## Linux的安全性

Linux安全系统的核心是用户账户。每个能进入Linux系统的用户都会被分配唯一的用户账户。用户对系统中各种对象的访问权限取决于他们登录系统时用的账户。 用户权限是通过创建用户时分配的用户ID（User ID，通常缩写为UID）来跟踪的。UID是数值，每个用户都有唯一的UID，但在登录系统时用的不是UID，而是登录名。登录名是用户用来登录系统的长八字符的字符串（字符可以是数字或字母），同时会关联一个对应的密码。 Linux系统使用特定的文件和工具来跟踪和管理系统上的用户账户。

注意：UID是数值，它关乎权限， 登录名是字符串，它关联一个密码

### 1./etc/passwd文件

Linux系统使用一个专门的文件来将用户的登录名匹配到对应的UID值。这个文件就是 /etc/passwd文件，它包含了一些与用户有关的信息。

root用户账户是Linux系统的管理员，固定分配给它的UID是0。如下所示。



Linux系统会为各种各样的功能创建不同的用户账户，而这些账户并不是真的用户。这些账户叫作系统账户，是系统上运行的各种服务进程访问资源用的特殊账户。所有运行在后台的服务都需要用一个系统用户账户登录到Linux系统上。如mysql



历史教训：

在安全成为一个大问题之前，这些服务经常会用root账户登录。遗憾的是，如果有

非授权的用户攻陷了这些服务中的一个，他立刻就能作为root用户进入系统。为了防止发生这种情况，现在运行在Linux服务器后台的几乎所有的服务都是用自己的账户登录。这样的话，即使有人攻入了某个服务，也无法访问整个系统。

现在的做法：

Linux为系统账户预留了500以下的UID值。有些服务甚至要用特定的UID才能正常工作。为普通用户创建账户时，大多数Linux系统会从500开始，将第一个可用UID分配给这个账户（并非所有的Linux发行版都是这样）。

7个字段含义：



登录用户名：jack 

用户密码:x

用户账户的UID（数字形式）:500 

用户账户的组ID（GID）（数字形式）:500 

用户账户的文本描述（称为备注字段） :centosB

用户HOME目录的位置:/home/jack 

用户的默认shell: /bin/bash

/etc/passwd文件中的密码字段都被设置成了x，这并不是说所有的用户账户都用相同的密码。 在早期的Linux上，/etc/passwd文件里有加密后的用户密码。但鉴于很多程序都需要访问 /etc/passwd文件获取用户信息，这就成了一个安全隐患。随着用来破解加密密码的工具的不断演 进，用心不良的人开始忙于破解存储在/etc/passwd文件中的密码。Linux开发人员需要重新考虑这个策略。 现在，绝大多数Linux系统都将用户密码保存在另一个单独的文件中（叫作shadow文件，位置 在/etc/shadow）。只有特定的程序（比如登录程序）才能访问这个文件。 /etc/passwd是一个标准的文本文件。你可以用任何文本编辑器在/etc/password文件里直接手动 进行用户管理（比如添加、修改或删除用户账户）。但这样做极其危险。如果/etc/passwd文件出现 损坏，系统就无法读取它的内容了，这样会导致用户无法正常登录（即便是root用户）。用标准的Linux用户管理工具去执行这些用户管理功能就会安全许多。

### 2. /etc/shadow文件

/etc/shadow文件对Linux系统密码管理提供了更多的控制。只有root用户才能访问/etc/shadow 文件，这让它比起/etc/passwd安全许多。 /etc/shadow文件为系统上的每个用户账户都保存了一条记录。记录就像下面这样：



