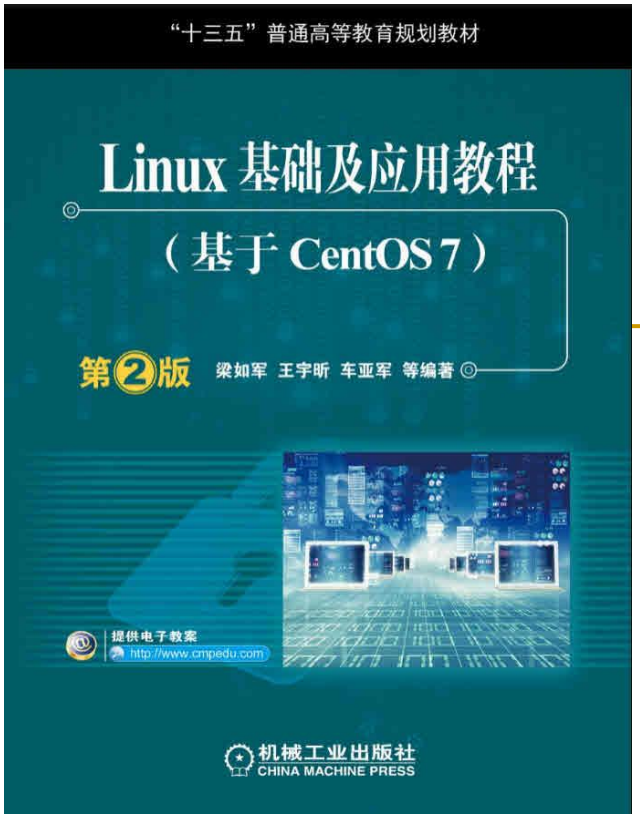
CentOS
The Community ENTerprise Operating System



第3章
多用户与多任务管理

主讲人： 梁如军

2015-05-05

本章内容要点

- 账户实质
- 账户文件
- 账户设置
- 口令管理
- 权限表示
- 权限设置
- 进程概述
- 进程管理
- 作业控制

梁如军（linuxbooks@126.com）

本章学习目标

- 熟悉账户配置文件
- 学会设置和管理口令
- 理解Linux系统的权限
- 掌握设置基本操作权限
- 学会设置特殊权限
- 学会设置ext2/3/4 的文件扩展属性
- 学会设置**FACL权限**
- 理解进程相关概念
- 掌握如何运行后台进程及注销后继续执行
- 掌握进程管理命令的使用
- 掌握**作业控制**的命令及快捷键的使用

梁如军（linuxbooks@126.com）

账户管理的相关概念

梁如军（linuxbooks@126.com）

账户实质

- 账户实质上就是一个用户在系统上的标识
  - 系统依据账户来区分每个用户的文件、进程、任务，给每个用户提供特定的工作环境（如用户的工作目录、shell版本、以及X-Window环境的配置等），使每个用户的工作都能独立不受干扰地进行。
- Linux中的账户包括
  - 用户账户
  - 组账户

用户

- Linux系统下的用户账户（简称用户）有两种
  - 普通用户账户：在系统上的任务是进行普通工作
  - 超级用户账户（或管理员账户）：在系统上的任务是对普通用户和整个系统进行管理。
- 每个用户都被分配了一个**唯一的用户ID号**（UID）
  - 超级用户：UID=0，GID=0
  - 普通用户：UID>=1000
  - 系统用户（伪用户，不可登录）：0<UID<1000

用户（续）

- 用户名和 UID 被保存在 /etc/passwd 这个文件中
- 当用户登录时，他们被分配了一个主目录和一个运行的程序（通常是 shell）
- 若无适当权限，用户无法读取、写入或执行彼此的文件

梁如军（linuxbooks@126.com）

组

CentOS
The Community ENTerprise Operating System

- 组是用户的集合。
- 每个组都被分配了一个唯一的组ID号（GID）
- 组和GID 被保存在 /etc/group 文件中
- 每个用户都有他们自己的私有组
- 每个用户都可以被添加到其他组中来获得额外的存取权限
- 组中的所有用户都可以共享属于该组的文件

梁如军（linuxbooks@126.com）

标准组和私有组

- 标准组
  - 标准组可以容纳多个用户
  - 若使用标准组，在创建一个新的用户时就应该指定他所属于的组
- 私有组
  - 私有组中只有用户自己
  - 当在创建一个新用户时， 若没有指定他所属于的组，RHEL/CentOS就建立一个和该用户同名的私有组，且用户被分配到这个私有组中
  - 优点：防止新文件归 “公共” 组所有
  - 缺点：可能会鼓励创建 “任何人都可以访问” 的文件

梁如军（linuxbooks@126.com）

用户和组的关系

CentOS
The Community ENTerprise Operating System

- 组是用户的集合。
- 一个标准组可以容纳多个用户。
- 同一个用户可以同属于多个组，这些组可以是私有组，也可以是标准组。
- 当一个用户同属于多个组时，将这些组分为：
  - 主组（初始组）：用户登录系统时的组。
  - 附加组：登录后可切换的其他组。

梁如军（linuxbooks@126.com）

Red Hat 的账户管理

- 默认启用shadow passwords功能。
  - /etc/passwd文件对任何用户均可读， 为了增加系统的安全性， 用户的口令通常用shadow passwords保护。
  - 经过shadow passwords保护的账户密码和相关设置信息保存在/etc/shadow文件里。 /etc/shadow只对root用户可读。
  - 默认使用sha512哈希算法存储用户的口令。
- 一般不设置组口令。因为绝大多数应用程序不使用它。
- 建议尽量使用私有组来提高系统安全性。
- 管理工具由 shadow-utils 软件包提供。
- 不建议管理员直接编辑修改系统账户文件来维护账户。

梁如军（linuxbooks@126.com）

账户验证信息文件

- 口令文件 /etc/passwd
  - 文件权限 (-rw-r--r--)
- 影子口令文件 /etc/shadow
  - 文件权限 (-r--------)
- 组账号文件 /etc/group
  - 文件权限 (-rw-r--r--)
- 组口令文件 /etc/gshadow
  - 文件权限 (-r--------)

口令文件 /etc/passwd

- 每一个用户一条记录
- 每条记录由用分号间隔的七个字段组成。

字段	说明
name	用户名
password	在此文件中的口令是X，这表示用户的口令是被/etc/shadow文件保护的
uid	用户的识别号，是一个数字。每个用户的UID都是唯一的
gid	用户的组的识别号，也是一个数字。每个用户账户在建立好后都会有一个主组。主组相同的账户其GID相同。
description	用户的个人资料，包括地址、电话等信息
home	用户的主目录，通常在/home下，目录名和账户名相同
shell	用户登录后启动的shell，默认是/bin/bash

