**Linux**

**学**

**习**

**笔**

**记**

**毛**

**立**

**夫**

**2019**

**年**

**7**

**月**

**目录**

[1 VMware使用与Linux安装 5](#_Toc14764539)

[1.1 VMware使用 5](#_Toc14764540)

[1.1.1 VMware使用技巧 5](#_Toc14764541)

[1.1.2 VMware使用步骤 6](#_Toc14764542)

[1.2 CentOS安装 6](#_Toc14764543)

[1.2.1 系统分区 6](#_Toc14764544)

[1.2.2 CentOS安装步骤 7](#_Toc14764545)

[1.2.3 远程登录管理 8](#_Toc14764546)

[1.3 注意事项 9](#_Toc14764547)

[1.3.1 Linux系统注意事项 9](#_Toc14764548)

[1.3.2 服务器注意事项 9](#_Toc14764549)

[2 Linux常用命令 10](#_Toc14764550)

[2.1 快捷键与命令格式 10](#_Toc14764551)

[2.2 文件处理命令 10](#_Toc14764552)

[2.2.1 目录处理命令 10](#_Toc14764553)

[2.2.2 文件处理命令 13](#_Toc14764554)

[2.2.3 链接命令 14](#_Toc14764555)

[2.3 权限管理命令 14](#_Toc14764556)

[2.4 文件搜索命令 16](#_Toc14764557)

[2.5 帮助命令 18](#_Toc14764558)

[2.6 用户管理命令 18](#_Toc14764559)

[2.7 打包压缩命令 19](#_Toc14764560)

[2.8 网络命令 21](#_Toc14764561)

[2.9 关机重启命令 23](#_Toc14764562)

[3 软件包管理 24](#_Toc14764563)

[3.1 文本编辑器Vim 24](#_Toc14764564)

[3.2 软件包分类 25](#_Toc14764565)

[3.3 RPM包管理：rpm命令管理 25](#_Toc14764566)

[3.4 RPM包管理：yum在线管理 28](#_Toc14764567)

[3.5 源码包管理 30](#_Toc14764568)

[4 用户和用户组管理 31](#_Toc14764569)

[4.1 用户配置文件 31](#_Toc14764570)

[4.2 用户管理相关文件 32](#_Toc14764571)

[4.3 用户管理命令 32](#_Toc14764572)

[4.4 用户组管理命令 34](#_Toc14764573)

[5 权限管理 36](#_Toc14764574)

[5.1 ACL权限 36](#_Toc14764575)

[5.2 文件特殊权限 38](#_Toc14764576)

[5.3 文件隐藏属性 40](#_Toc14764577)

[5.4 系统命令sudo权限 40](#_Toc14764578)

[6 分区和文件系统 42](#_Toc14764579)

[6.1 文件系统分类 42](#_Toc14764580)

[6.2 文件系统常用命令 42](#_Toc14764581)

[6.3 磁盘分区 45](#_Toc14764582)

[6.4 磁盘配额 49](#_Toc14764583)

[6.5 逻辑卷管理器LVM 51](#_Toc14764584)

[7 Shell基础 55](#_Toc14764585)

[7.1 Shell概述 55](#_Toc14764586)

[7.2 Shell脚本执行方式 55](#_Toc14764587)

[7.3 Bash的基本功能 56](#_Toc14764588)

[7.4 Bash的变量 59](#_Toc14764589)

[7.5 Bash的运算符 62](#_Toc14764590)

[7.6 Bash环境配置文件 63](#_Toc14764591)

[8 Shell编程 66](#_Toc14764592)

[8.1 正则表达式 66](#_Toc14764593)

[8.2 字符截取命令 66](#_Toc14764594)

[8.3 字符处理命令 69](#_Toc14764595)

[8.4 条件判断 70](#_Toc14764596)

[8.5 流程控制 71](#_Toc14764597)

[9 服务与系统管理 73](#_Toc14764598)

[9.1 服务的分类 73](#_Toc14764599)

[9.2 服务管理 73](#_Toc14764600)

[9.3 进程管理 75](#_Toc14764601)

[9.4 任务管理 77](#_Toc14764602)

[9.5 系统资源查看 77](#_Toc14764603)

[9.6 定时任务 78](#_Toc14764604)

[9.7 SELinux 79](#_Toc14764605)

[10 日志与启动管理 83](#_Toc14764606)

[10.1 日志管理 83](#_Toc14764607)

[10.2 启动管理 84](#_Toc14764608)

[10.3 备份与恢复 85](#_Toc14764609)

[参考文献 88](#_Toc14764610)

[附录A Linux目录树 89](#_Toc14764611)

# VMware使用与Linux安装

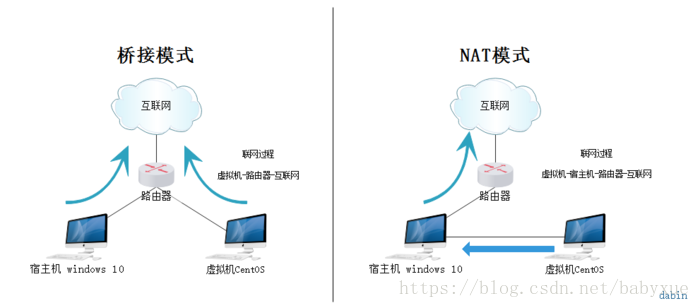
## VMware使用

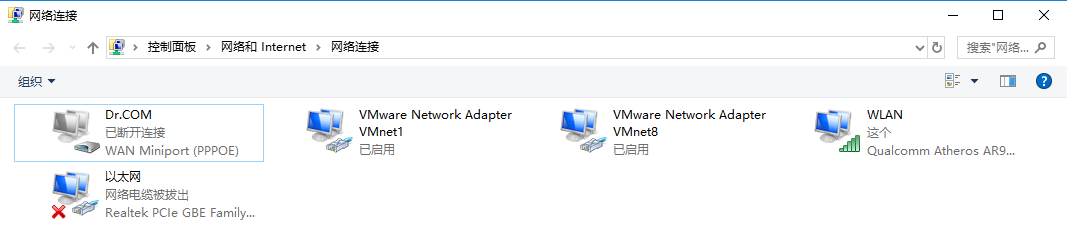
### VMware使用技巧

1. VMware虚拟机在Windows下安装与普通软件安装一样
2. 创建快照（保留虚拟机状态，以便返回）
3. 克隆（一般使用链接克隆，完整克隆相当于复制整个虚拟机）

虚拟机 –> 管理 -> 克隆

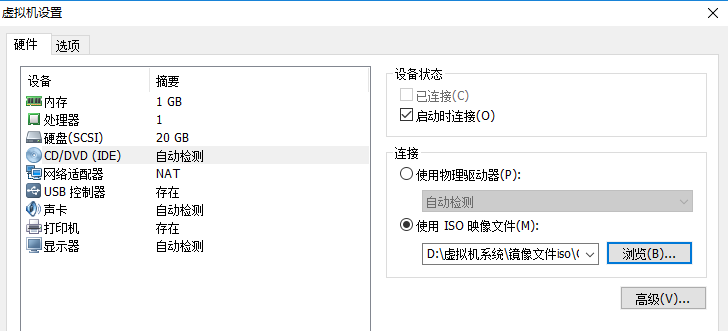
1. 三种网络连接区别（网络适配器中的设备状态要勾选已连接）
2. 桥接模式：虚拟机和宿主机在网络上是平级的关系，可以和宿主机、局域网其它计算机通信，具有独立IP地址和真实网卡
3. NAT模式：虚拟机必须先通过宿主机才能和外面进行通信，此时使用虚拟网卡VMnet8，不占用真实网段IP地址
4. 仅主机模式：虚拟机只能和宿主机进行通信，此时使用虚拟网卡VMnet1





### VMware使用步骤

1. 官网复制链接地址，使用迅雷下载ISO镜像文件（不要选功能少的Minimal）
2. 创建虚拟机 –> 典型安装 –> 稍后安装操作系统 -> Linux CentOS -> 选择位置 -> 完成
3. 虚拟机 -> 设置 -> 内存（不超过本机的一半，最小628M）-> CD/DVD –> 使用ISO镜像文件 –> 选择之前下载的ISO镜像文件 –> 声卡、打印机可以选择移除 -> 确定



## CentOS安装

### 系统分区

1. 分区类型：主分区、扩展分区、逻辑分区
2. 主分区加扩展分区最多4个（硬盘限制）
3. 扩展分区不能写入数据，只能包含逻辑分区，最多1个（操作系统限制）
4. 逻辑分区的设备文件名号码一定从5开始
5. 格式化：硬盘分为等大小数据块block，并建立iNode列表，目的是为了写入文件系统，扩展分区不能格式化
6. 建立设备文件名：在Linux系统中，每个设备都被当成一个文件对待

|  |  |
| --- | --- |
| 硬件 | 设备文件名 |
| SCSI/SATA/U盘 | /dev/sd[a-p] |
| 当前CD ROM/DVD ROM | /dev/cdrom |

1. 挂载：类似Windows分配盘符，指将一个设备挂接到一个已存在的目录上
2. 必须分区：/根分区、swap交换分区（建议内存2倍）
3. 推荐分区：/home分区、/boot启动分区（建议200M）

### CentOS安装步骤

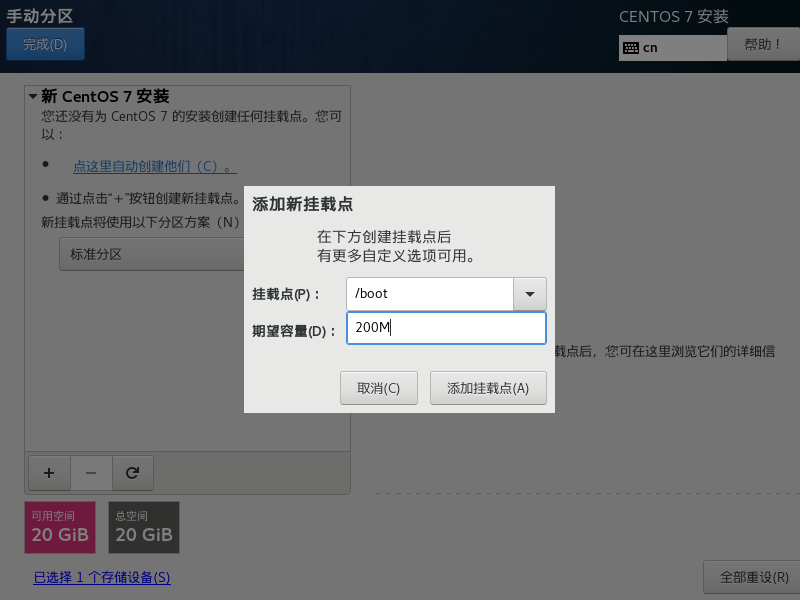
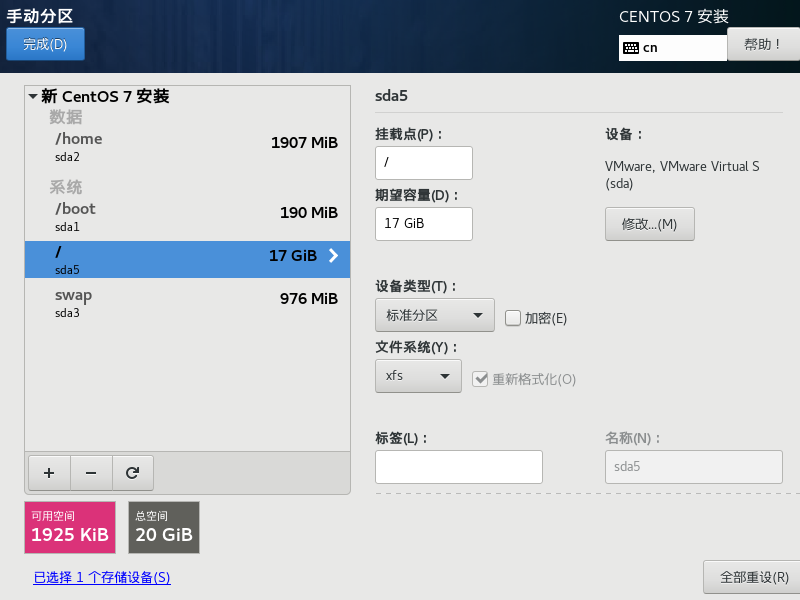
1. 开启虚拟机 –> Install CentOS直接回车安装 -> 选择中文 -> 继续
2. 日期和时间（亚洲上海） -> 软件选择 -> 安装位置 -> 我要配置分区 -> 完成 -> 配置分区 -> 完成 -> 接受更改 -> 网络和主机名 -> 右上角打开以太网，获取IP地址 -> 主机名默认或更改 -> 完成 -> 开始安装

软件选择：个人用户推荐桌面安装，作为服务器时推荐最小安装（此时很多命令使用不了，初学者不推荐），这里选择基本服务器安装

配置分区：选择标准分区 -> 点击“+”号，添加/boot分区，200M -> 继续添加swap分区，1G -> 继续添加/home分区，2G -> 添加根目录/分区，大小不填，系统自动将剩余空间分配给根目录

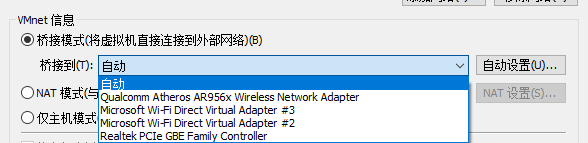
注：根目录/是sda5，不是sda4，即无论如何分区，逻辑分区一定是从5开始的，原因是当用户继续添加分区时，可以直接在sda4中划分

1. 设置ROOT密码 -> 典型密码：123456 -> 设置用户（可选） -> 完成

### 远程登录管理

1. 设置网络连接模式：确保本机网络能ping通Linux
2. 仅主机模式下ping不通 -> 未配置VMnet1的IP地址或防火墙未关闭
3. 桥接模式下ping不通 -> 宿主机存在有线和无线两块网卡，可能是有限通信桥接到无线 –> 选择编辑下的虚拟网络编辑器，更改设置



1. 远程连接管理工具Xshell：用于在Windows系统中操作Linux服务器
2. 有些Linux版本禁止root远程登录，需要创建普通用户
3. 注意一些谨慎操作，关机、重启、修改IP地址，连接会断掉
4. 需要输入Linux服务器IP地址、用户名和密码
5. 文件拷贝工具WinSCP：用于Windows系统与Linux系统间安全的复制文件

## 注意事项

### Linux系统注意事项

1. Linux严格区分大小写（Windows不区分小大写）
2. Linux中所有内容以文件形式保存，包括硬件
3. Linux不靠扩展名区分文件类型，有些扩展名只是为了方便管理员识别
4. Linux所有存储设备都必须手动挂载后才能使用，如硬盘、U盘、光盘（Windows是自动分配盘符，更人性化）
5. Windows下的程序不能直接在Linux中安装和运行

### 服务器注意事项

1. 远程服务器不允许关机，只能重启，重启前应该关闭服务
2. 不要在服务器访问高峰运行高负载命令
3. 远程配置防火墙时不要把自己踢出服务器
4. 合理分配权限，定期备份重要数据和日志
5. 指定合理的密码并定期更新