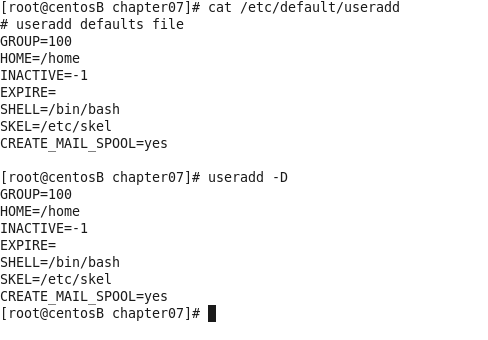
在/etc/shadow文件的每条记录中都有9个字段： 

1. 与/etc/passwd文件中的登录名字段对应的登录名 
2. 加密后的密码 
3. 自上次修改密码后过去的天数密码（自1970年1月1日开始计算） 
4. 多少天后才能更改密码 
5. 多少天后必须更改密码 
6. 密码过期前提前多少天提醒用户更改密码
7. 密码过期后多少天禁用用户账户 
8. 用户账户被禁用的日期（用自1970年1月1日到当天的天数表示） 
9. 预留字段给将来使用

使用shadow密码系统后，Linux系统可以更好地控制用户密码。它可以控制用户多久更改一 次密码，以及什么时候禁用该用户账户，如果密码未更新的话。

### 3. 添加新用户

来向Linux系统添加新用户的主要工具是useradd。这个命令简单快捷，可以一次性创建 新用户账户及设置用户HOME目录结构。useradd命令使用系统的默认值以及命令行参数来设置 用户账户。系统默认值被设置在/etc/default/useradd文件中。可以使用加入了-D选项的useradd 命令查看所用Linux系统中的这些默认值。



在创建新用户时，如果你不在命令行中指定具体的值，useradd命令就会使用-D选项所显示 的那些默认值。这个例子列出的默认值如下： 

1.新用户会被添加到GID为100的公共组； 

2.新用户的HOME目录将会位于/home/loginname； 

3.新用户账户密码在过期后不会被禁用； 

4.新用户账户未被设置过期日期； 

5.新用户账户将bash shell作为默认shell； 

6.系统会将/etc/skel目录下的内容复制到用户的HOME目录下； 

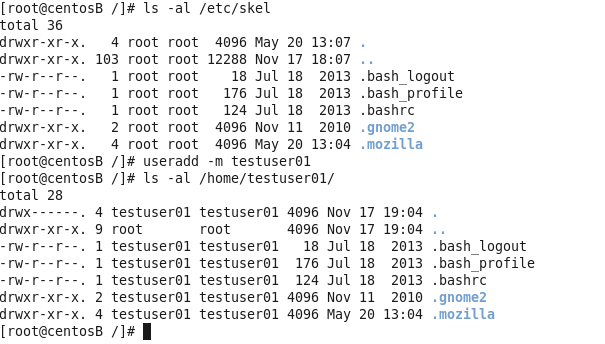
7.系统为该用户账户在mail目录下创建一个用于接收邮件的文件。

第6个值很有意思。useradd命令允许管理员创建一份默认的HOME目录配置，然后把 它作为创建新用户HOME目录的模板。这样就能自动在每个新用户的HOME目录里放置默认的系 统文件。在Centos Linux系统上，/etc/skel目录有下列文件：

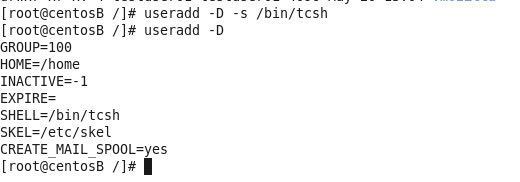
它们是bash shell环境的标准启动文 件。系统会自动将这些默认文件复制到你创建的每个用户的HOME目录。 可以用默认系统参数创建一个新用户账户，然后检查一下新用户的HOME目录。

默认情况下，useradd命令不会创建HOME目录，但是-m命令行选项会使其创建HOME目录。 你能在此例中看到，useradd命令创建了新HOME目录，并将/etc/skel目录中的文件复制了过来。