梁如军（linuxbooks@126.com）

影子口令文件 /etc/shadow

- 每一个用户一条记录
- 每条记录由用分号间隔的九个字段组成。

字段	说明
用户名	用户登录名
口令	用户的密码，是加密过的（MD5）
最后一次修改的时间	从1970年1月1日起，到用户最后一次更改密码的天数
最小时间间隔	从1970年1月1日起，到用户应该更改密码的天数
最大时间间隔	从1970年1月1日起，到用户必须更改密码的天数
警告时间	在用户密码过期之前多少天提醒用户更新
不活动时间	在用户密码过期之后到禁用账户的天数
失效时间	从1970年1月1日起，到账户被禁用的天数
标志	保留位

梁如军（linuxbooks@126.com）

组账号文件 /etc/group

- 每一个组一条记录
- 每条记录由用分号间隔的四个字段组成

字段	说明
组名	这是用户登录系统时的默认组名，它在系统中是唯一的
口令	组口令，由于安全性原因，已不使用该字段保存口令，用“x”占位
组ID	是一个整数，系统内部用它来标识组
组内用户列表	属于该组的所有用户名表，列表中多个用户间用“,”分隔

梁如军（linuxbooks@126.com）

组口令文件 /etc/gshadow

- 每一个组一条记录
- 每条记录由用分号间隔的四个字段组成。

字段	说明
组名	组名称，该字段与group文件中的组名称对应
加密的组口令	用于保存已加密的口令
组的管理员账号	管理员有权对该组添加删除账号
组内用户列表	属于该组的用户成员列表，列表中多个用户间用“,”分隔

梁如军（linuxbooks@126.com）

验证账号文件的一致性

- Red Hat 不建议管理员直接编辑修改系统账户文件来维护账户。
- 若用户直接编辑了账户文件， 建议使用账号文件的一致性检测命令。
- pwck
  - 验证用户账号文件，认证信息的完整性。
  - 该命令检测文件“/etc/passwd”和“/etc/shadow” 的每行中字段的格式和值是否正确。
- grpck
  - 验证组账号文件，认证信息的完整性。
  - 该命令检测文件“/etc/group”和“/etc/gshadow”的每行中字段的格式和值是否正确。

梁如军（linuxbooks@126.com）

用户默认环境配置及模板

- 用户默认配置文件
  - /etc/login.defs
  - /etc/default/useradd
- 新用户基本信息
  - /etc/skel
  - 如果手工创建用户，则需复制该目录到用户主目录

梁如军（linuxbooks@126.com）

用户和组管理工具

梁如军（linuxbooks@126.com）

用户和组管理工具

- 用户管理
  - useradd
  - usermod
  - userdel
- 组管理
  - groupadd
  - groupmod
  - groupdel

梁如军（linuxbooks@126.com）

添加用户账号（useradd）

- 格式：

  # useradd [<选项>] <用户名>

- 常用选项

-g group	指定新用户的主（私有）组。
-G group	指定新用户的附加组。
-d directory	指定新用户的自家目录。
-s shell	指定新用户使用的Shell，默认为bash。
-e expire	指定用户的登录失效时间，例如：08/10/2001
-M	不建立新用户的自家目录。

梁如军（linuxbooks@126.com）

useradd命令添加用户的过程

- 编辑账户验证信息文件
  - /etc/passwd, /etc/shadow
  - /etc/group, /etc/gshadow
- 创建主目录 /home/<username>
  - 根据骨架目录（Skeleton Directory） /etc/skel/ 的内容填充用户主目录
- 设置权限和拥有者

梁如军（linuxbooks@126.com）

设置用户口令

- 命令格式
  - **passwd [<用户账号名>]**
- 使用举例
  - 设置用户自己的口令

    $ passwd

    # passwd
  - root 用户设置他人的口令

    # passwd user1

梁如军（linuxbooks@126.com）

添加用户账号举例

- 例一：
  - # useradd -g group1 -e 12/31/2011 user1
  - # passwd user1
- 例二：
  - # useradd -G staff tom
  - # passwd tom
- 例三：
  - # useradd -G ftpgrp -d /var/ftp2 -s /sbin/nologin -M ftp1
  - # passwd ftp1

梁如军（linuxbooks@126.com）

useradd 命令参数的默认值

- 显示 useradd 命令参数的默认值
  - **useradd -D**
  - 从文件 /etc/default/useradd 中读取
- 更改 useradd 命令参数的默认值
  - 格式

```
# useradd -D [-g group] [-b base] [-s shell]  [-e expire ]
```

  - 举例

```
# useradd -D -s /bin/ksh
```


梁如军（linuxbooks@126.com）

修改用户账号（usermod）

- 格式：

  # usermod [<选项>] <用户名>

  - 选项与useradd命令基本相同
- 举例：
  - # usermod -l user2 user1
  - # usermod -G softgroup jjh
  - # usermod -L user1
  - # usermod -U user1

梁如军（linuxbooks@126.com）

删除用户账号（userdel）

- 格式：

  # userdel [<-r>] <用户名>

  - 选项-r用于删除用户的宿主目录

- 举例：
  - # userdel ftp1
  - # userdel –r user1

梁如军（linuxbooks@126.com）

添加组账号（ groupadd ）

- 格式

  # groupadd [<参数>] <*组账号名*>

- 常用参数
  - 参数-r用于创建系统组账号（GID小于500 ）
  - 参数-g用于指定GID
- 举例
  - # groupadd mygroup
  - # groupadd -r sysgroup
  - # groupadd -g 888 group2

梁如军（linuxbooks@126.com）

修改组账号（groupmod）

- 格式

  # groupmod [<参数>] <*组账号名*>

- 常用参数
  - 参数-g改变组账号的GID，组账号名保持不变。
  - 参数-n改变组账号名。
- 举例
  - # groupmod -g 503 mygroup
  - # groupmod –n **newgroup mygroup**

梁如军（linuxbooks@126.com）

删除组账号（ groupdel ）

- 格式

  # groupdel <*组账号名*>

- 举例

  # groupdel mygroup

- 注意
  - 被删除的组账号必须存在
  - 当有用户使用组账号作为私有组时不能删除
  - 与用户名同名的私有组账号在使用userdel命令删除用户时被同时删除，无需使用groupdel命令

梁如军（linuxbooks@126.com）

组成员管理

- 向标准组中添加用户
  - gpasswd -a <用户账号名> <组账号名>

```
# gpasswd -a user1 staff
```

  - usermod -G <组账号名> <用户账号名>

```
# usermod -G  staff  user1
```

- 从标准组中删除用户
  - gpasswd -d <用户账号名> <组账号名>

```
# gpasswd -d user1 staff
```


梁如军（linuxbooks@126.com）

批量用户管理

梁如军（linuxbooks@126.com）

批量用户管理工具

- 成批添加/更新一组账户
  - **newusers**
- 成批更新用户的口令
  - **chpasswd**
- 批量生成安全的口令
  - **pwgen**
  - **secpwgen**（由RPMForge仓库提供）

