

用户相关操作

管理员显示# 普通用户显示\$

- 查看当前的用户: whoami
- 添加用户: useradd 用户名

用户创建在/etc/passwd 目录中

- 设置登入密码: passwd 用户名
- 切换用户: su 用户名 (直接 su 回车直接切换到 root 用户)

注意: 管理员切换普通用户不需要密码, 普通用户切换需要密码

Shell 快捷键

- Ctrl+a: 跳到命令行开头
- Ctrl+e: 跳到命令行末尾
- Ctrl+u: 将光标到命令行开头的内容清除
- Ctrl+k: 将光标到命令行结尾的内容清除
- Ctrl+r: 在历史记录列表中搜索某一条命令
- Ctrl+l: 清屏

- 远程虚拟机: ssh 用户名@IP 地址

虚拟机查看 IP 地址: ip addr

- 查看历史命令: history
!【number】直接执行曾经执行过的命令

路径操作

- 查看当前所在的路径：pwd
- 查看当前路径下的文件：ls
 - l: 查看文件详细信息（同 ll）
 - a: 查看所有的文件（包括隐藏文件）
 - h: 显示文件大小
 - d: 只看当前目录的信息
(可组合使用)
- 查看某个目录下的详细信息：ls 目录
- 路径切换：cd 路径（/为根路径）
 - : 回到之前的目录
 - ..: 返回上一级目录
 - .: 当前目录
 - ~: 到用户目录
- 绝对路径/相对路径
- touch: 创建一个文件

目录结构

- `/boot` Linux启动时，需要的文件
 - `/dev` 设备文件
 - `/etc` 配置文件
 - `/home` 用户家目录
 - `/media` 媒体文件
 - `/mnt` 挂载文件
 - `/opt` 第三方软件
 - `/proc` 虚拟化文件
 - `/root` 管理员的家目录
 - `/run` 进程文件
-
- `/srv` 压缩过的文件
 - `/sys` 系统文件
 - `/usr` 安装的软件，共享库
 - `/var` 可变数据，日志文件
 - `/tmp` 临时文件
 - `/usr/bin` 普通用户可以使用的命令
 - `/usr/sbin` 超级用户可以使用的命令
 - `/usr/lib` 32位库文件
 - `/usr/lib64` 64位库文件

文件处理

Linux文件处理

```
[root@localhost ~]# ls -l /etc/passwd
-rw-r--r--. 1 root root 1090 Sep 18 12:58 /etc/passwd
```

- d: 表示目录
- : 表示文件
- l: 连接文件 I
- b: 设备文件，提供存储的接口设备
- c: 设备文件，提供串行的接口设备--键盘，鼠标

文件处理

活动	单 来源	多来源
复制文件	<code>cp file1 file2</code>	<code>cp file1 file2 file3 dir</code> ⁽⁵⁾
移动文件	<code>mv file1 file2</code> ⁽¹⁾	<code>mv file1 file2 file3 dir</code> ⁽⁴⁾
删除文件	<code>rm file1</code>	<code>rm -f file1 file2 file3</code> ⁽⁵⁾
创建目录	<code>mkdir dir</code>	<code>mkdir -p par1/par2/dir</code> ⁽⁶⁾
复制目录	<code>cp -r dir1 dir2</code> ⁽²⁾	<code>cp -r dir1 dir2 dir3 dir4</code> ⁽⁴⁾
移动目录	<code>mv dir1 dir2</code> ⁽³⁾	<code>mv dir1 dir2 dir3 dir4</code> ⁽⁴⁾
删除目录	<code>rm -r dir1</code> ⁽²⁾	<code>rm -rf dir1 dir2 dir3</code> ⁽⁵⁾
注:	<p>(1)结果为重命名。 (2)需要使用“递归”选项处理来源目录。 (3)如果 dir2 存在，则结果为移动。如果 dir2 不存在，则结果为重命名。 (4)最后一个参数必须是目录。 (5)请谨慎使用“force”选项，系统将不会提示您确认操作。 (6)使用“创建父级”选项时应小心：无法捕获键入错误。</p>	

- 如果是 root 用户删除文件会进行询问
- rm -f: 强制删除不会询问（目录不会删除）
- *通配符选择所有的非隐藏文件
- .选择所有的隐藏的文件
- rm -r: 删除目录（同 rmdir）
- 删除一个目录所有: `rm -fr /tmp/*`
- 创建一个目录: `mkdir` （只能创建一个新目录或者在已有的目录下创建一个新目录）
- 递归创建目录: `mkdir -p` （可创建多个不存在的嵌套的目录）
- 创建相应权限的目录: `mkdir -m 权限等级 目录名`
- 复制文件: `cp a b` 复制一个名为 b 的 a 文件副本
- 复制路径: `cp -r a b` 复制一个名为 b 的 a 路径副本（包括 a 路径中的文件路径）