# Linux常用命令

## 快捷键与命令格式

1. 重要的快捷键

|  |  |  |  |
| --- | --- | --- | --- |
| 快捷键 | 作用 | 快捷键 | 作用 |
| 上/下箭头 | 上/下一条命令 | Ctrl + insert | 复制 |
| Tab | 补全命令或文件 | Shift + insert | 粘贴 |
| Ctrl + C | 终止操作 | Ctrl + U | 剪切光标前的命令 |
| Ctrl + L | 清屏，相当于clear | Ctrl + Y | 粘贴剪切的命令 |
| Shift + Page Up/Dn | 向上/下翻页 |  |  |

1. 命令格式

# 命令 [-选项] [操作对象]

1. 一个命令可以包括多个选项， 操作对象也可以是多个
2. 通常选项前会带“-”号，但完整的选项名会带“--”号，如“--help”
3. 命令太长时，可以使用“\[Enter]”转义，就能到下一行继续输入

## 文件处理命令

### 目录处理命令

1. 查看文件或目录：# ls [-adil] 目录名

* -a：all，显示所有文件，包括隐藏文件 （以“.”开头）



* -l：long，显示详细信息（“ll”相当于“ls -l”）

注：-rw-r--r--：类型和权限，-是文件类型（-表示文件，d表示目录，l表示软链接），rw-r--r--分别是所有者、所属组、其他人的权限（r表示读，w表示写，x表示执行）

1：链接，表示有多少文件名链接到此inode结点

root root：所有者和所属组，文件与用户关系有3种：所有者user（创建文件者，只能1个）、所属组group、其他人other

1652：文件大小，单位字节（-lh 人性化显示文件大小）

5-12 09:22：文件最后一次修改的时间mtime（没有创建时间这个概念，但有状态时间ctime和读取时间atime）

anaconda-ks.cfg：文件名



* -d：directory，只查看目录属性



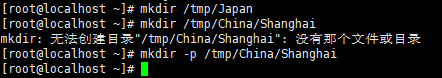
* -i：iNode，查看文件i结点



1. 创建目录：# mkdir [-p] 目录名

* -m：设置文件的权限，不使用默认权限
* -p： 递归创建目录

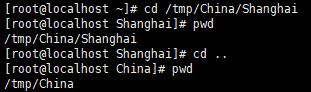
注：规划好目录结构，不要随便在根目录下创建，一般在tmp目录下



1. 切换目录：# cd 相对路径或绝对路径

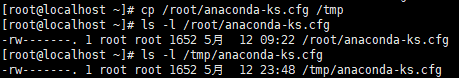
显示目前所在目录：# pwd

注：绝对目录一定由根目录/写起，相对路径则不是由/写起。“.”表示当前目录，“..”表示当前目录的上级目录，“-”表示前一个工作目录，“~”表示主目录（root用户位于/root，普通用户位于/home/用户名）

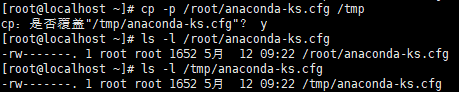


1. 复制文件或目录：# cp [-pr] 源文件 目标文件

注：文件的最后一次修改时间不同，使用-p指令解决（有时目标文件的权限、所有者、所属组也会因不同执行身份而改变，因此某些特殊文件，如密码文件复制时必须加上-p）。如果源文件有多个，则最后一个目标文件必须是目录

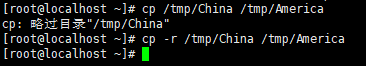


* -p：保留文件属性



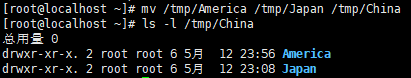
* -r ：复制目录

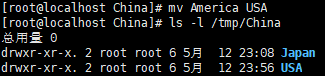
注：复制过程中可以同时修改文件名



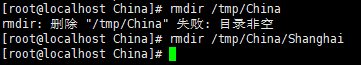
1. 剪切文件或改名：# mv源文件 目标文件

注：不用区分文件和目录，示例分别将两个文件同时剪切到一个目录下和改名





1. 删除空目录：# rmdir 目录名



1. 删除文件或目录：# rm [-fr] 文件或目录名

* -r：删除目录（慎重！）



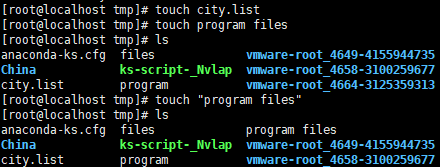
* -f：强制执行（不会出现提示是否删除）



### 文件处理命令

1. 创建文件：# touch 文件名

注：/不能作为文件名，同时不建议使用空格等特殊字符。该命令还可用于修改文件的读取时间和修改时间



1. 显示文件内容：# cat [-n] 文件名

* -n：显示行号



反向显示文件内容：# tac 文件名



1. 分页显示文件内容：# more 文件名

注：适合内容较多的文件，空格（向下翻页），b（向上翻页，只对文件有用，对管道无用），回车（翻行），q（退出），可使用“/加关键词”进行查找信息

可向上翻页的分页显示文件内容：# less 文件名

注：空格或PageDown（向下翻页），PageUp（向上翻页），上箭头（向上翻一行），同样可以使用“/加关键词”进行查找信息，n（next）查找下一个

1. 显示文件的前面几行：# head [-n] 文件名



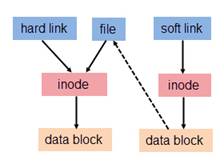
显示文件的后面几行：# tail [-fn] 文件名

* -n：指定显示的行数（默认10行）
* -f：动态显示文件末尾内容（可用于监控日志）

### 链接命令

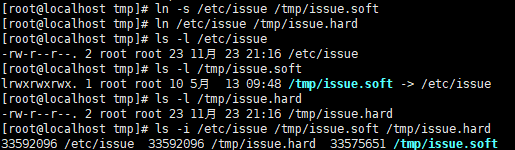
1. 生成链接文件：# ln [-s] 源文件 目标文件

* -s：创建软链接（默认创建硬链接）



注： 硬链接特征，相当于属性拷贝+同步更新，通过i结点识别（1个文件仅占有1个i结点， 1个i结点不一定只对应1个文件）。最大优点是安全，当删除任何一个链接文件时，只要文件计数不为0，真正的文件就不会被删除；缺点是不能跨分区，且不能链接目录（软链接可以跨分区、链接目录，因此使用较广）

软链接特征，类似Windows快捷方式，只是符号链接，指向源文件。它是一个独立的新文件，属性一定是lrwxrwxrwx，同时会占用inode，增加一定的磁盘空间，该文件很小，示例中仅占10字节，即是文件名的10个字母



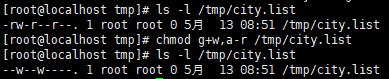
## 权限管理命令

1. 改变文件或目录权限：# chmod [ {u/g/o/a} {+/-/=} {r/w/x} ] [文件或目录]

* -R：递归修改

注：只有管理员root和文件所有者可以更改权限，另外可以用数字表示权限，其中r=4、w=2、x=1，则rwx=7，rw=6。目录与文件的权限意义并不是相同的，如删除一个文件的权限是对所在的目录有写权限，再如当开放目录给任何人浏览时，应该至少给予r和x权限，但w权限不可随便给，见下表。

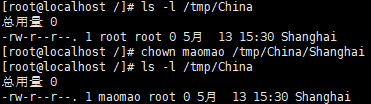
|  |  |  |  |
| --- | --- | --- | --- |
| 代表字符 | 权限 | 对文件的含义 | 对目录的含义 |
| r | 读 | 可以查看文件的内容 | 可以列出目录中的内容 |
| w | 写 | 可以修改文件的内容 | 可以在目录中创建删除文件 |
| x | 执行 | 可以执行文件 | 可以进入目录 |



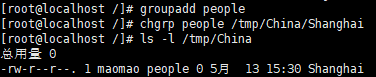


1. 改变文件或目录的所有者：# chown [用户] [文件或目录]

注：只有管理员root可以修改所有者，常见使用场景是复制文件给其他人时，新文件的权限不会改变，这时需要修改所有者和所属组



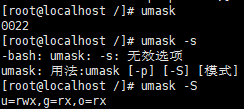
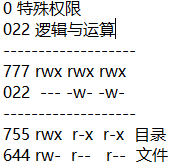
1. 改变文件或目录的所属组：# chgrp [用户组] [文件或目录]



1. 显示、设置文件的缺省权限：# umask [-S]

* -S：以rwx形式显示新建文件的缺省权限

注：Linux中会把新建文件的可执行权限x去掉，因为一般文件通常用于记录数据，不需要执行权限。但是x权限对目录非常重要，所以Linux没有去掉。root的umask默认是022，一般用户默认是002，同时不建议修改文件的缺省权限

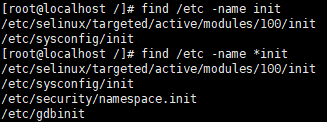
 

## 文件搜索命令

1. 文件搜索：# find [搜索范围] [选项] [匹配条件]

注：“\*”可以匹配任意字符，“?”匹配单个字符。“-a”表示连接与，“-o”表示连接或，“-exec/-ok 命令 {} \;”表示对搜索结果执行额外操作（不支持命令别名，{}表示找到的结果，-exec和\;表示额外命令的开始和结束，\用于转义）。find命令功能强大，但是查找速度慢，且相当消耗硬盘资源，优先使用locate和whereis命令

* -name：根据文件名搜索（-iname：不区分大小写）



* -size：根据大小搜索（+n大于、-n小于、n等于）

注：1数据块是512字节，即0.5K，如100MB=102400KB，204800数据块



* -user：根据所有者搜索

-group：根据所属组搜索



* -amin（access） 根据访问时间改变（单位：分钟）

-cmin（change）根据文件属性改变

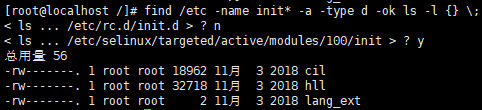
-mmin（modify）根据文件内容改变

注：相应地，有以天为单位的选项-atime、-ctime、-mtime

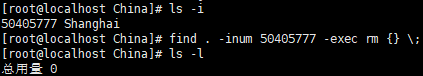


* -type：根据文件类型查找（f文件、d目录、l软链接文件）





* -inum：根据i结点查找



* -perm：根据文件权限查找（mode表示刚好等于，-mode表示必须包含全部mode权限，/mode表示包含任一mode权限）



1. 在文件资料库中查找文件：# locate [-i] 文件名

* -i：不区分大小写

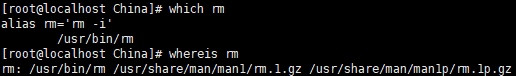
注：维护并定期更新文件资料库（命令updatedb手动更新），查找文件速度快，但是不能查找所有文件，有些文件没有收录在文件资料库中，如tmp目录



1. 搜索命令所在目录及别名信息：# which命令

搜索命令所在目录及帮助文档路径：# whereis 命令

注：whereis命令只在几个特定的目录中查找，因此查找速度快，有些文件也找不到



1. 在文件中搜索字串匹配的行并输出：# grep [-iv] [指定字串] [文件]

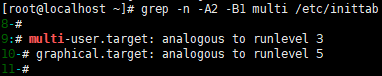
* -i：不区分大小写
* -n：输出行号



* -v：排除指定字串



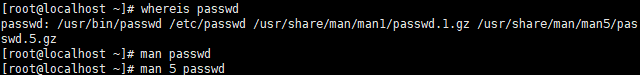
* -A/B：后面接数字，后续/前面n行也列出



## 帮助命令

1. 获得帮助信息：# man [命令或配置文件名]

注：空格或Page Down向下翻页，Page Up向上翻页，回车翻行，q退出，/加关键词查找，n继续查找。配置文件不要写绝对路径，只需要写配置文件的名称（1表示命令的帮助，5表示配置文件的帮助，如果同时存在，会优先选择命令）



1. 获得Shell内置命令的帮助信息：# help 命令

注：使用which找不到命令位置就是Shell内置命令，此时若使用man显示的是Shell帮助信息，不适用（查询选项信息使用“命令 --help”）

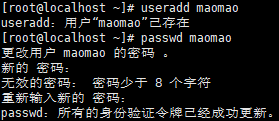


## 用户管理命令

1. 添加新用户：# useradd 用户名

设置用户密码：# passwd用户名

注：管理员root可以更改所有人的密码，密码格式无限制；用户只能改自己的密码，且密码不能过于简单

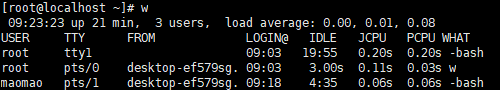


1. 查看登录用户信息：# who

注：登录用户名、登录终端（tty本地终端，pts远程终端）、登录时间



1. 查看登录用户详细信息：# w



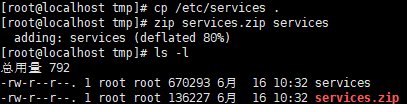
## 打包压缩命令

1. 压缩文件或目录：# zip [-r] [压缩后文件名] [文件或目录]

解压缩.zip的压缩文件：# unzip [压缩文件]

* -r：压缩目录

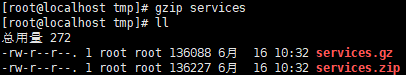
注：压缩效率最低，且保留原文件，文件格式最好指定为“.zip”，是Windows和Linux都支持的压缩格式



1. 压缩文件：# gzip [文件]

解压缩.gz的压缩文件：# gunzip [压缩文件]

注：只能压缩文件，不能压缩目录，压缩后不保留原文件，文件格式为“.gz”。gzip压缩效率比zip高，可以使用-d选项进行解压缩，下面两个压缩命令同样支持

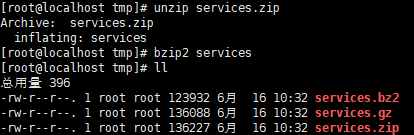


1. 压缩文件：# bzip2 [-kd] [文件]

解压缩.bz2的压缩文件：# bunzip2 [压缩文件]

* -k：压缩或解压缩时保留原文件，而不会删除

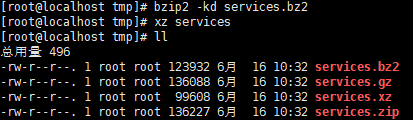
注：只能压缩文件，压缩效率比gzip好，文件格式为“.bz2”



1. 压缩文件：# xz [-kd] 文件

解压缩.xz的压缩文件：# unxz [压缩文件]

注：只能压缩文件，压缩效率比bzip2还好，但耗时较大，文件格式为“.xz”



1. 打包并压缩：# tar [-z / -j / -J] [cv] [-f 待建立的新文件名] [目录名]

查看文件名：# tar [-z / -j / -J] [tv] [-f 既有的tar文件名]

解压缩：# tar [-z / -j / -J] [xv] [-f 既有的tar文件名]

注：很多压缩命令仅能压缩文件，因此当文件较多时，可以利用打包将很多文件打包为一个文件，再进行压缩。选项的顺序一般是可以调换的，但是-f选项必须立刻接要处理的文件名，所以必须放在所有选项的最后一个，为避免出错，单独写出

* -c：打包（-x：解包）
* -t：查看打包文件中含有的文件
* -v：显示详细信息
* -f 文件名：指定文件名（或压缩文件名）
* -z / -j / -J：打包同时进行压缩或解压缩（分别使用gzip、bzip2、xz进行压缩或解压缩，文件名最好分别指定为“.tar.gz”、“.tar.bz2”、“.tar.xz”）
* -p：保留数据原本权限与属性，常用于备份