梁如军（linuxbooks@126.com）

newusers命令

- 格式
  - newusers <filename>
  - filename的格式与 /etc/passwd 一致
- 举例

vi userfile.txt

cat userfile.txt

user1:x:1001:1001::/home/user1:/bin/bash
user2:x:1002:1002::/home/user2:/bin/bash
ftpuser1:x:2001:2001::/home/ftpuser1:/sbin/nologin
ftpuser2:x:2002:2002::/home/ftpuser2:/sbin/nologin

newusers userfile.txt

梁如军（linuxbooks@126.com）

chpasswd命令

- 格式
  - chpasswd < <filename>
  - filename每一行的格式：
    - username:password
    - username必须是系统上已存在的用户
- 举例

```
# vi userpwdfile.txt; chmod 600 userpwdfile.txt
# cat userpwdfile.txt
user1:123456
user2:passwd
ftpuser1:123qaz
ftpuser2:xsw321
# chpasswd < userpwdfile.txt
```


梁如军（linuxbooks@126.com）

pwgen命令

- 格式

  pwgen [选项] [口令长度] [口令个数]
  - 默认的口令长度为 8
  - 默认生成的口令个数为 8*20
- 选项
  - -c：至少包含一个大写字母
  - -n：至少包含一个数字
  - -y：至少包含一个非字母和数字的特殊字符
  - -s：生成完全随机的安全口令
  - -1：每行显示一个口令

梁如军（linuxbooks@126.com）

pwgen命令举例

- # pwgen
- # pwgen 8 20
- # pwgen -1 8 20

- # pwgen -y
- # pwgen -sy
- # pwgen -ns 10
- # pwgen -nsy 12 20
- # pwgen -nsy1 12 20

梁如军（linuxbooks@126.com）

使用pwgen命令生成chpasswd命令所需的口令文件

```
# cat userfile.txt
user1:x:1001:1001::/home/user1:/bin/bash
user2:x:1002:1002::/home/user2:/bin/bash
ftpuser1:x:2001:2001::/home/ftpuser1:/sbin/nologin
ftpuser2:x:2002:2002::/home/ftpuser2:/sbin/nologin
```

- # cut -d\: -f 1 userfile.txt> pwdtemp1
- # pwgen -cn1 8 $(wc -l < userfile.txt) > pwdtemp2
- # paste -d \: pwdtemp1 pwdtemp2 > userpwdfile.txt
- # chmod 600 userpwdfile.txt; rm -f pwdtemp{1,2}
- # chpasswd < userpwdfile.txt

梁如军（linuxbooks@126.com）

口令维护和口令时效

梁如军（linuxbooks@126.com）

口令维护
——禁用、恢复和删除用户口令

- 禁用用户账户口令
  - # passwd -l <用户账号名>
- 查看用户账户口令状态
  - # passwd -S <用户账号名>
- 恢复用户账户口令
  - # passwd -u <用户账号名>
- 清除用户账户口令
  - # passwd -d <用户账号名>

梁如军（linuxbooks@126.com）

口令时效

- 口令时效是系统管理员用来防止机构内不良口令的一种技术。
- 防止口令被攻击的方法就是要**经常地改变口令**。
- 强制用户在一段时间之后更改口令的机制称为口令时效。

- 在默认状况下，口令不会过期
- 强制口令失效是强大安全策略的一部分

梁如军（linuxbooks@126.com）

设置新添用户的口令时效

- 修改 /etc/login.defs 的相关配置参数

PASS_MAX_DAYS	设定在多少天后要求用户修改口令。默认口令时效的天数为99999，即关闭了口令时效。明智的设定一般是60天（2个月）强制更改一次口令。
PASS_MIN_DAYS	设定在本次口令修改后，至少要经过多少天后才允许更改口令。
PASS_MIN_LEN	设定口令的最小字符数。
PASS_WARN_AGE	设定在口令失效前多少天开始通知用户更改口令（一般在用户刚刚登录系统时就会收到警告通知）。

梁如军（linuxbooks@126.com）

设置已存在用户的口令时效

- chage命令
  - chage [选项] [用户登录名]
- 举例
  - 查看用户jason当前的口令时效信息

    # chage -l jason
  - 使用户jason下次登录之后修改口令

    # chage -d 0 jason
  - 用户 jason 两天内不能更改口令，且口令最长的存活期为 30 天，并在口令过期前 5 天通知用户

    # chage -m 2 -M 30 -W 5 jason

梁如军（linuxbooks@126.com）

用户切换命令

CentOS
The Community ENTerprise Operating System

- su
  - 直接切换为超级用户
  - 普通用户要切换为超级用户必须知道超级用户的口令
  - 适用于系统中只有单个系统管理员的情况
- sudo
  - 直接使用 sudo 命令前缀执行系统管理命令
  - 执行系统管理命令时无需知道超级用户的口令，使用普通用户自己的口令即可
  - 由于执行系统管理命令时无需知晓超级用户口令，所以适用于系统中有多个系统管理员的情况，因为这样不会泄露超级用户口令。当然系统只有单个系统管理员时也可以使用。

梁如军（linuxbooks@126.com）

账户相关命令

id	显示用户当前的uid、gid和用户所属的组列表
groups	显示指定用户所属的组列表
whoami	显示当前用户的名称
w/who	显示登录用户及相关信息
newgrp	用于转换用户的当前组到指定的组账号，用户必须属于该组才可以正确执行该命令

梁如军（linuxbooks@126.com）

文件和目录的基本权限

梁如军（linuxbooks@126.com）

权限概述

- Linux是**多用户的操作系统**，允许多个用户同时在系统上登录和工作。
- 为了确保系统和用户的安全采取了如下安全措施
  - 通过UID/GID确定每个用户在登录系统后都做了些什么
  - 通过UID/GID来区别不同用户所建立的文件或目录
    - 每个文件或目录都属于一个UID和一个GID
  - 每个进程都使用一个UID和一个或多个GID来运行
    - 通常由被运行进程的用户决定
  - **超级用户具有一切权限，无需特殊说明**
  - 普通用户只能不受限制的操作主目录及其子目录下的所有文件，对系统中其他目录/文件的访问受到限制

梁如军（linuxbooks@126.com）

三种基本权限

权限	描述字符	对文件的含义	对目录的含义
读权限	r	可以读取文件的内容	可以列出目录中的文件列表
写权限	w	可以修改或删除文件	可以在该目录中创建或删除文件或子目录
执行权限	x	可以执行该文件	可以使用cd命令进入该目录

