# 目录

[目录 1](#_Toc502000027)

[第1章 认识linux 7](#_Toc502000028)

[1.1 linux简介 7](#_Toc502000029)

[1.1.1 linux简述 7](#_Toc502000030)

[1.1.2 linux诞生历程 7](#_Toc502000031)

[1.1.3 GNU NPL 简介 7](#_Toc502000032)

[1.1.4 linux特点 8](#_Toc502000033)

[1.1.5 涉及的领域 8](#_Toc502000034)

[1.2 Centos与Red Hat的区别和联系 8](#_Toc502000035)

[1.3 Centos的安装 8](#_Toc502000036)

[1.3.1 配置虚拟机 8](#_Toc502000037)

[1.3.2 网卡模式 8](#_Toc502000038)

[1.3.3 nat 模式设置步骤 9](#_Toc502000039)

[1.3.4 桥接模式设置 11](#_Toc502000040)

[1.4 克隆机的网卡配置 12](#_Toc502000041)

[1.5 xshell的优化 13](#_Toc502000042)

[1.5.1 终端 13](#_Toc502000043)

[1.5.2 外观 13](#_Toc502000044)

[1.5.3 文件路径 14](#_Toc502000045)

[第2章 linux基础命令 14](#_Toc502000046)

[2.1帮助命令 14](#_Toc502000047)

[2.1.1 man 14](#_Toc502000048)

[2.1.2 help 15](#_Toc502000049)

[2.2文件和目录操作命令 15](#_Toc502000050)

[2.2.1 mv 15](#_Toc502000051)

[2.2.2 mkdir 15](#_Toc502000052)

[2.2.3 touch 15](#_Toc502000053)

[2.2.4 rmdir 16](#_Toc502000054)

[2.2.5 rm 16](#_Toc502000055)

[2.2.6 cd 16](#_Toc502000056)

[2.2.7 cp 16](#_Toc502000057)

[2.2.8 pwd 17](#_Toc502000058)

[2.2.9 ls 17](#_Toc502000059)

[2.2.10 tree 17](#_Toc502000060)

[2.2.11 find 18](#_Toc502000061)

[2.3文件查看和内容处理命令 18](#_Toc502000062)

[2.3.1 cat 18](#_Toc502000063)

[2.3.2 head 19](#_Toc502000064)

[2.3.3 tail 19](#_Toc502000065)

[2.3.4 grep 19](#_Toc502000066)

[2.3.5 egrep 20](#_Toc502000067)

[2.3.6 sed 20](#_Toc502000068)

[2.3.7 awk 20](#_Toc502000069)

[2.3.8 vi 20](#_Toc502000070)

[2.3.9 vim 20](#_Toc502000071)

[2.3.10 more 20](#_Toc502000072)

[2.3.11 less 21](#_Toc502000073)

[2.3.12 wc 21](#_Toc502000074)

[2.3.13 file 21](#_Toc502000075)

[2.3.14 stat 21](#_Toc502000076)

[2.3.15 dumpe2fs 21](#_Toc502000077)

[2.4其他命令 22](#_Toc502000078)

[2.4.1 alias 22](#_Toc502000079)

[2.4.2 unalias 22](#_Toc502000080)

[2.4.3 seq 22](#_Toc502000081)

[2.4.4 echo 22](#_Toc502000082)

[2.4.5 xargs 22](#_Toc502000083)

[2.4.6 env 22](#_Toc502000084)

[2.4.7 ln 23](#_Toc502000085)

[2.4.8 basename 23](#_Toc502000086)

[2.4.9 dirname 23](#_Toc502000087)

[2.4.10 rename 23](#_Toc502000088)

[2.4.11 which 24](#_Toc502000089)

[2.4.12 whereis 24](#_Toc502000090)

[2.4.13 locate 24](#_Toc502000091)

[2.4.14 readlink 24](#_Toc502000092)

[2.4.15 md5sum 25](#_Toc502000093)

[2.5 系统相关的命令 25](#_Toc502000094)

[2.5.1 uname 25](#_Toc502000095)

[2.5.2 runlevel 26](#_Toc502000096)

[2.5.3 init 26](#_Toc502000097)

[2.5.4 history 26](#_Toc502000098)