- 重命名: mv a b 如果 b 路径不存在则 a 重命名为 b (a 可以是文件也可以是路径)
- 移动: mv a b 如果 b 路径存在则 a 被移动到 b 路径中 (b 路径中有重名文件则覆盖 b 路径中的文件。路径中有重名路径则不可以移动)

查看文件

- 查看文件: cat 文件名 (正序显示) tac 文件名 (倒序显示)
 - b: 列出行号 (不包括空白行)
 - n: 列出行号 (包括空白行)
 - E: 行尾显示\$换行符
 - T: 将 tab 键以^I 显示
 - v: 列出一些特殊字符
 - A: 整合命令-vET
- 查看文件: nl 文件名 (显示行号)

查看文件-nl

- nl
 - b
 - ba //无论是否有空行，都列出行号
 - bt //如果有空行，则不列出行号（默认）
 - n
 - nln //行号在屏幕最左方
 - nrn //行号在屏幕最右方，前面不加0
 - nrz //行号在屏幕最右方，前面加0
 - w //缩进多少位
- 查看文件：less 文件名

查看文件-less

- less (一页一页翻动)
 - 空格：向下翻动一页
 - pagedown：向下翻动一页
 - pageup：向上翻动一页
 - n：重复前一个搜索
 - N：反方向重复前一个搜索
 - q：退出
- 查看文件：more 文件名

查看文件-more

- **more** (一页一页翻动)
 - 空格: 向下翻一页
 - 回车: 向下翻一行
 - /字符串: 所搜
 - :f: 立刻显示文件名和行数
 - b: 翻到第一页
 - q: 离开
- 查看文件: head, tail 文件名

查看文件

head,tail: 显示文件头或尾几行

-n: 指定显示几行。默认是10行

日志文件: /var/log/messages。我们只需要看尾部的最新的几行即可
目录结构

wc: 显示文件的行数, 数字, 字节

-c: 只显示字节

-w: 只显示字数。一个字被定义为由空白、跳格或换行字符分隔的字符串。

-l: 只显示行

VIM 编辑器 (linux 内置 VI 编辑器, VIM 编辑器需要安装)

- VIM 编辑器会对文件进行颜色标记
- 创建: vim 文件名

Vim编辑器

i键插入

esc退出

u撤销

x删除

v选择文本

y复制

yy复制当前行

dd删除光标所在行

p粘贴

:w保存

:wq保存退出

:q!强退

gedit来进行编辑--需要x-windows的支持

光标上下左右移动: hjkl (数字+方向可以指定移动的位置)

Ctrl+f 屏幕向下翻一页

Ctrl+b 屏幕向上翻一页

Ctrl+d 屏幕向下翻半页

Ctrl+u 屏幕向上翻半页

+ 光标移动到下一行的第一个非空字符

- 光标移动到当前行的第一个非空字符

0(数字 0) 光标移动到当前行的第一个字符(可以为空字

符,注意与-区分)

\$ 光标移动到当前行的最后一个字符(可以为空
字符,注意与-区分)

G 光标移动到该文章最后一行的第一个非空字
符

gg	光标移动到该文章最后一行的第一个非空字符
nG(n 代表数字)	光标移动到该文章第 n 行的第一个非空字符
/word	在光标之后查找 word 字符串
?word	在光标之前查找 word 字符串

gedit 编辑器（直接在虚拟机上操作或使用第三方软件 x-manger）

软连接

文件处理-软链接

软连接=windows下的快捷方式

特点：

- 1、可以对不存在文件创建软链接
- 2、可以对文件或者目录创建软链接
- 3、删除了软链接，不会影响他的指向文件
- 4、删除了指向文件，该软链接就会变成死链接



创建软连接：ln -s 源文件 链接文件

硬链接

文件处理-硬链接

Inode编号

文件名：给人看的

Inode编号：给机器看的

硬链接：多个文件对应同一个**inode**编号

对多个文件中的某一个文件进行修改时，其他文件也会同时发生更改。多用户协同一致工作。

I

特点：

- 1、不能对目录进行硬链接的创建
- 2、只能对已经存在的文件进行硬链接的创建
- 3、删除一个硬链接文件，不影响其他相同的**inode**编号的文件

通过 ls -i 查看 inode 编号

创建硬链接：link 源文件 链接文件 或 ln 源文件 链接文件

关机

关机

- 要注意的事项：
 - 观察系统的使用状态
 - 通知在线使用者关机的时间
 - 正确的关机指令使用
- 关机的指令
 - 数据同步写入磁盘-sync
 - 常用的关机指令：shutdown
 - 重启，关机：reboot, halt, poweroff