梁如军（linuxbooks@126.com）

三种基本权限（续）

- 目录上只有执行权限，表示可以进入或穿越他进入更深层次的子目录
- 目录上只有执行权限，要访问该目录下的有读权限的文件，必须知道文件名才可以访问
- 目录上只有执行权限，不能列出目录列表也不能删除该目录
- 目录上执行权限和读权限的组合，表示可以进入目录并列出目录列表
- 目录上执行权限和写权限的组合，表示可以在目录中创建、删除和重命名文件

分配三种基本权限

- 文件和目录的使用者
  - 属主、同组人、其他人
- 权限分配
  - 属主的权限：用于限制文件或目录的创建者
  - 属组的权限：用于限制文件或目录所属组的成员
  - 其他用户的权限：用于限制既不是属主又不是所属组的能访问该文件或目录的其他人员
- 权限的优先顺序
  - 如果UID匹配，就应用用户属主（user）权限
  - 否则，如果GID匹配，就应用组（group）权限
  - 如果都不匹配，就应用其它用户（other）权限

查看文件/目录的权限

- 通过给三类用户分配三种基本权限，就产生了文件或目录的9个基本权限位

```
[osmond@soho ~]$ ls  -l
总计 12
-rw-rw-r--   1  osmond     family        0 06-16 20:43   abc
drwxr-xr-x   2  osmond     family     4096  06-16 20:43   docs
-rw-rw-r--   1  osmond     osmond     1155  06-16 20:44   mylist.txt
drwxr-xr-x   3  osmond     osmond     4096  05-16 13:32   nobp
```

文件类型

文件权限

硬链接数或目录包含的文件数

文件所有者

文件所有者所在的用户组

文件长度

文件上次修改的时间和日期

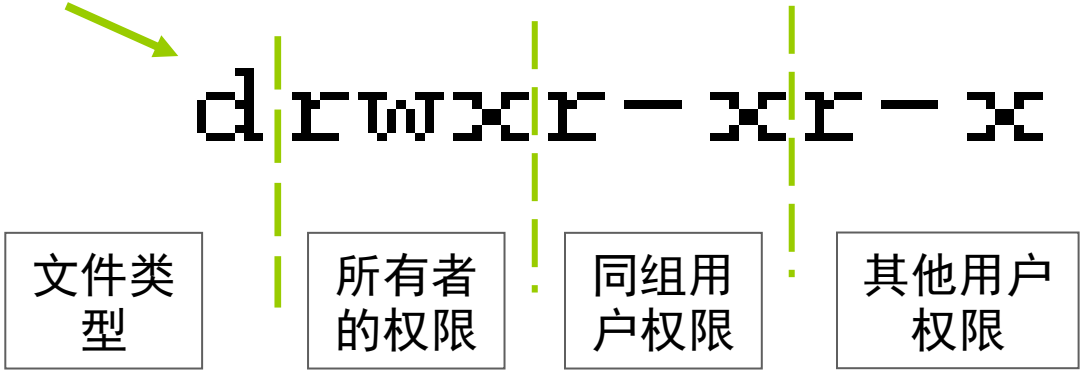
文件名

- 表示无权限

梁如军（linuxbooks@126.com）

文件/目录的权限

```
[osmond@soho ~]$ ls  -l  docs
drwxr-xr-x   2  osmond     family    4096  06-16 20:43   docs
```



- 在显示的结果中，第一个字段的第 2～10 个字符是用来表示权限。
- 这 9 个字符每 3 个一组，组成 3 套 权限控制
  - 第一套控制文件所有者的访问权限
  - 第二套控制所有者所在用户组的其他成员的访问权限
  - 第三套控制系统其他用户的访问权限

梁如军（linuxbooks@126.com）

常见的权限字符串及其含义

字符串	八进制数值	说明
-rw-------	600	只有属主才有读取和写入的权限。
-rw-r--r--	644	只有属主才有读取和写入的权限；同组人和其他人只有读取的权限。
-rwx------	700	只有属主才有读取、写入、和执行的权限。
-rwxr-xr-x	755	属主有读取、写入、和执行的权限；同组人和其他人只有读取和执行的权限
-rw		
-rw		
-rwxrwxrwx	777	每个人都能够读取、写入、和执行。
drwx------	700	只有属主能在目录中读取、写入。
drwxr-xr-x	755	每个人都能够读取目录，但是其中的内容却只能被属主改变。

每个用户都拥有自己的专属目录（主目录），通常放置在 /home 目录下，这些专属目录的默认权限通常为

drwx------

梁如军（linuxbooks@126.com）

与权限相关的命令

- chmod
  - 改变文件或目录的权限
- chown
  - 改变文件或目录的属主（所有者）
- chgrp
  - 改变文件或目录所属的组
- umask
  - 设置文件的**缺省生成掩码**

梁如军（linuxbooks@126.com）

修改文件/目录的权限

- 更改已有文件或目录的访问权限
  - 使用chmod命令
- chmod命令有两种设置方法
  - 文字设定法
    - 使用字母和操作符表达式来修改或设定文件的访问权限
    - chmod [-R] <文字模式> <文件或目录名>
  - 数值设定法
    - 使用八进制数字来设定文件的访问权限
    - chmod [-R] <八进制模式> <文件或目录名>

-R 选项表示对目录中的所有文件或子目录进行递归操作

梁如军（linuxbooks@126.com）

chmod 的文字设定法

操作对象		操作方法		访问权限	
u	属主（user）	+	添加某权限	r	读
g	同组（group）	-	删除某权限	w	写
o	其他（others）	=	直接赋予某权限并取消其他所有权限	x	执行
a	所有（all）			-	无权限

在一个命令行中可给出多个权限模式，其间用逗号间隔

梁如军（linuxbooks@126.com）

chmod 的文字设定法举例

- chmod u+rw myfile
- chmod a+rx,u+w myfile
- chmod u+rwx,g+rx,o+rx myfile
- chmod a+rwx ,g-w,o-w myfile
- chmod a=rwx myfile
- chmod go=rx myfile
- chmod u-wx,go-x myfile
- chmod a+x myfile

梁如军（linuxbooks@126.com）

chmod 的文字设定法举例续

- chmod +r myfile ←→ chmod a+r myfile
- chmod +x myfile ←→ chmod a+x myfile
- chmod +w myfile ←→ chmod u+w myfile
- chmod =r myfile ←→ chmod a=r myfile
- chmod =w myfile ←→ chmod u=w myfile
- chmod o= myfile ←→ chmod o=- myfile

- chmod u=rw,g=,o= projects
- chmod -R u=rwx,go= projects

梁如军（linuxbooks@126.com）

chmod 的数字设定法

CentOS
The Community ENTerprise Operating System

```
chmod n1n2n3 文件或目录名
```

- 使用三个数字模式来表示，分别代表用户（**n1**）、同组用户（**n2**）和其它用户（**n3** ）的访问权限。
- 每个数字模式（**n1|n2|n3**）由不同权限所对应的数字相加得到一个表示访问权限的八进制数字。