例子如下

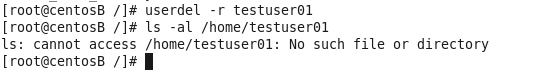


更改默认值则非常简单



### 4.删除用户

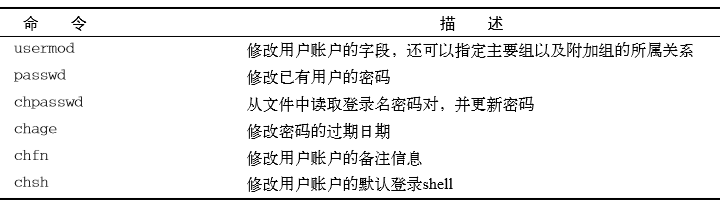
如果你想从系统中删除用户，userdel可以满足这个需求。默认情况下，userdel命令会只删除/etc/passwd文件中的用户信息，而不会删除系统中属于该账户的任何文件。 如果加上-r参数，userdel会删除用户的HOME目录以及邮件目录。然而，系统上仍可能存 有已删除用户的其他文件。这在有些环境中会造成问题。



特别注意：

在有大量用户的环境中使用-r参数时要特别小心。你永远不知道用户是否在其HOME目 录下存放了其他用户或其他程序要使用的重要文件。记住，在删除用户的HOME目录之 前一定要检查清楚！

### 5.修改用户



#### usermod

usermod命令是用户账户修改工具中强大的一个。它能用来修改/etc/passwd文件中的大部 分字段，只需用与想修改的字段对应的命令行参数就可以了。参数大部分跟useradd命令的参数 一样（比如，-c修改备注字段，-e修改过期日期，-g修改默认的登录组）。除此之外，还有另外 一些可能派上用场的选项。

-l修改用户账户的登录名。 

-L锁定账户，使用户无法登录。 

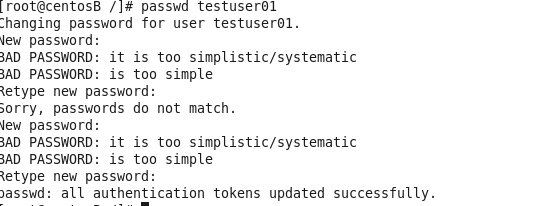
-p修改账户的密码。 

-U解除锁定，使用户能够登录。

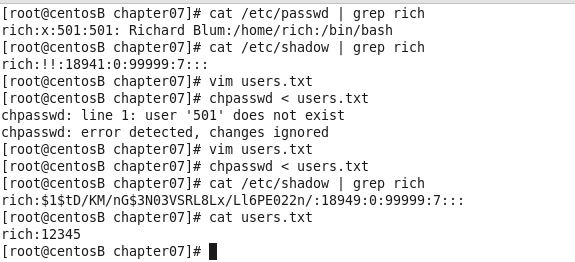
-L选项尤其实用。它可以将账户锁定，使用户无法登录，同时无需删除账户和用户的数据。 要让账户恢复正常，只要用-U选项就行了。

#### 2.passwd 和chpasswd

改变用户密码的一个简便方法就是用passwd命令。



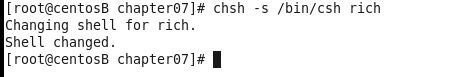
如果只用passwd命令，它会改你自己的密码。系统上的任何用户都能改自己的密码，但只 有root用户才有权限改别人的密码。 -e选项能强制用户下次登录时修改密码。你可以先给用户设置一个简单的密码，之后再强制 在下次登录时改成他们能记住的更复杂的密码。 如果需要为系统中的大量用户修改密码，chpasswd命令可以事半功倍。chpasswd命令能从 标准输入自动读取登录名和密码对（由冒号分割）列表，给密码加密，然后为用户账户设置。你 也可以用重定向命令来将含有userid:passwd对的文件重定向给该命令。实例如下