[2.5.5 shutdown 26](#_Toc502000099)

[2.5.6 halt 26](#_Toc502000100)

[2.5.7 chkconfig 26](#_Toc502000101)

[2.5.8 last 26](#_Toc502000102)

[2.5.9 lastb 27](#_Toc502000103)

[2.5.10 lastlog 27](#_Toc502000104)

[2.5.11 w 27](#_Toc502000105)

[2.5.12 du 28](#_Toc502000106)

[2.5.13 date 28](#_Toc502000107)

[2.6 用户相关的命令 29](#_Toc502000108)

[2.6.1 useradd 29](#_Toc502000109)

[2.6.2 userdel 30](#_Toc502000110)

[2.6.3 usermod 30](#_Toc502000111)

[2.6.4 groupadd 31](#_Toc502000112)

[2.6.5 groupdel 31](#_Toc502000113)

[2.6.6 chown 31](#_Toc502000114)

[2.6.7 chmod 31](#_Toc502000115)

[2.6.8 umask 32](#_Toc502000116)

[2.6.9 chattr 32](#_Toc502000117)

[2.6.10 lsattr 32](#_Toc502000118)

[2.6.11 id 33](#_Toc502000119)

[2.6.12 chage 33](#_Toc502000120)

[2.6.13 su 34](#_Toc502000121)

[2.6.17 sudo 34](#_Toc502000122)

[2.6.18 visudo 34](#_Toc502000123)

[2.7 系统常用开关机重启以及注销常用的命令 35](#_Toc502000124)

[2.8 备注 35](#_Toc502000125)

[第3章 linux基础入门中部 35](#_Toc502000126)

[3.1 linux目录的基础知识 35](#_Toc502000127)

[3.1.1 linux目录的特点 35](#_Toc502000128)

[3.1.2 linux树状目录结构图 36](#_Toc502000129)

[3.1.3 linux常用目录详解 36](#_Toc502000130)

[3.2 linux文件属性 38](#_Toc502000131)

[3.2.1 linux文件的概述 38](#_Toc502000132)

[3.2.2 inode 索引节点 39](#_Toc502000133)

[3.2.3 block 40](#_Toc502000134)

[3.2.4 相关的面试题 40](#_Toc502000135)

[3.2.5 linux软硬连接 40](#_Toc502000136)

[3.2.6 软连接 41](#_Toc502000137)

[3.2.7 文件连接的总结 42](#_Toc502000138)

[3.2.8 目录连接文件的总结 43](#_Toc502000139)

[3.2.9 软硬连接的区别 43](#_Toc502000140)

[3.2.10 linux文件的删除原理 43](#_Toc502000141)

[3.3 linux文件的权限 44](#_Toc502000142)

[3.3.1 rwx详解 44](#_Toc502000143)

[3.3.2 权限位图解 44](#_Toc502000144)

[3.3.3 linux普通文件权限的读、写、执行的解读 45](#_Toc502000145)

[3.3.4 目录权限的读、写、执行的解读 45](#_Toc502000146)

[3.3.5 访问文件的图解说明 以/tmp/oldboyfile为例 46](#_Toc502000147)

[3.3.6 umak 46](#_Toc502000148)

[3.3.7 suid 46](#_Toc502000149)

[3.3.8 sgid 47](#_Toc502000150)

[3.4 用户 47](#_Toc502000151)

[3.4.1 用户的分类 47](#_Toc502000152)

[3.4.2 用户的详解 47](#_Toc502000153)

[3.4.3 用户相关的配置文件 48](#_Toc502000154)

[3.4.4 /etc/passwd 配置文件内容的详解 48](#_Toc502000155)

[3.4.5 删除用户的经验 48](#_Toc502000156)

[3.5 有关用户管理的三个重要目录和文件 49](#_Toc502000157)

[3.5.1 linux用户管理重要初始化目录 /etc/skel 49](#_Toc502000158)

[3.5.2 /etc/login.defs配置文件的详解（了解即可） 50](#_Toc502000159)

[3.5.3 /etc/default/useradd配置文件的详解 52](#_Toc502000160)

[3.6 sudo配置文件/etc/sudoers的讲解 53](#_Toc502000161)

