第 3 讲　Linux 基础 (2)

王晓庆

wangxiaoqing@outlook.com

March 23, 2016

Outline

帐号

- 帐号 (account) 是用于记录用户或用户组的数据
- Linux 中每一个合法用户都必须拥有帐号才能登录使用系统
- Linux 依靠帐号来验证用户身份并设置用户权限

用户帐号和组帐号

- 用户帐号：每个用户帐号存储某个用户的数据
- 组帐号：每个组帐号存储某个用户组的数据
- CentOS 中最多可以建立 2**30 个帐号和 2**30 个组帐号

根据帐号位置分类

- 本机帐号 (local account)
  - 帐号数据存储于本机硬盘内
  - 帐号有效范围仅限于本机
  - 优点：简单易用，无需额外设置即可直接创建
  - 缺点：不具备可扩展性，设想一个数百台设备的环境，要让某个帐号在每台设备上都能登录，则必须在每台设备上建立该帐号
- 域帐号 (domain account)
  - 将大量计算机组织成一个域，并在其中一台设备上建立帐号数据，并通过某些通信协议实现帐号数据的远程访问
  - 优点：具备可扩展性
  - 缺点：配置域帐号必须要先建立域环境

根据帐号功能分类

- 用户帐号
  - 超级用户：用户名通常为 root，其 UID 一定为 0
    - 权限不受任何限制
    - 不要轻易以 root 用户身份使用 Linux 系统。
  - 普通用户：除超级用户外的其他所有帐号
    - 系统帐号
    - 仅提供给 Linux 系统本身使用，某些软件运行时需要一个普通用户身份
    - 在 CentOS 5 中，系统用户的 UID 为 1~499 之间
    - 真实用户
    - 真实用户就是可以用于登录系统的普通帐号
    - 真实用户的 UID 在 500 ～ 4,294,967,295 之间

根据帐号功能分类

- 组帐号
- 超级用户组
  - root 组，GID 为 0，但组成员不具备系统管理权力，受系统权限限制
- 系统组
  - 类似于系统帐号，仅提供给系统本身或某个软件使用
  - 在 CentOS 5 中，系统组的 GID 范围为 1 ～ 499
- 用户自定义组
  - 除 root 组和系统组之外的组，都可以由管理员自行定义
  - 用户自定义组的 GID 在 500 ～ 4,294,967,295 之间
  - 用户私有组 (User Private Group, UPG)
    1. 是指与用户账户名称相同的组，是用户默认的基本组
    2. 当建立新用户时，CentOS 会自动建立该用户的私有组
    3. 用户还可以加入多个其他的组，也叫用户的附加组

用户帐号

- 用户帐号数据和密码分别存储于/etc/passwd 和/etc/shadow 中

```
less /etc/passwd
less /etc/shadow
man 5 passwd
man 5 shadow
```

说明

1. 如果密码字段为空，则该用户无需密码即可登录系统
2. 如果密码字段为 x，则该用户名密码被存储至他处
3. 如果默认 shell 为/sbin/nologin，则不允许该用户登录

组帐号

- 组帐号数据和密码分别存储于/etc/group 和/etc/gshadow 中

```
less /etc/group
less /etc/gshadow
man 5 group
man 5 gshadow
```

说明

1. CentOS 不支持嵌套组，组成员只能是用户，不能是其他组
2. 组成员之间以逗号 (,) 分隔

添加用户帐号

useradd

```
useradd jack
tail -1 /etc/passwd
tail -1 /etc/group
tail -1 /etc/shadow
ls /home
ls /var/spool/mail
man useradd
```

说明

用 useradd 添加新用户时，如果未指定选项，则 useradd 会根据/etc/login.defs 与/etc/default/useradd 中的配置添加新用户。

设置/修改用户密码

示例

```
passwd jack      # 为指定用户设置密码 (仅 root)
passwd           # 修改当前用户密码
passwd -l jack   # 锁定用户 (仅 root)
passwd -u jack   # 解锁用户 (仅 root)
passwd -d jack   # 清除用户密码 (本地免密码登录)
```

说明

让用户免密码本地登录的两种方法：

1. 将/etc/passwd 文件中的密码字段清空
2. 将/etc/shadow 文件中的密码字段清空

远程用户必须提供密码才允许登录!