注：解压.tar.bz2需要先安装bzip2命令。为方便截图，示例中未使用-v选项



## 网络命令

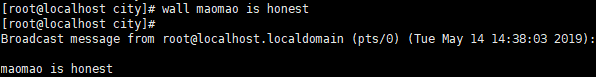
1. 给在线用户发消息：# write [用户名]

发广播消息：# wall [消息]

注：“Ctrl+D”保存结束，“Ctrl+退格键”删除写错的文字







1. 查看发送电子邮件：# mail [-s “邮件标题”] [用户名]

注：不需要用户在线，“Ctrl+D”保存结束，利用的是本地内存，并不是使用搭建的邮件服务器。可以直接使用mail接收。



1. 测试网络连通性：# ping [-c] 域名或IP地址

* -c：指定发送次数



1. 显示数据包到主机间的路径：# traceroute域名或IP地址



1. 查看和设置网卡信息：# ifconfig [网卡名] [IP地址]

配置网络：# setup

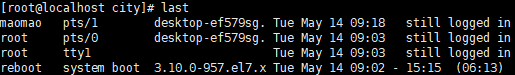
注：直接使用ifconfig可以查看网卡名，只能临时更改IP地址，重启后丢失，若要永久改变IP地址，需要改变配置文件选项。

setup使用窗口模式，只有“red hat”系列才有该命令，可以永久配置IP地址，配置后需要重启服务（命令service network restart）



1. 列出目前与过去登录系统的用户信息：# last

检查某特定用户上次登录的时间：# lastlog

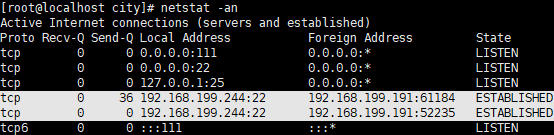




1. 显示网络相关信息：# netstat [-tulrn]

* -tlunp：查看本机监听的端口
* -an：查看本机所有的网络连接
* -rn：查看本机路由表

注：-t表示TCP协议，-u表示UDP协议，-l表示监听，-r表示路由，-n表示IP地址和端口号，-p表示网络服务进程PID



1. 下载文件：# wget [-c] 网址URL

* -c：支持断点续传

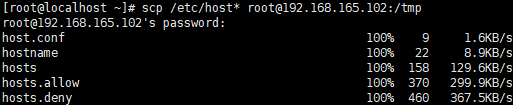


1. 文件异地直接上传：# scp [-pr] 文件名 [用户名@]主机:目录名

文件异地直接下载：# scp [-pr] [用户名@]主机:文件名 目录名

* -p：保留文件的权限属性
* -r：上传下载目录

注：可以直接将scp换为远程同步命令rsync。两者的区别是：scp是复制，若目的地文件不存在则新建，若存在则覆盖；rsync是同步，比较两边文件是否相同，若相同则什么都不做，若存在差异就直接更新。因此，同步文件时rsync会快一些。



## 关机重启命令

1. 关机或重启：# shutdown 时间

* -c：取消前一个关机命令
* -h：关机（halt）
* -r：重启（reboot）

注：now表示现在，20:30表示具体时间，10表示十分钟后执行操作，默认1分钟后自动执行。其他关机命令有systemctl poweroff；其他重启命令有systemctl reboot



1. 退出登录：# logout



# 软件包管理

## 文本编辑器Vim

1. Vim工作模式和常用操作



1. 一般命令模式：复制粘贴、查找替换等

|  |  |
| --- | --- |
| 移动光标 | |
| Page Dn/Page Up | Page Dn屏幕向下移动一页，Page Up屏幕向上移动一页 |
| gg，G | gg移动到第一行，G移动到最后一行 |
| nG，:n | 作用相同，移动到第n行 |
| 0，$ | 0移动到行首，$移动到行尾 |
| 查找、替换 | |
| /word | 搜索字符串，“:set ic”忽略大小写，“n/N”查找下/上一个 |
| ?word | 搜索字符串（从后往前），“n/N”查找上/下一个 |
| :1,$s/old/new/g | 全文替换指定字符串（gc询问确认） |
| :n1,n2s/old/new/g | 在第n1与n2行间替换指定字符串 |
| 删除 | |
| x，nx | x删除光标所在字符，nx删除光标后n个字符 |
| dd，ndd | dd删除光标所在行，ndd删除光标后n个行 |
| dG，D | dG删除光标所在行到行尾，D删除光标所在字符到行尾 |
| :n1,n2d | 删除第n1与n2之间的行 |
| 复制、剪切、粘贴 | |
| yy，nyy | yy复制当前行，nyy复制当前行以下n行 |
| dd，ndd | dd剪切当前行，ndd剪切当前行以下n行 |
| p，P | p粘贴在当前行下，P粘贴在当前行上 |
| u，. | u取消上一步操作，.重复上一个操作 |

1. 编辑模式：插入替换等

|  |  |
| --- | --- |
| a，A | a在光标所在字符后插入，A在光标所在行行尾插入 |
| i，I | i在光标所在字符前插入，I在光标所在行行首插入 |
| o，O | o在光标下插入新行，O在光标上插入新行 |
| r，R | r替换光标所在字符一次，R一直替换光标所在字符 |

1. 命令行模式：保存离开等

|  |  |
| --- | --- |
| :w [filename] | :w保存修改，:w filename另存为指定文件 |
| :q! | 不保存修改强制退出，“!”是强制的意思 |
| :wq，:wq! | :wq保存并退出，:wq!适用于文件权限为只读 |
| :set nu，:set nonu | :set nu显示行号，:set nonu取消行号 |

1. Vim使用技巧

|  |  |
| --- | --- |
| :r [filename] | 将filename文件内容加到光标所在行后面 |
| ! command | 暂时离开vi执行命令的显示结果，回车键返回vi |
| :r ! command | 直接将命令的执行结果加到光标所在行后面 |
| :map 快捷键 触发命令 | 如执行“:map [Ctrl+v+p] [I#<ESC>]”后，可以使用快捷键“Ctrl+p”注释当前行 |
| :n1,n2s/^/#/g  :n1,n2s/^#//g  :n1,n2s/^/\/\//g | 使用“#”连续行注释，“^”表示行首  去掉行首的注释，不加“^”会把中间的注释也去掉  使用“//”连续行注释，“\/”转义“/” |
| ab word1 word2 | 输入word1会自动替换为word2 |

注：快捷键命令重启会失效，需要写入配置文件~/.vimrc（默认不存在，需手动创建）

## 软件包分类

1. 源码包：优点是开源、自由选择所需功能；缺点是编译安装时间长、步骤多
2. 二进制包（RPM包、系统默认包）：优点是安装速度快；缺点是经过编译，不能看到源代码，功能选择不灵活，具有依赖性

## RPM包管理：rpm命令管理

1. RPM包命名规则

anaconda 软件包名 21.48.22.147软件版本

1软件发布的次数 el7.centos适合的Linux平台

X86\_64适合的硬件平台 rpm包扩展名



注：包全名：操作没有安装的软件包时，使用包全名，如安装、升级；

包名：操作已经安装的软件包时，使用包名，如查询、卸载

1. RPM包依赖性

树形依赖：a->b->c

环形依赖：a->b->c->a

模块依赖：查询网站[www.rpmfind.net](http://www.rpmfind.net)

1. RPM安装：# rpm –ivh 包全名

* -i：安装（install）
* -v：显示详细信息（verbose）
* -h：显示进度（hash）

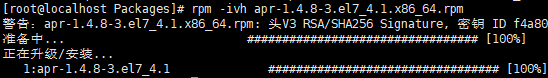
注：安装前需要先挂载光盘镜像，示例中安装Apache，需要在Packages路径下执行，目的是使用Tab键可以找到需要安装的包，第1个是主包，后面3个是附加包。



注：安装主包时提示存在依赖，需要先安装其它4个包或模块（.so），然后再安装主包，卸载时顺序相反。安装模块需要打开网站搜索“libapr-1.so.0”，根据CentOS版本找到对应包安装。另外，“/etc/mime.types”文件在“mailcap”







1. RPM升级：# rpm -Uvh 包全名

RPM卸载：# rpm -e 包名

* -U：升级（upgrade）
* -e：卸载（erase）

注：使用-Uvh升级时，若该软件未安装，系统会直接安装；使用-Fvh则不会被安装，即只有已安装的软件才会被升级

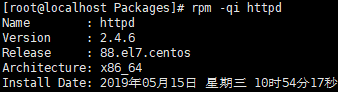
1. RPM查询
2. 查询是否安装：# rpm [-qa] [包名]

* -q：查询是否安装（query）
* -a：查询所有（all）



1. 查询详细信息：# rpm -qi 包名

* -i：查询详细信息（information）
* -p：查询未安装包信息（包全名package）



1. 查询包中文件安装位置：# rpm -ql 包名

* -l：列表（list）



1. 查询系统文件名属于哪个包：# rpm -qf 系统文件名

查询软件包的依赖性：# rpm -qR 包名

* -f：文件（file）
* -R：依赖性（requires）



1. RPM包校验：rpm -V 已安装的包名

注：应用场景如不小心误删或丢失了某个软件的文件时，就可以用来验证原来的文件，它使用/var/lib/rpm下的数据库来比对当前系统环境下的所有安装文件。另外，rpm命令基本上用于查询和校验，安装主要还是使用yum

* -V：校验指定RPM包中的文件（verify）

|  |  |
| --- | --- |
| S | 文件大小是否改变 |
| M | 文件类型或权限（rwx）是否改变 |
| 5 | 文件MD5校验和是否改变（可以看成文件内容是否改变） |
| D | 设备的主、次代码是否改变 |
| L | 文件路径是否改变 |
| U | 文件的所有者是否改变 |
| G | 文件的所属组是否改变 |
| T | 文件的修改时间是否改变 |

|  |  |
| --- | --- |
| c（config file） | 配置文件 |
| d（documentation） | 普通文档 |
| g（ghost file） | 鬼文件（少见，即该文件不应该被RPM包含） |
| l（license file） | 授权文件 |
| r（read me） | 描述文件 |



1. RPM包文件提取：# rpm2cpio 包全名 | cpio -idv . 文件绝对路径

注：主要用于修复丢失的文件。rpm2cpio命令将rmp包转换为cpio格式，cpio命令用于创建软件档案文件和从档案文件中提取文件，“.”表示当前路径。

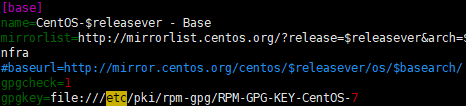
* -i：copy-in模式，还原
* -d：还原时自动新建目录
* -v：显示还原过程

## RPM包管理：yum在线管理

1. 网络yum源配置文件：# vim /etc/yum.repos.d/CentOS-Base.repo

注：yum解决了包依赖问题，自动安装，除了CentOS外其它版本需要收费。默认使用“CentOS-Base.repo”，yum源后缀名为“.repo”





注：[base]表示软件源的名字，name表示软件源的含义，mirrorlist表示软件源可以使用的镜像站，baseurl表示软件源的实际地址（最重要），enable表示是否启用，gpgcheck表示是否查看数字签名，gpgkey表示数字签名的公钥文件位置

1. yum命令
2. 查询所有可用软件包列表：# yum list

搜索服务器上和关键字相关的包：# yum search 关键字



1. 软件包安装：# yum -y install 包名

软件包升级（慎用）：# yum -y update 包名

软件包卸载（慎用）：# yum -y remove 包名

* -y：自动回答yes

注：gcc是C语言编辑器包，升级时不加包名会更新所有，包括内核，容易导致服务器崩溃；卸载时可能会误删依赖的内核包，也容易导致服务器崩溃。



1. 列出所有可用的软件组列表：# yum grouplist

软件组安装：# yum groupinstall 软件组名

软件组卸载：# yum groupremove软件组名

1. 光盘yum源搭建步骤

注：没有网络也可以使用，且下载快，但是版本可能不是最新的。Linux配置文件的注释#一定是顶格写，否则会导致一些命令报错

* 挂载光盘：# mount /dev/cdrom /mnt/cdrom
* 让网络yum失效：# cd /etc/yum.repos.d

注：再使用命令“mv”将后缀“.repo”改为“.repo.bak”使其暂时失效（除了CentOS-Media.repo）

* 修改光盘yum源文件：# vim CentOS-Media.repo

注：首先将“baseurl=file:///media/CentOS/”改为光盘挂载地址，如“baseurl=file:///mnt/ cdrom /”，然后注释下面两行（顶格注释），最后将“enabled=0”改为“enabled=1”进行启用



## 源码包管理

1. 源码包和RPM包的区别（安装位置不同）

注：RPM包会自动安装在系统默认路径，除了使用绝对路径，安装的服务还可以使用命令service来管理，其实质还是使用绝对路径，如apache的启动方法有“# /etc/rc.d/init.d/httpd start”和“service httpd start”。

源码包需要手动指定位置，一般是“/usr/local/软件名”，因此不能使用命令service，只能使用绝对路径进行服务的管理

1. 源码包安装（略）和脚本安装

注：脚本安装是人为把安装过程写成自动安装的脚本，简化源码包安装的过程，类似于Windows下软件的安装方式。

# 用户和用户组管理

## 用户配置文件

1. 用户信息文件/etc/passwd

* 查看配置文件帮助信息：# man 5 passwd



注：命令不能写绝对路径。管理员使用帐号，系统使用用户ID，二者关系类似与域名和IP地址；用户UID分为超级用户（0）、系统用户（1-999，不能登录或删除）、普通用户（1000-60000）；所属组分为初始组（只能1个，不建议修改）和附加组（可以多个）；家目录分为超级用户（/root）、普通用户（/home/用户名）

使用命令groups第一个输出的用户组即为有效用户组，应用场景如用户maomao初始组为maomao，附加组为group1、group2，那么用户新建一个文件或目录时，新文件的用户组就会是有效用户组。另外使用命令“newgrp 用户组”可更改有效用户组，命令exit退出环境

* 显示文件内容：# vim /etc/passwd



1. 密码影子文件/etc/shadow

* 查看文件权限：# ll /etc/shadow

注：该文件只有root才可以读写



* 查看配置文件：# man 5 shadow

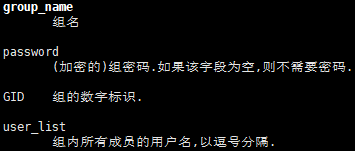
注：第1字段为用户名；第2字段为加密密码（字段前加\*或!让密码暂时无效）；第3字段为密码最后一次修改日期（时间戳）；第4字段为两次密码的修改间隔时间（与第3字段相比）；第5字段为密码有效期（与第3字段相比）、第6字段为密码修改到期前的警告天数（与第5字段相比）、第7字段为密码到期后的宽限时间（与第5字段相比）、第8字段为账号失效时间（时间戳）、第9字段保留

* 显示文件内容：# vim /etc/shadow



1. 组信息文件/etc/group和组密码文件/etc/gshadow

* 查看配置文件：# man 5 group



* 显示文件内容：# vim /etc/group



## 用户管理相关文件

1. 用户的家目录：/root（超级用户）、/home/用户名（普通用户）

注：超级用户权限550，普通用户权限700。将普通用户修改为超级用户，只需使用“vim /etc/passwd”将对应用户的UID修改为0

1. 用户的邮箱：/var/spool/mail/用户名
2. 用户模板目录：/etc/skel/

注：目录中含有隐藏文件，在创建用户时会复制过去

## 用户管理命令

1. 添加用户：# useradd [-cG] 用户名

注：命令useradd实际上操作的是上面介绍的7个文件或目录，用户默认值文件位于“/etc/default/useradd”和“/etc/login.defs”，即useradd添加用户时的默认值



