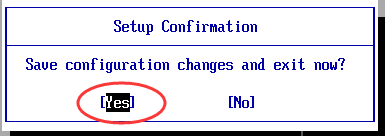
### 5.5.1 第一种方法

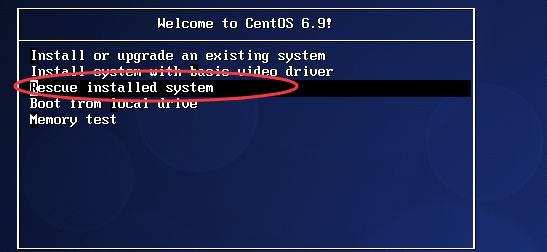




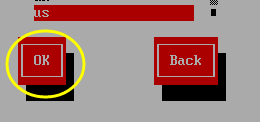


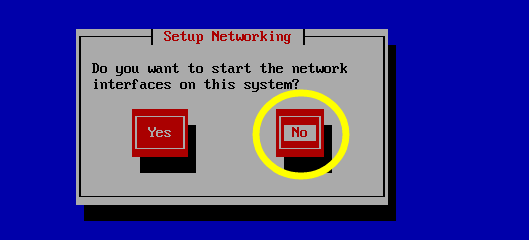
之后保存退出

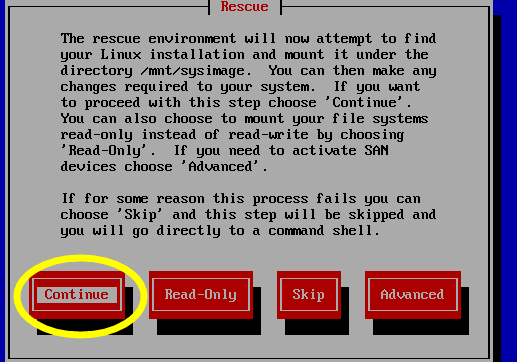


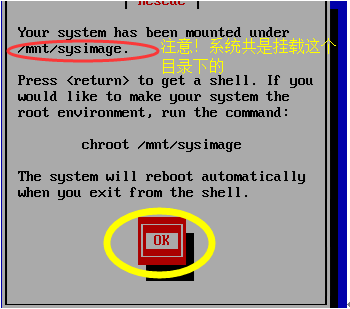




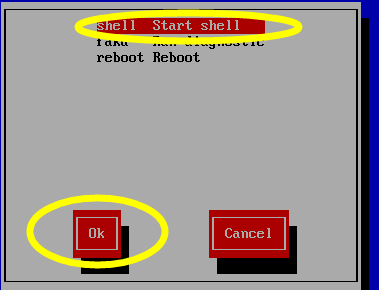


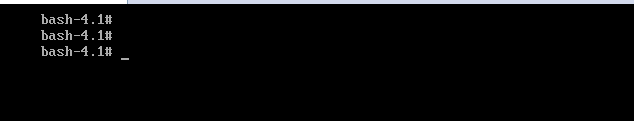












* 之后开始进行编辑 vi /mnt/sysimage/etc/passwd 只需把root的密码给去掉即可，然后重启，在把启动项修改为默认的就行

### 5.5.2 第二种方法

* 操作流程和第一种方法一样，不同的就是在最后编辑的时候，首先执行下面的命令：

#choot /mnt/sysimage/

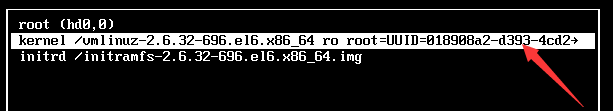
#passwd root

### 5.5.3 第三种方法

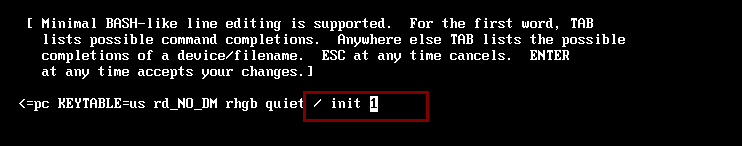
* 在开机时按e键



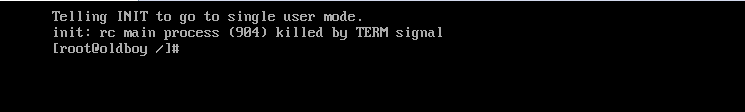
* 之后在按e键



* 之后进入，然后输入 /init 1 或者 single 或者 / 1 都行



* 然后按enter键 之后按b键之后就可以进行修改密码了



## 5.6 修改主机名的规范步骤

### 5.6.1 修改步骤

* 首先临时修改主机名

hostname 主机名

* 编辑/etc/sysconfig/network配置文件修改HOSTNAME=主机名
* 编辑/etc/hosts配置文件 在127.0.0.1后加上主机名

### 5.6.2 实例

[root@oldboy ~]# hostname oldboy 临时修改

[root@oldboy ~]# vim /etc/sysconfig/network 永久修改

[root@oldboy ~]# cat /etc/sysconfig/network

NETWORKING=yes

HOSTNAME=oldboy

[root@oldboy ~]# vim /etc/hosts

[root@oldboy ~]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4 oldboy

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6 oldboy

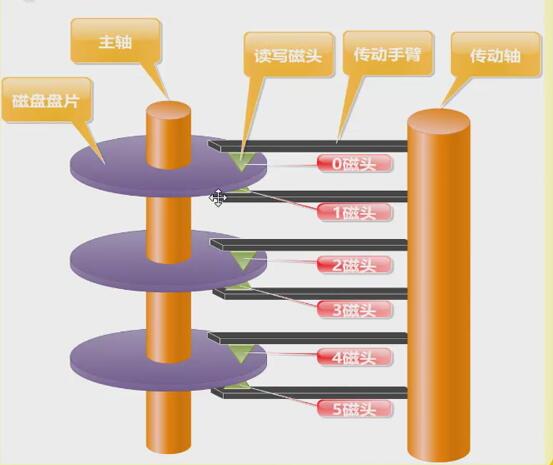
# 第6章 磁盘

## 6.1 linux磁盘设备的基础知识

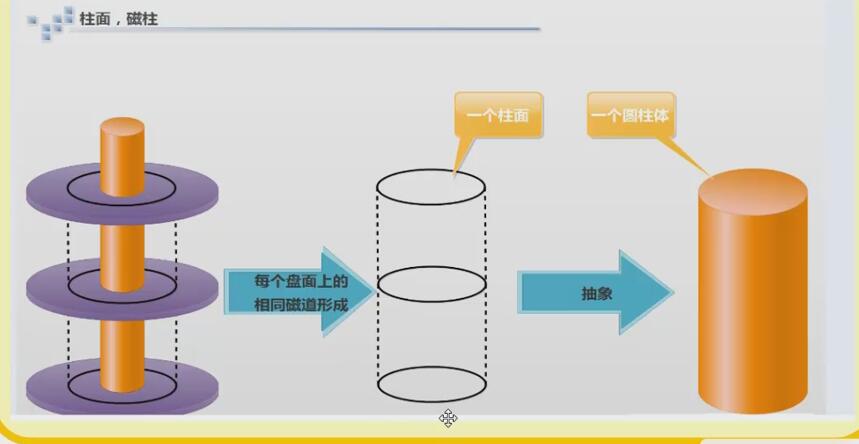
### 6.1.1 linux磁盘的历史简述

* 第一台磁盘存储系统是在1956年9月由IBM公司推出的IBM 350 RAMAC，容量大小只有5M
* IBM 350 RAMAC是磁盘的开山鼻祖
* 现代磁盘的真正原型是1973年IBM推出的Winchester(温氏)磁盘，其特点是磁头在高速旋转的盘片上方做径向运动，现在的机械硬盘依旧沿用这种模式

### 6.1.2 磁盘的图形说明



### 6.1.3 柱面的图形说明



### 6.1.4 有关磁盘的重要名词的说明

* 盘片：盘片是储存数据的真正载体
* 柱面：所有盘面上相同半径的磁道从上往下组成的圆柱体叫做柱面
* 磁头：悬浮在盘面的上方，在高速旋转的盘片上做径向运动
* 磁道：相同半径的圆组成的轨迹叫做磁道
* 扇区：操作系统是以扇区为单位将信息存储在磁盘上的，每个扇区的大小是512字节