权限	对应数字
r	4
w	2
x	1
-	0

```
-rw-r--r--  <--->  644
drwx--x--x  <--->  711
drwx------  <--->  700
-rwxr-xr-x  <--->  755
```


梁如军（linuxbooks@126.com）

chmod 的数字设定法举例


```
chmod 644 myname.txt
```

设定文件 myname.txt 的权限属性为：-rw-r--r--

```
chmod 750 myname.txt
```

设定文件 myname.txt 的权限属性为：-rwxr-x---

```
chmod 700 mydata/
```

设定目录 mydata 的权限属性为：drwx------

梁如军（linuxbooks@126.com）

改变文件/目录属主或组

- 只有**root用户才能改变文件的所有者**
- 只有root用户或所有者才能改变文件所属的组
- 用 chown 命令改变属主 和/或 组

  chown [-R] <用户名[<.|:>组名]> <文件丨目录>
- chgrp 被用来改变所属组

  chgrp [-R] <组名> <文件丨目录>

梁如军（linuxbooks@126.com）

改变属主或组举例

- chown soft myfile
- chgrp softgrp myfile
- chown .softgrp myfile
- chown -R soft mydir
- chgrp softgrp mydir
- chown -R :softgrp mydir
- chown -R soft.softgrp mydir

梁如军（linuxbooks@126.com）

设置生成文件/目录时的默认权限

- 创建新文件或新目录时，系统都会为它们指定默认的访问权限，这个缺省的访问权限就由 umask 值来决定。
- 用户可以使用 umask 命令设置文件的默认生成掩码。默认生成掩码告诉系统当创建一个文件或目录时不应该赋予其哪些权限。
- 系统不允许用户在创建一个普通文件时就赋予它可执行权限，必须在创建后用 chmod 修改。目录则允许设定可执行权限。

梁如军（linuxbooks@126.com）

umask命令

- 查看当前 umask 值
  - 格式：umask [-S]
- 修改当前 umask 值
  - 格式：umask u1u2u3
    - u1表示的是不允许属主有的权限
    - u2表示的是不允许同组人有的权限
    - u3表示的是不允许其他人有的权限
- RHEL/CentOS默认的 umask 值
  - 普通用户的 umask 是 002
  - root用户的 umask 是 022

梁如军（linuxbooks@126.com）

umask 值与文件/目录权限

- 常见 umask 值与新建文件/目录的权限对应表

umask值	新建目录的访问权限	新建文件的访问权限
022	777-022= 755	666-022= 644
027	777-027= 750	666-027= 640
002	777-002= 775	666-002= 664
006	777-006= 771	666-006= 660
007	777-007= 770	666-007= 660

梁如军（linuxbooks@126.com）

设置umask值的方法

- 使用umask命令临时设置
- 在Shell环境配置文件中设置
  - RHEL/CentOS 默认在 /etc/bashrc 中设置
  - 用户可以在~/. bashrc 中重新设置
- 在 /etc/fstab 的文件系统挂装参数中指定
  - 例如：

- 在 /etc/fstab 的挂装选项中加入 umask 值

```
/dev/sda10  /home  ext3  noauto,umask=022,iocharset=cp936,ro,users  0 0
```


梁如军（linuxbooks@126.com）

文件权限的设置准则

CentOS
The Community ENTerprise Operating System

- 尽量使用私有组，保护用户各自的文件或目录。
- 把权限设置为 777或666 的世界可读写的权限是不明智的，应该尽量避免使用。
- 应随时了解指定给文件和目录的权限，定期检查文件和目录以确保指定了正确的权限。
  - 如果在目录下发现陌生的文件请向系统管理员或安全人员报告。
- 为文件和目录指定权限时请慎重考虑只有在具有充分的理由时再将访问权限授予他人。
  - 例如处理小组项目时组员可能需要访问特定的文件或目录需要让他人访问

梁如军（linuxbooks@126.com）

三种特殊权限

梁如军（linuxbooks@126.com）

特殊权限

- 可执行文件的特殊权限
  - **suid**：使用命令的所属用户的权限来运行，而不是命令执行者的权限
  - **sgid**：使用命令的组权限来运行
- 目录的特殊权限
  - **sgid**：在设置了 sgid 权限的目录中创建的文件会具备该目录的组权限
  - **sticky-bit**：在带有粘滞位的目录中的文件只能被文件的**所属用户和root用户删除**，不管该目录的写入权限是如何设置的

梁如军（linuxbooks@126.com）

特殊权限的文字表示方法

- **SUID和SGID用s表示**；Sticky-bit用t表示
  - SUID是占用属主的x位置来表示
  - SGID是占用组的x位置来表示
  - sticky-bit是占用其他人的x位置来表示
- 例如

```
-rwsr-xr-x 1 root root  23420  2010-08-11  /usr/bin/passwd
-rwxr-sr-x 1 root tty   11084  03-10 21:28 /usr/bin/write
-rwsr-sr-x 1 root root  315416 2010-01-06  /usr/bin/crontab
drwxrws--- 3 root admin 4096   06-18 01:01 /admin/sales
drwxrwxrwt 5 root root  4096   06-18 01:01 /tmp
```


梁如军（linuxbooks@126.com）

特殊权限的数值表示方法

CentOS
The Community ENTerprise Operating System

chmod n0n1n2n3 文件或目录名

- 使用一个单独的数字模式（n0）由不同权限所对应的数字相加得到一个表示特殊权限的八进制数。

权限	对应数字
SUID	4
SGID	2
Sticky-bit	1
-	0

-rwsr-xr-x ⟷ 4755

-rwxr-sr-x ⟷ 2755

-rwsr-sr-x ⟷ 6755

drwxrws--- ⟷ 2770

drwxrwxrwt ⟷ 1777

-rwxr-xr-x ⟷ 0755

梁如军（linuxbooks@126.com）

使用chmod设置特殊权限

- 为程序 ~/bin/myapp 添加 SUID权限

  # chmod u+s ~/bin/myapp

  # chmod 4755 ~/bin/myapp

- 为目录 /home/groupspace 添加 **SGID权限**

  # chmod g+s /home/groupspace

  # chmod 2755 /home/groupspace

- 为目录 /home/share 添加 sticky-bit 权限

  # chmod o+t /home/share

  # chmod 1755 /home/share

梁如军（linuxbooks@126.com）

CentOS
The Community ENTerprise Operating System

EXT2/3/4 的文件扩展属性

梁如军（linuxbooks@126.com）

ext2/3/4 的文件扩展属性

- Linux内核中有大量安全特征。EXT2/3/4 文件系统的扩展属性（Extended Attributes）可以在某种程度上保护系统的安全。
- 常见的扩展属性