* -c：指定用户的说明
* -G：指定用户的附加组



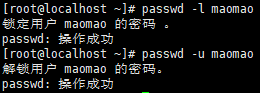
1. 修改用户密码：# passwd [-luS] 用户名

注：设置自身密码时可以省略用户名，若root不小心省略就会修改自己的密码。另外，root设置的密码系统都会接受，而普通用户设置的密码必须符合规范

* --stdin：通过前一个管道数据，作为密码输入（shell脚本）
* -S：查询用户的密码状态（仅root）



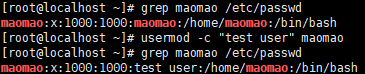
* -l：锁定用户（实际上是在/etc/shadow加密字串前加“!!”）
* -u：解锁用户（去掉“!!”）



1. 修改用户信息：# usermod [-LU] 用户名

注：选项和useradd基本相同，区别在于useradd操作的是新用户，而usermod操作的是已经存在的用户

* -L：锁定用户（实际上是在/etc/shadow加密字串前加“!”）
* -U：解锁用户（去掉“!”）



1. 修改用户密码状态：# chage -d 0 用户名

* -d：修改密码的最后一次更改日期（shadow第3字段）

注：示例场景如用户一旦登陆就要修改密码，也可修改其它字段（man chage）



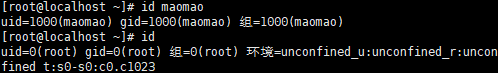
1. 删除用户：# userdel -r 用户名

* -r：删除用户同时删除用户家目录

注：userdel实际上删除的仍然是那7个文件或目录，为了确保完全删除，可以先使用“find / -user 用户名”找出整个系统属于该用户的文件，然后再删除



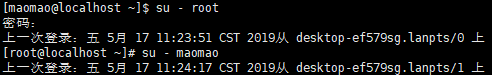
1. 查询用户UID等信息：id [用户名]



1. 用户身份切换：# su - 用户名

* -：使用“-”代表连带用户的环境变量一起切换（否则容易报错）

注：使用exit才能退出su环境，若要执行完毕就自动恢复原来身份，可使用-c选项。su命令的缺点是切换为root时需要输入root密码，容易造成密码泄露，此时可使用sudo来处理



* -c：仅执行一次命令，而不切换用户身份

注：示例场景比如暂时使用root身份添加新用户



## 用户组管理命令

1. 添加用户组：# groupadd 组名



1. 修改用户组：# groupmod 组名

* -n：后接新组名，修改组名



1. 删除用户组：# groupdel 组名

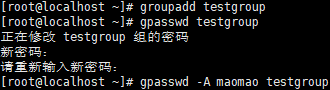
注：若该组是某个用户的初始组（附加组不影响），必须先修改或删除该用户

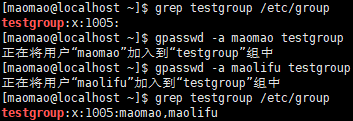


1. 添加用户组管理员：# gpasswd -A 用户名 组名

添加用户入组或从组中删除：# gpasswd [-ad] 组名

* -a：将用户加入组（也可以在“/etc/group”中手动添加）
* -d：将用户从组中删除





# 权限管理

## ACL权限

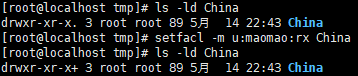
1. ACL权限简介与开启

ACL（Access Control List）主要目的是提供传统的owner、group、others的rwx权限之外的具体权限设置，针对单一用户、单一文件或目录进行rwx权限设置。Linux中很多分区默认开启了ACL权限

注：应用场景比如一个教学环境下，一个文件的所有者是授课老师、权限rwx，所属组是学生、权限rwx，其它人无权限，此时有一个听课老师要分配rx权限，就只能用ACL实现

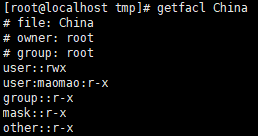
1. 设定与查看ACL权限
2. 设定ACL命令：# setfacl -m [u/g]:[用户名/组名]:权限 文件名

* -m：设定ACL权限
* -x：删除指定ACL权限
* -b：删除所有ACL权限
* -d：设定默认ACL权限
* -k：删除默认ACL权限
* -R：递归设定ACL权限



1. 查看ACL命令：# getfacl 文件名

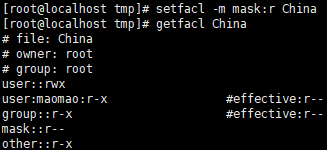
注：mask是用来指定最大有效权限的，即如果给用户赋予了ACL权限，用户的真正权限是需要和mask权限相与才能得到



1. 最大有效权限与删除ACL权限

（1）修改最大有效权限：# setfacl -m m:权限 文件名

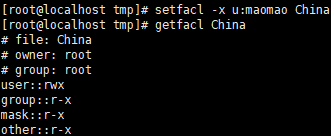
注：示例中右侧是实际的权限，x权限不会生效



（2）删除ACL权限

删除指定用户/用户组ACL权限：# setfacl -x [u/g]:[用户名/组名] 文件名

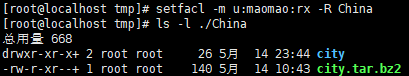
删除文件所有ACL权限：# setfacl -b 文件名



1. 默认ACL权限与递归ACL权限

（1）递归ACL权限：# setfacl -m u:用户名:权限 -R 目录名

注：针对现有的文件，即在父目录设定ACL权限时，所有现有的子目录和子文件也会拥有相同的ACL权限。



（2）默认ACL权限：# setfacl -m d:u:用户名:权限 目录名

注：针对新建的文件，即在父目录设定了默认ACL权限，那么所有新建的子目录和子文件都会继承父目录的ACL权限。

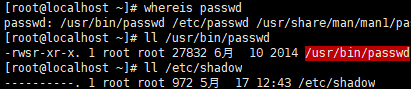


## 文件特殊权限

1. Set UID：权限s出现在文件所有者的x权限上

只有可以执行的二进制程序才能设定SUID权限，命令的执行者要对该程序拥有x权限，在执行程序时获得该程序文件所有者的身份，且仅在程序执行中有效。

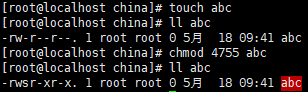
注：SUID不能用于shell脚本，且对目录无效。应用场景比如记录密码的文件/etc/shadow的权限是000，普通用户不能读写，但是普通用户又要可以修改自己的密码，即写入文件，因此使用passwd命令可以将用户身份暂时转换为root，因此随意设置SUID有风险。

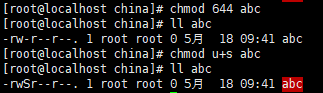




* 设定SUID：# chmod 4755/u+s 文件名
* 取消SUID：# chmod 755/u-s 文件名

注：4代表SUID，2代表SGID，1代表SBIT。示例中在没有x执行权限的基础上添加SUID权限，会报错（大写的S表示）

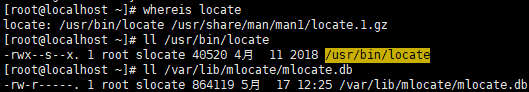




1. Set GID：权限s出现在文件所属组的x权限上

（1） SGID针对文件的作用（与SUID类似）

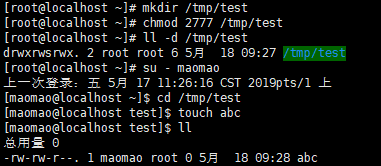
注：示例中“/usr/bin/locate”是可执行二进制程序，可以赋予SGID，普通用户对其拥有x执行权限，在执行命令时，组身份会升级为slocate，而slocate 组对“/var/lib/mlocate/mlocate.db”数据库拥有r读权限，所以普通用户可以使用命令locate查询mlocate.db数据库



（2）SGID针对目录的作用

普通用户必须对此目录拥有r和x权限，才能进入此目录，此时普通用户在此目录的有效组变成目录的所属组，若普通用户对目录拥有w权限，则新建的文件默认所属组与此目录的所属组相同

* 设定SGID：# chmod 2755/g+s 文件名
* 取消SGID：# chmod 755/g-s 文件名

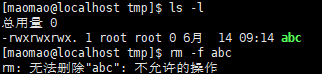


1. Sticky BIT：权限t出现在文件其他人的x权限上

SBIT只对目录有效，普通用户对该目录拥有w和x权限。若没有SBIT，普通用户可以删除此目录下的所有文件，包括其它用户建立的文件，一旦赋予SBIT，除了root可以删除，普通用户即便拥有w权限也只能删除自己建立的文件

* 设定SBID：# chmod 1755/o+s 目录名
* 取消SBID：# chmod 755/o-s 目录名





## 文件隐藏属性

1. 设置文件隐藏属性：# chattr [+-=] [ai] 文件或目录名

* i：如果对文件设置，那么不允许对文件进行删除、改名、添加或修改数据（包括root，相当于锁定文件）；如果对目录设置，那么只能修改目录下文件的数据，而不允许建立和删除文件
* a：如果对文件设置，那么只能在文件中增加数据，而不能删除或修改数据（相当于锁定文件现有数据）；如果对目录设置，那么只允许在目录中建立和修改文件，而不允许删除

注：以上选项命令只有root才能执行，另外chattr命令只能在ext传统文件系统上生效，其它文件系统，如xfs仅支持部分参数

1. 显示文件隐藏属性：# lsattr [-ad] 文件或目录名

* -a：将隐藏文件的属性也显示出来
* -d：若是目录，仅列出目录本身的属性，而不是子文件的



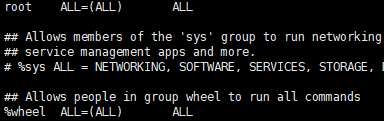
## 系统命令sudo权限

1. sudo作用

root把本来只能超级用户执行的命令赋予普通用户执行，sudo的操作对象是系统命令

1. sudo使用：# visudo

注：“root或%wheel”表示用户名或组名（%），“ALL=(ALL)”表示被管理主机的地址=可切换的身份，“ALL”表示授权命令（绝对路径）。命令visudo实际修改的是etc/sudoers文件，地址不是来源IP地址，而是被管理的主机地址。必须按格式添加sudo权限，如示例中授权用户maomao可以重启系统（命令前加!表示不可执行）





1. 普通用户执行sudo赋予的命令

* 查看可用的sudo命令：# sudo -l
* 前面加sudo执行命令：# sudo /sbin/shutdown -r now



# 分区和文件系统

## 文件系统分类

1. Ext系列：索引式文件系统，包括ext2、ext3、ext4。其中ext2是ext文件系统的升级版本；ext3是CentOS 5默认文件系统，最大支持16TB的分区和2TB的文件，最大的区别是带日志功能；ext4是CentOS 6默认文件系统，最大支持1EB的分区和16TB的文件（1EB = 210PB = 220TB）
2. XFS：CentOS 7默认文件系统，最大支持8EB减1字节的单个文件。由于Ext系列需要预先规划出所有inode、区块等数据，导致当磁盘容量很大时，格式化超慢，因此改为较适合高容量磁盘的XFS文件系统
3. FAT：用于Windows系统和U盘，这种文件系统没有inode存在，因此当文件写入的数据区块过于分散时，文件的读取性能会变差，这时需要通过“碎片整理”将同一文件所属区块集合在一起

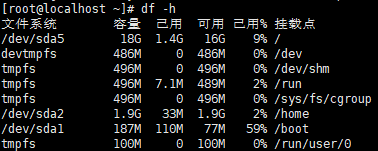
## 文件系统常用命令

1. df、du、xfs\_info命令

（1） 文件系统查看：# df [-ahT] [文件或目录名]

* -a：显示所有文件系统信息，包括特殊文件系统
* -h：使用习惯单位显示容量
* -T：显示文件系统类型

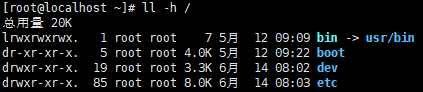
注：示例中/dev/shm目录是利用内存虚拟出来的磁盘空间，通常是总物理内存的一半。在该目录下新建文件，访问速度非常快，但是重启后会消失



（2） 统计文件或目录大小：# du [-ahs] [文件或目录名]

* -a：显示每个子文件的磁盘占用量，默认只统计子目录的占用量
* -s：统计总占用量，而不列出子目录和子文件的占用量

注：ls –l具有局限性，当它统计目录时，只会统计一级子目录或子文件名占用的空间，而不是文件里的数据。du命令是面向文件的，只会计算文件或目录占用的空间；而df命令是从文件系统考虑的，除了文件，还有被命令或程序占用的空间，因此统计结果可能会大





1. 显示磁盘状态：# xfs\_info 分区设备文件名

注：示例中第1行表示inode大小为512B，存储区群组（allocation group）有4个，每个存储区群组有122048个区块，配合第4行区块大小4K，因此整个文件系统容量为4\*122048\*4K



1. 挂载命令

（1） 查询与自动挂载

注：分区也是硬件，在开机时会自动挂载，但是光盘、U盘、硬盘不是自动挂载，因为不能确定每次开机都插入了硬盘，因此需要手动挂载

* 依据配置文件/etc/fstab的内容，自动挂载：# mount -a
* 查询系统中已经挂载的设备，显示卷标名：# mount [-l]



（2） 挂载命令格式

# mount [-t 文件系统] [-o 特殊选项] 设备文件名/UUID=’ ’ 挂载点

* -t：加入文件系统类型来指定挂载的类型，如ext4、xfs等文件系统（默认以最恰当的文件系统尝试挂载）
* -o：可以指定挂载的额外选项，如remount重新挂载，noexec不执行

1. 挂载光盘与U盘

（1） 挂载光盘

* 建立挂载点：# mkdir /mnt/cdrom
* 挂载光盘：# mount -t iso9660 /dev/cdrom /mnt/cdrom（或# mount /dev/sr0 /mnt/cdrom）

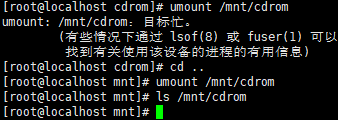
注：一般目录/mnt用于挂载光盘，目录/media用于挂载U盘。光盘的文件系统默认iso9660，因此也可以不写，“/dev/cdrom”软链接到“/dev/sr0”





* 卸载光盘：# umount 设备文件名或挂载点

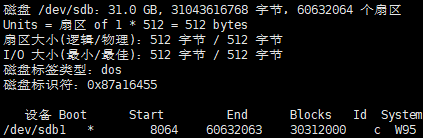
注：光盘挂载后必须先卸载才能退出，卸载前必须先退出当前光盘目录



（2） 挂载U盘

* 查看U盘设备文件名：# fdisk –l

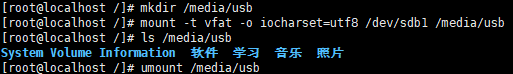
注：U盘的设备文件名与硬盘相同，因此是自动分配的。插入U盘前不能在远程终端界面，只能在虚拟机里，且鼠标必须点进去，否则会由Windows识别，之后才可以到远程终端继续操作。也可以使用命令blkid查看UUID进行挂载