### 6.1.5 磁道、扇区、柱面的总结

* 一块磁盘有2-14个盘片，每个盘片有两个面，盘面数=磁头数
* 不同盘面的磁道被划分为多个扇形区域，每个扇形区域就是一个扇区，大小为512字节
* 同一个盘面，以盘片中心为圆心，每个不同半径的圆形轨迹就是一个磁道
* 不同盘面相同半径的磁道从上往下组成的圆柱形就是柱面
* 一个柱面包含多个磁道，一个磁道包含多个扇区
* 数据信息的记录可以表示为：某磁头、某柱面、某扇区

### 6.1.6 磁盘的工作原理

* 磁盘是按照柱面为单位进行读写数据的，只有同一柱面的所有磁头全部读写完毕后，磁头才转移到下一个柱面（即寻道）
* 不同磁头间的切换是电子切换，切换到不同的磁道是机械运动，速度要慢很多
* 磁头寻道是机械运动，磁头切换是电子切换
* 磁头在高速旋转的片面上做径向运动

### 6.1.7 磁盘大小的计算

#### 6.7.1 计算公式

* 盘面大小=磁道大小\*磁道数（磁道数=柱面数）
* 磁道大小=512字节\*扇区数
* 磁盘大小=盘面大小\*盘面数（盘面数=磁头数）
* 磁盘大小=磁道大小\*磁道数\*磁头数
* 磁盘大小=512字节\*扇区数\*磁道数\*磁头数
* 磁盘大小=柱面大小\*柱面数

#### 6.7.2 详细的计算过程

[root@oldboy ~]# fdisk -l

Disk /dev/sda: 10.7 GB, 10737418240 bytes

255 heads（磁头数）, 63 sectors/track（每个磁道的扇区数）, 1305 cylinders（柱面数）

Units（单元块） = cylinders of 16065 \* 512 = 8225280 bytes（一个柱面的大小）

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x000e0552

10.7 GB=（512\*255\*63\*1305）/1000/1000/1000或

10.7GB=（1305\*8225280）/1000/1000/1000

### 6.1.8 磁盘的接口类型

#### 6.1.8.1 常见的磁盘接口类型

SATA、SCSI、SAS、光钎FC通道

#### 6.1.8.2 几种磁盘接口的优点

* SATA：可靠、简单、支持热插拔、性能高，是当前pc机和服务器的主流
* SCSI：应用范围广、多任务、带宽大、CPU占用率低、支持热插拔
* SAS：结合了SATA和SCSI两者的优点
* 光钎通道：支持热插拔、高度带宽、远程连接、连接设备数量大

#### 6.1.8.3 SAS技术的优势

* SAS技术降低了磁盘阵列的成本
* 串行接口让传输性能提高
* 具有更好的扩展性能
* 安装变得更加简单
* 具有更好的兼容性

### 6.1.9 生产环境中主流磁盘信息的对比

* 企业生产场景普及程度：SAS>SATA>SSD
* 单位容量对比性能和价格：SSD>SAS>SATA
* 单位价格购买的磁盘容量：SATA>SAS>SSD

### 6.1.10 磁盘选购的方法

1）线上（用户正在使用的环境 给用户提供服务）的业务，用SAS盘

2）线下（自己人用户环境）的业务，选SATA盘

3）线上高并发、小容量（很多人都想看的图片）的业务，选用固态硬盘SSD

4）成本思想：根据访问的热度，智能分析分层存储，SATA+SSD

http://oldboy.blog.51cto.com/2561410/926983/

当老板让你购买服务器时：你要选dell的2u（R720/R730）的或1u（R410/R610）的，选用4核4线程 ，8核16线程的，SAS 15000/M的

### 6.1.11 内存的查看方法

[root@oldboy ~]# free -m

total used free shared buffers cached

Mem: 980 118 862 0 10 32

-/+ buffers/cache: 75 905

Swap: 767 0 767

##################################注意#########################################

linux系统的特性是将系统不用的物理内存作为缓存区或缓冲区使用，因此862不是真实的内存，而905才是

buffers为写入缓冲区，sync将缓冲区数据写入磁盘

cache为读取数据的缓存区

硬盘是机械的，无论是写入或读取都太慢了，所以读取或写入都采取了缓冲或者缓存技术

门户网站或架构网站都会使用缓存技术，来让用户写入和读取尽可能的不接触磁盘

### 6.1.12 磁盘的分类

* 机械硬盘和固态硬盘

### 6.1.13 两者之间的比较

|  |  |  |
| --- | --- | --- |
| 项目 | 固态硬盘 | 机械硬盘 |
| 容量 | 较小 | 大 |
| 价格 | 高 | 低 |
| 随机存取 | 极快 | 一般 |
| 写入次数 | SLC:：10万次  MLC：1万次  特制的可以到达100-500万次 | 无限制 |
| 盘内阵列 | 可 | 极难 |
| 工作噪音 | 无 | 有 |
| 工作温度 | 极低 | 较明显 |
| 防震 | 很好 | 较差 |
| 数据恢复 | 难 | 可以 |
| 重量 | 轻 | 重 |

## 6.2 RAID

### 6.2.1 raid简介

* raid就是把多块独立的物理磁盘按照不同的技术组合成一个磁盘组，在逻辑上看起来是一块盘。具有比单块盘更高的容量、更高的储存性能、还可以提供数据的冗余备份的功能

### 6.2.2 raid的级别

#### 6.2.2.1 raid常见的级别

* raid常见的级别有raid0、raid1、raid2、raid3、raid4、raid5、raid6、raid10、raid53

#### 6.2.2.2 生产环境中常用的级别

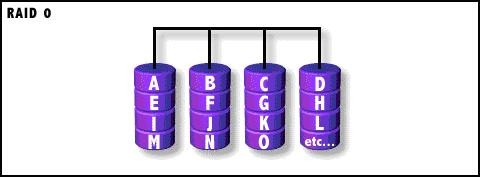
* raid0、raid1、raid5、raid10

### 6.2.3 raid0详解

#### 6.2.3.1 raid0简单的描述

* raid0又称为stripe（条带化）或striping（条带模式），在所有的raid级别中具有最高的储存性能，其提高储存性能的原理就是把连续的数据分散到多个磁盘上存取
* raid0同时对多块磁盘进行读写操作，提高读写的速度
* 制作raid0最少需要一块物理磁盘，一般用来做raid的磁盘大小最好一样

#### 6.2.3.2 raid0的图解



#### 6.2.3.3 raid0的总结（以4块盘为例）

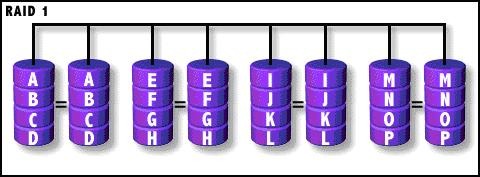
|  |  |
| --- | --- |
| 关注点 | 描述 |
| 容量 | 是4块盘容量的总和，在raid级别中具有最高的存储性能，原理是把连续的数据分散到多个磁盘上存取 |
| 性能 | 理论上磁盘读写速度比单盘提升4倍，磁盘越多倍数越小 |
| 冗余 | 无任何冗余，任何一块盘损坏，整个raid就不能用了 |
| 场合 | 适合于大规模并发读写，但对数据安全要求不高的情况，如集群中的RS，做虚拟机的实体机，负载均衡下面多个相同的RS节点服务器，mysql主从复制的多个slave服务器 |
| 特点 | 读写速度快，无冗余，磁盘容量无损失 |

### 6.2.4 raid1详解

#### 6.2.4.1 raid1的简单描述

* raid1又称为Mirror（镜像），他的宗旨就是最大限度的保证用户数据的可用性和可修复性，raid1的操作方式就是把用户写入一个磁盘的数据百分之百的自动复制到另外一个磁盘上，从而实现存储双份的数据
* 做raid1最少需要2块盘，磁盘的容量是最小的那块盘的容量（不过使用的磁盘大小和厂商最好保持一样），写性能会下降