修改用户帐号

usermod

```
usermod -u 1001 bob
usermod -g 1001 bob
usermod -G root,users bob # 设置 bob 的附加组
usermod -G '' bob         # 设置 bob 不属于任何附加组
usermod -s /bin/csh bob
usermod -d /home/bobie
usermod -l bobie bob
usermod -L alice          # 锁定帐号
usermod -U alice          # 解锁帐号
```

说明

usermod 与 useradd 的选项类似，还增加了-U、-L、-l 等选项。

查看和删除用户帐号

id

```
id              # 查看当前用户信息
id jack         # 查看指定用户信息
```

userdel

```
userdel jack      # 删除用户
userdel -r jack   # 删除用户及其主目录和邮箱
```

创建、修改组

groupadd 创建组

```
groupadd proj1
groupadd -g 1001 proj2
```

groupmod 修改组

```
groupmod -g 1000 students
groupmod -n projectA proj1
```

查询、删除组

groups 查询组

```
groups
groups mary
```

groupdel 删除组

```
groupdel projectA
```

注意

注意：删除某用户的基本组之前，需要先删除该用户！

组成员管理

- gpasswd
- gpasswd 本用于设置组密码，但该功能极少使用，实际上更多用于组成员管理

示例

```
gpaswd -a mary students    # 将 mary 加入 students 组
gpasswd -d mary students   # 从 students 组删除 mary
gpasswd -A mary students   # 将 mary 设置为 studens 组管理员
```

切换组身份

newgrp

```
newgrp students   # 切换至 students 组身份
exit              # 返回原先组身份
```

说明

如果用户不是root，下列情况下将会被提示输入组密码：
1. 如果用户没有密码而要切换的组有密码
2. 如果用户不属于要切换的组，而该组有密码
如果用户不属于要切换的组，且该组未设密码，则拒绝切换

加密算法

- DES
  传统 UNIX 使用的加密算法，只能支持 8 个字符以内的密码，如果超过 8 个字符，则 DES 会忽略第 8 个字符以后的密码，其安全性比较差。
- MD5
  CentOS 默认使用的加密算法，支持 255 个字符的密码，比 DES 更安全。

查看与设置帐号秘密有效期

chage

```
chage -l mike   # 查看用户帐号密码有效期信息
# 注意：普通用户只能查看自己的帐号密码有效期信息
chage -m 3 mike   # 设置设定密码后至少经过几天才能修改密码
chage -M 30 mike # 设置设定密码后最长多少天后必须修改密码
chage -d 2016-01-01 mike # 设置密码最后修改时间
chage -E 2020-12-31 mike # 设置密码过期时间
chage -W 7 mike   # 设置密码过期前几天提醒
chage -I 3 mike   # 设置密码过期后多少天被锁定
```

文件权限和目录权限

- 三种基本权限

r–	-w-	–x
100	010	001
4	2	1
文件可读	文件可写	文件可执行
目录可读	目录可写	目录可搜索

说明

1. 目录可读：可以通过 ls 查看目录中的文件列表
2. 目录可写：可以在目录内创建、删除、移动文件
3. 目录可访问：可以通过 cd 进入该目录或者搜索其内的文件

三种用户

三种用户的权限

```
ls -l ~/students.db
```

```
-rw-rw-r--. 1 wxq wxq 115 3 月 8 22:35 /home/wxq/students.db
```

- 文件所有者 (默认为创建该文件的用户)
  ls -l 输出结果第 3 列，第一组权限 (rw-)
- 文件所属组
  ls -l 输出结果第 4 列，第二组权限 (rw-)
- 其他用户
  既不是文件所有者，也不属于文件所属组，第三组权限 (r–)

修改文件和目录权限

- chmod 仅文件所有者和 root 用户有权修改文件权限

示例

```
chmod u=rw-,g=r--,o=r-- file
chmod g+w,o-r file
chmod a+x file
chmod 644 file
chmod 600 file
chmod 700 dir
chmod -R 755 dir
```

想一想

1. 若用户对某个文件只有读权限，他有没有可能删除该文件？
2. 可否允许其他用户修改某文件，却不能删除或更名该文件？

查看和设置默认权限掩码

umask

```
umask       # 掩码中某一位为 1 表示该位对应权限被禁止
umask -S
umask 0026
```

说明

新建文件/目录的权限等于 0666/0777 与掩码的反码按位相与

算一算

1. 若将 umask 值设置为 0011，则新建文件的默认权限是什么？
2. 若希望新建文件的权限是所有者可读写，属组用户可读，其他用户无权限，则 umask 值应设为多少？

权限设置实例

问题

xiaobai 想在自己的主目录内建立一个公共目录 pub，希望所有人都可以只读访问该目录内的文件，但不希望其他人能访问其主目录内的其他文件和目录，请问要怎么配置相关文件/目录的权限才能达到目的呢？

chown 和 chgrp

- 仅 root 用户有权更改文件所有者和所属组

示例

```
chown root file          # 令 file 为 root 所有
chown root:users file    # 令 file 为 root 所有, 属 users 组
chown root: file         # 令 file 为 root 所有, 属 root 组
chown :users file        # 令 file 属 users 组
chown -R root:users dir  # 递归处理
chgrp users file         # 令 file 属 users 组
chgrp -R users dir       # 递归处理
```