属性	说明
A（Atime）	告诉系统不要修改对这个文件的最后访问时间。
S	
a (	个目录下建立和修改文件，而不允许删除任何文件。
i（Immutable）	系统不允许对这个文件进行任何的修改。如果目录具有这个属性，那么任何的进程只能修改目录之下的文件，不允许建立和删除文件。

使用A属性可以提高一定的性能。

使用S属性能够最大限度的保障文件的完整性。

a属性和i属性对于提高文件系统的安全性和保障文件系统的完整性有很大的好处。

显示ext2/3/4 的文件扩展属性

- lsattr命令格式

  lsattr [-adR] [文件|目录]

  - -a: 全部文件，包含隐含文件
  - -d: 目录本身，而非目录中的内容
  - -R: 递归方式，包含子目录
- 举例
  - lsattr myfile
  - lsattr -R /etc
  - lsattr -d /etc

梁如军（linuxbooks@126.com）

修改ext2/3/4 的文件扩展属性

- chattr命令格式

  chattr [-R] [[-+=][属性]] <文件|目录>

  - 设置方式：添加（+）、删除（-）、直接设置（=）
  - 常用属性：A（Atime）、S（Sync）、a（Append Only）、i（Immutable）
- 举例
  - chattr +a my.log
  - chattr +i /etc/passwd
  - chattr -i /etc/passwd

梁如军（linuxbooks@126.com）

设置文件扩展属性注意事项

- 虽然属性能够提高系统的安全性，但是它并不适合所有的目录。
- 为了保证系统的正常运行，注意如下目录的扩展属性
  - / 文件系统不能设置immutable属性
  - /dev 目录不能设置immutable和append-only属性
  - /tmp 目录不能设置immutable和append-only属性
  - /var 目录不能设置immutable属性

梁如军（linuxbooks@126.com）

CentOS
The Community ENTerprise Operating System

POSIX文件访问控制列表

梁如军（linuxbooks@126.com）

文件访问控制列表简介

- IEEE POSIX 1003.1e 制定了ACL标准。
- FACL是访问控制列表（File Access Control Lists）的缩写，简称 ACL。
- ACL给予用户和管理员更灵活的控制文件读写和权限赋予的能力。
  - ACL 是标准UNIX文件属性（r，w）的附加扩展。
  - ACL 可以针对任意指定的用户/组分配RWX权限。
  - ACL 允许用户共享文件避免使用冒险的 777 权限。
- 支持FACL的操作系统
  - 主流的商业 UNIX 系统
  - FreeBSD、Linux 系统

梁如军（linuxbooks@126.com）

Linux的FACL支持

- ACL需要内核和文件系统的同时支持
  - Linux从2.6内核开始提供了对EXT2/EXT3, JFS, XFS, ReiserFS等文件系统的ACL支持。
- ACL的文件系统支持
  - 通过文件系统的挂装选项实现 ACL 支持

```
# mount -t ext4 -o acl <device name> <partition>
```

或修改 /etc/fstab 的挂装选项

  - 查看ext4文件系统的默认选项

```
# tune2fs -l /dev/sda1 | grep options
Default mount options:    user_xattr acl
```

(在RHEL/CentOS中，ACL是默认挂装选项)

梁如军（linuxbooks@126.com）

ACL类型

- 存取 ACL（access ACLs）
  - 是对指定文件或目录的存取控制列表。
- 默认 ACL（default ACLs）
  - 只能和目录相关。
  - 若目录中的文件没有存取 ACL，就会使用该目录的默认 ACL。但是访问ACL的优先级更高。
  - 默认 ACL 是可选的。

梁如军（linuxbooks@126.com）

ACL工具

- 在RHEL/CentOS中由 acl 软件包提供
  - getfacl – 获取文件的FACL信息

    getfacl <file|directory>
  - setfacl – 设置文件的FACL信息
- 自动获得ACL权限
  - 若目录已设置了默认ACL，则新创建的文件将从其目录继承默认ACL设置。
  - 使用 mv 命令和 cp -p 命令操作文件时将保持 ACL 设置。

梁如军（linuxbooks@126.com）

setfacl命令

- 语法
  - **setfacl [-R] {-m|-x} <rules> <files or directory>**
- 说明
  - -R选项用于对目录进行递归操作
  - -m选项表示修改ACL权限
  - -x选项表示删除ACL权限
  - rules为要设置的ACL规则

梁如军（linuxbooks@126.com）

setfacl命令中的ACL规则

[d:]u:uid:perms — 为指定的用户（使用 UID 或用户名）设置ACL权限
[d:]g:gid:perms — 为指定的组（使用 GID 或组名）设置ACL权限
[d:]o:[:]perms — 为其他用户设置ACL权限
[d:]m:[:]perms — 设置有效的访问掩码

- 当使用d:前缀时用于设置默认ACL，当使用d:前缀时只能对目录设置
- perms为r、w、x、-或其组合

梁如军（linuxbooks@126.com）

setfacl命令举例

- setfacl -m u:gandolf:rwx file|directory
- setfacl -m g:nazgul:rw file|directory
- setfacl -m d:u:frodo:rw directory
- setfacl -x u:samwise file|directory
- setfacl -R -m g:doc:rwx,d:g:doc:rwx,g:everyone:--- /data/share/

进程相关概念

梁如军（linuxbooks@126.com）

程序、进程和作业

- 程序（ program）
  - 程序是机器指令的集合， 文件形式存储
- 进程（ process ）
  - 进程是一个程序在其自身的虚拟地址空间中的一次执行活动
- 作业/任务（ job/task ）
  - 用户通过操作系统用户接口（Shell或X环境）提交给计算机进行加工处理的程序。如用户发出一个打印命令，就产生一个打印作业/任务。

进程和程序概念的比较

- 程序只是一个静态的指令集合；而进程是一个程序的动态执行过程，它具有生命期，是动态的产生和消亡的。
- 进程是资源申请、调度和独立运行的单位，因此，它使用系统中的运行资源；而程序不能申请系统资源、不能被系统调度、也不能作为独立运行的单位，因此，它不占用系统的运行资源。
- 程序和进程无一一对应的关系。一方面一个程序可以由多个进程所共用，即一个程序在运行过程中可以产生多个进程；另一方面，一个进程在生命期内可以顺序的执行若干个程序。

梁如军（linuxbooks@126.com）

进程

CentOS
The Community ENTerprise Operating System

- 进程是一个动态实体。
- 进程是处理器通过操作系统调度的基本单位。
- 每个进程的执行都独立于系统中的其它进程。
- 进程之间可以通过称为进程间通信（IPC）的机制进行交互。
- 当进程之间共享数据时，操作系统使用了同步技术来保证共享的合法性。