#### 6.2.4.2 raid1的图解



#### 6.2.4.3 raid1的总结（以两块盘为例）

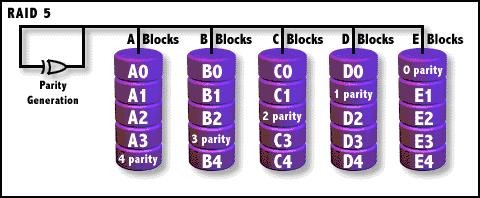
|  |  |
| --- | --- |
| 关注点 | 描述 |
| 容量 | 损失50%的数据容量，即只剩下一块盘的容量 |
| 性能 | 不能提高磁盘的存储性能 |
| 冗余 | 100%的冗余 |
| 场合 | 适用于存放重要的数据，如服务器系统分区和对性能要求不高的数据库存储等领域 |
| 特点 | 100%的冗余，磁盘容量损失一半 |

### 6.2.5 raid5详解

#### 6.2.5.1 raid5简单的描述

* raid5是一种存储性能、数据安全和存储成本兼顾的存储解决方案
* 做raid5至少需要三块物理磁盘，采用奇偶校验，并且奇偶校验信息和相对应的数据分别存储在不同的磁盘上，可以允许一块磁盘损坏
* raid5是把数据和相对应的奇偶校验信息存储在组成raid5的各个磁盘上，并且奇偶校验信息和相对应的数据分别存储在不同的磁盘上，当一块磁盘的数据发生损坏后，利用剩下的数据和奇偶校验信息，能够恢复损坏的数据

#### 6.2.5.2 raid5的图解



#### 6.2.5.3 raid5的总结

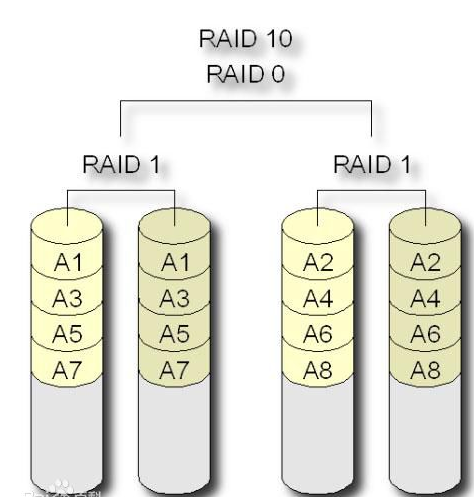
|  |  |
| --- | --- |
| 关注点 | 描述 |
| 容量 | 损失一块盘的数据容量 |
| 性能 | raid5和raid0具有相似的读取速度，只是多了一个奇偶检验信息，写入数据相对于raid0稍微慢点 |
| 冗余 | 可以损失一块盘，安全性能比raid1低，但是空间利用率比raid1高 |
| 场合 | raid5是raid1和raid0的折中方案，适合对性能和冗余都有一定的要求，又都不是十分高的情况，如mysql的主从复制，存储也可以，普通的服务器为了减少维护成本，又要保持一定的冗余和性能的话可以做raid5 |
| 特点 | 容量损失一块盘，写数据通过奇偶校验，是raid0和raid1的折中方案 |

### 6.2.6 raid10详解

#### 6.2.6.1 raid10的简单描述

* raid10是raid1和raid0两者的结合体，具有raid0和raid1所有的优点
* 做raid10最少需要4块物理磁盘

#### 6.2.6.2 raid10的图解



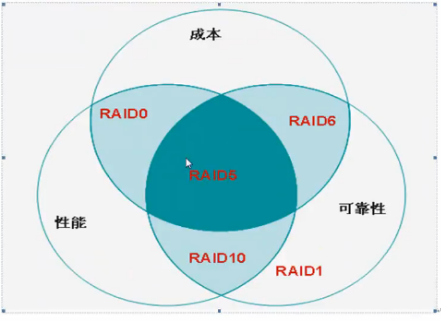
#### 6.2.6.3 raid10的总结（以4块盘为例）

|  |  |
| --- | --- |
| 关注点 | 描述 |
| 容量 | 损失一半的磁盘容量 |
| 性能 | 读写性能最快，100%的数据冗余 |
| 冗余 | 100%的数据冗余 |
| 场合 | 主要用户对数据的安全性和读写速度快的场景，如mysql的数据分区 |
| 特点 | 读写性能最快，100%的数据冗余，磁盘利用率为50%。成本高 |

#### 6.2.6.4 raid10和raid01的区别

* raid10又称raid1+0，先做raid1后做raid0
* raid01又称raid0+1，先做raid0后做raid1

### 6.2.7 常用RAID选择成本、性能、可靠性的对比



### 6.2.8 常见raid的比较

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| raid级别 | 磁盘要求 | 关键优点 | 关键缺点 | 实际应用场景 |
| 0 | 至少一块盘 | 读写速度最快 | 没有冗余 | mysql的slave服务器，集群的RS节点 |
| 1 | 至少两块（偶数） | 100%冗余（镜像） | 读写性能一般，成本高，利用率为50% | 单独的，数据重要，且不能宕机的业务，像系统盘等 |
| 5 | 至少3块 | 具备一定的性能和冗余，读性能和raid0差不多 | 写性能不高，利用率为（n-1）/n | 一般的业务都可以使用，如应用服务器 |
| 10 | 至少4块（偶数） | 读写速度和raid0一样，100%冗余 | 利用率为50%，成本高 | 性能和冗余要求都很好的业务 |

### 6.2.9 DELL R710使用6块盘的实际制作

* raid0支持一块盘到多块盘，容量是所有盘的容量之和
* raid1只支持2块盘，容量损失一块盘
* raid5最少三块盘，不管硬盘数多少，只损失一块盘的容量
* raid10最少4块盘，必须是偶数盘，不管硬盘多少，都损失一半的容量，不支持虚拟磁盘

## 6.3 分区知识

### 6.3.1 常用的分区工具

#### 6.3.1.1 fdisk

* fdisk的分区实质就是改变0磁道0柱面1扇区前446字节之后的64字节的分区表的信息，实际的数据并不会被修改
* fdisk存在的问题就是无法对大于2T的磁盘进行分区，如果大于2T就用parted进行分区
* 【分区实例】

[root@oldboy ~]# fdisk -cu /dev/sdb

WARNING: GPT (GUID Partition Table) detected on '/dev/sdb'! The util fdisk doesn't support GPT. Use GNU Parted.

Command (m for help): m

Command action

a toggle a bootable flag 设定硬盘的启动区

b edit bsd disklabel

c toggle the dos compatibility flag

d delete a partition 删除硬盘的分区信息

l list known partition types

m print this menu 显示命令的帮助信息

n add a new partition 创建新的分区

o create a new empty DOS partition table

p print the partition table 显示分区的信息

q quit without saving changes 退出但是不保存

s create a new empty Sun disklabel

t change a partition's system id 改变硬盘分过区的属性

u change display/entry units

v verify the partition table

w write table to disk and exit 保存

x extra functionality (experts only)

=============================分区信息==================================

Command (m for help): n

Command action

e extended 扩展分区（扩展分区不能用，必须创建逻辑分区 逻辑分区的序号从5开始）

p primary partition (1-4) 主分区（主分区最多有4个）

p

Partition number (1-4): 1 选择分区的序号

First sector (2048-208895, default 2048): 指定分区的起始扇区 一般默认即可

Using default value 2048

Last sector, +sectors or +size{K,M,G} (2048-208895, default 208895): +10M 指定分区的大小

Command (m for help): p 显示分区的信息

Disk /dev/sdb: 106 MB, 106954752 bytes

64 heads, 32 sectors/track, 102 cylinders, total 208896 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

Device Boot Start End Blocks Id System

/dev/sdb1 2048 22527 10240 83 Linux

#### 6.3.1.2 parted

* parted是一个万能的分区工具，并且分区是实时的，即不用像fdisk那样分完区还要进行保存
* 用parted进行分区，主分区的个数不在进行限制，可以有很多个
* 【分区实例】

[root@oldboy ~]# parted /dev/sdb

GNU Parted 2.1

Using /dev/sdb

Welcome to GNU Parted! Type 'help' to view a list of commands.

