19.用户和用户组管理

Linux 用户和用户组管理

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。

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。**

这里有必要提一下：我们在windows下几乎都习惯了，就使用电脑中的默认的管理员用户，很少在自己的电脑上添加很多个用户，而且还给用户分配不同的权限，这里不要把这个习惯带到Linux系统上来，注意本质区别，我们平时用的是个人计算机，Linux是服务器的操作系统！用户和用户组管理不是防备网络上的普通用户，是防备内部系统管理人员的！这个基本道理要认识清楚！

一、Linux系统用户账号的基本管理

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，直到你给它设置了密码后，它才可以用来登录系统。

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（一般情况下，我们不会用他们，了解一下就ok）：
  - -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，会自动创建主目录。
  - -g 用户组　　指定用户所属的用户组，**这个组是初始组。只能是**

一个，建议不要改它，默认组名和用户名同名。

- -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，每个用户都有对应的一个shell，不同的shell功能不一样的，/bin/bash 基本shell。
- -u 用户号　　指定用户的用户号（就是用户的id），注意linux中区别用户不是根据用户名的，是根据uid的。

- 用户名:
  - 指定新账号的登录名。

实例1

```
# useradd -d /sam sam
```

此命令创建了一个用户sam，其中-d选项用来为登录名sam产生一个主目录/sam

实例2

```
# useradd -s /bin/bash -g sam -G sam,root tom
```

此命令新建了一个用户tom，该用户的登录Shell是 /bin/bash，它属于sam 用户组，同时又属于sam 和root用户组，其中sam 用户组是其主组,也叫初始组。刚才说了-g这个初始组不建议改，要将这个新用户添加到其他主建议使用-G

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/etc/shadow, /etc/group等文件。

添加的新用户一般情况下不使用任何选项，用默认选择值，两个文件中设置这些默认值：

文件1：/etc/default/useradd

```
1 # useradd defaults file
2 GROUP=100  已经失效
3 HOME=/home  默认家目录
4 INACTIVE=-1  用户密码过期后的宽限天数，-1表示永不过期，在实际的工作中建议修改为0表示密码过期后就失效
5 EXPIRE=  密码失效时间，一般不用，它是个时间戳，2019-2-30，
6 SHELL=/bin/bash
7 SKEL=/etc/skel  指定新用户的模板目录，当添加新用户的时候，这个目录下的所有文件会自动被复制到新添加的用户的家目录下
8 CREATE_MAIL_SPOOL=yes  给每个新建的用户都创建邮箱，注意这里只是打开要建的开关，建在哪儿下面的文件决定
```

文件2：/etc/login.defs

15 MAIL_DIR /var/spool/mail 新建用户要建邮箱的话，就建在这里

25 PASS_MAX_DAYS 99999 密码的有效期，单位是天，永久

26 PASS_MIN_DAYS 0 是否可修改密码,0表示可修改,非0表示多少天后可修改，修改密码后必须等这么多天后才能在修改

27 PASS_MIN_LEN 5 密码最小长度，但已经失效，不管它

28 PASS_WARN_AGE 7 密码失效前多少天在用户登录时通知用户修改密码

33 UID_MIN 1000

34 UID_MAX 60000

这两行代表创建用户时， 最小 UID 和最大的 UID 的范围

36 SYS_UID_MIN 201

37 SYS_UID_MAX 999

这两行代表创建系统用户时， 最小 UID 和最大的 UID 的范围，这个用户是系统内部用来启动服务用的，不能登录的内部用户

42 GID_MIN 1000

43 GID_MAX 60000

这两行指定了 GID 的最小值和最大值之间的范围

45 SYS_GID_MIN 201

46 SYS_GID_MAX 999

这两行代表创建系统用户的所属组， 最小 GID 和最大的 GID 的范围

60 CREATE_HOME yes 这行指定建立用户时是否自动建立用户的家目录，默认是建立

64 UMASK 077 这行指定的是建立的用户家目录的默认权限，权限掩码077，所以新建的用户家目录的权限是 700

71 ENCRYPT_METHOD SHA512 这行指定 Linux 用户的密码使用 SHA512 散列模式加密算法

3、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 -r 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。