三种基本权限够用吗？

- mike 用户对文件 F 具有只读权限，则 mike 用编辑器 vim 打开文件 F 时，vim 对文件 F 也仅具备只读权限。

结论

一个程序对文件具备何种权限由启动该程序的用户决定

问题

mike 用户对/etc/shadow 不具备任何权限，那 mike 如何通过执行 passwd 程序修改存放在/etc/shadow 内的登录密码呢？

setUID 权限

- setUID 权限：可执行程序被设置了 setUID 权限后，无论被谁执行，该程序都将具备其所有者的权限，而不是执行者的权限。

示例

```
ls -l /usr/bin/passwd
chmod u+s file    # 设置 setUID 权限
chmod u-s file    # 删除 setUID 权限
chmod 4751 file   # 设置 setUID 权限
chmod 0751 file   # 删除 setUID 权限
```

setGID 权限

- setGID 权限：可执行程序被设置了 setGID 权限后，无论被谁执行，该程序都将具备其所属组的权限，但没有其所有者的权限。

示例

```
ls -l /bin/locate
chmod g+s file
chmod g-s file
chmod 2751 file
```

setGID 权限 (2)

- setGID 权限常针对目录设置：目录设置了 setGID 权限后，无论谁在该目录内创建的新文件，默认都会属于该目录所属的组，而不是创建者所属的组。

示例

```
mkdir proj_dir
groupadd proj
chgrp proj proj_dir
chmod g+s proj_dir
```

Sticky bit 权限

- Sticky bit：目前仅对目录有效，目录设置该权限后，该目录内的所有文件仅文件所有者和 root 用户才有权力进行删除/更名/移动等操作，

示例

```
ls -ld /tmp
chmod o+t dir
chmod o-t dir
chmod 1751 dir
```

查看进程

ps

```
ps                  # 查看当前 shell 执行的进程
ps aux              # 查看所有进程 (BSD 风格)
ps -ef              # 查看所有进程 (System V 风格)
ps -ft /dev/pts/0 # 查看与指定终端关联的进程
ps -fp 1234     # 查看进程号为 1234 的进程的详细信息
ps -fu xiaobai # 查看用户 xiaobai 运行的进程信息
ps -fC evince   # 查看名称为 eninсe 的所有进程信息
```

pstree

```
pstree              # 打印系统进程树
```

查看进程

pgrep

```
pgrep evince   # 查找名为 evince 的进程的进程号
```

top

```
top       # 实时系统状态监控
h/?       # 帮助
d 5       # 设置刷新周期为 5 秒
k 1234    # 杀死进程号为 1234 的进程
P         # 按 cpu 使用率排序
M         # 按 memory 使用率排序
T         # 按 cpu 使用时间排序
u tom     # 仅显示用户 tom 的进程
q         # 退出 top
man top
```

进程控制

- 前后台进程
- 一个终端一次只能运行一个前台进程，但可以并发运行多个后台进程。

前后台作业控制

```
Ctrl-z                  # 挂起（暂停）前台进程的执行
jobs [-l]               # 查看作业
fg [n]                  # 将作业恢复至前台执行
bg [n]                  # 将作业恢复至后台执行
kill %n/pid             # 杀死指定的作业/进程
killall xeyes           # 杀死所有 xeyes 进程
pkill eye               # 查找包含 eye 的进程名并杀死
xeyes -center red &     # 启动后台进程
nohup ./longjob &       # 保持后台进程在用户注销后继续运行
nohup ./longjob & >longjob.out
```

进程和信号

- 信号
  内核用信号通知进程发生的异常事件和实现进程间通信，进程也可以给其他进程。每种信号可以用名称或整数标识。

1. 硬件异常：进程让硬件执行错误操作，如进程进行除 0 运算，内核会给它发送 SIGFPE(8)
2. 软件状态：把异常的软件状态通知进程，如进程终止时，内核会向父进程发送 SIGCHLD（17），当调整 X 图形应用程序的窗口大小时，该程序会收到 SIGWINCH(28)
3. 终端中断：用户敲入的终端控制组合键会把信号发送给 shell 前台进程，如 Ctrl-c 发送 SIGINT(2)，Ctrl-z 发送 SIGSTP(20)
4. 其他进程：进程可以通过 kill 命令给同一用户拥有的其他任何进程发送任何信号