[3.6.1 /etc/sudoer文件中各选项的作用 53](#_Toc502000162)

[3.6.2 配置sudo配置文件需要注意的事项 53](#_Toc502000163)

[3.6.3 用户的提权配置说明 53](#_Toc502000164)

[3.6.4 sudo执行过程图解 54](#_Toc502000165)

[3.6.5 sudo的审计功能 54](#_Toc502000166)

[sudo配置实例 55](#_Toc502000167)

[3.7 环境变量 56](#_Toc502000168)

[第四章 linux基础入门下 57](#_Toc502000169)

[4.1 linux通配符 57](#_Toc502000170)

[4.1.1 linux通配符与正则表达式的区别 57](#_Toc502000171)

[4.1.2 常用通配符 57](#_Toc502000172)

[4.1.3 举例 58](#_Toc502000173)

[4.2 linux 正则表达式 60](#_Toc502000174)

[4.2.1 linux基础正则符号 60](#_Toc502000175)

[4.2.2举例 60](#_Toc502000176)

[4.2.3 linux扩展的正则表达式 65](#_Toc502000177)

[4.2.4 举例 65](#_Toc502000178)

[4.2.5 元字符 66](#_Toc502000179)

[4.2.5 POSIX字符类 66](#_Toc502000180)

[4.3 linux 定时任务之crond 67](#_Toc502000181)

[4.3.1 什么是crond 67](#_Toc502000182)

[4.3.2 为什么要使用crond定时任务 67](#_Toc502000183)

[4.3.3 linux系统的定时任务 67](#_Toc502000184)

[4.3.4 crond的基本格式 67](#_Toc502000185)

[4.3.5 crond语法格式中时间段的含义表及取值范围 68](#_Toc502000186)

[4.3.6 crond语法格式中的特殊符号及其含义 68](#_Toc502000187)

[4.3.7 crontab 指令的说明 69](#_Toc502000188)

[4.3.8 使用者权限及定时任务文件 69](#_Toc502000189)

[4.3.9 crontab设置的规范 69](#_Toc502000190)

[4.3.10 crond /dev/null 2>&1的作用 70](#_Toc502000191)

[4.3.11 生产环境中调试定时任务的方法 70](#_Toc502000192)

[第5章 linux性能优化 71](#_Toc502000193)

[5.1 关闭selinux 71](#_Toc502000194)

[5.1.1 selinux的配置文件 71](#_Toc502000195)

[5.1.2 临时关闭 71](#_Toc502000196)

[5.1.3 永久关闭 71](#_Toc502000197)

[5.2 关闭防火墙 72](#_Toc502000198)

[5.2.1 临时关闭 72](#_Toc502000199)

[5.2.2 永久关闭 72](#_Toc502000200)

[5.2.3 查看iptables的状态 72](#_Toc502000201)

[5.3 linux中文显示设置 72](#_Toc502000202)

[5.3.1 配置文件 72](#_Toc502000203)

[5.3.2 设置中文字符 72](#_Toc502000204)

[5.3 隐藏linux版本信息 72](#_Toc502000205)

[5.3.1 版本信息的配置文件 72](#_Toc502000206)

[5.3.2 清空该配置文件中的内容 73](#_Toc502000207)

[5.4 精简开机自启动服务 73](#_Toc502000208)

[5.4.1 需要保留的重要开机自启动服务 73](#_Toc502000209)

[5.4.2 开启需要的服务 74](#_Toc502000210)