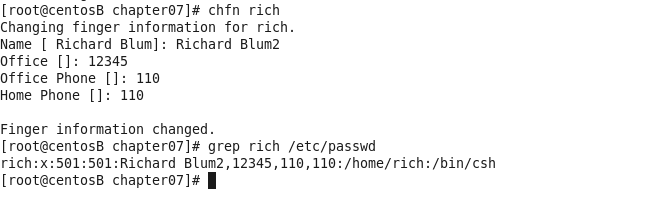
#### chsh, chfn和chage

chsh、chfn和chage工具专门用来修改特定的账户信息。

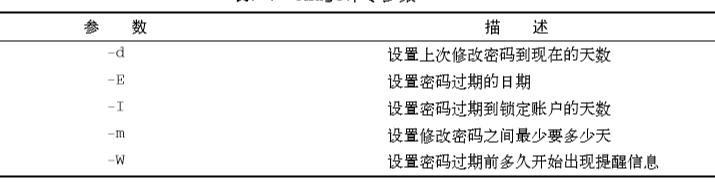
chsh命令用来快速修改默认的用户登录shell。使用时必须用shell的全路径名作为参数，不能只用shell名。



chfn命令提供了在/etc/passwd文件的备注字段中存储信息的标准方法。chfn命令会将用于 Unix的finger命令的信息存进备注字段，而不是简单地存入一些随机文本（比如名字或昵称之 类的），或是将备注字段留空



chage命令用来帮助管理用户账户的有效期。你需要对每个值设置多个参数，命令参数如下：



chage命令的日期值可以用下面两种方式中的任意一种： 

YYYY-MM-DD格式的日期 

代表从1970年1月1日起到该日期天数的数值

chage命令中有个好用的功能是设置账户的过期日期。有了它，你就能创建在特定日期自动过期的临时用户，再也不需要记住删除用户了！过期的账户跟锁定的账户很相似：账户仍然存在， 但用户无法用它登录。

## 2.使用Linux组

用户账户在控制单个用户安全性方面很好用，但涉及在共享资源的一组用户时就有问题。为了解决这个问题，Linux系统采用了另外一个安全概念——组（group）。

组权限允许多个用户对系统中的对象（比如文件、目录或设备等）共享一组共用的权限。 Linux发行版在处理默认组的成员关系时略有差异。

有些Linux发行版会创建一个组，把所有用户都当作这个组的成员。遇到这种情况要特别小心，因为文件很有可能对其他用户也是可读的。

有些发行版会为每个用户创建单独的一个组，这样可以更安全一些。

每个组都有唯一的GID——跟UID类似，在系统上这是个唯一的数值。除了GID，每个组还有唯一的组名。Linux系统上有一些组工具可以创建和管理你自己的组。

### 1./etc/group文件

与用户账户类似，组信息也保存在系统的一个文件中。/etc/grou文件包含系统上用到的每个组的信息。和UID一样，GID在分配时也采用了特定的格式。系统账户用的组通常会分配低于500的GID 值，而用户组的GID则会从500开始分配。



/etc/group文件有4个字段：

组名 

组密码 

GID 

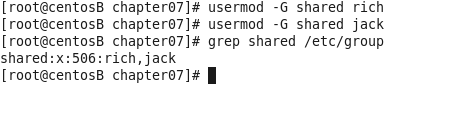
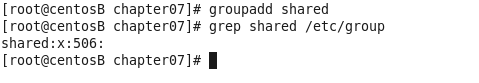
属于该组的用户列表

组密码允许非组内成员通过它临时成为该组成员。这个功能并不很普遍，但确实存在。 千万不能通过直接修改/etc/group文件来添加用户到一个组，要用usermod命令。在添加用户到不同的组之前，首先得创建组。

说明：用户账户列表某种意义上有些误导人。你会发现，在列表中，有些组并没有列出用户。 这并不是说这些组没有成员。当一个用户在/etc/passwd文件中指定某个组作为默认组时， 用户账户不会作为该组成员再出现在/etc/group文件中。

### 2.创建新组

groupadd命令可在系统上创建新组。在创建新组时，默认没有用户被分配到该组。groupadd命令没有提供将用户添加到组中的 选项，但可以用usermod命令来弥补这一点。



shared组现在有两个成员：jack和rich。usermod命令的-G选项会把这个新组添加到该用 户账户的组列表里。

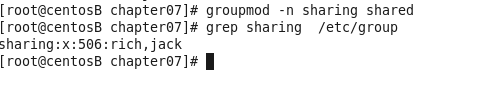
说明: 如果更改了已登录系统账户所属的用户组，该用户必须登出系统后再登录，组关系的更 改才能生效。