- Shutdown
 - t 添加秒数，几秒后关机
 - k 不是真关机，而是发出告警信息
 - r 在系统服务都停止后，重启
 - h 在系统服务都停止后，关机
 - f 关闭并且开机以后，强行略过磁盘检查
 - F 重启后，强制进行进行磁盘检查
 - c 取消已经在进行的shutdown指令内容

- shutdown -h now
- shutdown -h 12:00
- shutdown -h +10
- shutdown -r now
- shutdown -r +30 'the system will reboot'
- shutdown -k now 'this system will reboot'
- 服务等级
 - init0 //关机
 - init6 // 重启
 - init3 //纯文本模式
 - init5 //含有图形接口

查看 cpu 信息：lscpu

查看磁盘空间：df

-h: 显示总大小

-i: 显示索引节点索引的空间

帮助文件

1. Pinfo 2. Ls /usr/share/doc 3. man

帮助文件-man

章节	内容类型
1	用户命令（可执行命令和 shell 程序）
2	系统调用（从用户空间调用的内核例程）
3	库函数（由程序库提供）
4	特殊文件（如设备文件）
5	文件格式（用于许多配置文件和结构）
6	游戏（过去的有趣程序章节）
7	惯例、标准和其他（协议、文件系统）
8	系统管理和特权命令（维护任务）
9	Linux 内核 API（内核调用）

帮助文件-man

导航 man page	
命令	结果
空格键	向前（向下）滚动一个屏幕
PageDown	向前（向下）滚动一个屏幕
PageUp	向后（向上）滚动一个屏幕
向下箭头键	向前（向下）滚动一行
向上箭头键	向后（向上）滚动一行
d	向前（向下）滚动半个屏幕
u	向后（向上）滚动半个屏幕
/string	在 man page 中向前（向下）搜索 string
n	在 man page 中重复之前的向前（向下）搜索
N	在 man page 中重复之前的向后（向上）搜索
g	转到 man page 的开头。
G	转到 man page 的末尾。
q	退出 man，并返回到命令 shell 提示符

文件权限

Linux文件权限

```
[root@localhost ~]# ls -l /etc/passwd  
-rw-r--r--. 1 root root 1090 Sep 18 12:58 /etc/passwd
```

权限	连接	所有者	所属组	容量-默认单位B	修改的日期	文件名
----	----	-----	-----	----------	-------	-----

权限：

文件的属性：

d: 表示目录

-: 表示文件

l: 连接文件

b: 设备文件，提供存储的接口设备

c: 设备文件，提供串行的接口设备—键盘，鼠标

权限：
文件的权限：所有者，所属组，其他人
rwx , 读、写、执行，没有权限就是-
第一个组 rwx : 文件所有者的权限
第二个组 rwx : 文件所属组的权限
第三个组 rwx : 文件其他人的权限

权限：

目录的权限：

r: 具有读取目录结构列表的权限，可以查看目录下有哪些文件

w: 该权限对于目录来说是很大的，

- o 1、可以在该目录下新建新的文件和目录
- o 2、可以删除已经存在的文件和目录
- o 3、将已经存在的文件和目录重命名
- o 4、移动该目录内的文件和目录的位置

x: 是否可以进入该目录

权限修改

- chown: 修改文件的拥有者, 前提是要有该拥有者
 - chown 拥有者 文件/目录
 - -R 递归修改
- chgrp: 修改文件所属组, 前提是要有该组
 - -R 递归修改
- chmod: 修改拥有者和所属组的权限
 - 加减法: u/g/o +/=/= r/w/x|
 - 数字法: r=4, w=2, x=1

文件或目录的拥有者即使没有 w 权限, 也可以编辑文件或目录中的文件。所属组和其他人不可以

文件权限-getfacl和setfacl



acl: 针对单一使用者, 设置单一文件或目录来进行rwx的权限修改

getfacl: 获取单一使用者, 针对单一文件或目录的权限

setfacl: 配置单一使用者, 对单一文件或目录的权限

-m 配置acl参数

-x 删除单个文件的acl

-b 删除[acl](#)的配置

-R 递归配置[acl](#)参数

文件权限-getfacl和setfacl

setfacl用法

setfacl -m u:用户名:rwx 文件或者目录

setfacl -m g:组名:rwx 文件或者目录

setfacl -m m::rwx 修改mask值

getfacl 文件名

mask的定义—权限上限

- 1、权限的集合（特点用户，特定组，所属组的并集）
- 2、如果mask中没有的权限，即使getfacl设定了，也不会有

```
[test@localhost tmp]$ ls -l abc  
-rw-rwrxr--+ 1 test test 0 Mar 24 03:38 abc
```