* 建立挂载点：# mkdir /media/usb
* 挂载U盘：# mount -t vfat /dev/sdb1 /media/usb

注：加上选项“-o iocharset=utf8”可以支持中文，否则会出项乱码

* 卸载U盘：# umount 设备文件名或挂载点



1. 支持NTFS文件系统

注：Linux默认不支持NTFS文件系统，即移动硬盘。与Windows需要手动安装驱动不同，Linux内核已经包括了常用的驱动，因此不需要手动安装

（1） 下载NTFS-3G插件：<http://www.tuxera.com/community/ntfs-3g-download/>

（2） 安装NTFS-3G：

* 解压：# tar -zxvf ntfs-3g\_ntfsprogs-2017.3.23.tgz
* 进入解压目录：# cd ntfs-3g\_ntfsprogs-2017.3.23
* 编译准备：# ./configure
* 编译：# make
* 编译安装：# make install

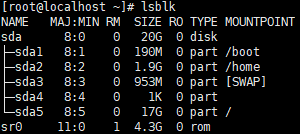
（3）使用：# mount -t ntfs-3g 分区设备文件名 挂载点

## 磁盘分区

1. 查看磁盘分区状态

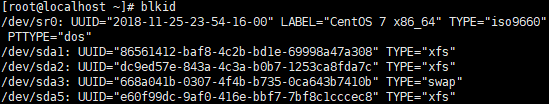
（1） 查看所有磁盘列表：# lsblk [-p] [设备文件名]

* -p：列出设备的完整文件名



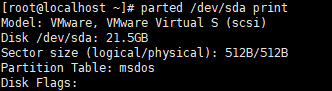
（2） 查看设备UUID等参数：# blkid

注：UUID是硬盘通用唯一识别码，Linux会将系统内所有设备都给予一个唯一标识符，可用于挂载或使用设备或文件系统



（3） 查看磁盘分区表类型和分区信息：# parted [设备文件名]

注：Linux磁盘分区表有两种类型，分别是MBR（MS-DOS）和GPT。GPT分区已经没有了主、扩展、逻辑分区的概念，它更适合容量大的磁盘。示例中使用的仍然使用MBR分区表



1. 磁盘分区：fdisk/gdisk

（1） 添加新硬盘

注：由于硬盘空间已经全部分配，因此需要添加新硬盘，添加前需要关闭虚拟机（虚拟机 -> 设置 -> 添加 -> 默认下一步 –> 确认）

（2） 查看新硬盘：# fdisk –l

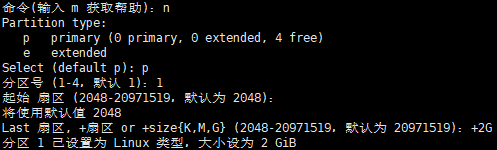
注：MBR分区表使用fdisk命令，GPT使用gdisk命令，否则命令按错，分区记录可能会全部消失



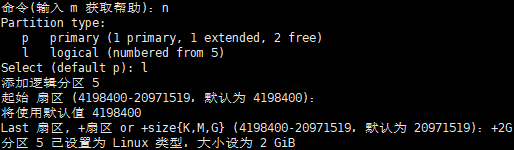
（3） 分区：# fdisk /dev/sdb

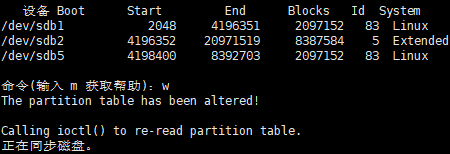
|  |  |
| --- | --- |
| fdisk交互指令说明 | |
| d/n | 删除/新建一个分区 |
| l | 显示已知的文件系统类型（82为Linux swap分区，83为Linux分区） |
| m | 显示帮助菜单 |
| t | 改变一个分区的系统ID |
| q/w | 不保存退出/保存退出 |
| p | 显示分区列表 |

注：使用Ctrl+backspace删除打错的命令，结束后记得w保存退出





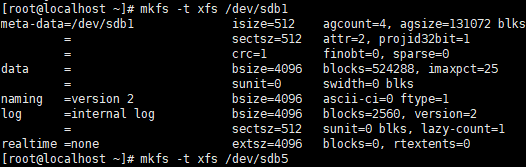




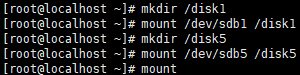
（4） 重新读取分区表信息：# partprobe

（5） 格式化分区：# mkfs -t xfs /dev/sdb1

注：扩展分区/dev/sdb2不能格式化，也不能写入数据。可使用“mkfs[Tab][Tab]”查看系统支持哪些文件系统的格式化功能



（6） 建立挂载点并挂载：# mkdir /disk1; # mount /dev/sdb1 /disk1



1. 分区自动挂载

注：前面使用命令手动挂载重启后分区会消失，要想永久生效，在开机时自动挂载，需要写入/etc/fstab文件。由于/etc/fstab是重要的系统文件，万一写错内容会导致系统崩溃，此时要使用fstab文件修复



注：第1字段是设备文件名或UUID；第2字段是挂载点；第3字段文件系统名称；第4字段是挂载参数；第5字段是指定分区是否被dump备份（0表示不备份，1表示每天备份，2表示不定期备份）；第6字段是指定分区是否被fsck检测（0表示不检测，其它数字表示检测的优先级，1优先级大于2）

* 查看UUID：# blkid /dev/sdb1



* 分区自动挂载：# vim /etc/fstab



* 依据配置文件自动挂载：# mount –a

注：用于检查挂载，无报错即挂载成功，以避免系统崩溃

1. fstab文件修复：# mount -o remount,rw /

注：仅能修复文件报错，开机时报错会提示输入root密码，然后使用命令以读写方式重新挂载，否则根目录会被系统挂载为只读，文件不能修改，最后使用命令vim修改fstab文件

## 磁盘配额

1. 磁盘配额简介

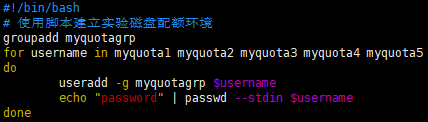
磁盘配额（quota）针对xfs文件系统可分别限制某一用户、用户组、目录的最大磁盘配额，通过限制inode或block使用量来限制文件数量或磁盘容量（常见）。限制值通常有两个：软限制soft、硬限制hard，硬限制表示用户使用量绝对不能超过这个值，而高于软限制时系统会发出警告信息，且给予一个宽限时间（grace time），若该时间内仍未进行磁盘管理，软限制会即刻替换硬限制

注：ext文件系统仅能针对整个文件系统，而不能针对单一目录。磁盘配额只对一般身份用户有效，且默认情况下仅能针对/home进行设置（受限于SELinux）

1. 磁盘配额实例

（1） 实例要求

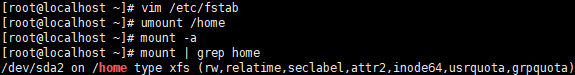
五个用户myquota1-5容量限制250M/300M，用户组myquotagrp容量限制950M/1G，宽限时间grace time设置为14天



（2） 文件系统支持与查看

xfs文件系统的磁盘配额需要写入/etc/fstab配置文件中，在/home所在行的defaults后加上“,usrquota,grpquota”，然后重新挂载。限制的选项主要有三种：

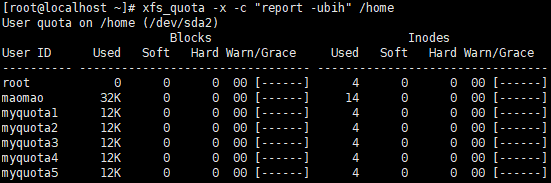
* uquota/usrquota/quota：针对用户设置
* gquota/grpquota：针对用户组设置
* pquota/prjquota：针对单一目录设置（不能与grpquota同时存在）



（3） 查看磁盘配额数据：# xfs\_quota -x -c “命令” [挂载点]

* pirnt：列出文件系统磁盘配额挂载参数支持
* report：列出磁盘配额限制值，选项-ugr（user/group/project）及-bi（block/inode）
* state：列出目前支持磁盘配额的文件系统信息

注：选项-x表示专家模式，后面才能接选项-c，后面再接命令



（4） 限制值设置方式

* limit [-ug] b[soft/hard]=N i[soft/hard=N] 用户名或组名：限制容量
* timer [-ug] [-bir] Ndays：限制宽限时间





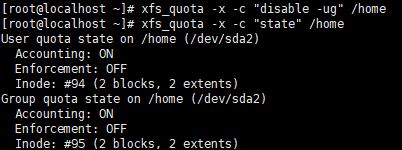


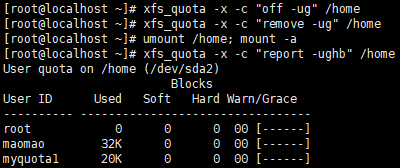


（5） 磁盘配额管理

* disable/enable [-ug]：暂时取消磁盘配额限制/恢复正常管制
* off [-ug]：完全关闭磁盘配额限制（重启quota需要重新挂载）
* remove [-ug]：删除磁盘配额的限制设置（必须先执行off）

注：使用disable仍然在计算（account）磁盘配额，只是没有管制（enforcement）





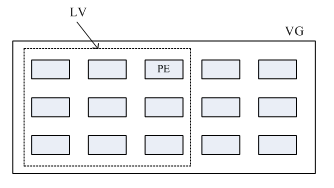
## 逻辑卷管理器LVM

1. LVM简介

普通的磁盘分区管理方式在逻辑分区划分好之后就无法改变其大小，当一个逻辑分区存放不下某个文件时，这个文件因为受上层文件系统的限制，不能跨越多个分区来存放，所以也不能同时放到别的磁盘上。

LVM全称逻辑卷管理器（Logical Volume Manager），它可以弹性地调整文件系统的容量，即可以整合多个物理分区，让这些分区看起来就像一个磁盘一样，未来还可以新增或删除其它物理分区，实现了文件系统跨越不同磁盘和分区。

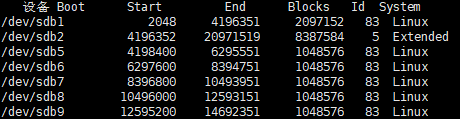
具体做法是将多个物理卷PV整合为卷组VG，然后将其划分为可使用的逻辑卷LV，划分的最小存储单位是物理扩展块PE（默认4MB）。如图所示，VG内的PE划分给虚线部分的LV，若VG要扩充，加上其它PV即可；若LV要扩充，加入PE即可



1. LVM实例

（1） 磁盘分区：# fdisk /dev/sdb

注：重新创建了2G的主分区、8G的扩展分区，以及5个1G的逻辑分区



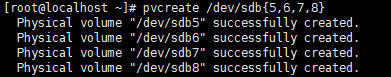
（2） 创建PV：# pvcreate 分区名

查找目前系统中任何具有PV的磁盘：# pvscan

显示出目前系统上的PV状态：# pvdisplay 分区名

将PV属性删除：# pvremove

注：VG和LV都有类似的命令，后面不再单独列出

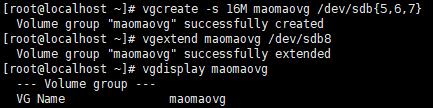


（3） 创建VG：# vgcreate [-s PE的大小] VG名 PV名

在VG内增加额外的PV：# vgextend VG名 分区名

在VG内删除PV：# vgreduce

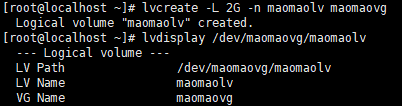
设置VG是否启动：# vgchange



（4） 创建LV：# lvcreate [-lL] [-n LV名] VG名

* -L：后接容量，必须是PE大小的倍数，若不符，系统会计算相近值
* -l：后接PE的个数

注：创建LV后，后面的操作都必须使用LV全名



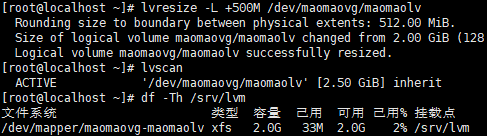
（5） 格式化分区并挂载



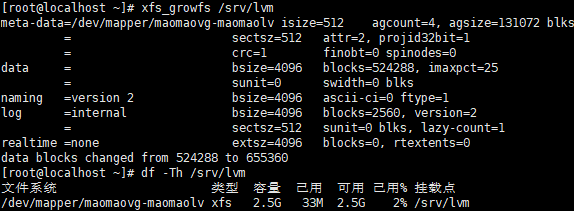


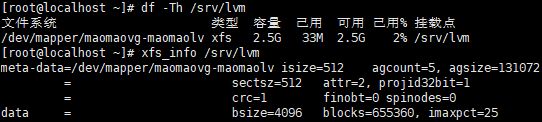
1. 放大LV容量：# lvresize [-L +/-PE容量] LV全名

注：若VG剩余容量足够，可直接使用lvresize命令放大LV容量；若VG容量不足，最简单的方法是增加硬盘，再利用pvcreate和vgextend命令增加到VG内。另外也可使用lvextend命令进行扩容



注：示例中LV放大到2.5G，但是文件系统却没有相对增加。这是因为Linux实际使用的不是LV，而是LV设备内的文件系统，需要使用xfs\_growfs命令来增加区块群组agcount，从而放大文件系统（xfs暂不支持文件系统容量缩小）



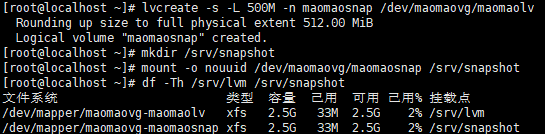


1. LVM快照

（1） 创建快照：# lvcreate -s [-lL] -n 快照名 LV全名

* -s：快照（snapshot）

LVM快照区与系统区共享所有的PE数据，因此最初它们的内容是一模一样的，PE内数据修改前系统会将该区域的数据移动到快照区，达到备份的目的。快照区与系统区必须在同一个VG上，因此需要先使用vgdisplay命令查看VG的剩余容量



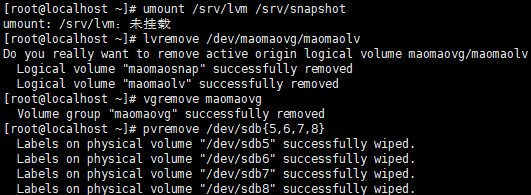
注：xfs不允许相同的UUID文件系统挂载，因此加上nouuid参数，让系统忽略相同的UUID造成的问题。也可以先卸载原来的系统区，再挂载快照区

（2） 使用快照

由于最初创建快照时，快照区与系统区的内容一模一样，因此我们可以使用快照区来做测试，测试完毕后将快照删除，再重新挂载系统区即可（若卸载）

1. 删除LVM

删除的步骤：卸载LVM文件系统 –> 使用lvremove删除LV -> 使用vgremove删除VG -> 使用pvremove删除PV



# Shell基础

## Shell概述

1. Shell作用

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

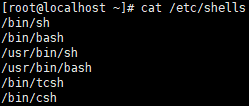
1. Shell的分类

（1） Bourne Shell（sh）：主要包括sh、ksh、Bash、psh、zsh

（2） C Shell：主要包括csh、tcsh

注：Bourne Shell与C Shell语法彼此不兼容，目前Linux使用Bash作为用户基本Shell

1. Linux支持的Shell：# cat /etc/shells



## Shell脚本执行方式

1. 输出命令：# echo [-e] [输出内容]