(parted) help 显示帮助信息，可以理解为fdisk的m

align-check TYPE N check partition N for TYPE(min|opt) alignment

check NUMBER do a simple check on the file system

cp [FROM-DEVICE] FROM-NUMBER TO-NUMBER copy file system to another partition

help [COMMAND] print general help, or help on COMMAND

mklabel,mktable LABEL-TYPE create a new disklabel (partition table)

mkfs NUMBER FS-TYPE make a FS-TYPE file system on partition NUMBER

mkpart PART-TYPE [FS-TYPE] START END make a partition

mkpartfs PART-TYPE FS-TYPE START END make a partition with a file system

move NUMBER START END move partition NUMBER

name NUMBER NAME name partition NUMBER as NAME

print [devices|free|list,all|NUMBER] display the partition table, available devices, free space, all found partitions, or a

particular partition

quit exit program

rescue START END rescue a lost partition near START and END

resize NUMBER START END resize partition NUMBER and its file system

rm NUMBER delete partition NUMBER

select DEVICE choose the device to edit

set NUMBER FLAG STATE change the FLAG on partition NUMBER

toggle [NUMBER [FLAG]] toggle the state of FLAG on partition NUMBER

unit UNIT set the default unit to UNIT

version display the version number and copyright information of GNU Parted

(parted) mklabel gpt 将格式改为gpt的格式

注意：将gpt格式转化为mbr格式parted -s /dev/sda mklabel msdos

或者在交互式命令下 输入 mklabel msdos

Warning: The existing disk label on /dev/sdb will be destroyed and all data on this disk will be lost. Do you want to continue?

Yes/No? Yes 选择Yes

Warning: /dev/sdb contains GPT signatures, indicating that it has a GPT table. However, it does not have a valid fake msdos

partition table, as it should. Perhaps it was corrupted -- possibly by a program that doesn't understand GPT partition tables. Or

perhaps you deleted the GPT table, and are now using an msdos partition table. Is this a GPT partition table?

Yes/No? Yes 选择Yes

(parted) mkpart primary ext4 0 10 创建一个大小为10M的主分区

Warning: The resulting partition is not properly aligned for best performance.

Ignore/Cancel? Ignore 选择Ignore

(parted) mkpart primary linux-swap 11 21 创建一个大小为10M的swap分区

Warning: The resulting partition is not properly aligned for best performance.

Ignore/Cancel? Ignore 选择Ignore

(parted) p

Model: VMware, VMware Virtual S (scsi)

Disk /dev/sdb: 107MB

Sector size (logical/physical): 512B/512B

Partition Table: gpt

Number Start End Size File system Name Flags

1 17.4kB 10.0MB 9983kB primary

2 11.0MB 21.0MB 10.0MB primary

(parted) mkpart logical ext4 22 32 创建一个大小为10M的逻辑分区

Warning: The resulting partition is not properly aligned for best performance.

Ignore/Cancel? Ignore

(parted) p 显示刚才创建的分区信息

Model: VMware, VMware Virtual S (scsi)

Disk /dev/sdb: 107MB

Sector size (logical/physical): 512B/512B

Partition Table: gpt

Number Start End Size File system Name Flags

1 17.4kB 10.0MB 9983kB primary

2 11.0MB 21.0MB 10.0MB primary

3 22.0MB 32.0MB 10.0MB logical

(parted) quit 退出

Information: You may need to update /etc/fstab.

[root@oldboy ~]# ls /dev/sd\* 查看刚才的分区信息 未保存，说明是实时的

/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sdb /dev/sdb1 /dev/sdb2 /dev/sdb3

* 【实时分区的命令】

parted /dev/sdb mklabel gpt Yes

parted /dev/sdb mkpart primary 0 10 Ignore

parted /dev/sdb mkpart primary linux-swap 11 21 Ignore

parted /dev/sdb mkpart logical ext4 22 32 Ignore

parted /dev/sdb p

### 6.3.2 swap分区的构建

#### 6.3.2.1 使用物理分区构建swap

* 【命令】

fdisk /dev/sdb 首先创建一个物理分区 假设为/dev/sdb1

mkswap /dev/sdb1 创建swap交换分区

free -m 查看swap分区的大小

swapon /dev/sdb1 开启交换分区 只有超级管理员才能使用该命令

swapoff /dev/sdb1 关闭交换分区 只有超级管理员才能使用该命令

swapon -s 查看swap分区的信息

* 【创建过程】

[root@oldboy ~]# mkswap /dev/sdb1

Setting up swapspace version 1, size = 104428 KiB

no label, UUID=3e4b2002-9977-451e-9135-49e29d695025

[root@oldboy ~]# free -m

total used free shared buffers cached

Mem: 980 116 864 0 8 30

-/+ buffers/cache: 76 903

Swap: 869 0 869

[root@oldboy ~]# swapon /dev/sdb1

[root@oldboy ~]# swapon -s

Filename Type Size Used Priority

/dev/sda2 partition 786428 0 -1

/dev/sdb1 partition 104428 0 -2

[root@oldboy ~]# swapoff /dev/sdb1

[root@oldboy ~]# swapon -s

Filename Type Size Used Priority

/dev/sda2 partition 786428 0 -1

#### 6.3.2.2 使用文件构建swap

* 【命令】

dd if=/dev/zero of=/tmp/swap bs=1M count=128 创建一个交换分区文件

ls -lh /tmp/swap 查看是否创建成功

mkswap /tmp/swap 创建swap分区

swapon /tmp/swap 开启swap分区

swapon -s 查看swap分区信息

* 【创建过程】

[root@oldboy ~]# dd if=/dev/zero of=/tmp/swap bs=1M count=128

128+0 records in

128+0 records out

134217728 bytes (134 MB) copied, 0.899452 s, 149 MB/s

[root@oldboy ~]# ls -lh /tmp/swap

-rw-r--r-- 1 root root 128M Jan 14 18:32 /tmp/swap

[root@oldboy ~]# mkswap /tmp/swap

mkswap: /tmp/swap: warning: don't erase bootbits sectors

on whole disk. Use -f to force.

Setting up swapspace version 1, size = 131068 KiB

no label, UUID=056a7938-1487-4989-aa0e-5ae0175aae9c

[root@oldboy ~]# swapon -s

Filename Type Size Used Priority

/dev/sda2 partition 786428 0 -1

[root@oldboy ~]# swapon /tmp/swap

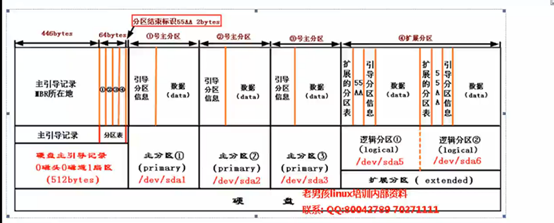
[root@oldboy ~]# swapon -s

Filename Type Size Used Priority

/dev/sda2 partition 786428 0 -1

/tmp/swap file 131068 0 -2

### 6.3.3 分区表的详解



### 6.3.4 文件系统

#### 6.3.4.1 文件系统的简单定义

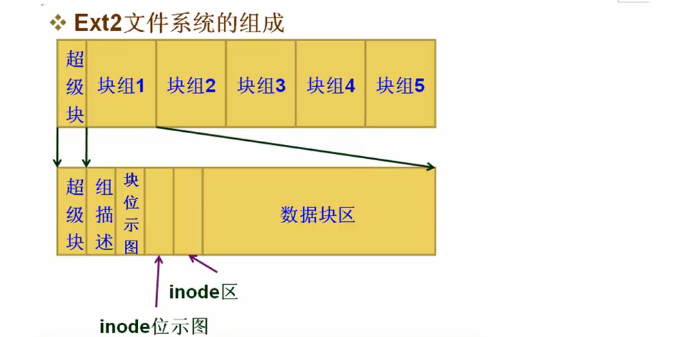
* 文件系统就是一种存储和组织计算机数据文件的机制或方法，他使得对计算机文件系统内的数据的访问和查找变的更加简单和容易
* 在分区完成后，磁盘还不能进行挂载，只有被格式化后才能被挂载存放数据，因为在未被格式化之前，操作系统还不能识别磁盘上的分区格式