个别用户权限设置好后会显示+号

默认权限

默认权限-umask

- umask: 查看当前用户的umask权限； -S 选项
- 0022---拿走的权限
 - 第一个数字表示特殊权限
 - 022=rwxr-xr-x
- 默认创建文件和目录的权限，文件会拿走X权限
- 临时的修改: umask 0000
- 永久修改: /etc/bashrc (不建议)

特殊权限

文件的特殊权限-SUID SGID SBIT

www.51xxw.net

```
[root@localhost ~]# ls -ld /tmp/ ; ls -l /usr/bin/passwd
drwxrwxrwt. 14 root root 4096 10月 23 19:32 /tmp/
-rwsr-xr-x. 1 root root 27832 6月 10 2014 /usr/bin/passwd
```

当s出现在拥有者的x权限位置时候，表示拥有者有SUID的权限（Set UID）

当s出现在所属组的x权限位置时候，表示所属组有SGID的权限（Set GID）

当t出现在其他人的x权限位置时候，表示其他人有SBIT的权限（Sticky Bit）

SUID，临时获取文件拥有者的权限—只能针对文件

SGID，即使可以作用于目录，也可以作用于文件

 作用于文件：SUID一样

 作用于目录：继承父集目录—目录会不停的继承 I

SBIT指的是，只有文件的拥有者，才能删除，修改该目录下的文件—只能针对目录

S和T都有大写和小写之分

大写说明：没有x权限

小写说明：有x权限

SUID=4 SGID=2 SBIT=1

账户管理

Linux用户账户管理

www.51xxw.net

- 账户和组名：人看的
 - UID和GID：Linux看的
 - id：查看当前登入用户的UID和GID
-
- 当我们使用ls -l的时候，系统会根据/etc/passwd和/etc/group文件的内容，找到UID和GID对应的名称，进行显示
-
- 如果随意修改了/etc/passwd下的用户的UID会发生什么情况
-
- /etc/passwd：记录了Linux上所有的账号
 - /etc/shadow：记录了账户对应的密码
 - /etc/group：记录了所有的组

/etc/passwd

```
lewis:x:1000:1000:lewis:/home/lewis:/bin/bash
```

- Lewis: 用户名
- X: 早期这个部分放的是用户登入密码，现在密码放入了/etc/shadow中了
- UID: 0表示系统管理员，1-999保留给系统使用的ID，1000以上给一般使用者
- GID: 0表示系统管理员，1-999保留给系统使用的ID，1000以上给一般使用者
- Lewis: 使用者信息说明
- /home/lewis: 用户家目录，用登入时，所在的目录
- /bin/bash: 用户在登入的时候，是否可以使用shell，如果不能使用shell，则会显示/sbin/nologin

/etc/shadow

```
root:s6$nrFl5rJDPhQUgDD$uZY7DAbR0Q24FIoExQcwulLTAawBmSPwmglnmVV8k1kJdovtefm  
qTpL3tjmS0KzduUuqcAQ/CdyDCiDddiro.:.:0:99999:7:::
```

- 1、Root: 用户名
- 2、一串红色的字：经过加密的密码
- 3、最近更改过密码的日期：Linux中的日期，是通过1970年1月1号开始累加的日期
- 4、密码不能修改的天数：0表示随时可以修改
- 5、密码需要重新被修改的天数：通过修改该值，可以强制用户修改密码
- 7、密码需要变更的告警天数：7天内系统会向用户发出告警
- 8、密码到期后，账号还可以使用的时间
- 9、账号失效日期：通过1970年1月1号开始累加的日期，到了时间后，无论密码是否过期，该账号就不能使用了
- 10、保留

I

/etc/group

```
lewis:x:1000:
```

- 1、Lewis: 组名
- 2、X: 组密码，一般不需要
- 3、1000: GID
- 4、附属组

这些文件不建议直接进行修改，建议使用命令进行修改。

用户创建修改删除

Linux用户创建-useradd



- useradd

- u: 设置UID
- g: 设置主要组
- G: 设置要附属组
- c: 设置用户说明
- d: 指定用户家目录
- s: 指定用户shell
- e: 账号失效日期, 格式为: YYYY-MM-DD

^I-f: 指定密码是否失效, 0表示立刻失效, 1表示永不失效

Linux用户创建-usermod



- usermod可以对用户账户的信息进行细微的修改

- c: 账号说明
- d: 修改家目录
- g: 修改主要组
- G: 修改附属组
- a: 与-G一起用, 增加附属组
- l: 修改用户名
- u: 修改UID
- L: 冻结账号
- U: 解冻

