Linux 用户和用户组管理详解

Linux 系统是一个多用户多任务的分时操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统。

用户的账号一方面可以帮助系统管理员对使用系统的用户进行跟踪，并控制他们对系统资源的访问；另一方面也可以帮助用户组织文件，并为用户提供安全性保护。

每个用户账号都拥有一个惟一的用户名和各自的口令。

用户在登录时键入正确的用户名和口令后，就能够进入系统和自己的主目录。

实现用户账号的管理，要完成的工作主要有如下几个方面：

- 用户账号的添加、删除与修改。
- 用户口令的管理。
- 用户组的管理。

一、Linux 系统用户账号的管理

用户账号的管理工作主要涉及到用户账号的添加、修改和删除。

添加用户账号就是在系统中创建一个新账号，然后为新账号分配用户号、用户组、主目录和登录 Shell 等资源。刚添加的账号是被锁定的，无法使用。

1、添加新的用户账号使用 useradd **命令，其语法如下：**

useradd 选项用户名

参数说明：

选项:

-c comment 指定一段注释性描述。

-d 目录 指定用户主目录，如果此目录不存在，则同时使用-m 选项，可以创建主目录。

-g 用户组 指定用户所属的用户组。

-G 用户组，用户组 指定用户所属的附加组。

-s Shell 文件 指定用户的登录 Shell。

-u 用户号 指定用户的用户号，如果同时有-o 选项，则可以重复使用其他用户的标识号。

用户名:

指定新账号的登录名。

实例 1

useradd –d /usr/sam -m sam

此命令创建了一个用户 sam，其中-d 和-m 选项用来为登录名 sam 产生一个主目录/usr/sam（/usr 为默认的用户主目录所在的父目录）。

实例 2

useradd -s /bin/sh -g group –G adm,root gem

此命令新建了一个用户 gem，该用户的登录 Shell 是 /bin/sh，它属于 group 用户组，同时又属于 adm 和 root 用户组，其中 group 用户组是其主组。

这里可能新建组：#groupadd group 及 groupadd adm

增加用户账号就是在/etc/passwd 文件中为新用户增加一条记录，同时更新其他系统文件如/etc/shadow, /etc/group 等。

Linux 提供了集成的系统管理工具 userconf，它可以用来对用户账号进行统一管理。

2、删除帐号

如果一个用户的账号不再使用，可以从系统中删除。删除用户账号就是要将/etc/passwd 等系统文件中的该用户记录删除，必要时还删除用户的主目录。

删除一个已有的用户账号使用 userdel 命令，其格式如下：

userdel 选项用户名

常用的选项是-r，它的作用是把用户的主目录一起删除。

例如：

userdel sam

此命令删除用户 sam 在系统文件中（主要是/etc/passwd, /etc/shadow, /etc/group 等）的记录，同时删除用户的主目录。

3、修改帐号

修改用户账号就是根据实际情况更改用户的有关属性，如用户号、主目录、用户组、登录 Shell 等。

修改已有用户的信息使用 usermod 命令，其格式如下：

usermod 选项用户名

常用的选项包括-c, -d, -m, -g, -G, -s, -u 以及-o 等，这些选项的意义与useradd 命令中的选项一样，可以为用户指定新的资源值。

另外，有些系统可以使用选项：-l 新用户名

这个选项指定一个新的账号，即将原来的用户名改为新的用户名。

例如：

usermod -s /bin/ksh -d /home/z –g developer sam

此命令将用户 sam 的登录 Shell 修改为 ksh，主目录改为/home/z，用户组改为 developer。

5、用户口令的管理

用户管理的一项重要内容是用户口令的管理。用户账号刚创建时没有口令，但是被系统锁定，无法使用，必须为其指定口令后才可以使用，即使是指定空口令。

指定和修改用户口令的 Shell 命令是 passwd。超级用户可以为自己和其他用户指定口令，普通用户只能用它修改自己的口令。命令的格式为：

passwd 选项用户名

可使用的选项：

-l 锁定口令，即禁用账号。

-u 口令解锁。

-d 使账号无口令。

-f 强迫用户下次登录时修改口令。

如果默认用户名，则修改当前用户的口令。

例如，假设当前用户是 sam，则下面的命令修改该用户自己的口令：

$ passwd Old password:******New password:*******Re-enter new password:*******

如果是超级用户，可以用下列形式指定任何用户的口令：

passwd sam New password:*******Re-enter new password:*******

普通用户修改自己的口令时，passwd 命令会先询问原口令，验证后再要求用户输入两遍新口令，如果两次输入的口令一致，则将这个口令指定给用户；而超级用户为用户指定口令时，就不需要知道原口令。