* -e：支持反斜杠控制的字符转换



1. 第一个脚本：# vim helloworld.sh

注：第1行“#!/bin/bash”不是注释，而是shell脚本标识，第2行才是注释，第4行是要输出的内容



1. 脚本执行

（1） 赋予执行权限，直接运行：# chmod 755 helloworld.sh; ./ helloworld.sh



（2） 通过bash调用执行脚本：# sh [-nx] helloworld.sh（或使用bash）

* -n：不执行脚本，仅查询语法的问题
* -x：显示使用到的脚本内容，用于debug

注：一些在Windows下编写的脚本文件，在Linux下可能不能执行，因为换行符的表示不同（Windows下换行符为^M$，Linux下换行符为$）。此时必须使用命令“# dos2unix 文件名”进行转换后执行，但是该命令默认没有安装，使用命令“# yum -y install dos2unix”进行安装



## Bash的基本功能

1. 历史命令：# history [-cw] [历史命令保存文件]

* -c：清空历史命令
* -w：把缓存中的历史命令写入历史命令保存文件~/.bash\_history

注：历史命令默认保存1000条，可以修改配置文件/etc/profile



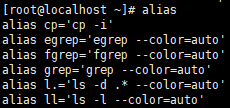
1. 定义命令别名：# alias 别名='原命令'

删除别名：# unalias 别名

查询命令别名：# alias

别名永久生效：# vim ~/.bashrc

注：命令的执行顺序，第一执行用绝对路径或相对路径的命令，第二执行别名alias，第三执行bash的内部命令，第四执行按照$PATH环境变量定义的目录查找顺序找到的第一个命令



1. 输入输出重定向

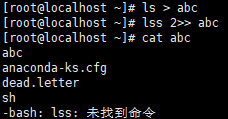
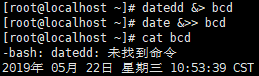
（1） 标准输入输出

|  |  |  |  |
| --- | --- | --- | --- |
| 设备 | 设备文件名 | 文件描述符 | 类型 |
| 键盘 | /dev/stdin | 0 | 标准输入 |
| 显示器 | /dev/stdout | 1 | 标准输出 |
| 显示器 | /dev/stderr | 2 | 标准错误输出 |

（2） 输出重定向

|  |  |  |
| --- | --- | --- |
| 类型 | 符号 | 作用 |
| 标准输出重定向 | 命令 > 文件 | 把命令的正确输出写入文件（覆盖） |
| 命令 >> 文件 | 把命令的正确输出写入文件（追加） |
| 标准错误输出重定向 | 命令 2> 文件 | 把命令的错误输出写入文件（覆盖） |
| 命令 2>> 文件 | 把命令的错误输出写入文件（追加） |
| 正确输出和错误输出同时保存 | 命令 &> 文件 | 把正确或错误输出写入同一文件（覆盖） |
| 命令 &>> 文件 | 把正确或错误输出写入同一文件（追加） |
| 命令 >> 文件1 2>> 文件2 | 把正确输出追加到文件1中，把错误输出追加到文件2中（覆盖类似） |

注：输出重定向是将命令的结果不输出到屏幕，而输出到文件；输入重定向则是使用文件作为命令的输入，而不是键盘，使用较少。另外使用“2> /dev/null”可以将执行的错误信息丢掉，/dev/null相当于回收站的设备

1. 多命令顺序执行与管道符

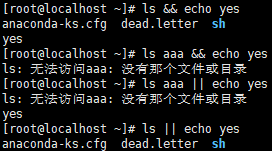
（1） 多命令顺序

|  |  |  |
| --- | --- | --- |
| 多命令执行符 | 格式 | 作用 |
| ; | 命令1;命令2 | 顺序执行，命令间无任何逻辑联系 |
| && | 命令1&&命令2 | 当命令1正确执行，命令2才执行 |
| || | 命令1||命令2 | 当命令1执行不正确，命令2才执行 |

注：可以使用两个date命令计算耗时命令的执行时间，使用“命令 && echo yes || echo no”来判断命令执行正确与否







（2） 管道符：# 命令1 | 命令2

注：命令1的正确输出作为命令2的操作对象，若命令1没有正确执行，则命令2不会执行。同时命令2必须能够接受标准输入的数据，这样的命令才能是管道命令，如less、grep等，而如ls、cp、mv等就不是管道命令



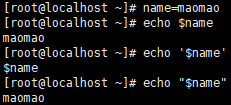
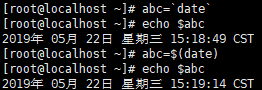
1. 通配符与其它特殊符号

（1）通配符

|  |  |
| --- | --- |
| 通配符 | 作用 |
| ? | 匹配任意一个字符 |
| \* | 匹配0个或任意多个字符 |
| [] | 匹配括号中任意一个字符，如[abc]表示匹配a或b或c，[a-z]表示匹配一个小写字母，[^0-9]表示匹配一个不是数字的字符 |

（2）其它特殊符号

|  |  |
| --- | --- |
| 符号 | 作用 |
| ‘ ’ | 单引号，在单引号内所有特殊符号都没有特殊含义 |
| “ ” | 双引号，$、~、\拥有“调用变量”、“引用命令”、“转义”的特殊含义 |
| ` `/$() | 反引号，在反引号内是系统命令，推荐使用$() |
| $ | 用于调用变量的值 |
| \ | 转义符，特殊符号将失去特殊含义 |

## Bash的变量

1. 变量设置规则

（1） 变量名由字母、数字、下划线组成，不能由数字开头

（2） Bash中变量默认类型是字符串型，若要数值运算，须指定变量为数值型

（3） 变量用等号连接值，等号左右两侧不能有空格

（4） 变量值如果有空格，需要使用单引号或双引号

（5） 环境变量名建议大写，便于区分

1. 变量分类

（1） 用户自定义变量：变量只在当前shell中生效，变量名、变量值都可自由定义

（2） 环境变量：变量在当前shell和所有子shell中生效，若写入配置文件则所有shell生效。系统环境变量只能修改变量的值，允许用户自定义环境变量，变量名和变量的值都可以修改

（3） 位置参数变量：主要用来向脚本当中传递参数或数据，变量名不能自定义，且变量作用固定，只能修改变量的值

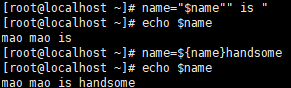
（4） 预定义变量：bash已经定义好的变量，位置参数变量本质上是预定义变量

1. 用户自定义变量

（1） 变量调用：$变量名



（2） 变量叠加：”$变量名”/${变量名}



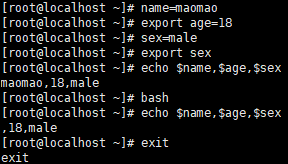
（3） 变量查看：# set

（4） 变量删除：# unset变量名

1. 环境变量

（1） 环境变量定义：# export 变量名=变量值

注：子进程仅会继承父进程的环境变量，而不会继承自定义变量。进入子shell直接输入shell名，命令exit退出子shell，命令pstree以树状结构显示进程的父子关系（安装：# yum install psmisc）



（2） 变量查看：# env

注：set可以查看所有类型的变量，env只能查看环境变量

（3） 系统常用环境变量

* PATH：系统查找命令的路径

注：可使用“PATH=”$PATH”:/root/sh”临时叠加PATH变量



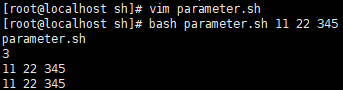
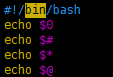
* PS1：定义系统提示符的变量



1. 位置参数变量

|  |  |
| --- | --- |
| 位置参数变量 | 作用 |
| $n | n为数字，$0代表脚本文件名，$1-${10}代表第1-10个参数 |
| $\* | 代表命令行中所有参数，它把所有参数看成一个整体 |
| $@ | 代表命令行中所有参数，它把每个参数区分对待 |
| $# | 代表命令行中所有参数的个数 |

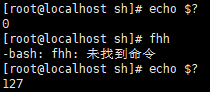
 

1. 预定义变量

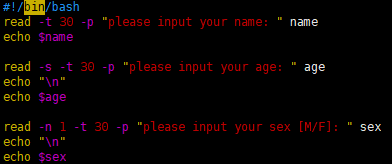
（1） 预定义变量

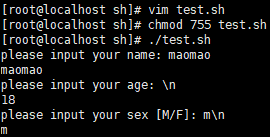
|  |  |
| --- | --- |
| 预定义变量 | 作用 |
| $? | 最后一次执行命令的返回状态，值为0表示上一个命令正确执行，非0表示上一个命令不正确执行 |
| $$ | 当前进程的进程号（PID） |
| $! | 后台运行的最后一个进程的进程号 |



（2） 接收键盘输入：# read [选项] [变量名]

* -p “提示信息”：在等待输入时，输出提示信息
* -t 秒数：指定输入等待时间，默认一直等待用户输入
* -n 字符数：只接受指定的字符数，不用输入回车自动执行
* -s：隐藏输入的数据，适用于机密信息



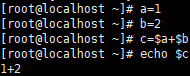
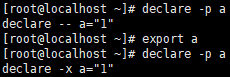


## Bash的运算符

1. 数值运算与运算符

（1） 声明变量类型：# declare [+/-] [-ixp] 变量名

* +/-：取消/设定变量的类型属性（+取消）
* -i：将变量声明为整数型
* -x：将变量声明为环境变量
* -p：显示指定变量的被声明类型

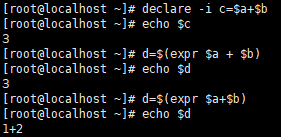
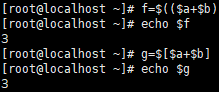
 

（2） 数值运算

* 方法1：declare -i c=$a+$b
* 方法2：d=$(expr $a + $b)

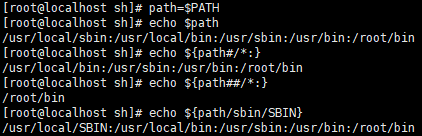
注：“+”号左右两侧必须有空格，一般单括号表示系统命令，双括号数值运算

* 方法3：f=$(($a+$b)) 或 g=$[$a+$b]

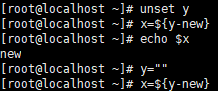
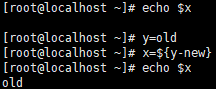
1. 变量内容的删除与替换

|  |  |
| --- | --- |
| 变量设置方式 | 说明 |
| ${变量#关键词}  ${变量##关键词} | 从头开始匹配关键词，删除最短  从头开始匹配关键词，删除最长 |
| ${变量%关键词}  ${变量%%关键词} | 从尾向前匹配关键词，删除最短  从尾向前匹配关键词，删除最长 |
| ${变量/旧字串/新字串}  ${变量//旧字串//新字串} | 替换第一个旧字符串  替换全部的旧字符串 |



1. 变量测试与内容替换

|  |  |  |  |
| --- | --- | --- | --- |
| 变量置换方式 | 变量Y没有设置 | 变量Y为空值 | 变量Y设置值 |
| X=${Y-新值} | X=新值 | X为空 | X=$Y |
| X=${Y:-新值} | X=新值 | X=新值 | X=$Y |
| X=${Y+新值} | X为空 | X=新值 | X=新值 |
| X=${Y:+新值} | X为空 | X为空 | X=新值 |
| X=${Y=新值} | X=新值，Y=新值 | X为空，Y值不变 | X=$Y，Y值不变 |
| X=${Y:=新值} | X=新值，Y=新值 | X=新值，Y=新值 | X=$Y，Y值不变 |
| X=${Y?新值} | 新值输出 | X为空 | X=$Y |
| X=${Y:?新值} | 新值输出 | 新值输出 | X=$Y |

## Bash环境配置文件

1. 环境变量配置文件

环境变量配置文件中主要是定义对系统操作环境生效的系统默认环境变量，如PATH、HISTSIZE、PS1、HOSTNAME等，这也是一进入bash就可以取得有用变量的原因。环境变量配置文件大致有如下5类：“/etc/profile”、“/etc/profile.d/\*.sh”、“/etc/bashrc”、“~/.bash\_ profile”、“~/.bashrc”，其中前3类对任何用户都生效，后2类只对当前用户有效（与用户家目录~有关）。另外使用命令“# source 配置文件名”或“# . 配置文件名”可以使修改的配置文件立即生效，而不用退出登录



* /etc/profile：（login shell才会读）是系统整体的设置，不建议修改。包括MAIL、PATH、umask等变量，同时调用/etc/profile.d/\*.sh等外部配置文件
* ~/.bash\_ profile：（login shell才会读）属于用户个人设置，PATH变量以叠加的方式加入“:~/bin”，同时调用~/.bashrc配置文件
* ~/.bashrc：（non-login shell会读）定义默认别名，调用/etc/bashrc文件
* /etc/bashrc：设置PS1、PATH、umask等变量，调用/etc/profile.d/\*.sh文件

注：取得bash时需要输入账号密码，就称为login shell，否则就称为non-login shell，如输入bash进入子进程时没有输入账号密码，则子进程是non-login shell。另外，文件/etc/locale.conf记录的是系统默认的语系，使用命令“# locale”还可以查看Linux支持的语系

1. 其它配置文件

* 注销时生效：~/.bash\_logout
* 记录历史命令：~/.bash\_history
* 规定man命令的查找路径：/etc/man\_db.conf

1. 登录欢迎信息文件

* 本地终端欢迎信息：/etc/issue

|  |  |  |  |
| --- | --- | --- | --- |
| 转义符 | 作用 | 转义符 | 作用 |
| \d | 显示当前系统日期 | \n | 显示主机名 |
| \S | 显示操作系统名称 | \o | 显示域名 |
| \l | 显示登录的终端号 | \r | 显示内核版本 |
| \m | 显示硬件体系结构 | \u | 显示当前登录用户的序列号 |

* 远程终端欢迎信息：/etc/issue.net

注：转义符在该文件中不能使用，且默认不显示欢迎信息，需要在SSH的配置文件“/etc/ssh/sshd\_config”中加入“Banner /etc/issue.net”，然后重启SSH服务

* 登录后的欢迎信息：/etc/motd

注：不管是本地登录，还是远程登录，都可以显示此欢迎信息

# Shell编程

## 正则表达式

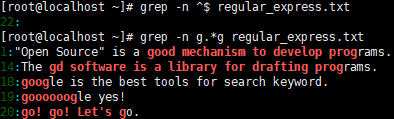
1. 正则表达式与通配符

* 正则表达式用来在文件中匹配符号条件的字符串，正则是包含匹配。grep、awk、sed等命令可以支持正则表达式
* 通配符用来匹配符合条件的文件名，通配符是完全匹配，ls、find、cp等命令不支持正则表达式，所以只能使用Shell自己的通配符

1. 基础正则表达式

|  |  |
| --- | --- |
| RE字符 | 作用 |
| \* | 重复前一个字符0次或任意多次，如oo\*表示至少1个o |
| . | 匹配除了换行符外任意一个字符 |
| ^ | 匹配行首，如^hello匹配以hello开头的行 |
| $ | 匹配行尾，如hello$匹配以hello结尾的行 |
| [] | 匹配括号中指定的任意一个字符 |
| [^] | 匹配除括号的字符以外的任意一个字符 |
| \ | 转义符，用于取消特殊符号的含义 |
| \{n\} | 表示其前面的字符恰好出现n次（\转义） |
| \{n,m\} | 表示其前面的字符至少出现n次，最多出现m次 |