Linux用户创建-useradd

- 使用了 useradd 后，会默认做以下几个操作

- 1、在 /etc/passwd 下建立相关的资料
- 2、在 /etc/shadow 下写入密码的相关的参数
- 3、在 /etc/group 中写入和账号名一样的组
- 4、在 /home 下创建用户的家目录

- id 命令
- 显示用户的 uid 和 gid

删除一个用户-userdel

- 要彻底的删除一个账号，比较麻烦

- 1、删除 /etc/passwd;/etc/shadow 文件中的内容
- 2、删除 /etc/group;/etc/gshadow
- 3、删除 /home/username;/var/spool/mail/username
- 4、删除该用户曾经来该 Liunx 中创建的文件
 find 进行搜索，先搜索，在使用 userdel

-r 连同家目录和邮箱一起删除

创建用户组：groupadd 用户组名

删除用户组：groupdel 用户组名

查看用户组：groups 用户名

密码创建修改

Linux用户创建-passwd

- **passwd** 用户名 //所有人都可以通过该命令来修改自己的密码
 - l: 锁住该账号，在`/etc/shadow`中放密码的位置加个!
 - u: 解锁
 - S: 显示账号的密码参数
 - n: 接天数，设置多久可以不修改密码
 - x: 接天数，设置多久内必须修改密码
 - w: 接天数，设置密码过期前警告天数
 - i: 接天数，设置密码失效天数

Linux用户创建-chage

- **chage** 可以修改和密码有关的时间参数
 - l: 查看一个账户和密码有关的时间参数
 - d: YYYY-MM-DD, 修改最近一次更改密码的时间
 - E: YYYY-MM-DD, 修改账号的失效时间
 - I: 天数, 设置密码几天后失效
 - m: 天数, 设置密码至少保留几天
 - M: 天数, 设置密码多久后, 需要更新
 - W: 天数, 设置密码过期前警告时间

进程

Linux中的进程

进程：已经启动的可执行程序的运行实体

- 1、PID: 进程的ID (每一个新进程都有一个唯一的PID)
- 2、PPID: 父进程的ID
- 3、任何一个进程都可以创建一个子进程
- 4、在redhat 7以上，所有进程的父进程：`systemd`
在redhat 5,6中，所有进程的父进程：`init`

进程状态和优先级

Linux中的进程

运行中：

R: 该进程正在运行或等待运行

睡眠：

S: 正在休眠但是可以被唤醒

D: 正在休眠，而且不可以被唤醒，该进程被中断，可能会导致设备的异常状态

K: 正在休眠，而且不可以被唤醒，该进程可以被中断。

已停止：

T: 进程被停止，但是可以通过其他进程来进行恢复

T: 正在被调试的进程

僵停：

Z: 子进程在退出时向父进程发出信号，除PID外，所有资源全部释放

X: 父进程获取了子进程的结构，子进程可以完全释放，该状态进程中是看不到的

Linux中的进程

进程优先级

1、< 高优先级

2、n 低优先级

3、s 包含子进程

4、+ 位于后台的进程组

ps -aux参数解释

1、%CPU: 占用的 CPU 使用率

2、%MEM: 占用的记忆体使用率

3、VSZ: 占用的虚拟记忆体大小

4、RSS: 占用的记忆体大小

5、TTY: 终端的次要装置号码 (minor device number of tty)

6、STAT: 该行程的状态

7、START: 行程开始时间

8、TIME: 执行的时间

9、COMMAND: 所执行的指令

查看进程

Linux中的进程

- `ps` 用于显示当前进程状态

常用选项

-aux: 列出所有进程

-ef: 列出所有进程

-l: 列出和当前用户有关的进程

-u 用户: 查看某一用户的进程状态

- `top` 可以查看实时的进程状态

会话和作业

linux中的会话和作业

- **进程:** Linux自身运行的独立的程序

交互进程: 由一个shell启动的进程, 可以再前台运行, 也可以在后台运行

批处理进程: 是一个进程序列, 和终端没有联系

监控进程(守护进程): Linux系统启动时, 启动的进程, 并且在后台运行

- **作业:** 一个正在执行的进程, 而且作业可以包含一个或多个进程。

• **作业控制:** 控制正在运行的进程的行为。如: 挂起一个进程, 等一会再执行。这样用户就可以在多个作业之间切换。

• **&:** 在命令后面使用该符号, 可以让命令在后台执行

• **jobs:** 可以查看正在后台运行的作业

linux中的会话和作业

- `sleep 10000` 延迟几秒
 - `ctrl+c`: 中断
 - `ctrl+z`: 挂起
- 案例: `date ; sleep 5 ; date`
- `jobs`: 查看当前后台的作业状态
 - `-l`: 除了列出作业号外, 同时列出PID
 - `-r`: 列出仅仅在后台运行的作业
 - `-s`: 列出仅仅在后台暂停的作业

linux中的会话和作业

- `fg`: 将后台的命令调到前台来继续执行, 不能放回后台了
 %作业编号
- `bg`: 将后台暂停的命令继续执行
 %作业编号
- `ps -j` 显示当前作业进程信息
 - TGID: 线程组ID号
 - SID: 会话ID号