4、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，-g，-G，-s，-u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。

例如：

```
# usermod -s /bin/sh -d /home/z  -G root user1
```

此命令将用户user1的登录Shell修改为sh，主目录改为/home/z，加入用户组root。

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。
- --stdin: 可以将通过管道符输出的数据作为用户的密码。

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
```

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
```

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 这个锁定实质上是修改了保存密码的文件/etc/shadow，在密码前加了一个！！，使得密码失效

passwd -u sam 解开锁定

便于在脚本中批量修改密码，Linux还给我提供了通过管道符输出的数据作为用户的密码的功能：

echo "123456" | passwd --stdin user1 这样避免了人机交互的手工修改密码的麻烦，一步到位

一般情况下，批量添加用户，或者批量修改用户的密码，都是设置的初始密码，这个密码肯定不安全的，所以，就需要要求用户第一次登录的时候，必须修改自己的初始密码，这个情况可以通过命令来实现：

chage -d 0 user1 执行以后，user1下次登录的时候，就必须修改密码后才能使用系统

6、切换用户命令

su 命令可以切换成不同的用户身份，命令格式如下：

su [选项] 用户名

选项：

-: 选项只使用“-”代表连带用户的环境变量一起切换

“-”不能省略，如果省略，你查看环境变量（env）就会发现，表面上切换了用户，但根子上没换彻底，注意。

二、Linux系统用户组的管理

每个用户都有至少属于一个用户组，这个组就是初始组，每个用户必须只能拥有一个初始组，用户是依附于初始组生存的。同时用户可以没有附加组，也可以同时属于很多个附加组。

用户组的管理涉及用户组的添加、删除。组的增加、删除，实际上就是对/etc/group文件进行修改。

1、增加一个新的用户组使用groupadd命令。其格式如下：

groupadd 用户组的组名

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。

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。

注意：当删除初始组的时候，有个前提：必须把和该组关联的那个初始用户删除后，这个组才能删除！

3、把用户添加进组或从组中删除： gpasswd。其语法如下：

gpasswd ［选项］ 组名 这里的组是指的是附加组

选项：

-a 用户名： 把用户加入组

-d 用户名： 把用户从组中删除

例如：

groupadd grouptest

gpasswd -a user1 grouptest

gpasswd -d user1 grouptest

强调：#usermod -G grouptest user1

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。（了解）

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标用户组。这个东西体现在新建文件的时候，所属组会变！

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。

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。linux一切皆文件嘛！

与用户和用户组相关的信息都存放在一些系统文件中，这些文件中最核心的包括/etc/passwd, /etc/shadow, /etc/group

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:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:998:User for polkitd:/:/sbin/nologin
abrt:x:173:173::/etc/abrt:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
chrony:x:998:996::/var/lib/chrony:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
user1:x:1000:1000::/home/user1:/bin/bash
user2:x:1001:1001::/home/user2:/bin/bash
saslauth:x:997:76:Saslauthd user:/run/saslauthd:/sbin/nologin
```

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uid:组标识号gid:注释性描述:家目录:登录Shell

1）"用户名"是代表用户账号的字符串

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”

由于/etc/passwd文件对所有用户都可读，密码放这里是一个安全隐患。因此，Linux 系统使用了shadow技术，把真正的加密后的用户口令存放到/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“x”。

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、不同的家目录以及不同的登录Shell等。

0　超级用户 UID。如果用户 UID 为 0，代表这个账号是管理员账号。

1-999 系统用户（伪用户）UID。这些 UID 账号是系统保留给系统用户的 UID，也就是

说 UID 是 1-999 范围内的用户是不能登录系统的，而是用来运行系统或服务的。其中 1-200 是系统保留的账号，系统自动创建。201-999　是预留给用户创建系统账号的。

1000-60000 普通用户 UID。建立的普通用户 UID 从1000 开始，最大到 60000。

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。

6）“家目录”，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。

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bash为默认的登录Shell，即这个字段的值为/bin/bash。

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为空或/sbin/nologin。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。

2、/etc/shadow中的记录行与/etc/passwd中的一一对应

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

下面是/etc/shadow的一个例子：