为了系统安全起见，用户应该选择比较复杂的口令，例如最好使用 8 位长的口令，口令中包含有大写、小写字母和数字，并且应该与姓名、生日等不相同。

为用户指定空口令时，执行下列形式的命令：

passwd -d sam

此命令将用户 sam 的口令删除，这样用户 sam 下一次登录时，系统就不再询问口令。

passwd 命令还可以用-l(lock)选项锁定某一用户，使其不能登录，例如：

passwd -l sam

二、Linux 系统用户组的管理

每个用户都有一个用户组，系统可以对一个用户组中的所有用户进行集中管理。不同 Linux 系统对用户组的规定有所不同，如 Linux 下的用户属于与它同名的用户组，这个用户组在创建用户时同时创建。

用户组的管理涉及用户组的添加、删除和修改。组的增加、删除和修改实际上就是对/etc/group 文件的更新。

1、增加一个新的用户组使用 groupadd 命令。其格式如下：

groupadd 选项用户组

可以使用的选项有：

-g GID 指定新用户组的组标识号（GID）。

-o 一般与-g 选项同时使用，表示新用户组的 GID 可以与系统已有用户组的 GID 相同。

实例 1：

groupadd group1

此命令向系统中增加了一个新组 group1，新组的组标识号是在当前已有的最大组标识号的基础上加 1。

实例 2：

groupadd -g 101 group2

此命令向系统中增加了一个新组 group2 ,同时指定新组的组标识号是 101。

2、如果要删除一个已有的用户组，使用 groupdel 命令，其格式如下：

groupdel 用户组

例如：

groupdel group1

此命令从系统中删除组 group1。

3、修改用户组的属性使用 groupmod 命令。其语法如下：

groupmod 选项用户组

常用的选项有：

- -g GID 为用户组指定新的组标识号。

-o 与-g 选项同时使用，用户组的新 GID 可以与系统已有用户组的 GID 相同。

-n 新用户组 将用户组的名字改为新名字

实例 1：

groupmod -g 102 group2

此命令将组 group2 的组标识号修改为 102。

实例 2：

groupmod –g 10000 -n group3 group2

此命令将组 group2 的标识号改为 10000，组名修改为 group3。

4、如果一个用户同时属于多个用户组，那么用户可以在用户组之间切换，以便具有其他用户组的权限。

用户可以在登录后，使用命令 newgrp 切换到其他用户组，这个命令的参数就是目的用户组。例如：

$ newgrp root

这条命令将当前用户切换到 root 用户组，前提条件是 root 用户组确实是该用户的主组或附加组。类似于用户账号的管理，用户组的管理也可以通过集成的系统管理工具来完成。

三、与用户账号有关的系统文件

完成用户管理的工作有许多种方法，但是每一种方法实际上都是对有关的系统文件进行修改。

与用户和用户组相关的信息都存放在一些系统文件中，这些文件包括 /etc/passwd, /etc/shadow, /etc/group 等。

下面分别介绍这些文件的内容。

1、/etc/passwd 文件是用户管理工作涉及的最重要的一个文件。

Linux 系统中的每个用户都在/etc/passwd 文件中有一个对应的记录行，它记录了这个用户的一些基本属性。

这个文件对所有用户都是可读的。它的内容类似下面的例子：

```
# cat /etc/passwd


root:x:0:0:Superuser:/:
```

```
daemon:x:1:1:System daemons:/etc:
bin:x:2:2:Owner of system commands:/bin:
sys:x:3:3:Owner of system files:/usr/sys:
adm:x:4:4:System accounting:/usr/adm:
uucp:x:5:5:UUCP administrator:/usr/lib/uucp:
auth:x:7:21:Authentication administrator:/tcb/files/auth:
cron:x:9:16:Cron daemon:/usr/spool/cron:
listen:x:37:4:Network daemon:/usr/net/nls:
lp:x:71:18:Printer administrator:/usr/spool/lp:
sam:x:200:50:Sam san:/usr/sam:/bin/sh
```

从上面的例子我们可以看到，/etc/passwd 中一行记录对应着一个用户，每行记录又被冒号(:)分隔为 7 个字段，其格式和具体含义如下：

用户名:口令:用户标识号:组标识号:注释性描述:主目录:登录 Shell

1）"用户名"是代表用户账号的字符串。

通常长度不超过 8 个字符，并且由大小写字母和/或数字组成。登录名中不能有冒号(:)，因为冒号在这里是分隔符。

为了兼容起见，登录名中最好不要包含点字符(.)，并且不使用连字符(-)和加号(+)打头。

2）“口令”一些系统中，存放着加密后的用户口令字。

虽然这个字段存放的只是用户口令的加密串，不是明文，但是由于/etc/passwd 文件对所有用户都可读，所以这仍是一个安全隐患。因此，现在许多 Linux 系统（如 SVR4）都使用了 shadow 技术，把真正的加密后的用户口令字存放到/etc/shadow 文件中，而在/etc/passwd 文件的口令字段中只存放一个特殊的字符，例如“x”或者“*”。