中断进程

Linux中断进程

- 信号: 传递给Linux进程的操作
- `kill -l` 显示可以传递给Linux进程的所有信号
- 常用:
 - `kill -9 PID` 杀死一个进程—强制
 - `kill -15 PID` 正常的方式终止一个进程
 - `kill -2` 可以`control+c`的操作是一样的

控制服务和守护进程

控制服务和守护进程

systemd: 是所有进程的父进程 (Linux内核3.0以上版本)

systemctl: 用户管理各种类型的systemd对象，这些对象称为：单元

常用的单元: .service (服务单元) .socket (套接字) .path (路径单元)

ssh: 是一个协议 sshd: 是一个进程

systemctl status sshd.service 显示中有几个关键字—红帽7

loaded: 单元配置文件以处理 active (running): 正在运行 active (exited): 配置成功

active (waiting): 运行中，但正在等待事件 inactive: 不在运行 enable: 开机自启动

disabled: 开机不自启 static: 无法启动，但可以通过某一个已经启动的单元来启动

红帽5,6

Service sshd status

Chkconfig ssh on/off 开机自动启动

控制服务和守护进程

- **start:** 在系统中启动一项服务
- **stop:** 等待程序需处理完毕后再**stop**,
- **restart:** 直接关闭程序 在开启
- **reload:** 重新加载配置文件，进程暂停，然后把配置文件加载进去后，继续执行后续操作。进程的PID不会发生改变
- **enable:** 设置开机自启
- **disable:** 关闭开机自启
- **status:** 查看某一单元的状态

日志

分析和存储日志

- 日志：用于系统审核和故障排除---Linux中的“黑匣子”
- 日志文件都是保存在/var/log目录中的
- 在RedHat 7中，系统日志消息由两个服务负责处理。他们是systemd-journald和rsyslogd。

/var/log/messages	//大多数系统日志消息记录的日志
/var/log/secure	//安全和身份验证的消息和错误日志
/var/log/maillog	//与邮件服务器相关的日志
/var/log/cron	//与定期执行任务相关的日志文件
/var/log/boot/log	//记录和系统启动有关的日志

- 许多程序使用syslog协议将事件记录到系统。每一个日志都会根据消息类型和严重性分类

系统日志优先级概述

编码	优先级	严重性
0	emerg	系统不可用。
1	alert	必须立即采取措施。
2	crit	严重状况。
3	err	非严重错误状况。
4	warning	警告状况。
5	notice	正常但重要的事件。
6	info	信息性事件。
7	debug	调试级别消息。

日志分析

- 大多数日志由四个部分组成
- 1、记录在日志的时间
 - 2、发送该日志的主机
 - 3、发送该日志消息的程序或进程
 - 4、发送的实际消息

日志监控

```
tail -f /var/log/secure //显示最后10行，如果有新的内容加入，那么会继续输出
```

打包

可以打包文件也可以打包目录 `tar cf a.tar /etc/`

打包和压缩

打包程序: **tar**

c: 创建文档;
t: 列出存档内容;
x: 提取存档;
f filename: 要操作的存档的文件名;
v: 详细信息

注意:

- 1、选项前不用加-
- 2、创建之前请检查有木有重名文件（**覆盖且不提示**）
- 3、要使tar可以打包选定的文件，执行**tar命令的用户必须要能够读取这些文件**

Linux 中的文件后缀名只起到表示作用

压缩

压缩可选 zcf、jcf、Jcf 解压缩用 xf

打包和压缩

tar支持三种不同压缩方式:

gzip: 压缩速度最快, 历史最久, 应用最广泛;
bzip2: 压缩成的存档文件小, 可用性不如gzip;
xz : 最新的方式, 提供最佳的压缩率。

实际环境中3种情况都可能遇到, 所以要创建不一样格式的归档文件就有自己的选项。

z用于gzip压缩: filename.tar.gz

j用于bzip2压缩: filename.tar.bz2

J用于xz压缩: filename.tar.xz

xz同样不支持压缩目录

xz 1.txt	压缩文件，压缩后源文件消失
du -sh 1.txt.xz	查看压缩文件后大小
xz -d 1.txt.xz	解压缩，解压缩后压缩包消失
unxz 1.txt.xz	解压缩文件同xz -d
xz -c 1.txt > ./2.txt.xz	压缩文件到指定目录，支持重命名压缩后的文件名，压缩后源文件不消失
xzcat 2.txt.xz more	查看压缩包中文件的内容
xz -c -d 2.txt.xz > ./2.txt	解压缩文件到指定目录支持重命名，压缩包不消失
unxz -c 2.txt.xz > ./3.txt	同上