梁如军（linuxbooks@126.com）

Linux是多用户多任务系统

- 当多个用户同时在一个系统上工作时，Linux 要能够同时满足用户们的要求，而且还要使用户感觉不到系统在同时为多个用户服务，就好像每一个用户都单独拥有整个系统一样。
- 每个用户均可同时运行多个程序。为了区分每一个运行的程序，Linux 给每个进程都做了标识，称为进程号（process ID），每个进程的进程号是唯一的。
- Linux 给每个进程都打上了运行者的标志，用户可以控制自己的进程：给自己的进程分配不同的优先级，也可以随时终止自己的进程。
- 进程从执行它的用户处继承UID、GID，从而决定对文件系统的存取和访问。

Linux的多任务实现——分时技术

- Linux 不可能在一个 CPU 上同时处理多个任务（作业）请求，而是采用 “分时” 技术来处理这些任务请求。
- 分时技术
  - 所有的任务请求被排除一个队列，系统按顺序每次从这个队列中抽取一个任务来执行，这个任务执行很短的时间（几毫秒）后，系统就将它排到任务队列的末尾，然后读入队列中的下一个任务，以同样的方式执行。这样经过一段时间后，任务队列中的所有任务都被执行一次，然后又开始下一轮循环。

梁如军（linuxbooks@126.com）

Linux中的进程识别

- 使用PID区分不同的进程
  - 系统启动后的第一个进程是init，它的PID是1。
  - init是唯一一个由系统内核直接运行的进程。
  - 除了init之外，每个进程都有父进程（PPID标识）
- 每个进程还有四个与用户和组相关的识别号
  - 实际用户识别号　　（real user ID，RUID）
  - 实际组识别号　　（real group ID，RGID）
  - 有效用户识别号　　（effect user ID，EUID）
  - 有效组识别号　　（effect group ID，EGID）

梁如军（linuxbooks@126.com）

用户、进程与权限

- RUID和RGID的作用
  - 识别正在运行此进程的用户和组。
    - 一个进程的RUID和RGID就是运行此进程的UID和GID。
- EUID和EGID的作用
  - 确定一个进程对其访问的文件的权限。
    - 除了产生进程的程序被设置SUID和SGID权限位之外，一般EUID、EGID和RUID、RGID相同。
    - 若程序被设置了SUID或SGID权限位，则此进程相应的EUID和EGID，将和运行此进程的文件的所属用户的UID或所属组的GID相同。

梁如军（linuxbooks@126.com）

进程类型

- 交互进程
  - 由一个Shell启动的进程。
  - 交互进程既可以在前台运行，也可以在后台运行。
- 批处理进程
  - 不与特定的终端相关联，提交到等待队列种顺序执行的进程。
- 守护进程（Daemon）
  - 在Linux在启动时初始化，需要时运行于后台的进程。

进程的启动方式

- 手工方式：使用操作系统提供的用户接口
  - 前台
  - 后台 （&）
- 调度方式：按照**预先指定的时间**执行
  - at
  - batch
  - **cron**

前台与后台

- 前台进程
  - 指一个程序控制着标准输入/输出，在程序运行时，shell 被暂时挂起，直到该程序运行结束后，才退回到 shell。在这个过程中，用户不能再执行其它程序。
- 后台进程
  - 用户不必等待程序运行结束就可以执行其它程序。
  - 运行后台进程的方法是在命令行最后加上 “&”
    - 例如：$ sleep 10000 &

在一个终端里只能同时存在一个前台任务，但可以有多个后台任务。

梁如军（linuxbooks@126.com）

进程管理

梁如军（linuxbooks@126.com）

查看系统中的进程

- 使用ps命令查看进程状态信息
  - 显示哪些进程正在执行和执行的状态
  - 进程是否结束、进程有没有僵死
  - 哪些进程占用了过多的系统资源等
- ps（ Process Status ）命令格式
  - ps [选项]
  - 不带任何选项的ps命令
    - 显示当前用户所在终端中的所有进程
    - 输出项包括：识别号(PID)、终端(TTY)、运行时间(TIME)、产生该进程所运行的命令(CMD)

梁如军（linuxbooks@126.com）

ps命令的常用选项

选项	说明
-e	显示所有进程，等价于 -A。
-f	完全（FULL）显示。增加显示用户名、PPID、进程起始时间。
f/-H	显示进程树，等价于 --forest。
a	显示终端上的所有进程，包括其他用户地进程。
x	显示没有控制终端地进程。
u	面向用户的显示格式。增加显示用户名，进程起始时间，CPU和内存占用百分比等信息。
-u <username>	仅显示指定用户的进程。
l/-l	长格式显示。增加显示进程的UID、PPID和优先权值。
w[w]/-w[w]	加宽显示。通常用于显示完整的命令行。
o/-o <format>	由用户自定义输出列。
--sort <order>	指定按哪/哪些列排序，order格式为：[+|-]key[,[+|-]key[,...]]

ps命令使用举例

CentOS
The Community ENTerprise Operating System

$ ps -e

$ ps -ef

$ ps -eH

$ ps -elw

$ ps au

$ ps aux

$ ps axf

$ ps auxw

梁如军（linuxbooks@126.com）

ps 常见的输出标记

- ps 的输出依赖于用户所给的选项

UID	用户 ID	**START**	进程启动时间
USER	用户名	**TIME**	执行时间
PID	进程 ID	**STAT**	进程状态
PPID	父进程的 ID	**NI**	优先权值 / nice 值
TTY	启动进程的终端	**CMD**	命令名（COMMAND）
RSS	进程所用内存块数	**%CPU**	进程所用CPU时间百分比 **(pcpu)**
VSZ	进程所用虚拟内存块数	**%MEM**	进程所有MEM百分比 **(pmem)**

梁如军（linuxbooks@126.com）

ps命令的进程状态列

CentOS
The Community ENTerprise Operating System

- "STAT" 或 "S" 列的输出

R	正在运行或处在运行队列中
S	休眠
T	停止或被追踪
W	进程在 RAM 中没有驻留页（2.6.xx 的内核无效）
D	不可中断的睡眠，通常指 I/O
Z	僵尸进程（已结束但未被父进程收回）
X	已死进程 （这个状态不会出现）
<	具有最高优先权
N	具有较低的优先权

梁如军（linuxbooks@126.com）

ps命令使用举例（2）

- 指定输出列

  $ ps o user,pid,ppid,pcpu,pmem,nice,cmd

  $ ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm

  $ ps axo stat,euid,ruid,tty,tpgid,sess,pgrp,ppid,pid,pcpu,comm

  $ ps -eo “%p %y %x %n %c“ —— AIX风格

- 对指定列排序

  $ ps -ef --sort user,-time

  $ ps aux --sort -pcpu

  $ ps aux --sort -pmem

  $ ps o user,pid,ppid,pcpu,pmem,nice,cmd --sort nice

梁如军（linuxbooks@126.com）

搜索进程

- 搜索指定的进程

  $ ps aux | grep httpd

  $ ps -fp $(pgrep -d, -x httpd)