# 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
* 有大量的平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

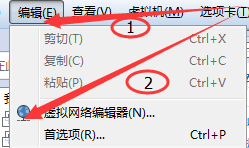
* 简介：物理主机移动时，虚拟机需要更改网络配置

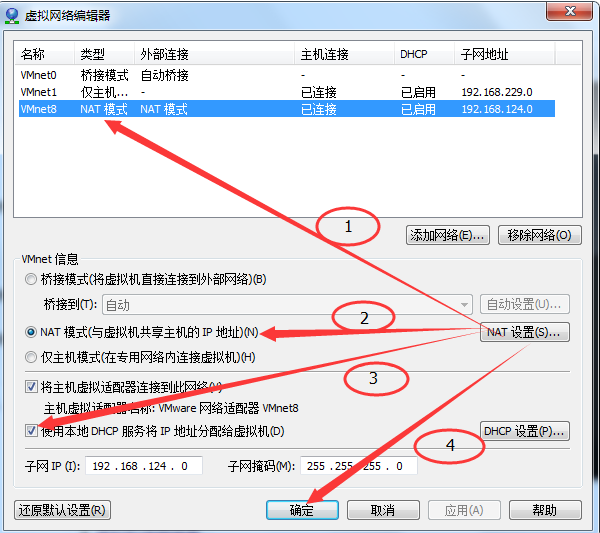
#### 1.3.3 Host-only 模式 仅主机模式

* 简介：虚拟机和物理主机可以互访，但是出不去

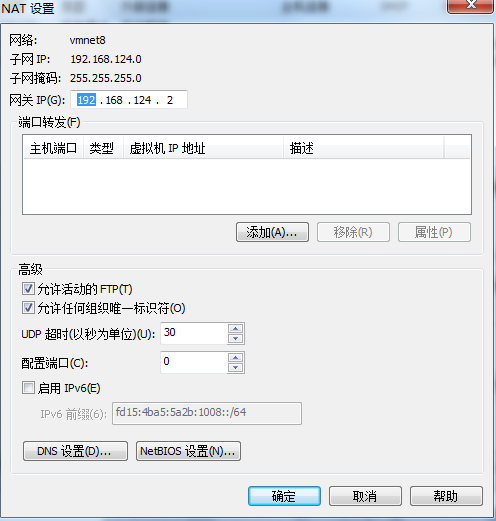
### 1.3.3 nat 模式设置步骤

#### 1.3.3.1 虚拟网络编辑器设置



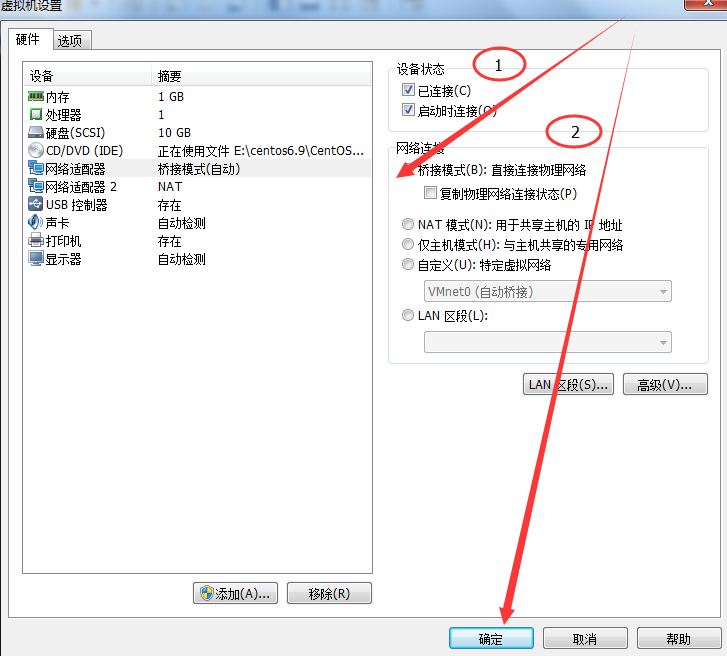


#### 1.3.3.2 nat设置

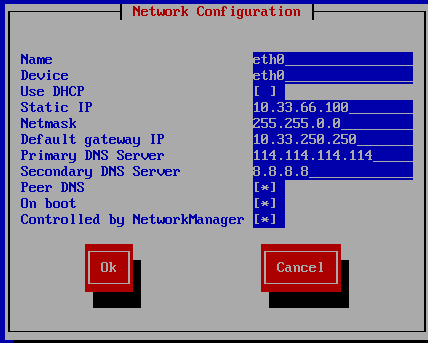


### 1.3.4 桥接模式设置

#### 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

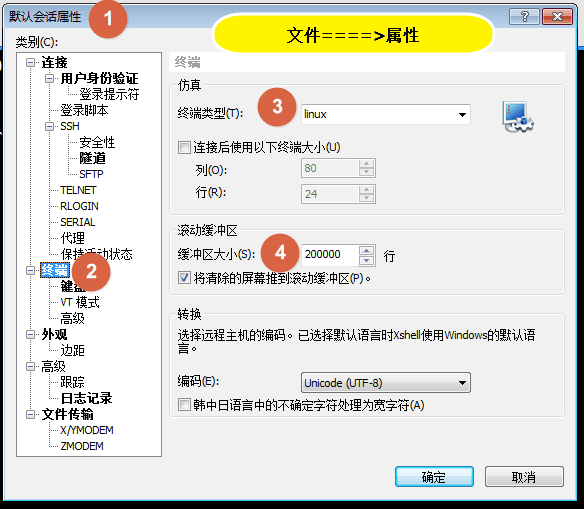
* 重新启动克隆机

[root@zabbix-1 ~]# reboot

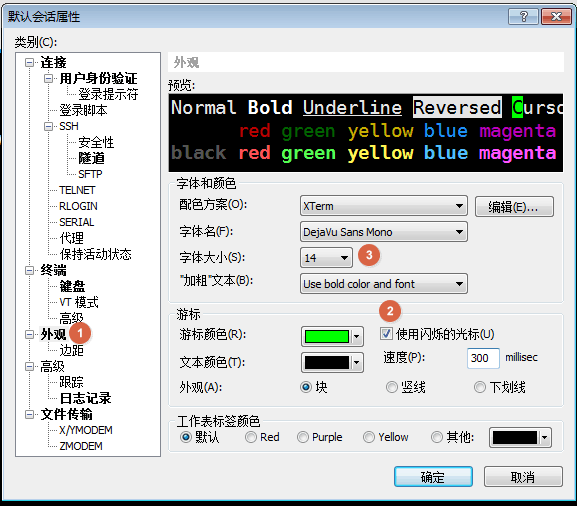
* 【说明】以上步骤可以简记为 两删一清空一重启

## 1.5 xshell的优化

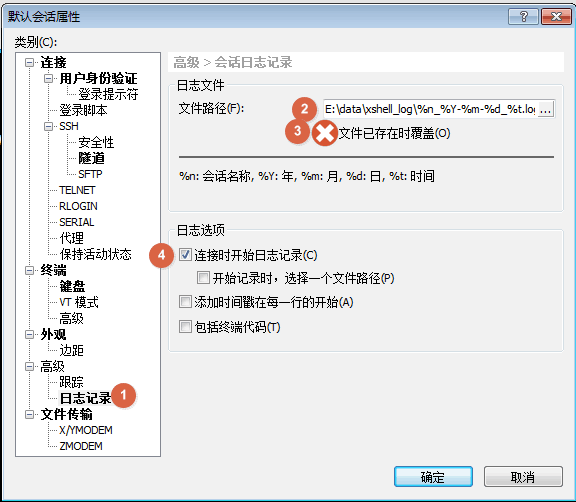