警告 为用户账户分配组时要格外小心。如果加了-g选项，指定的组名会替换掉该账户的默认 组。-G选项则将该组添加到用户的属组的列表里，不会影响默认组。

### 3.修改组

在/etc/group文件中可以看到，需要修改的组信息并不多。groupmod命令可以修改已有组的 GID（加-g选项）或组名（加-n选项）。

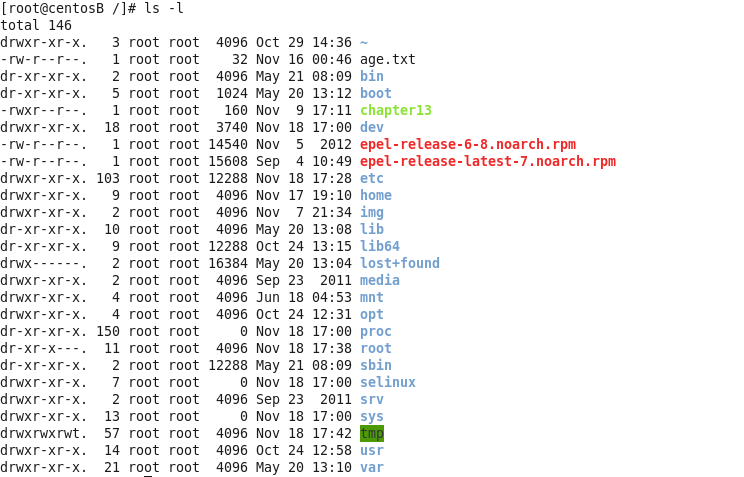
修改组名时，GID和组成员不会变，只有组名改变。由于所有的安全权限都是基于GID的， 你可以随意改变组名而不会影响文件的安全性。



## 3.理解文件权限

### 1.使用文件权限符

ls命令可以用来查看Linux系统上的文件、目录和设备的权限。



输出结果的第一个字段就是描述文件和目录权限的编码。这个字段的第一个字符代表了对象的类型

-代表文件 d代表目录l代表链接c代表字符型设备b代表块设备n代表网络设备

之后有3组三字符的编码。每一组定义了3种访问权限：

r代表对象是可读的 w代表对象是可写的x代表对象是可执行的

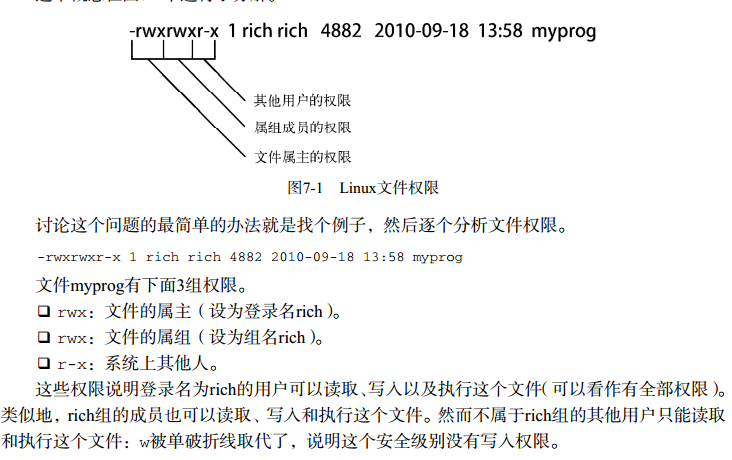
若没有某种权限，在该权限位会出现单破折线。这3组权限分别对应对象的3个安全级别：