#### 6.3.4.2 linux下常见的文件系统

* ext2、ext3、ext4、NFS、XFS

#### 6.3.4.3 ext2文件系统

* 操作系统中的数据分为文件内容和文件属性信息，文件的内容存放在blcok中，文件的属性信息存放在inode中，并且操作系统还会记录文件系统的整体信息到超级块（superblock）中
* 超级块又称为metadata（元数据），所以超级块是存放元数据信息的
* 【ext2的文件系统组成】



#### 6.3.4.4 常规文件系统的选择小结

* Centos选则默认的文件系统就行（Centos5x ext3/Centos6x ext4/Centos7 XFS）
* Bfs、reiserfs默认操作系统是不支持的需要单独安装，海量服务器使用多种文件系统维护可能不方便
* 大并发做磁盘的优化是一方面，但是更多的还是在网站前端追加缓存服务器（squid、varmish），存储数据库的前端增加缓存服务器（memcache、redis）

#### 6.3.4.5 文件系统的选型总结（SAS/SATA硬盘）

* 大量的小文件作业首选reiserfs
* 有门户的数据库mysql业务的选择xfs
* 视频下载、流媒体、数据库，小文件，默认等，选择ext4
* CDN网站加速服务，蓝汛的cache业务选择ext2

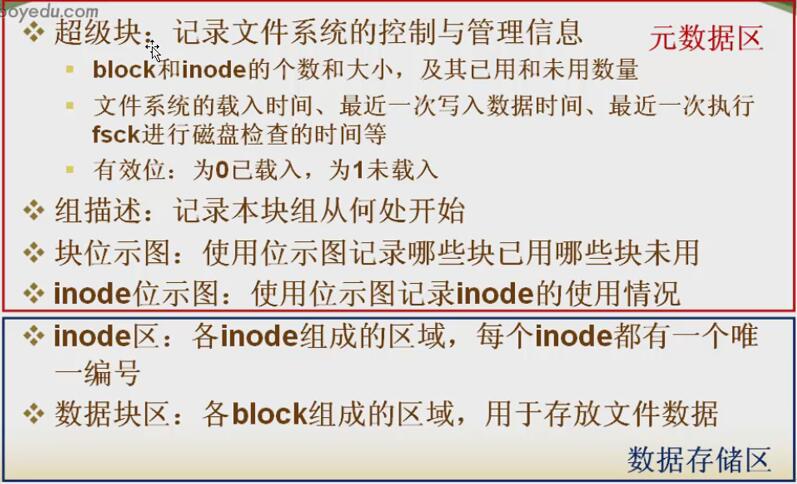
#### 6.3.4.6 查看linux系统支持的文件系统类型

[root@oldboy ~]# ls /lib/modules/`uname -r`/kernel/fs

autofs4 cachefiles configfs dlm exportfs ext3 fat fuse jbd jffs2 mbcache.ko nfs\_common nls ubifs xfs

btrfs cifs cramfs ecryptfs ext2 ext4 fscache gfs2 jbd2 lockd nfs nfsd squashfs udf

#### 6.3.4.7 超级块信息表



### 6.3.5 格式化与挂载

#### 6.3.5.1 格式化

[root@oldboy ~]# mkfs.ext4 /dev/sdb1

Writing inode tables: done

Creating journal (4096 blocks): done

Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 21 mounts or

180 days, whichever comes first. Use tune2fs -c or -i to override. 挂载21次或者180天后对磁盘进行检查

==============================解决办法=================================

不让系统对我们新建的分区进行磁盘检查

[root@oldboy ~]# tune2fs -c 0 -i 0 /dev/sdb1

tune2fs 1.41.12 (17-May-2010)

Setting maximal mount count to -1

Setting interval between checks to 0 seconds

#### 6.3.5.2 挂载

[root@oldboy ~]# mount /dev/sdb1 /mnt 挂载

[root@oldboy ~]# df -h 检查是否挂载上

Filesystem Size Used Avail Use% Mounted on

/dev/sda3 8.8G 1.6G 6.8G 19% /

tmpfs 491M 0 491M 0% /dev/shm

/dev/sda1 190M 35M 146M 19% /boot

/dev/sdb1 95M 1.6M 89M 2% /mnt

#### 6.3.5.3 开机自动挂载 /etc/fstab

[root@oldboy ~]# cat /etc/fstab

#

# /etc/fstab

# Created by anaconda on Thu Nov 16 15:52:25 2017

#

# Accessible filesystems, by reference, are maintained under '/dev/disk'

# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info

#

UUID=018908a2-d393-4cd2-918f-6547e16fabed / ext4 defaults 1 1

UUID=30f41846-e5c1-4c3e-8c57-b4fc0ca4a1b1 /boot ext4 defaults 1 2

UUID=e881b76e-7399-449f-a836-42d76279b69d swap swap defaults 0 0

tmpfs /dev/shm tmpfs defaults 0 0

devpts /dev/pts devpts gid=5,mode=620 0 0

sysfs /sys sysfs defaults 0 0

proc /proc proc defaults 0

第一列：磁盘分区

第二列：挂载点

第三列：文件系统类型

第四列：挂载参数

第五列：是否进行dump备份。dump备份是在电脑蓝屏或死机时把内存中的数据写入到磁盘中

第六列：是否进行fsck检查磁盘

注意：磁盘在未出现问题时，最好不要进行磁盘检查，因为这样磁盘容易出现问题，就像人没病去打针吃药一样，没病也会搞出来病

===========================查看UUID==================================

[root@oldboy ~]# blkid

/dev/sda1: UUID="30f41846-e5c1-4c3e-8c57-b4fc0ca4a1b1" TYPE="ext4"

/dev/sda2: UUID="e881b76e-7399-449f-a836-42d76279b69d" TYPE="swap"

/dev/sda3: UUID="018908a2-d393-4cd2-918f-6547e16fabed" TYPE="ext4"

/dev/sdb1: UUID="08c431a0-a15b-4b2f-afff-95e7facc97db" TYPE="ext4"

#### 6.3.5.4 特殊设备loop的挂载的方式

ls -lh /root/centos6.9\_x86\_64.iso

mkdir /tmp/centos\_dvd

mount -o loop /root/centos6.9\_x86\_64.iso /tmp/centos\_dvd

### 6.3.6 常见的linux磁盘操作的总结

#### 6.3.6.1 磁盘格式化

mkfs -t ext4（文件系统类型）/dev/sdb1（设备名）

#### 6.3.6.2 磁盘检查

fsck badblocks

[root@oldboy ~]# fsck -C -f -t ext4 /dev/sdb1 切记没坏的磁盘一定不要用

[root@oldboy ~]# badblocks -sv /dev/sdb1 该命令可以被fsck替代

#### 6.3.6.3 磁盘的挂载与卸载

mount /dev/sdb1(设备名) /mnt(挂载点)

df -hTi 查看是否挂载上

umount /mnt(挂载点)

### 6.3.7 安装xfs文件系统的操作实践

yum install xfsprogs kmod-xfs xfsdump xfsprogs-devel

rpm -qa xfsprogs kmod -xfs xfsdump xfsprogs-devel

modprobe xfs

lsmod |grep xfs

mkfs.xfs /dev/sdc

dd if=/dev/zero of=/dev/sdc bs=1M count=100

mount -t xfs -o loop /dev/sdc /mnt

### 6.3.8 企业案例

* 【问题】

java环境内存不够用

* 【原因】

程序写的有问题，导致内存泄漏，可能会占用swap

* 【解决的办法】

从根上解决的办法：协调开发，修改源代码程序

运维的临时解决方法：增大swap

创建一个块文件

[root@oldboy ~]# dd if=/dev/zero of=/tmp/lishiswap bs=1M count=100

100+0 records in

100+0 records out

104857600 bytes (105 MB) copied, 2.06999 s, 50.7 MB/s

查看块文件是否创建成功

[root@oldboy ~]# ls -lh /tmp/lishiswap

-rw-r--r-- 1 root root 100M Jan 14 20:09 /tmp/lishiswap

构建swap分区

[root@oldboy ~]# mkswap /tmp/lishiswap

mkswap: /tmp/lishiswap: warning: don't erase bootbits sectors

on whole disk. Use -f to force.