注：正则表达式的RE字符与命令行输入命令的通配符并不相同，如\*在通配符中表示任意个字符，而在正则表达式中表示重复前一个字符任意次。示例中的含义分别是查找空白行、查找以g开头和结尾的字符串

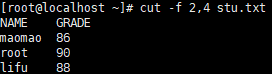


## 字符截取命令

1. cut命令：# cut [-df] 文件名

* -f 列号：提取第几列
* -d 分隔符：按照指定分隔符分隔列

注：grep是提取符号条件的行，cut是提取符号条件的列，一般配合grep一起使用。cut不适用识别空格（awk命令可以），但是可以识别制表符、冒号等

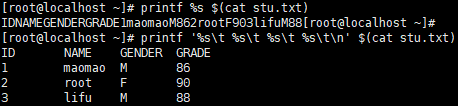
 

1. printf命令：# printf ‘输出类型 输出格式’ 输出内容

* %ns：输出n个字符
* %ni：输出n个数字
* %m.nf：输出浮点数，m代表总位数（包括小数点），n代表小数位数

注：printf命令不是管道命令，其中输出格式与C语言相同，常用\n换行，\t制表符。另外使用数字n可将字段长度固定，解决未对齐的情况

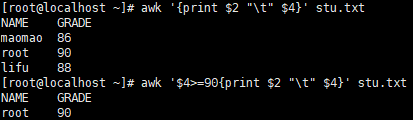


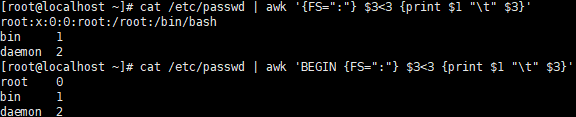


1. awk命令：# awk ‘条件1{动作1} 条件2{动作2}’ 文件名

注：示例中$n表示第n列，在awk命令的输出中支持print和printf命令，区别在于print会自动换行，且只能在awk命令中使用，因为Linux默认没有print命令。另外awk的默认分隔符是空格键或Tab键，若要指定分隔符为冒号，条件动作可使用“BEGIN{FS=”:”}”（不加BEGIN不会处理第一行）

|  |  |
| --- | --- |
| 内置变量 | 含义 |
| NF | 每一行拥有的字段总数 |
| NR | 当前awk所处理的是第几行 |
| FS | 当前分隔字符，默认空格键 |





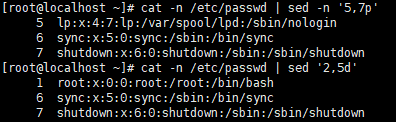
1. sed命令：# sed [-ein] ‘[动作]’ 文件名

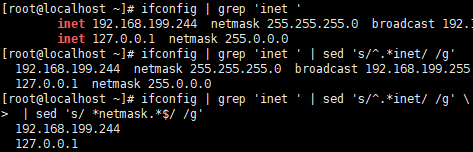
注：awk主要处理每一行的字段内的数据，而sed常常作用于一整行的处理，是用来将数据进行选取、替换、删除、新增的管道命令

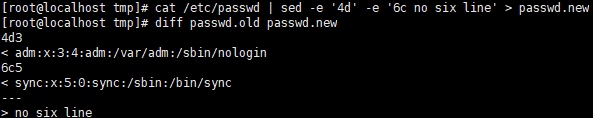
* -n：只把sed命令处理的行输出（默认所有数据都输出）
* -e：sed后面接两个以上动作时，每个动作前加-e
* -i：修改结果直接写入文件，而不是由屏幕输出
* [n1[,n2]] 操作：选择操作的行数，操作字符如下表

|  |  |  |  |
| --- | --- | --- | --- |
| 字符 | 动作 | 字符 | 动作 |
| a | 追加 | d | 删除 |
| c | 行替换 | p | 打印 |
| i | 插入 | s | 字串替换 |

注：c是整行替换，行内部分数据替换可以使用“# sed ‘s/old/new/g’”，另外可以使用命令“# diff 原始文件 目标文件”来比较同一文件或软件的新旧版本的差异



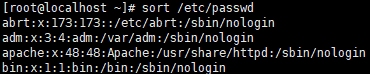


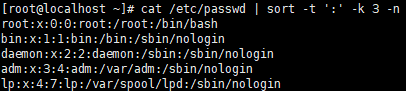


## 字符处理命令

1. 排序命令：# sort [-fnrtk] 文件名

* -f：忽略大小写
* -n：使用数值排序（默认使用字符串）
* -r：反向排序
* -t：指定分隔符（默认制表符）
* -k n,m：指定排序范围（不写m默认行尾）

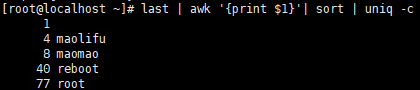




1. 去重命令：# uniq [-ic]

* -i：忽略大小写
* -c：进行计数

注：uniq命令一般配合sort等其它命令一起使用。示例中统计用户登录的次数



1. 统计命令：# wc [-lwm] 文件名

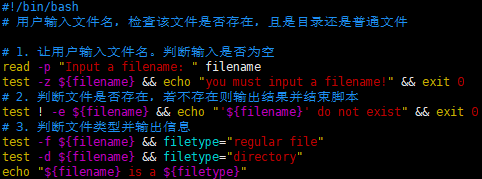
* -l：只统计行数
* -w：只统计单词数
* -m：只统计字符数

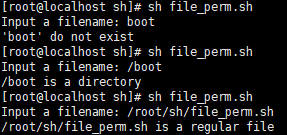


## 条件判断

1. 使用命令test：# test -e 文件名

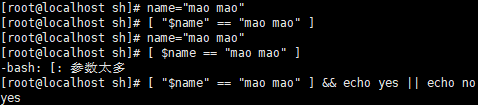
|  |  |
| --- | --- |
| 测试选项 | 作用 |
| 1、按照文件类型进行判断 | |
| -d 文件 | 判断该文件是否存在，并且是否为目录 |
| -e 文件 | 判断该文件是否存在 |
| -f 文件 | 判断该文件是否存在，并且是否为普通文件 |
| 2、按照文件权限进行判断 | |
| -r 文件 | 判断该文件是否存在，并且拥有读权限 |
| -w 文件 | 判断该文件是否存在，并且拥有写权限 |
| -x 文件 | 判断该文件是否存在，并且拥有执行权限 |
| 3、两个文件之间进行比较 | |
| 文件1 -nt 文件2 | 判断文件1的修改时间是否比文件2的新 |
| 文件1 -ot 文件2 | 判断文件1的修改时间是否比文件2的旧 |
| 文件1 -ef 文件2 | 判断两个文件的inode号是否一致（即判断硬链接） |
| 4、两个整数之间比较 | |
| 整数1 -eq / -ne整数2 | 判断整数1是否和整数2相等/不相等 |
| 整数1 -gt / -lt整数2 | 判断整数1是否大于/小于整数2 |
| 整数1 -ge / -ke整数2 | 判断整数1是否大于等于/小于大于等于整数2 |
| 5、按照字符串进行判断 | |
| -z 字符串 | 判断字符串是否为空，为空则为true |
| -n字符串 | 判断字符串是否为非空，为空则为false |
| 字符串1 == 字符串2 | 判断字符串1是否和字符串2相等 |
| 字符串1 != 字符串2 | 判断字符串1是否和字符串2不相等 |
| 6、多重条件判断 | |
| 判断1 -a 判断2 | 逻辑与 |
| 判断1 -o 判断2 | 逻辑或 |
| ! 判断 | 逻辑非 |





1. 使用判断符号[ ]：# [ -e 文件名 ]

注：由于中括号用在了很多地方，包括了通配符与正则表达式等，因此中括号两端需要有空格来分隔，另外中括号内的变量或常量，最好用双引号括起来。示例中报参数太多的错误，原因是判断[ mao mao == “mao mao”]时有三个参数



## 流程控制

1. if语句

|  |  |
| --- | --- |
| 单分支if条件语句：  if [ 条件判断式 ]; then  程序  fi | 多分支if条件语句：  if [ 条件判断式1 ]; then  程序1  elif [ 条件判断式2 ]; then  程序2  else  程序3  fi |

1. case语句

|  |
| --- |
| case $变量名 in  "值1"）  程序1  ;;  "值2"）  程序2  ;;  \*）  程序3  ;;  esac |

1. for循环

|  |  |
| --- | --- |
| 循环次数未知：  for 变量 in 值1 值2 值3  do  程序  done | 循环次数已知：  for (( 初始值; 循环控制条件; 变量变化 ))  do  程序  done |

1. while循环与until循环

|  |
| --- |
| while/util [ 条件判断式 ]  do  程序  done |

注：until循环与while循环相反，只要条件判断式不成立则执行程序，一旦循环条件成立，则终止循环

# 服务与系统管理

## 服务的分类

1. 服务的分类

早期的服务管理使用System V的机制，通过service、chkconfig等命令来管理服务的启动和关闭，从CentOS 7开始采用systemd的机制，此机制最大功能是并行处理，并采用单一命令（systemctl）管理，启动速度加快。systemctl将各服务定义为unit，而unit有如下类型：

|  |  |
| --- | --- |
| 扩展名 | 主要服务功能 |
| .service | 一般服务类型，主要是系统服务，最常见 |
| .socket | 内部程序数据交换的socket服务 |
| .target | 执行环境类型，其实是一群unit的集合 |
| .mount | 文件系统挂载相关的服务 |
| .path | 检测特定文件或目录类型 |
| .timer | 循环执行的服务 |

注：daemon是程序执行后的进程，完成某个服务（service）需要一个daemon在后台运行，因此很多服务名称之后会加字母d，如atd、crond、httpd等

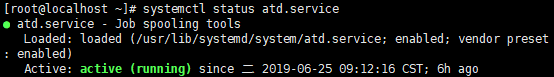
1. 常见服务介绍

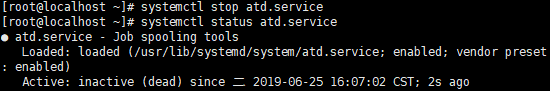
|  |  |
| --- | --- |
| 服务名称 | 简易说明 |
| 默认自启动服务（系统） | |
| atd | 仅执行一次的计划任务，配置文件为/etc/at.{allow,deny} |
| crond | 循环执行的计划任务，配置文件为/etc/crontab |
| cpupower | 提供CPU运行规范，可参考/etc/sysconfig/cpupower |
| rsyslog | 记录系统产生的各项信息，包括/var/log/messages等日志文件 |
| 其它服务（网络） | |
| httpd | 网站服务器 |
| named | 域名服务器DNS |
| vsftpd | 文件传输服务器FTP |
| sshd | 远程连接服务器，与telnet相比传输数据时会加密，不要关闭 |
| Postfix | 邮件发送主机，不要关闭 |

## 服务管理

1. 服务的启动与开机自启动：# systemctl [command] [unit]

* start/stop/restart/reload：启动/关闭/重启/重新加载配置文件
* enable/disable：开机自启动/开机不启动
* status：查看服务状态

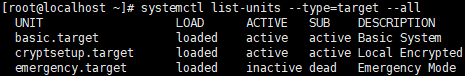




注：Loaded状态有包括开机自启动（enabled）、开机不启动（disabled）、不可自启动（static）、无法被启动（mask）；Active状态包括正在运行（running）、等待执行（waiting）、仅执行一次就结束（exited）、没有执行（dead）。另外由于服务之间有依赖性，因此一些已经关闭的服务会被其它服务唤醒

1. 查看系统上的服务：# systemctl [command] [--type=unit] [--all]

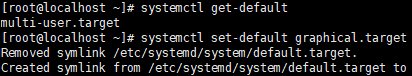
* list-units：根据unit类型显示已启动的服务（默认），加--all没启动的也列出
* list-unit-files：根据/usr/lib/systemd/system内的文件显示



1. 管理target操作模式：# systemctl [command] [unit.target]

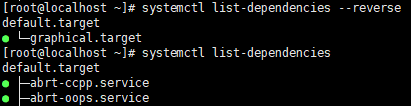
* get-default：取得当前的操作模式
* set-default：设置默认操作模式
* isolate：切换为操作模式

注：systemd取消了以前的运行级别概念，转而使用不同的target操作环境。其中正常模式有graphical.target（命令加图形界面）、multi-user.target（纯命令行），恢复模式有rescue.target、emergency.target



1. 分析服务间的依赖性：# systemctl list-dependencies [unit] [--reverse]

注：--reverse表示反向追踪谁使用这个unit，示例中说明multi-user.target主要是被graphical.target所使用



## 进程管理

1. 查看进程

（1） 查看系统所有进程：# ps aux

仅查看自己bash相关进程：# ps -l



注：字段依次表示进程属于哪个用户、进程ID号、占用CPU百分比、占用物理内存的百分比、占用虚拟内存大小、占用实际物理内存大小、哪个终端运行（tty1-tty6是本地字符终端，tty7是图形终端，pts/0-255是虚拟终端，?表示与终端无关）、进程状态（R/S/T/Z）、进程启动时间、占用CPU运行时间、进程实际命令

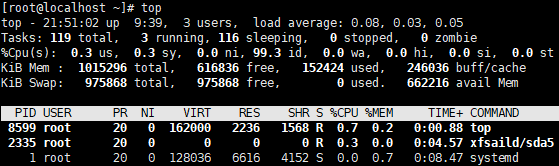
（2） 查看进程树：# pstree [-pu]

* -p：显示进程PID
* -u：显示进程所属用户



注：systemd进程PID为1，是系统第一个执行的进程，它是所有进程的父进程

（3） 动态查看进程变化：# top



注：前5行依次表示任务队列信息、进程信息、CPU信息、物理内存信息、交换分区信息，主要看负载、CPU空闲率、内存空闲率。top默认使用CPU使用率作为排序依据，可使用Q退出

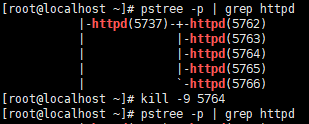
1. 终止进程

（1） 查看可用的进程信号：# kill -l



|  |  |  |
| --- | --- | --- |
| 信号代号 | 信号名称 | 说明 |
| 1 | SIGHUP | 重新读取配置文件后重启 |
| 9 | SIGKILL | 强制终止进程 |
| 15 | SIGTERM | 正常结束进程 |

（2） 按照进程PID终止进程：# kill -1/9/15 PID



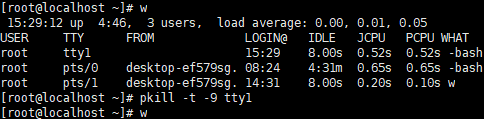
1. 按照进程名终止进程：# killall [-iI] [信号] 进程名

* -i：交互式询问
* -I：忽略进程名大小写



（4）按照进程名终止进程：# pkill [-t] [信号] 进程名

* -t 终端号：按照终端号踢出用户

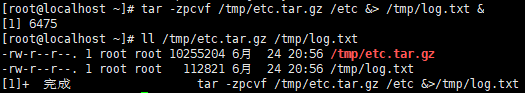


## 任务管理

1. 进程放入后台执行（关掉终端会停止运行）：# 命令 &

进程放入后台执行（关掉终端也会继续运行）：# nohup命令 &

注：[1]表示任务号，6475表示进程PID，可以利用数据流重定向将数据输出至某个文件，不至于影响前台。另外在命令执行过程中使用“Ctrl+Z”，两者区别在于&放入后台还在执行，而使用快捷键放入后台就不会执行，即暂停状态