对象的属主

对象的属组

系统其他用户

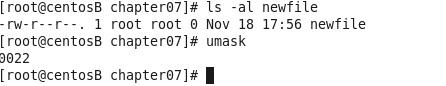
这个概念在图7-1中进行了分解



### 2.默认文件权限

这些文件权限从何而来，答案是umask。umask命令用来设置所创建文件和目录

的默认权限

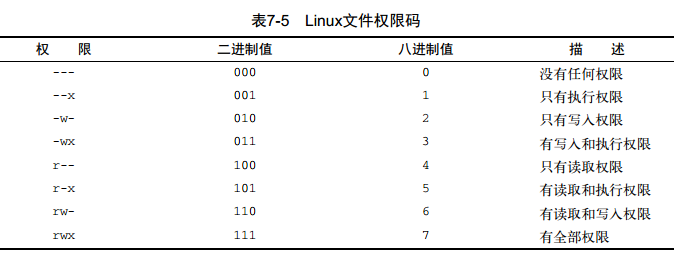


touch命令用分配给我的用户账户的默认权限创建了这个文件。umask命令可以显示和设置这个默认权限。

说明：第一位代表了一项特别的安全特性，叫作粘着位（sticky bit）后面的3位表示文件或目录对应的umask八进制值。

八进制模式的安全性设置。

八进制模式的安全性设置先获取这3个rwx权限的值，然后将其转换成3位二进制值，用一个八进制值来表示。在这个二进制表示中，每个位置代表一个二进制位。因此，如果读权限是唯一置位的权限，权限值就是r--，转换成二进制值就是100，代表的八进制值是4。表7-5列出了可能会遇到的组合。



八进制模式先取得权限的八进制值，然后再把这三组安全级别（属主、属组和其他用户）的八进制值顺序列出。因此，八进制模式的值664代表属主和属组成员都有读取和写入的权限，而其他用户都只有读取权限。

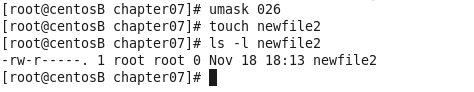
了解八进制模式权限是怎么工作的之后，umask值反而更叫人困惑了。我的Linux系统上默认的八进制的umask值是0022，而我所创建的文件的八进制权限却是644，这是如何得来的呢？

umask值只是个掩码。它会屏蔽掉不想授予该安全级别的权限。接下来我们还得再多进行一些八进制运算才能搞明白来龙去脉。

要把umask值从对象的全权限值中减掉。对文件来说，全权限的值是666（所有用户都有读和写的权限）；而对目录来说，则是777（所有用户都有读、写、执行权限）。

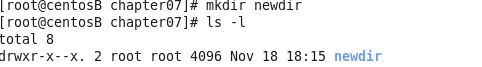
所以在上例中，文件一开始的权限是666，减去umask值022之后，剩下的文件权限就成了644。

在大多数Linux发行版中，umask值通常会设置在/etc/profile启动文件中（参见第6章），不过有一些是设置在/etc/login.defs文件中的（如Ubuntu）。可以用umask命令为默认umask设置指定一个新值。



在把umask值设成026后，默认的文件权限变成了640，因此新文件现在对组成员来说是只读的，而系统里的其他成员则没有任何权限.

umask值同样会作用在创建目录上。



由于目录的默认权限是777，umask作用后生成的目录权限不同于生成的文件权限。umask值026会从777中减去，留下来751作为目录权限设置。

## 改变安全性设置

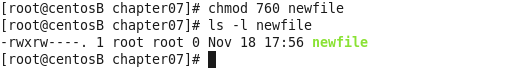
### 改变权限

chmod命令用来改变文件和目录的安全性设置。该命令的格式如下：

chmod options mode file

mode参数可以使用八进制模式或符号模式进行安全性设置。

1. 八进制模式设置非常直观，直接用期望赋予文件的标3位八进制权限码即可。



八进制文件权限会自动应用到指定的文件上。符号模式的权限就没这么简单了。

与通常用到的3组三字符权限字符不同，chmod命令采用了另一种方法。下面是在符号模式下指定权限的格式。

[ugoa…][[+-=][rwxXstugo…]