Setting up swapspace version 1, size = 102396 KiB

no label, UUID=db8fec73-11fc-44f9-b8d1-098bba946139

查看swap内存信息

[root@oldboy ~]# free -m

total used free shared buffers cached

Mem: 980 390 590 0 24 264

-/+ buffers/cache: 100 879

Swap: 767 0 767

开启swap交换分区

[root@oldboy ~]# swapon /tmp/lishiswap

在进行查看是否加上

[root@oldboy ~]# free -m

total used free shared buffers cached

Mem: 980 390 590 0 24 264

-/+ buffers/cache: 100 879

Swap: 867 0 867

开机自动挂载

方法1：将swapon /tmp/lishiswap 放入到/etc/rc.local中去

方法2：将其放到/etc/fstab文件中去

# 第7章 linux网络基础知识

## 7.1 硬件知识

### 7.1.1 网卡

* 【作用】

通过网卡调制或接收不同的信号，在单位时间内调制解调出相应数量信号的个数

* 【类型】

10M、100M、1000M

100Mbps等于每秒可以传输100M个bit

### 7.1.2 网线

* 网线至少需要4根铜线，其中第1、2脚起着接收信号的作用，3、6起着发信号的作用
* 【制作】

568A：俗称交叉线 绿橙蓝棕 46交换

568B：俗称直通线 橙绿蓝棕 46交换

* 【注意】

直通线是用来连接地位不平等的设备的，像交换机连接电脑

交叉线是用来连接地位平等的设备的，像电脑与电脑

只要1236四根线是通的，其他的通不通都无所谓

### 7.1.3 HUB

* 【功能】

多端口的转发器

* 【特点】

多端口的信号放大设备，共享介质的局域网

* 【缺点】

同一时刻只能一个方向传输数据

网络规模在10台一下

不能隔离冲突域和广播域

### 7.1.4 交换机

* 【定义】

实现多台设备间相互通信的设备

* 【特点】

工作在OSI的二层（数据链路层）

依据链路层的MAC地址，将以太网数据帧在端口之间进行转发

提供更多的接口，隔离冲突域

自行学习和维护MAC地址

* 【缺点】

不能隔离广播域

### 7.1.5 路由器

* 【功能】

实现不同网段之间的通信

* 【特点】

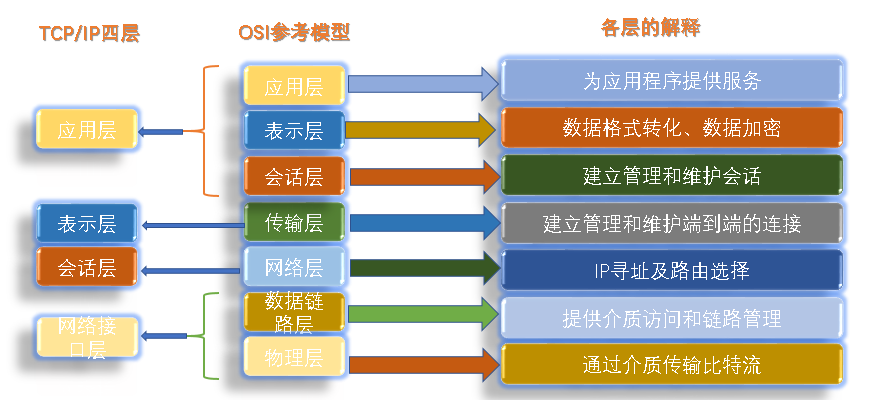
隔离广播

提供丰富的接口类型

支持丰富的链路层协议

## 7.2 网络基础原理

### 7.2.1 OSI七层模型及TCP/IP四层模型的简单总结

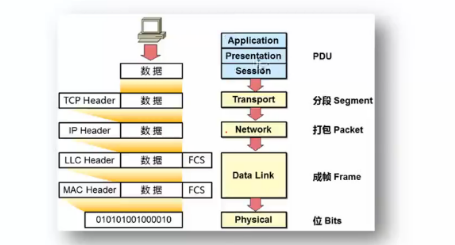


### 7.2.2 linux常用端口及对照的服务

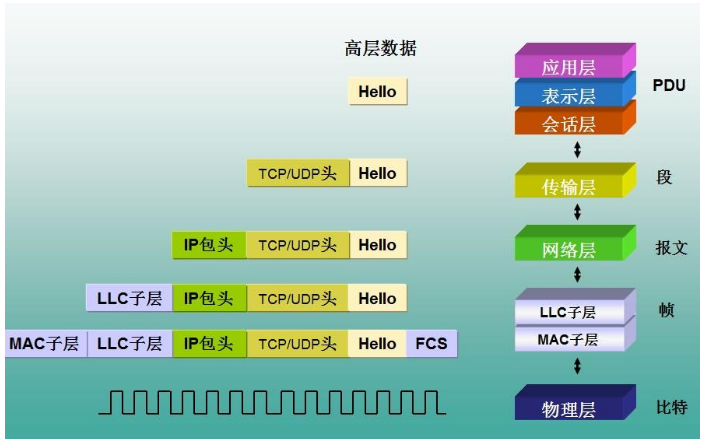
|  |  |  |
| --- | --- | --- |
| 端口号 | 服务名 | 作用 |
| 20 | ftp | 用于ftp的数据传输 |
| 21 | ftp | 用于控制文件传输信息的 |
| 22 | ssh | 用于远程加密连接 |
| 23 | telnet | 用于telnet远程连接 不加密 |
| 25 | smtp | 简单的邮件传输协议 |
| 53 | DNS | 域名解析服务 |
| 69 | tftp | 文件传输协议 |
| 80 | http | 超文本传输协议 |
| 109 | POP2 | 邮局协议2 用于接收邮件 |
| 110 | POP3 | 邮局协议3 用于接收邮件 |
| 115 | sftp | 安全文件传输协议 |
| 123 | ntp | 网络时间同步协议 |
| 143 | IMAP | 用于接收邮件的 |
| 161 | snmp | 简单的网络管理协议 |
| 443 | https | 安全的超文本传输协议 |
| 873 | rsync | rsync文件传输服务 |
| 2049 | nfs | 网络文件系统 |
| 3306 | mysql | mysql数据服务 |

### 7.2.3 数据包的封装与解封装的过程图解

#### 7.2.3.1 封装过程

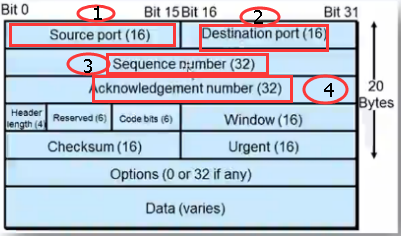


#### 7.2.3.2 解封装过程



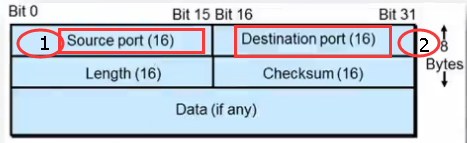
### 7.2.4 TCP/UDP的总结

#### 7.2.4.1 TCP



* 源端口：本地发起连接的端口（是随机分配的一般是大于1023的）
* 目的端口：即要访问服务的端口（例如22/23等）
* 序列号：因为在传输层会对上层的数据进行分段，因此需要对分段的数据进行编号，便于对数据的重组
* 校验和：用于对数据进行验证