### 1.5.1 终端

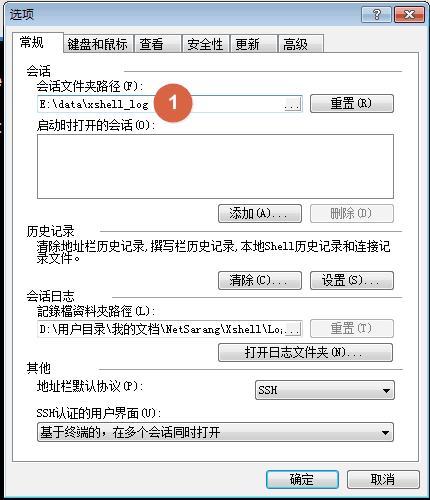


### 1.5.2 外观



### 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

### 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

* 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

* 【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

### 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

* 【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

## 2.4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/

### 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

## 2.5 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】

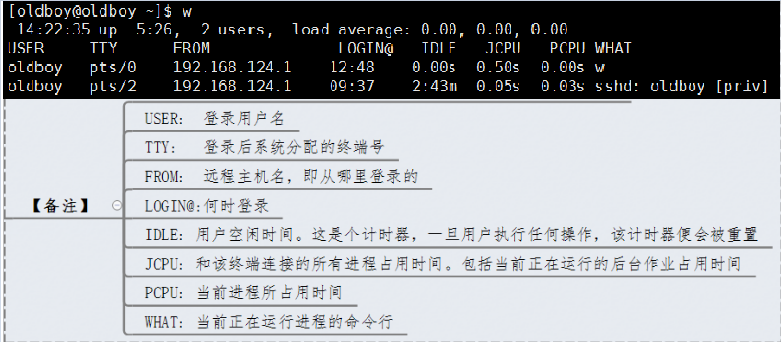
-h：不显示标题信息

-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

* 【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