- 查找符合条件的进程PID

  $ pgrep httpd

  $ pidof httpd

  $ ps -C httpd -o pid=

  $ pgrep -U apache httpd

  $ pgrep -G student -l

梁如军（linuxbooks@126.com）

注销后继续运行进程

- 通常当用户注销后，所有属于该用户的进程将全部被终止。
- 如果希望程序在退出系统后仍然能够继续运行，可以使用 **nohup** 命令启动该进程。

```
nohup 命令 [选项] [参数] [输出文件] &
```

- 例如：

```
$ nohup ~/bin/mirror_yumrepo_with_rsync.sh --centos \
  --arch i386 --exclude-iso &
```

  - 若程序有结果输出，输出结果将会被保存到当前目录下的一个文件名为 nohup.out 的文件中，
  - 若用户在当前目录没有写的权限，则结果将会被保存到用户主目录下的 nohup.out 文件中。

梁如军（linuxbooks@126.com）

进程调度的优先权

- 进程的优先权决定对CPU的使用
- 进程在运行时可以享有不同等的优先权
- 进程的优先权受进程的nice值的影响
  - 这个值的范围是 -20到19，默认是 0
  - 值越小说明对CPU的使用越优先
- 查看进程优先级（看 NI 列的值）
  - ps -l
  - ps -o comm,nice

梁如军（linuxbooks@126.com）

改变进程调度优先权
——在启动进程时指定

- 在启动进程时就指定优先级： **nice**

```
nice –优先级改变量 命令 [&]
```

是指优先级的增量

- 若为正，表示增加nice值，即降低进程优先权
- 若为负，表示减小nice值，即提高优先权
- 若缺省，则默认为 10，即 nice值 增加 10

梁如军（linuxbooks@126.com）

nice命令举例

例：

```
nice -5  lp  paper.pdf &
```

注：使用 nice 同样可以改变前台任务的优先级。

例：超级用户（root）忙着打印一份演讲稿：

```
nice --10 lp report.pdf
```

注：只有 root 才有权限提高一个进程的优先权。

梁如军（linuxbooks@126.com）

改变进程调度优先权
——在进程运行过程中调整

CentOS The Community ENTerprise Operating System

- 进程运行后调整优先级： **renice**
  - 在系统资源紧张时，可以通过降低其它不着急的进程的优先权，从而使得急用的进程能分得更多的 CPU 时间。
  - root 可以提高进程的优先权，但普通用户没这个权限。

调整指定进程的优先级

```
renice 优先级 [-p pid] [-u user] [-g gid]
```

```
renice 5 -p 2345          # -p 可以省略
```

注：普通用户一旦增加某个进程的优先级（即降低优先值）后，就无法再回复到原来的优先级。

梁如军（linuxbooks@126.com）

进程信号（signal）

- 进程信号是在软件层次上对中断机制的一种模拟，在原理上，一个进程收到一个信号与处理器收到一个中断请求可以说是一样的。
- 进程信号是**最基本的进程间通讯方式**
  - 可以在进程之间直接发送，而不需要用户界面
  - 可以在Shell中通过kill命令发送给进程
- Linux对每种进程信号都规定了默认关联动作。
- 查看可用的进程信号

```
$ kill  -l
$ man 7 signal
```


梁如军（linuxbooks@126.com）

进程信号和信号发送

- 常见的进程信号
  - SIGTERM | TERM （15） —— 正常终止 (默认)
  - SIGKILL | KILL （9） —— 立即终止
  - SIGHUP | HUP （1） —— 重读配置文件
- 给进程发送信号
  - 按PID： kill [信号] PID …
  - 按名称：killall [信号] COMM …
  - 按模式：pkill [-信号] 模式
- 发送信号可以使用名称或数字号码

梁如军（linuxbooks@126.com）

杀死进程

- 为什么要杀死进程
  - 该进程占用了过多的CPU时间
  - 该进程锁住了一个终端，使其他前台进程无法运行
  - 运行时间过长，但没有预期效果或无法正常退出
  - 产生了过多到屏幕或磁盘文件的输出
- kill/killall 命令举例

  $ kill 1234　　OR　　$ kill -9 1234

  $ killall myprog　　OR　　$ killall -9 myprog

注：(1) 使用 **kill** 前需要先用 **ps** 查看需要终止的进程的pid；

(2) **kill -9** 很霸道，它在杀死一个进程的同时，将杀死其所有子进程，使用时要谨慎。如错杀 login 进程或 shell 进程等。

梁如军（linuxbooks@126.com）

作业控制

CentOS
The Community ENTerprise Operating System

- 作业控制是指控制当前正在运行的进程的行为，也被称为进程控制。
- 暂时停止某个运行程序
  - 使用Ctrl-z或发送信号 17 （STOP）
- 管理后台作业或暂停的作业
  - 列举作业号码和名称：jobs
  - 在后台恢复运行：bg [%作业号码]
  - 在前台恢复运行：fg [%作业号码]
  - 发送信号：kill -[信号] [%作业号码]

梁如军（linuxbooks@126.com）

本章思考题

CentOS
The Community ENTerprise Operating System

- Linux系统是**如何标识用户和组的**？
- 什么是标准组？什么是私有组？为什么使用了私有组？
- 什么是主组？什么是附加组？以主组登录后如何切换到附加组？
- 简述私有组和主组的关系，简述标准组和附加组的关系。
- 简述Linux的4个账户系统文件及其各个字段的含义。
- 举例说明创建一个用户账号的详细过程。
- 举例说明如何将一个用户账号添加到一个当前还不存在的组中。
- 如何设置用户口令？如何锁定用户账号？如何设置用户口令时效？

梁如军（linuxbooks@126.com）

本章思考题

- Linux文件系统的三种基本权限是什么？
- Linux文件系统的三种特殊权限是什么？何时使用它们？
- 简述chmod命令的两种设置权限的方法。
- 如何更改文件或目录的属主和/或同组人？
- 为什么使用ACL？ACL的**两种类型**及其作用。

本章思考题

- 什么是进程？它与程序有何关系？
- 进程的类型？进程的启动方式？
- 什么是前台进程？什么是后台进程？
- 如何查看进程？
- 如何删除进程？
- 如何更改进程优先级？
- 什么是作业控制？

梁如军（linuxbooks@126.com）

本章实验

- 学会管理用户和组账号。
- 学会设置用户口令并管理用户口令时效。
- 学会设置文件和目录的操作权限。
- 学会设置和使用ACL权限。
- 学会显示和杀死进程。
- 学会显示和管理守护进程。
- 学会实施作业控制。

梁如军（linuxbooks@126.com）

进一步学习

CentOS
The Community ENTerprise Operating System

- 学习SUN的集中式的账户系统NIS服务的配置和使用。
- 学习使用**OpenLDAP**实现的集中式账户管理和应用。

梁如军（linuxbooks@126.com）