#### 7.2.4.2 UDP



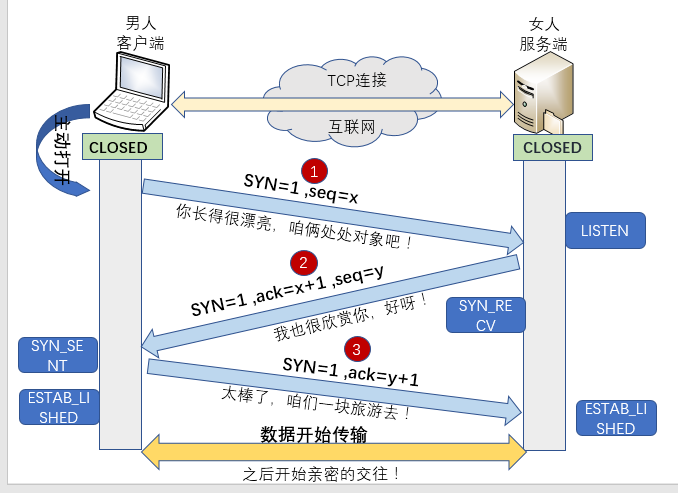
* 源端口与目的端口同上

#### 7.2.4.3 TCP与UDP的对比

|  |  |
| --- | --- |
| 传输控制协议TCP | 用户数据报协议UDP |
| 面向连接 | 无连接 |
| 可靠传输 | 不可靠传输 |
| 传输大量数据 | 传输少量数据 |
| 传输数据速度慢 | 传输数据速度快 |

### 7.2.5 TCP的三次握手与四次断开详解

#### 7.2.5.1 TCP的三次握手



* 【文字描述】

在最开始，客户端和服务器都处于CLOSE状态

在打开服务器时，服务器会创建scoket[源端口、目的端口、源IP、目的IP、TCP/UDP的标识符]开始监听，此时服务器处于LISTEN状态

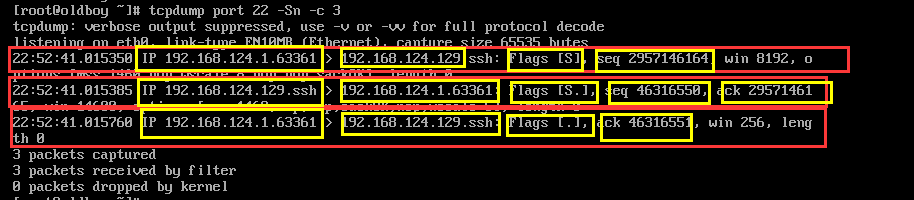
客户端向服务器发送SYN、seq，请求建立连接，此时的状态处于SYN\_SENT

服务器收到客户端的SYN，回复ack和syn报文，此时处于SYN\_RECV或SYN\_RCVD

客户端在收到服务端的SYN和ack后，马上恢复ack报文，此时处于ESTAB\_LISHED

服务器收到客户端的ack后，会直接进入ESTAB\_LISHED

* 【抓包的详细过程】



第一次握手：

客户端IP:192.168.124.1 服务器IP:192.168.124.129 Flags [s] seq 2957146164

第二次握手：

服务器IP:192. 168.124.129 客户端IP:192.168.124.1 Flags [s.] seq 46316550 ack 2957146165

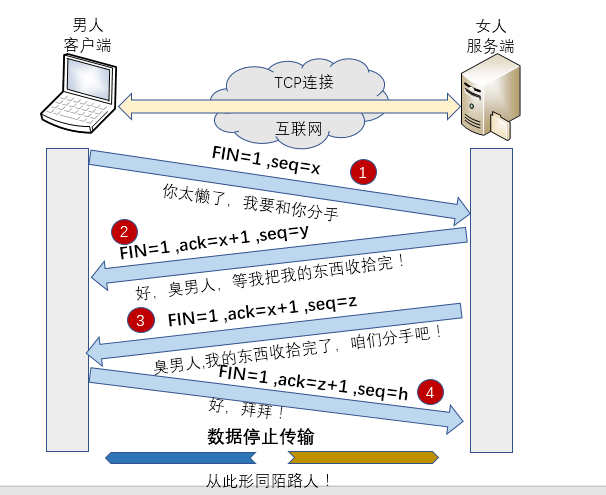
第三次握手

客户端IP:192.168.124.1 服务器IP:192.168.124.129 Flags [.] ack 46316551

* 【注意】

在三次握手的过程中可能会出现SYN泛洪的问题，产生的原因是在一端回应后，另一端不在进行回应，导致目标长期等待至超时，占用大量的目标资源，从而导致出现SYN泛洪问题

#### 7.2.5.2 TCP连接的四次断开



#### 7.2.5.3 TCP的包头及常用字段的解释



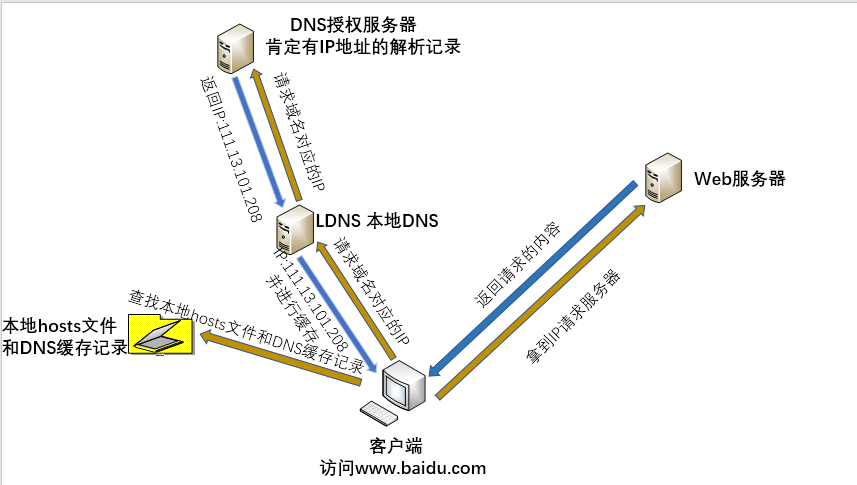
* ACK：表示验证字段
* SYN：位数置1，表示建立TCP连接
* FIN：位数置1，表示断开TCP连接

#### 7.2.5.4 TCP11种状态转移总结



### 7.2.6 访问网站的基本流程

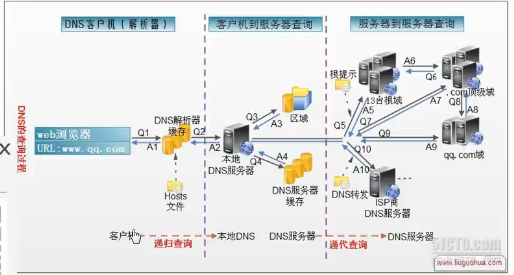
#### 7.2.6.1 图解



#### 7.2.6.2 文字的基本描述

* 客户端在浏览器中输入[www.baidu.com](http://www.baidu.com)网站网址按回车后，系统会首先查询本地hosts文件和DNS缓存记录，若有IP地址解析记录，则直接返回IP然后访问网站。
* 如果本地hosts文件和DNS缓存记录没有对应的IP地址解析记录，系统会把请求交给LDNS（即本地设置的DNS 像8.8.8.8等）进行解析，若LDNS中有相应的解析记录，则返回IP，如果没有，LDNS会请求其他的DNS服务器。
* LDNS经过一系列的请求，会找到网站的授权服务器，然后会把网站对应的IP返回给LDNS,并把域名对应的IP发送给客户端浏览器，并且LDNS会把域名对应的IP缓存起来，以便下次更快的返回相同的解析。
* 客户端在接收到网站对应的IP后，会请求IP地址对应的Web服务器，并将网站对应的IP在本地进行DNS缓存，Web服务器接收到请求并响应处理，并将客户端请求的内容返回给客户端浏览器