3）“用户标识号”是一个整数，系统内部用它来标识用户。

一般情况下它与用户名是一一对应的。如果几个用户名对应的用户标识号是一样的，系统内部将把它们视为同一个用户，但是它们可以有不同的口令、不同的主目录以及不同的登录 Shell 等。

通常用户标识号的取值范围是 0～65 535。0 是超级用户 root 的标识号，1～99 由系统保留，作为管理账号，普通用户的标识号从 100 开始。在 Linux 系统中，这个界限是 500。

4）“组标识号”字段记录的是用户所属的用户组。

它对应着/etc/group 文件中的一条记录。

5)“注释性描述”字段记录着用户的一些个人情况。

例如用户的真实姓名、电话、地址等，这个字段并没有什么实际的用途。在不同的 Linux 系统中，这个字段的格式并没有统一。在许多 Linux 系统中，这个字段存放的是一段任意的注释性描述文字，用做 finger 命令的输出。

6)“主目录”，也就是用户的起始工作目录。

它是用户在登录到系统之后所处的目录。在大多数系统中，各用户的主目录都被组织在同一个特定的目录下，而用户主目录的名称就是该用户的登录名。各用户对自己的主目录有读、写、执行（搜索）权限，其他用户对此目录的访问权限则根据具体情况设置。

7)用户登录后，要启动一个进程，负责将用户的操作传给内核，这个进程是用户登录到系统后运行的命令解释器或某个特定的程序，即 Shell。

Shell 是用户与 Linux 系统之间的接口。Linux 的 Shell 有许多种，每种都有不同的特点。常用的有 sh(Bourne Shell), csh(C Shell), ksh(Korn Shell), tcsh(TENEX/TOPS-20 type C Shell), bash(Bourne Again Shell)等。

系统管理员可以根据系统情况和用户习惯为用户指定某个 Shell。如果不指定 Shell，那么系统使用 sh 为默认的登录 Shell，即这个字段的值为/bin/sh。

用户的登录 Shell 也可以指定为某个特定的程序（此程序不是一个命令解释器）。

利用这一特点，我们可以限制用户只能运行指定的应用程序，在该应用程序运行结束后，用户就自动退出了系统。有些 Linux 系统要求只有那些在系统中登记了的程序才能出现在这个字段中。

8)系统中有一类用户称为伪用户（psuedo users）。

这些用户在/etc/passwd 文件中也占有一条记录，但是不能登录，因为它们的登录 Shell 为空。它们的存在主要是方便系统管理，满足相应的系统进程对文件属主的要求。

常见的伪用户如下所示：

伪用户含义

bin 拥有可执行的用户命令文件

sys 拥有系统文件

adm 拥有帐户文件

uucp UUCP 使用

lp lp 或 lpd 子系统使用

nobody NFS 使用

四、拥有帐户文件

1、除了上面列出的伪用户外，还有许多标准的伪用户，例如：audit, cron, mail, usenet **等，它们也都各自为相关的进程和文件所需要。**

由于/etc/passwd 文件是所有用户都可读的，如果用户的密码太简单或规律比较明显的话，一台普通的计算机就能够很容易地将它破解，因此对安全性要求较高的 Linux 系统都把加密后的口令字分离出来，单独存放在一个文件中，这个文件是/etc/shadow 文件。 有超级用户才拥有该文件读权限，这就保证了用户密码的安全性。

2、/etc/shadow **中的记录行与**/etc/passwd **中的一一对应，它由** pwconv **命令根据** /etc/passwd **中的数据自动产生**

它的文件格式与/etc/passwd 类似，由若干个字段组成，字段之间用":"隔开。这些字段是：

登录名:加密口令:最后一次修改时间:最小时间间隔:最大时间间隔:警告时间:不活动时间:失效时间:标志

"登录名"是与/etc/passwd 文件中的登录名相一致的用户账号

"口令"字段存放的是加密后的用户口令字，长度为 13 个字符。如果为空，则对应用户没有口令，登录时不需要口令；如果含有不属于集合 { ./0-9A-Za-z } 中的字符，则对应的用户不能登录。

"最后一次修改时间"表示的是从某个时刻起，到用户最后一次修改口令时的天数。时间起点对不同的系统可能不一样。例如在 SCO Linux 中，这个时间起点是 1970 年 1 月 1 日。