周期性计划作业

周期性计划作业--crontab



crontab:是一个进程。可以让linux周期性的执行某一命令。

I

crontab是一个命令，可以设置linux周期性的执行某一命令。

-u : 设置某一个用户的周期性工作—root权限；

-e : 编辑 crontab 的工作内容

-l : 查阅 crontab 的工作内容

-r : 移除所有的 crontab 的工作内容，若仅要移除一项，请用 -e 去编辑

先要-e 再-u

```
[root@localhost ~]# crontab -ue lewis
crontab: user `e' unknown
[root@localhost ~]# crontab -eu lewis
crontab: installing new crontab
```

crontab的格式讲解

一行代表一个任务

minute hour day month week command

- minute: 表示分钟，可以是从0到59之间的任何整数。
- hour: 表示小时，可以是从0到23之间的任何整数。
- day: 表示日期，可以是从1到31之间的任何整数。
- month: 表示月份，可以是从1到12之间的任何整数。
- week: 表示星期几，可以是从0到7之间的任何整数，这里的0或7代表星期日。
- command: 要执行的命令，可以是系统命令，也可以是自己编写的脚本文件。

crontab的格式讲解

特殊字符	代表含义
*(星号)	代表任何时候都接受的意思,*代表的是任何时候
,	代表分隔字段的意思 例如: 15,30,45 * * * * command 代表的是每月每日每小时的15分,30分,45分的时候执行命令
-	代表一段时间范围内 例如: 10 7-10 * * * command 代表的是每月每日的7点到10点的10分整时执行命令
/n	那个 n 代表数字，也就是每隔 n 单位间隔的意思 例如 /5 * * * * command 代表的是每月每日每时每个5分钟执行一次

crontab—配置举例

实例	
每1分钟执行一次	/1 * * * * command
每小时的第15,30,45分钟执行	15,30,45 * * * * command
7点到10点的第10,50分钟执行	10,50 7-10 * * * command
每隔一天的7点到11点的第20和40分钟执行	20,40 7-11 */1 * * command
每周一的7点到11点的第25和第50分钟执行	25,50 7-11 * * 1 command
每天的3:30执行	30 3 * * * command
每月1、10、20日的3:30执行	30 3 1,10,20 * * command
每周六、周日的3:30执行	30 3 * * 6,7 command
每天9点到16点之间每隔15分钟执行	/15 9-16 * * * command

/etc/crontab配置文件讲解

```
[root@localhost tmp]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# ----- minute (0 - 59)
# |----- hour (0 - 23)
# | |----- day of month (1 - 31)
# | | |----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | |----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# * * * * * user-name command to be executed
```

第一行SHELL变量指定了系统要使用哪个shell，这里是bash。

第二行PATH变量指定了系统执行命令的路径。

第三行MAILTO变量指定了crond的任务执行信息将通过电子邮件发送给root用户，如果MAILTO变量的值为空，则表示不发送任务执行信息给用户。

crontab配置原理

- 当使用者使用 crontab 这个命令来创建工作排程之后，该项工作就会被纪录到 /var/spool/cron 里面去了，而且是以帐号来作为判别的
- 如：lewis 用户使用 crontab 后，他的工作会被纪录到 /var/spool/cron/lewis 中。另外，cron 运行的每一项工作都会被纪录到 /var/log/cron 这个日志文件中。
- crond 服务每分钟检测一次，所以 cron 会每分钟去读取一次 /etc/crontab 与 /var/spool/cron 里面的数据内容，因此，只要你编辑完 /etc/crontab 这个文件，并且将他储存之后，那么 cron 的配置就自动的会来运行了。
- 如果你修改完的 crontab 或者添加的 crontab 没有马上执行起来，那么你可以用 systemctl restart crond.service 重启

输入输出重定向

将输出或报错的内容放到一个文件中查看

输入输出重定向

- > file //标准输出重定向到文件—覆盖
- >>file //标准输出重定向到文件—追加
- 2> //标准错误重定向到文件—覆盖
- 2>> //标准错误重定向到文件—追加
- 2>/dev/null //标准错误重定向到回收站
- &>file //标准输出和标准错误重定向到文件—覆盖
- >>file 2>&1 //标准输出和标准错误重定向到文件—追加