1. 查看后台进程：# jobs [-l]

* -l：显示进程的PID

注：示例中使用“Ctrl+Z”将任务放入后台并暂停，其中“+”表示最后一个放入后台，默认第一个恢复；“-”表示倒数第二个放入后台



1. 后台暂停的进程恢复到前台执行：# fg %任务号

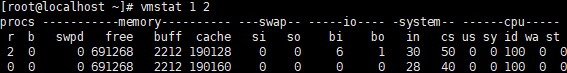
后台暂停的进程恢复到后台执行：# bg %任务号

注：%可以省略，任务号与PID有区别。若命令执行和前台有交互，则不能恢复到后台执行，如top、vim等



## 系统资源查看

1. 监控系统资源：# vmstat [刷新延时 刷新次数]



注：作用与命令top类似，主要观察内存和CPU资源

1. 开机时内核检测信息：# dmesg

查看CPU信息：# cat /proc/cpuinfo



1. 查看内存使用状态：# free



注：缓存cache是用来加速数据从硬盘中“读取”的，缓冲buffer是用来加速数据“写入”硬盘的

1. 查看系统与内核相关信息：# uname –a

判断系统的位数：# file /bin/ls



1. 查看进程打开或使用的文件信息：# lsof [-cpu]

* -c 字符串：只列出以字符串开头的进程打开的文件
* -u 用户名：只列出某个用户的进程打开的文件
* -p PID：只列出某个PID进程打开的文件



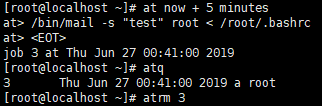
## 定时任务

1. 仅执行一次的计划任务：# at [-mldc] [时间]

* -m：发email通知用户任务完成（默认不发）
* -l：查询用户所有的定时任务（atq）
* -d 任务号：取消任务（atrm 任务号）
* -c 任务号：查询任务的命令内容

注：建议使用绝对路径来执行命令，“Ctrl+D”结束命令的输入，常见的时间格式如“HH:MM [YYYY-MM-DD]”、“now + 5 [minutes/hours/days]”。at定时任务会在后台执行，由atd服务负责，使用/etc/at.allow与/etc/at.deny两个文件来限制使用at，建议只保留一个，系统默认保留/etc/at.deny

* /etc/at.allow：只有在文件中的用户才能使用at，不在的用户不能使用
* /etc/at.deny：在文件中的用户不能使用at，不在的用户可以使用

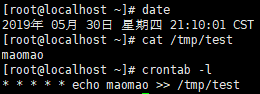
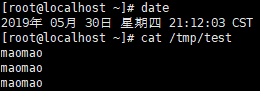


1. 循环执行的计划任务：# crontab [-elr]

* -e：编辑crontab定时任务（格式：\* \* \* \* \* 命令）
* -l：查询crontab任务
* -r：删除当前用户所有的crontab任务（若仅删除一项，使用-e编辑）

|  |  |  |
| --- | --- | --- |
| 格式说明 | | |
| 项目 | 含义 | 范围 |
| 第1个\* | 一小时当中的第几分钟 | 0-59 |
| 第2个\* | 一天当中的第几小时 | 0-23 |
| 第3个\* | 一个月当中的第几天 | 1-31 |
| 第4个\* | 一年当中的第几月 | 1-12 |
| 第5个\* | 一周当中的星期几（0和7都表示星期日） | 0-7 |
| 说明：“\*”表示任何时间，“，”表示不连续的时间，“-”表示连续的时间，“\*/n”表示每隔多久执行一次。at与crontab定时任务都是以分钟为基本单位的，另外不建议周与日月同时并存，易混淆 | | |

注： crontab定时任务由crond服务负责，使用/etc/cron.allow与/etc/cron.deny两个文件来限制使用crontab，作用类似。另外与系统运行相关的配置文件有/etc/crontab以及/etc/cron.d/\*，与用户自己的任务有关的配置文件有/var/spool/cron

## SELinux

1. SELinux简介

SELinux全称Security Enhanced Linux（安全强化的Linux），由美国国家安全局NSA开发，2.6及以上版本的Linux内核都已经集成了SELinux模块。SELinux日志文件位于/var/log/audit/audit.log，且需要借助auditd这个服务。

在未使用SELinux的操作系统中，读取文件系统的方式是自主访问控制（DAC），就是依据进程拥有者与文件资源的rwx权限来决定，这种方式的缺点是root具有最高权限，不受任何管制，且当某个目录权限被系统管理员不小心设为777时，将被任何人所任意读写。因此SELinux引入了强制访问控制（MAC），这种方式针对的主体变成了进程而不是用户，就是决定一个文件资源能否被访问除了上述因素外，还要判断进程是否拥有该类资源的访问权限

1. SELinux基本概念

（1） 主体：SELinux主要管理进程，因此主体等同于进程

（2） 目标：主体进程读写的目标资源一般是文件系统

（3） 策略：由于进程和文件数量庞大，通常我们只是选择性地对某些进程进行管制，而哪些进程需要管制、要怎么管制由策略决定。一套策略里面有多个规则，部分规则可以按照需求启用或禁用。在CentOS 7中提供了三种主要策略：

* targeted：针对网络服务限制较多，针对本机限制较少（默认策略）
* minimum：由target自定义而来，仅针对选择的进程来保护
* mls：完整的SELinux限制，对所有进程进行管制，限制方面较为严格

（4） 安全上下文：主体与目标的安全上下文必须一致才能够顺利读写，安全上下文放置在文件的inode节点内。可使用“ls -Z”查看文件的安全上下文





注：第1字段是身份识别，常见的有：unconfined\_u（不受限的用户）、system\_u（系统用户）；第2字段是角色，常见的有：object\_r（文件等资源）、system\_r（进程）；第3字段类型（最重要），在目标文件资源上称为类型（Type），在主体进程上称为域（domain），domain需要与Type搭配，该进程才能顺利读取文件资源

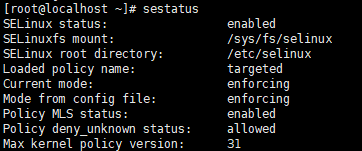
1. SELinux工作模式

（1） 查看工作模式：# getenforce



注：SELinux有3种工作模式，分别是强制模式enforcing：SELinux运行中，且已经正确开始限制domain/type；宽容模式permissive，SELinux运行中，但仅会有警告信息并不会实际限制domain/type的读写，一般为调试用；关闭模式disabled，SELinux没有运行

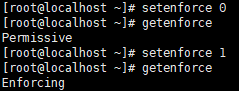
（2） 查看策略：# sestatus



注：SELinux的配置文件是“/etc/selinux/config”，如果由enforcing或permissive切换为disabled，或由disabled切换为其它两个，必须要重新启动，且从disabled转到启动模式时重启需要很长时间。

（3） 强制模式与宽容模式间切换：# setenforce [0/1]

* 0/1：转为permissive宽容模式/enforcing强制模式



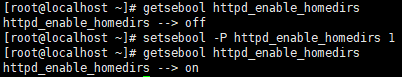
1. SELinux策略内规则管理

（1） 查询系统中各个规则的布尔值：# getsebool [-a] 规则名



（2） 修改规则布尔值：# setsebool [-P] 规则名 0/1

* -P：直接将设置值写入配置文件



1. SELinux安全上下文操作

（1） 查询文件的安全上下文：# ls –Z 文件或目录名

查询进程的安全上下文：# ps –eZ | grep 进程名



注：文件的安全上下文由文件的创建位置和创建进程决定，且系统有一套默认值，单纯的移动文件不会改变文件的上下文

（2） 恢复文件默认的安全上下文：# restorecon -Rv 文件或目录名

* -R：连同子目录一起修改
* -v：显示过程的详细信息

（3） 修改目录的安全上下文：# semanage fcontext [-lamd]

* -l/a/m/d：查询/增加/修改/删除

# 日志与启动管理

## 日志管理

1. 确定日志服务启动：# ps aux | grep rsyslogd



注：rsyslogd日志服务默认开机自启动，常见日志文件默认保存在“/var/log”，包括RPM安装的系统服务日志。

1. 配置文件/etc/rsyslog.conf

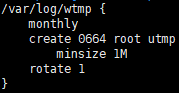
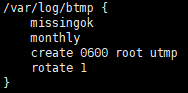


注：字段依次表示服务名称、连接符号、日志等级、日志记录位置

|  |  |
| --- | --- |
| 连接符号 | 含义 |
| . | 只要比后面日志等级高（包括该等级）的日志都记录下来 |
| .= | 只记录所需等级的日志，其它等级日志都不记录 |
| .! | 除了该等级日志外，其它等级日志都记录下来 |
| 日志等级：debug < info < notice < warning < err < crit < alert < emerg，\*表示所有等级 | |

1. 日志轮替

（1） 配置文件/etc/logrotate.conf

注：日志轮替指将日志按时间分开记录，同时定期用新日志替换旧日志，防止日志文件占用太多硬盘空间。只有源码包安装的服务需要手动添加日志轮替，如将用源码安装的apache日志加入轮替，只需根据格式编辑该配置文件

|  |  |
| --- | --- |
| 参数 | 说明 |
| daily/weekly/monthly | 轮替周期每天/每周/每月 |
| rotate 数字 | 保留日志文件的个数，0指没有备份 |
| create mode owner group | 建立新日志，同时指定日志的权限、所有者和所属组 |
| missingok | 如果日志不存在，则忽略日志的警告信息 |
| minsize 大小 | 日志轮替的最小值，只有达到这个值，日志才轮替 |
| dateext | 使用日期作为日志轮替文件的后缀 |

（2） logrotate命令：# logrotate [-vf] 配置文件名

* -v：显示日志轮替过程
* -f：强制进行日志轮替



## 启动管理

1. 系统启动流程

（1） 加载BIOS的硬件信息与进行自检，并根据设置取得第一个可启动的设备

（2） 读取并执行第一个启动设备内MBR的启动引导程序Boot loader，即grub2

（3） 根据启动引导程序的设置加载Kernel，内核会检测硬件与加载驱动程序

（4） Kernel主动调用systemd程序，并以default.target流程启动



注：内核需要加载SATA硬盘的驱动程序（相当于钥匙），才能挂载根目录（相当于房间），但是SATA的驱动程序在/lib目录下（钥匙在房间里），无法挂载根目录就不能读取驱动程序。解决方法是使用虚拟文件系统，文件名一般使用/boot/initrd或/boot/initramfs，它通过boot loader加载到内存，并解压缩模拟成一个根目录，加载启动过程中最基本驱动程序

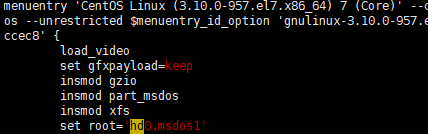
1. 启动引导程序grub2

（1） grub2中分区表示

|  |  |  |  |
| --- | --- | --- | --- |
| 硬盘 | 分区 | Linux中设备文件名 | grub2中设备文件名 |
| 第一块  SCSI硬盘 | 第1个主分区 | /dev/sda1 | (hd0,1) |
| 第2个主分区 | /dev/sda2 | (hd 0,2) |
| 扩展分区 | /dev/sda3 | (hd 0,3) |
| 第1个逻辑分区 | /dev/sda5 | (hd 0,5) |
| 第二块  SCSI硬盘 | 第1个主分区 | /dev/sdb1 | (hd 1,1) |
| 第2个主分区 | /dev/sdb2 | (hd 1,2) |
| 扩展分区 | /dev/sdb3 | (hd 1,3) |
| 第1个逻辑分区 | /dev/sdb5 | (hd 1,5) |

注：以查找顺序作为硬盘编号（从0开始），每块硬盘的第一分区代号为1。为了区分不同分区格式，分区号前加上msdos与gpt，分别表示MBR与GPT格式

（2） 配置文件：# vim /boot/grub2/grub.cfg



## 备份与恢复

1. 备份概述

（1） 需要备份的目录：/root、/home、/etc、/var/spool/mail、/var/lib、其它目录

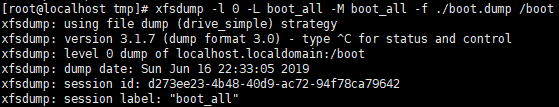
不需要备份的目录：/dev、/tmp、/proc、/sys、/run、/mnt、/media

（2） 备份策略：完全备份、增量备份、差异备份

1. XFS文件系统备份：# xfsdump [-LMIl] [-f 备份后的文件名] 待备份数据

* -L：后面接针对文件系统的简易说明
* -M：后面接存储媒介的简易说明
* -l：后面接备份级别0-9（默认0表示完全备份，1-9表示增量备份）
* -I：列出当前备份的信息状态

注：xfsdump只能备份已挂载的xfs文件系统，不支持备份文件目录（可使用命令tar）。经过完整备份（-l 0）后才能进行增量备份（-l 1-9）





1. XFS文件系统恢复：# xfsrestore [-f 备份文件] [-ILsri] 待恢复目录

* -I：列出当前备份的信息状态
* -L：完全备份数据恢复，后面接针对文件系统的简易说明，可用-I查询
* -s：后面接某特定的目录，即仅恢复某一个文件或目录
* -r：增量备份数据恢复
* -i：交互模式，可选择需要恢复的文件

注：选项-L、-r、-i不能混用，恢复时必须按照备份流程进行，如备份流程level 0 -> level 1 -> level 2，则恢复也应该依次进行

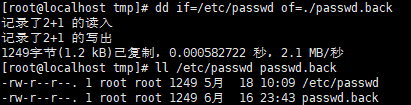


1. 其他备份命令

（1）# dd if=input\_file of=output\_file bs=block\_size count=number

* if：输入文件或输入设备
* of：输出文件或输出设备
* bs：设置一个block的大小（默认512B）
* count：共多少个bs，故bs\*count就是文件的大小

注：dd命令还可以用来建立空文件，使用“if=/dev/zero”指定会一直输出0的设备





（2）备份：# cpio -ovcB > 文件或设备名

还原：# cpio -ivcdu < 文件或设备名

查看：# cpio -ivct < 文件或设备名

* -o：将数据复制输出到文件或设备上
* -i：将数据自文件或设备复制出来到系统当中
* -u：自动将较新的文件覆盖较旧的文件

注：cpio命令可以备份任何东西，包括设备文件。但它不会主动去找文件来备份，所以一般要配合类似find等可以查找文件命令来使用。另外，备份时需要切换目录，使用相对路径查找，否则如果加上绝对路径的/，那么还原式就会覆盖原来的/boot



参考文献

1. 腾讯课堂. Linux视频教程[DB/OL]. <https://www.bilibili.com/video/av18156598>，2018.
2. 鸟哥. 鸟哥的Linux私房菜：基础学习篇[M]. 北京：人民邮电出版社，2018.
3. GitHub,Inc. 命令行的艺术[DB/OL].

<https://github.com/jlevy/the-art-of-command-line/blob/master/README-zh.md?tdsourcetag=s_pcqq_aiomsg>，2017

附录A Linux目录树