## 2.6 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

sudo的配置文件是 /etc/sudoers

## 2.7 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

## 2.8 备注

vi，vim，三剑客，find 命令详解见别处

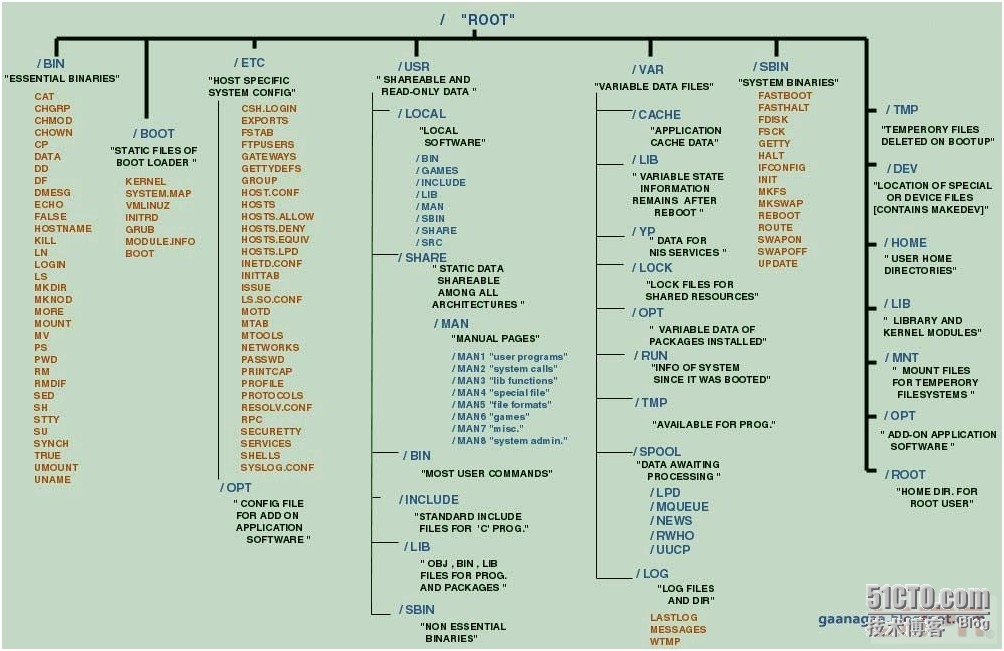
# 第3章 linux基础入门中部

## 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一且皆文件）
* 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， |

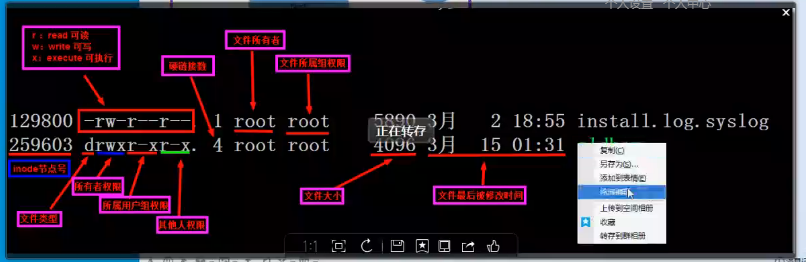
#### 3.1.3.4 /proc 目录

|  |  |
| --- | --- |
| 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本事是有大小的，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。
* block是用来存放实际数据的，常见的一个block的大小位4K，其引导分区为1K,其它的普通分区位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，可以理解为时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机会消耗一个磁盘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