```
# cat /etc/shadow
root:$6$/mVAoIY1$LDmGc5XLnuyfVVFFumjGWYO42IQyTzC5NwuL7feJCYEQARpMv2P/TlliqJ1RHR3ZjNUsYH7QStilJQfU8tlhj.:17922:0:99999:7:::
bin:*:17834:0:99999:7:::
daemon:*:17834:0:99999:7:::
adm:*:17834:0:99999:7:::
lp:*:17834:0:99999:7:::
sync:*:17834:0:99999:7:::
shutdown:*:17834:0:99999:7:::
halt:*:17834:0:99999:7:::
mail:*:17834:0:99999:7:::
operator:*:17834:0:99999:7:::
games:*:17834:0:99999:7:::
```

1）"登录名"是与/etc/passwd文件中的登录名相一致的用户账号

2）"密码"，这时用户的密码经过加密算法加密后的密码密文

我们可以在密码密文前加入“！”或“*”改变加密值让密码失效，使这个用户无法登陆，达到禁止用户登录的效果。

注意所有伪用户的密码都是“！！”或“*”，代表没有密码是不能登录的。

新创建的用户如果不设定密码，它的密码项也是“！！”，代表这个用户没有密码，不能登录

3）"最后一次修改密码的时间" 表示从世界标准时间1970-1-1 0:0:0，到用户最后一次修改密码时的间隔天数。这个字段如果改成0，那么用户在下次登录的时候，就必须修改密码，前面学习的# chage -d 0 user1命令其实就是修改这个字段的值

知道一个日期，得到天数：# echo $(($(date --date="2019/02/25" +%s)/86400+1))

反过来，知道天数，得到日期：# date -d "1970-01-01 15775 days"

4）"最小时间间隔" 最后一次修改密码（第 3 字段中记录的时间）后，必须等多少天后，才能在修改密码。默认是0，没间隔

这个字段值就是前面我们讲的用户默认配置文件2：/etc/login.defs中的PASS_MIN_DAYS配置的值就是这个的默认值

5）"最大时间间隔"　第 3 字段中记录的时间后多少天密码失效。默认值99999，几乎就是永久

默认值和配置文件：/etc/login.defs中的PASS_MAX_DAYS配置项关联

6）"警告时间" 第5字段设置是有效期到期前多少天，开始在用户登录的时候发出警告，让用户修改密码

默认值和配置文件：/etc/login.defs中的PASS_WARN_AGE配置项关联

7）"不活动时间"　第5字段设置是有效期到期后账号仍能用的最大天数。

默认值和文件：/etc/default/useradd的INACTIVE配置关联，-1表示到器后，用户仍然能用，显然改0合理一些

8）"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。这里同样要写时间戳，也就是用 1970 年 1 月 1 日进行时间换算

默认值配置文件/etc/default/useradd的的EXPIRE配置项关联

9）保留字段，现在没意义

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

1）"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。

2）"组密码位"存放的是一个标记，比如字母x，真正的密码放在文件/etc/gshadow中。这个密码一般情况下都不设置！

3）"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。

4）"组内用户列表"是属于这个组的所有用户的列表，不同用户之间用逗号(,)分隔。这里显示的是付属组内用户。

例子如下：

```
# cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
```

4 组密码文件/etc/gshadow（了解）

如果给用户组设定了组管理员，并给该用户组设定了组密码，组密码就保存在这个文件当中。

组管理员就可以利用这个密码管理这个用户组了。

四、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
user002
user003
user004
user005
user006
```

（2）以root身份执行命令 ，创建用户：

```
# for i in `cat user.txt`; do useradd $i; done
```

然后可以执行命令 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）编辑每个用户的密码对照文件。

范例文件 passwd.txt 内容如下：

user001:123

user002:123

user003:123

user004:123

user005:123

user006:123

（4）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