### 7.2.7 DNS解析原理



### 7.3 相关命令总结

#### 7.3.1 windows相关的命令

C:\Users\Administrator>ipconfig /all 查看ip

Windows IP 配置

主机名 . . . . . . . . . . . . . : MS-20171106GPCB

主 DNS 后缀 . . . . . . . . . . . :

**...**

默认网关. . . . . . . . . . . . . :

DHCPv6 IAID . . . . . . . . . . . : 486559830

DHCPv6 客户端 DUID . . . . . . . : 00-01-00-01-21-92-06-7E-54-EE-75-6E-04-5

C:\Users\Administrator>ipconfig /displaydns 查看dns缓存

Windows IP 配置

get.sogou.com

----------------------------------------

记录名称. . . . . . . : get.sogou.com

记录类型. . . . . . . : 1

生存时间. . . . . . . : 137

数据长度. . . . . . . : 4

部分. . . . . . . . . : 答案

A (主机)记录 . . . . : 123.125.125.86

C:\Users\Administrator>ipconfig /flushdns 清空dns缓存

Windows IP 配置

已成功刷新 DNS 解析缓存。

#### 7.3.2 linux相关的命令

* 【查看地址解析】

================================ping==================================

[root@oldboy ~]# ping [www.baidu.com](http://www.baidu.com)

PING www.a.shifen.com (61.135.169.125) 56(84) bytes of data.

64 bytes from 61.135.169.125: icmp\_seq=1 ttl=128 time=19.0 ms

===================================dig================================

[root@oldboy ~]# dig @8.8.8.8 www.baidu.com +trace

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.62.rc1.el6 <<>> @8.8.8.8 www.baidu.com +trace

; (1 server found)

;; global options: +cmd

. 17717 IN NS a.root-servers.net.

**...**

. 17717 IN NS l.root-servers.net.

. 17717 IN NS m.root-servers.net.

**...**

a.shifen.com. 1200 IN NS ns2.a.shifen.com.

;; Received 228 bytes from 61.135.165.235#53(61.135.165.235) in 54 ms

注意：用dig的时候也可以什么也不加，直接 dig 域名

=================================host==================================

[root@oldboy ~]# host www.baidu.com

www.baidu.com is an alias for www.a.shifen.com.

www.a.shifen.com has address 61.135.169.125

www.a.shifen.com has address 61.135.169.121

===============================nslookup================================

[root@oldboy ~]# nslookup

> baidu.com

Server: 192.168.124.2

Address: 192.168.124.2#53

Non-authoritative answer:

Name: baidu.com

Address: 111.13.101.208

Name: baidu.com

Address: 123.125.114.144

Name: baidu.com

Address: 220.181.57.217

【查看路由】

[root@oldboy ~]# route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.124.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

169.254.0.0 0.0.0.0 255.255.0.0 U 1002 0 0 eth0

0.0.0.0 192.168.124.2 0.0.0.0 UG 0 0 0 eth0

[root@oldboy ~]# netstat -rn

Kernel IP routing table

Destination Gateway Genmask Flags MSS Window irtt Iface

192.168.124.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth0

0.0.0.0 192.168.124.2 0.0.0.0 UG 0 0 0 eth0

[root@oldboy ~]# ip route show

192.168.124.0/24 dev eth0 proto kernel scope link src 192.168.124.129

169.254.0.0/16 dev eth0 scope link metric 1002

default via 192.168.124.2 dev eth0

[root@oldboy ~]# ip r s

192.168.124.0/24 dev eth0 proto kernel scope link src 192.168.124.129

169.254.0.0/16 dev eth0 scope link metric 1002

default via 192.168.124.2 dev eth0

【增加删除网关】 临时修改

[root@oldboy ~]# route del default gw 192.168.124.2

[root@oldboy ~]# route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.124.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

169.254.0.0 0.0.0.0 255.255.0.0 U 1002 0 0 eth0

[root@oldboy ~]# route add default gw 192.168.124.2

[root@oldboy ~]# route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.124.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

169.254.0.0 0.0.0.0 255.255.0.0 U 1002 0 0 eth0

0.0.0.0 192.168.124.2 0.0.0.0 UG 0 0 0 eth0

【增加删除路由】 **临时修改**

================================添加路由===============================

[root@oldboy ~]# route add -net 10.0.0.0/16 gw 192.168.124.2

[root@oldboy ~]# route add -host 10.0.0.1 dev eth0

================================删除路由===============================

[root@oldboy ~]# route del -net 10.0.0.0/16

[root@oldboy ~]# route del 10.0.0.1

============================增加默认路由===============================

[root@oldboy ~]# route del default gw 192.168.124.2

#### 7.3.3 配置和查看网络相关的信息

* 【ifconfig】

ifconfig [设备名] 查看网刊信息

ifconfig eth0: x ip/mask up 配置别名ip并启用

ifconfig eth0:x ip netmask 掩码 up 配置别名ip并启用

ifconfig eth0:x down 停用

配置别名永久生效的方法：编辑/etc/sysconfig/network-scripts/ifcfg-eth1:x

* 【ip】

ip addr 查看网络网卡信息

ip addr add ip/mask dev eth0:x 一个网卡配置多个ip的方法

### 7.4 tcpdump

* 【功能】一款强大的抓包工具
* 【常用参数】

|  |  |
| --- | --- |
| 选项 | 参数说明**：** |
| **-i** | 是interface的含义  是指我们有义务告诉tcpdump希望去监听哪一个网卡。这在一台服务器有多块网卡时很有必要 |
| **-nn** | 意思是说当tcpdump遇到协议号或端口号时，不要将这些号码转换成对应的协议名称或端口名称。  比如，众所周知21端口是FTP端口，我们希望显示21，而非tcpdump自作聪明的将它显示成FTP。 |
| **-X** | 告诉tcpdump命令，需要把协议头和包内容都原原本本的显示出来（tcpdump会以16进制和ASCII的形式显示），这在进行协议分析时是绝对的利器。  如果不进行原本的显示输出，当将收到的信息放入到其它专业分析包软件中分析时会有问题 |
| **-c** | 是Count的含义  这设置了我们希望tcpdump帮我们抓几个包。设置的是1，所以tcpdump不会帮我再多抓哪怕一个包回来 |
| **-w** | 将流量保存到文件中  tcpdump的-w方式是把raw packets（原始网络包）直接存储到文件中了，也就是存储的都是结构体形式，  而非是分析之后的文本格式的信息，因此是无法直接通过less命令查看的。 |
| **-r** | 读取raw packets文件 |
| **-n** | 指明协议 |

* 【应用实例】

[root@oldboy ~]# tcpdump tcp port 22 -c4 -i eth0 -nn

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode

listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes

18:23:31.592883 IP 192.168.124.129.22 > 192.168.124.1.58274: Flags [P.], seq 3537615929:3537616125, ack 3597215233, win 304, length 196

18:23:31.593276 IP 192.168.124.1.58274 > 192.168.124.129.22: Flags [.], ack 196, win 256, length 0

18:23:31.595035 IP 192.168.124.129.22 > 192.168.124.1.58274: Flags [P.], seq 196:472, ack 1, win 304, length 276

18:23:31.597068 IP 192.168.124.129.22 > 192.168.124.1.58274: Flags [P.], seq 472:636, ack 1, win 304, length 164

4 packets captured

4 packets received by filter

0 packets dropped by kernel

[root@oldboy ~]# tcpdump tcp port 22 and host 192.168.124.129 -c2

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode

listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes

18:24:49.155438 IP 192.168.124.129.ssh > 192.168.124.1.58274: Flags [P.], seq 3537619685:3537619881, ack 3597218077, win 304, length 196

18:24:49.155970 IP 192.168.124.1.58274 > 192.168.124.129.ssh: Flags [.], ack 196, win 253, length 0

2 packets captured

2 packets received by filter

0 packets dropped by kerne

### 7.5 开启路由转发功能

[root@oldboy ~]# vim /etc/sysctl.conf 修改配置文件

# Controls IP packet forwarding

net.ipv4.ip\_forward = 0 此项参默认是0 将其改为1即可

[root@oldboy ~]# sysctl -p 进行查看