输入输出重定向—举例

- [root@localhost tmp]# date > abc
- [root@localhost tmp]# date >> abc
- [root@localhost tmp]# cat /etc/passwd > abc

- [root@localhost tmp]# 1234 2> abc
- [root@localhost tmp]# 1234 2> /dev/null

- [root@localhost tmp]# 1234 >> abc 2>&1

- 配合计划任务
*/1 * * * * date >> /tmp/abc

管道符

管道符

字符： | 。这就是管道符。

作用有两个：

- 1、承上启下：把上一个指令的输出作为下一个指令的输入来执行。
- 2、搭配grep 字符实现过滤功能。

```
[root@localhost tmp]# ls -l /etc/ | more
[root@localhost tmp]# ps -aux | grep cron
[root@localhost tmp]# ps -aux | grep cron > abc
```

正则表达式和通配符

正则表达式和通配符

在linux中，有通配符和正则表达式，这是两个不同的概念

通配符：它是由shell解析，并且一般用于匹配文件名。如：ls

正则表达式：是一个字符匹配标准，可以匹配文本中的内容

一些命令工具按此标准实现字符匹配，常用于支持正则表达式的工具，如grep, sed等。一般用于匹配文件中的内容

常用的通配符

- *：匹配任意多个字符
- ?: 匹配任意一个字符 I
- [...]: 匹配中括号内出现的任意一个字符
- [...]!: 不匹配中括号内出现的任意一个字符

正则表达式

分组()：

- (ab)*：匹配ab这个分组出现任意次
- \1：引用第一个左括号以及与之对应的右括号所包括的所有内容，同理还有\2,\3

特殊子字符类：

- [:alnum:]：任何字母和数字
- [:alpha:]：任何字母
- [:cntrl:]：控制字符，在ASCII表中对应八进制000 到 037, 和177('DEL').
- [:digit:]：任何数字
- [:graph:]：匹配打印字符，相当于'[:alnum:]' +'[:punct:]'.
- [:lower:]：小写字母
- [:print:]：可打印字符，相当于'[:alnum:]', '[:punct:]', 和space.
- [:punct:] 标点符号，'! "#\$%&()' *+, -./; <=> ? @[\]^_`{|}~'
- [:space:] 空白字符，tab, newline, vertical tab, form feed, carriage return, and space.
- [:upper:] 大写字母
- [:xdigit:] 任何16进制的数字，相当于[0-9a-fA-F]

正则表达式



字符匹配

- .. 匹配任意单个字符
- *：匹配其前面一个字符出现任意次
- ?: 匹配其前面的字符1次或0次
- +：匹配其前面一个字符出现至少一次（在扩展正则表达式中）

位置匹配

- ^：锚定行首
- \$：锚定行尾
- \<或\b：锚定词首，其后面的任意字符必须作为单词首部出现
- \>或\b：锚定词尾，其前面的任意字符必须作为单词尾部出现
- \B：非单词的开头或结尾
- ^\$：空白行
- \：通常用于打开或关闭后续字符的特殊含义

查找和替换

查找和替换

www.51zxw.net

- grep只能用于查找文件中的内容
- sed可以查找，然后替换或者插入想要的内容

a：新增，a的后面可以接字串,而这些字串会在新的一行出现(目前的下一行);

d：删除，因为是删除啊，所以d后面通常不接任何东西的；

i：插入，i的后面可以接字串,而这些字串会在新的一行出现(目前的上一行);

p：列印，亦即将某个选择的资料印出。通常 p 会与参数 sed

s：取代，可以直接进行取代的工作！

以上操作不会对源文件修改，用-i 修改源文件

查找语句find

需求：

找出`/var` 目录下查找大于 5M 的文件，并且他们拷贝到`/tmp/lewisfile` 目录中

用法： `find [路径] [命令参数] [表达式]`

• 参数：

- `-name "文件名"`： 查找指定名称文件；
- `-user 用户名`： 查找指定用户拥有的文件；
- `-group 组名`： 查找指定组拥有的文件；
- `-mtime n`： 查找在N 天前被修改过的文件；
- `-atime n`： 查找在N 天前被访问过的文件；

查找语句find

需求：

找出`/var` 目录下查找大于 5M 的文件，并且他们拷贝到`/tmp/lewisfile` 目录中

用法： `find [路径] [命令参数] [表达式]`

• 参数：

- `-type d/f/b/l/p`： 查找指定类型的文件；
- `-empty`： 查找为空的文件；
- `-size`： 按容量大小查找；
- `-perm mode`： 查找指定属性的文件；
- `-exec command {} \;`： 查找指定的文件并执行指定的命令；
- `-newer 文件名`： 查找比指定文件新的文件