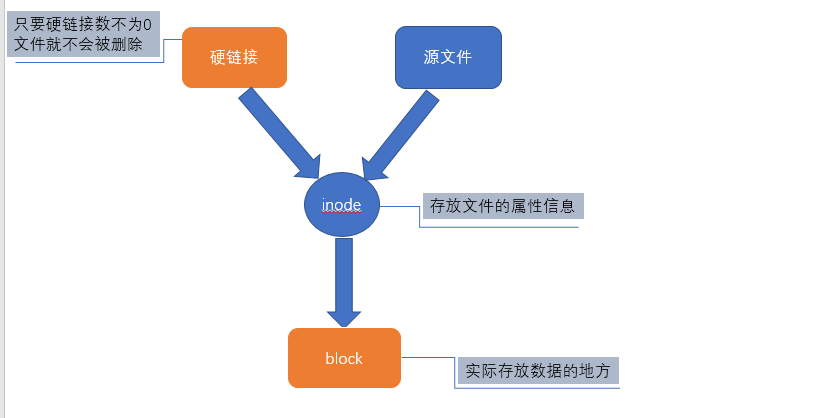
第二种：inode满了，可以用df -i查看，看IUse%，删除一些小的（不用的）文件即可

### 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

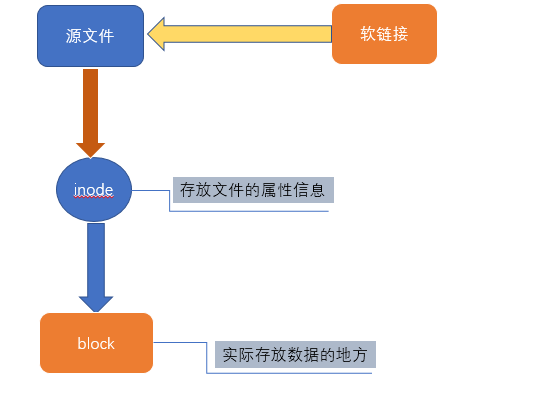
* 多个文件共享同一个索引节点，不占用实际空间
* 硬链接是一个文件的多个入口（可以理解为一个超市具有多个入口一样）
* 只要连接数不为0，文件就存在，所以硬链接可以防止文件误删除
* 删除源文件或硬链接的任意一个，则文件实体并不会被删除，只有同时删除源文件和其所有硬链接，文件实体才会被删除
* 硬链接文件是普通文件，可以用rm命令删除
* 对于静态文件（即没有被进程调用的文件），当i\_link数为0时，文件就被删除了（i\_link的查看方法ls –l 的第三列）

### 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

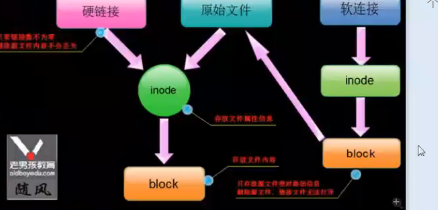


#### 3.2.6.3 软连接的特点

* 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

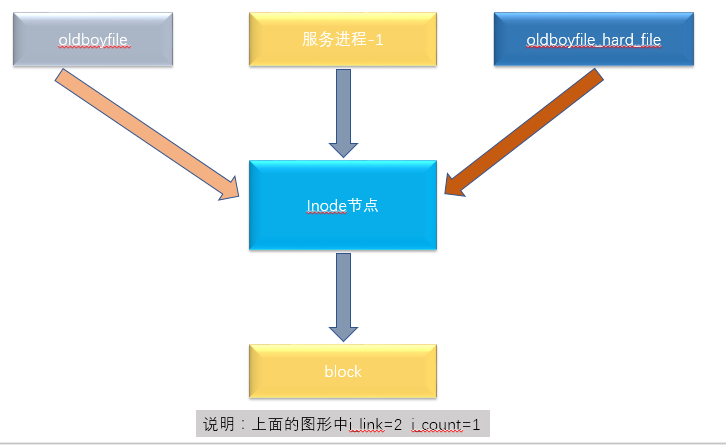
* 目录不可以创建硬链接，但是可以创建软连接
* 目录的硬链接不能跨越文件系统（从硬链接的原理进行理解）
* 每个目录下面都有一个硬链接 **.** 号，和对应上级目录的硬链接 **..**
* 在父目录下创建一个子目录，则父目录的硬链接数加1（因为在子目录里面都有一个 **..** 指向父目录），但是在下面创建文件，父目录的硬链接数不会增加