"最小时间间隔"指的是两次修改口令之间所需的最小天数。

"最大时间间隔"指的是口令保持有效的最大天数。

"警告时间"字段表示的是从系统开始警告用户到用户密码正式失效之间的天数。

"不活动时间"表示的是用户没有登录活动但账号仍能保持有效的最大天数。

"失效时间"字段给出的是一个绝对的天数，如果使用了这个字段，那么就给出相应账号的生存期。期满后，该账号就不再是一个合法的账号，也就不能再用来登录了。

下面是/etc/shadow 的一个例子：

```
# cat /etc/shadow


root:Dnakfw28zf38w:8764:0:168:7:::
daemon:*::0:0::::
bin:*::0:0::::
sys:*::0:0::::
adm:*::0:0::::
uucp:*::0:0::::
nuucp:*::0:0::::
auth:*::0:0::::
cron:*::0:0::::
listen:*::0:0::::
lp:*::0:0::::
sam:EkdiSECLWPdSa:9740:0:0::::
```

3、用户组的所有信息都存放在/etc/group **文件中。**

将用户分组是 Linux 系统中对用户进行管理及控制访问权限的一种手段。

每个用户都属于某个用户组；一个组中可以有多个用户，一个用户也可以属于不同的组。

当一个用户同时是多个组中的成员时，在/etc/passwd 文件中记录的是用户所属的主组，也就是登录时所属的默认组，而其他组称为附加组。

用户要访问属于附加组的文件时，必须首先使用 newgrp 命令使自己成为所要访问的组中的成员。

用户组的所有信息都存放在/etc/group 文件中。此文件的格式也类似于/etc/passwd 文件，由冒号(:)隔开若干个字段，这些字段有：

组名:口令:组标识号:组内用户列表

"组名"是用户组的名称，由字母或数字构成。与/etc/passwd 中的登录名一样，组名不应重复。

"口令"字段存放的是用户组加密后的口令字。一般 Linux 系统的用户组都没有口令，即这个字段一般为空，或者是*。

"组标识号"与用户标识号类似，也是一个整数，被系统内部用来标识组。

"组内用户列表"是属于这个组的所有用户的列表/b]，不同用户之间用逗号(,)分隔。这个用户组可能是用户的主组，也可能是附加组。

/etc/group 文件的一个例子如下：

```
root::0:root
bin::2:root,bin
sys::3:root,uucp
adm::4:root,adm
daemon::5:root,daemon
lp::7:root,lp
```

```
users::20:root,sam
```

五、添加批量用户

添加和删除用户对每位 Linux 系统管理员都是轻而易举的事，比较棘手的是如果要添加几十个、上百个甚至上千个用户时，我们不太可能还使用 useradd 一个一个地添加，必然要找一种简便的创建大量用户的方法。Linux 系统提供了创建大量用户的工具，可以让您立即创建大量用户，方法如下：

（1）先编辑一个文本用户文件。

每一列按照/etc/passwd 密码文件的格式书写，要注意每个用户的用户名、UID、宿主目录都不可以相同，其中密码栏可以留做空白或输入 x 号。一个范例文件 user.txt 内容如下：

```
user001::600:100:user:/home/user001:/bin/bash
user002::601:100:user:/home/user002:/bin/bash
user003::602:100:user:/home/user003:/bin/bash
user004::603:100:user:/home/user004:/bin/bash
user005::604:100:user:/home/user005:/bin/bash
user006::605:100:user:/home/user006:/bin/bash
```

（2）以 root 身份执行命令 /usr/sbin/newusers，从刚创建的用户文件 user.txt 中导入数据，创建用户：

newusers < user.txt

然后可以执行命令 vipw 或 vi /etc/passwd 检查 /etc/passwd 文件是否已经出现这些用户的数据，并且用户的宿主目录是否已经创建。

（3）执行命令/usr/sbin/pwunconv**。**

将 /etc/shadow 产生的 shadow 密码解码，然后回写到 /etc/passwd 中，并将/etc/shadow 的 shadow 密码栏删掉。这是为了方便下一步的密码转换工作，即先取消 shadow password 功能。

pwunconv

（4）编辑每个用户的密码对照文件。

范例文件 passwd.txt 内容如下：

```
user001:密码
user002:密码
user003:密码
user004:密码
user005:密码
user006:密码
```

（5）以 root **身份执行命令** /usr/sbin/chpasswd**。**

创建用户密码，chpasswd 会将经过 /usr/bin/passwd 命令编码过的密码写入 /etc/passwd 的密码栏。

chpasswd < passwd.txt

（6）确定密码经编码写入/etc/passwd **的密码栏后。**

执行命令 /usr/sbin/pwconv 将密码编码为 shadow password，并将结果写入 /etc/shadow。

pwconv

这样就完成了大量用户的创建了，之后您可以到/home 下检查这些用户宿主目录的权限设置是否都正确，并登录验证用户密码是否正确。