向进程发信号

- 信号：

kill 发送信号给进程

```
kill -l                  # 打印信号值列表
man 7 signal             # 查看信号相关帮助
Ctrl-c                   # 发送信号 2 给前台进程
Ctrl-z                   # 发送信号 20 给前台进程
kill -19 12121           # 暂停
kill -18 12121           # 恢复
kill -15 12345           # 发送信号 15 给进程 12345
kill 12345               # 同上
kill -9 11111            # 强行杀死进程 11111
kill -s SIGTERM 1234     # 发送信号 SIGTERM 给进程 1234
```

进程优先级

- 优先级 (priority) 与谦让值 (nice)

  top 命令的输出中，PR 是内核用于进程调度的动态优先级，而 NI 是用户可以设置的谦让值。NI 可以影响 PR，PR 与 NI 越小则进程优先级越高。nice 值可以从-20 至 19，默认为 0。

nice 以指定 nice 值启动进程

```
xeyes -center red &          # 默认 nice 值为 0
nice xeyes -center blue &    # 默认设置 nice 值为 10
nice -n 5 xeyes -center green &
ps -l    # 查看进程的 PR 与 NI 值
```

注意

只有 root 用户可以在 nice 命令中为进程指定负 nice 值

进程优先级 (2)

renice 调整进程 nice 值

```
renice 15 12707       # 将进程 12707 的 nice 值调整为 15
renice -5 12675       # 将进程 12675 的 nice 值调整为-5
renice 8 -u mike      # 将用户 mike 的进程 nice 值调整为 8
renice 0 -g users     # 将用户组 users 的进程 nice 值调整为 0
```

注意

只有 root 用户可以通过 renice 减小进程的 nice 值

守护进程 (Daemon)

- 守护进程 (daemon)
- 守护进程是那些在后台运行的进程，脱离控制终端，守护进程与系统服务像关联，如日志守护进程 syslogd、安全 shell 守护进程 sshd 等，守护进程名一般以 d 结尾。
- 守护进程通常作为系统启动过程的一部分被启动，或者由 root 用户启动，有些守护进程以 root 用户身份运行，有些守护进程则以特定的系统用户身份运行。

at/batch 一次性计划任务

- atd 守护进程：运行用户提交稍后运行的作业

用 at 命令提交作业

```
at 2:00 am
at> find / -mtime 0 -exec cp -p {} /bak \;
at> Ctrl-d
at 22:30 <task1
at -f task2 now+1h
atq                 # 查看作业队列，同 at -l
atrm 2              # 删除作业 2，同 at -d 2
man at
```

at/batch 一次性计划任务 (2)

- 用 batch 提交作业
- batch 与 at 的语法与 at 相同，但 batch 作业不在特定时间运行，而是等到系统负载较轻时运行。atd 会监控系统平均负载，等待它降低到 0.8 以下，然后开始运行作业任务。

cron 重复性计划任务

- cron 允许用户配置要定期运行的命令，用户用 crontab 命令配置自己的任务计划 (crontab)。
- cron 在/var/spool/cron 目录下搜索以用户名命名的 crontab 文件，以及/etc/crontab 文件和/etc/cron.d 目录下的 crontab 文件，cron 每分钟醒来一次检查是否有需要运行的作业。

crontab 命令

```
crontab -e     # 编辑计划任务
crontab file   # 从 file 创建计划任务
crontab -l     # 列出计划任务
crontab -e     # 删除计划任务
```

cron 重复性计划任务 (2)

计划任务格式

分钟 小时 几号 几月 星期几 [用户] 命令
注：用户计划任务无需用户字段

示例

```
SHELL=/bin/sh      #使用/bin/sh执行命令
MAILTO=mike        #发邮件给mike
 5  0  *  *  *   $HOME/bin/daily.job >>$HOME/tmp/out 2>&1
15 14  1  *  *   $HOME/bin/monthly.job
 0 22  *  * 1-5  $HOME/bin/workday.job
23 0-23/2 * * *  $HOME/bin/twohours.job
 5  4  *  * sun  $HOME/bin/sunday.job
```

查看和修改 shell 环境变量

- 查看 shell 环境变量

```
env
echo $SHELL
```

- 修改 shell 环境变量

```
export HISTSIZE=500
```

说明

上述修改仅对当前会话有效！

bash 配置文件

系统级配置文件

1. /etc/profile 用户登录时运行
2. /etc/bashrc 启动 bash 时运行

用户级配置文件

1. .bash_profile 用户登录时运行
2. .bashrc 启动 bash 时运行
3. .bash_logout 用户注销时运行
4. .bash_history 保存用户命令历史