### 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
* 删除盈软连接不影响源文件，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

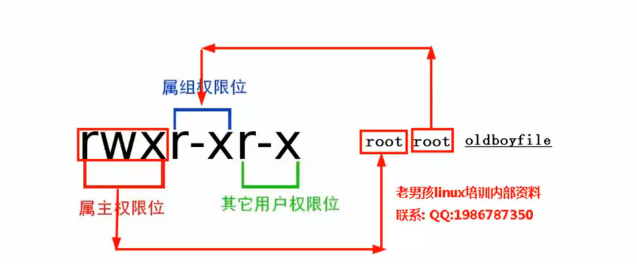
**删除文件的权限是受父目录的控制和文件本省的权限无关**

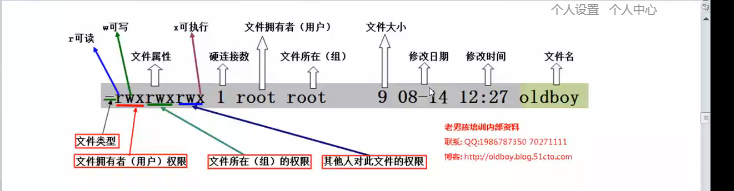
## 3.3 linux文件的权限

### 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

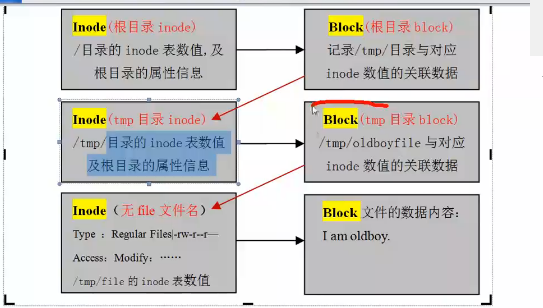
### 3.3.3 linux普通文件权限的读、写、执行的解读

* 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权限的最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
* 【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

## 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UID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/目录吓得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目录吓得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：

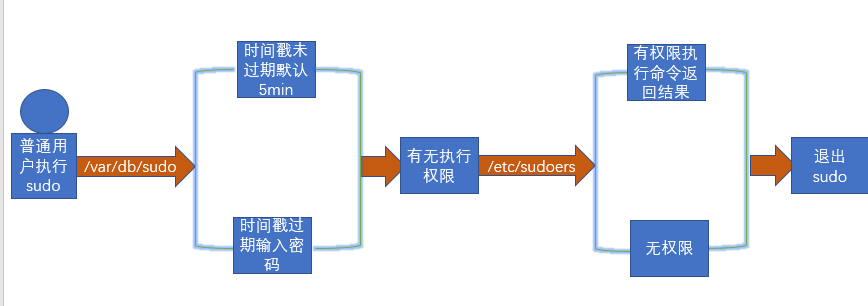
用户 机器 角色 命令

* 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 ~]#

[root@oldboy ~]#

[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

## 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

#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

# 第四章 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 |
| > | 重定向，会覆盖原数据 |
| >> | 追加重定向，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

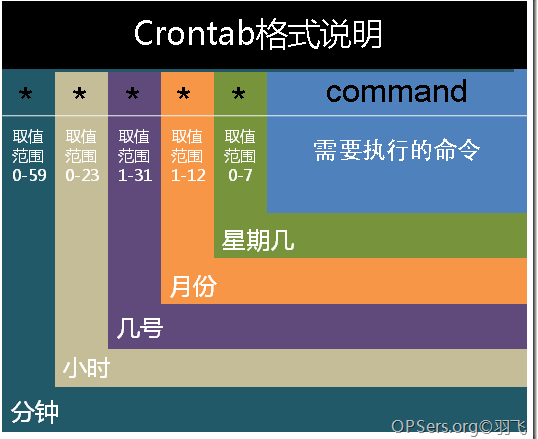
\* \* \* \* \* cmd

* 【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
* 现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 "s#SELINUX=enable#SELINUX=disabled#" /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

## 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 第一种方法：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