非常有意义，不是吗？第一组字符定义了权限作用的对象：

 u代表用户

 g代表组

 o代表其他

 a代表上述所有

下一步，后面跟着的符号表示你是想在现有权限基础上增加权限(+)，还是在现有权限基础上移除权限(-)，或是将权限设置成后面的值(=)。

最后，第三个符号代表作用到设置上的权限。你会发现，这个值要比通常的rwx多。额外的设置有以下几项。

X：如果对象是目录或者它已有执行权限，赋予执行权限。

s：运行时重新设置UID或GID。

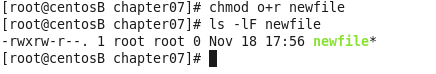
t：保留文件或目录。

u：将权限设置为跟属主一样。

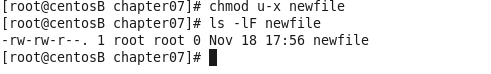
g：将权限设置为跟属组一样。

o：将权限设置为跟其他用户一样。

像这样使用这些权限。



不管其他用户在这一安全级别之前都有什么权限，o+r都给这一级别添加读取权限。



u-x移除了属主已有的执行权限。注意ls命令的-F选项，它能够在具有执行权限的文件名后加一个星号。

options为chmod命令提供了另外一些功能。-R选项可以让权限的改变递归地作用到文件和子目录。你可以使用通配符指定多个文件，然后利用一条命令将权限更改应用到这些文件上。

### 2. 改变所属关系

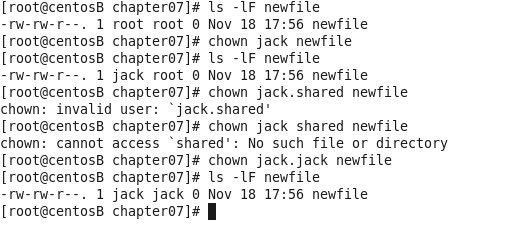
有时你需要改变文件的属主，比如有人离职或开发人员创建了一个在产品环境中需要归属在系统账户下的应用。Linux提供了两个命令来实现这个功能：chown命令用来改变文件的属主，chgrp命令用来改变文件的默认属组。

#### a. chown命令

chown命令的格式如下。

chown options owner[.group] file

可用登录名或UID来指定文件的新属主。chown命令也支持同时改变文件的属主和属组。



如果你不嫌麻烦，可以只改变一个目录的默认属组。

# chown .rich newfile

# ls -l newfile

-rw-rw-r-- 1 dan rich 0 Sep 20 19:16 newfile

#

最后，如果你的Linux系统采用和用户登录名匹配的组名，可以只用一个条目就改变二者。

# chown test. newfile

# ls -l newfile

-rw-rw-r-- 1 test test 0 Sep 20 19:16 newfile

#

chown命令采用一些不同的选项参数。-R选项配合通配符可以递归地改变子目录和文件的所属关系。-h选项可以改变该文件的所有符号链接文件的所属关系。

说明 只有root用户能够改变文件的属主。任何属主都可以改变文件的属组，但前提是属主必须是原属组和目标属组的成员。

chgrp命令可以更改文件或目录的默认属组。

$ chgrp shared newfile

$ ls -l newfile

-rw-rw-r-- 1 rich shared 0 Sep 20 19:16 newfile

用户账户必须是这个文件的属主，除了能够更换属组之外，还得是新组的成员。现在shared组的任意一个成员都可以写这个文件了。这是Linux系统共享文件的一个途径。然而。

## 5.共享文件

Linux系统上共享文件的方法是创建组。但在一个完整的共享文件的环境中，事情会复杂得多。创建新文件时，Linux会用你默认的UID和GID给文件分配权限。想让其他人也能访问文件，要么改变其他用户所在安全组的访问权限，要么就给文件分配一个包含其他用户的新默认属组。

如果你想在大范围环境中创建文档并将文档与人共享，这会很烦琐。幸好有一种简单的方法可以解决这个问题。

Linux还为每个文件和目录存储了3个额外的信息位。

设置用户ID（SUID）：当文件被用户使用时，程序会以文件属主的权限运行。

设置组ID（SGID）：对文件来说，程序会以文件属组的权限运行；对目录来说，目录中

创建的新文件会以目录的默认属组作为默认属组。

粘着位：进程结束后文件还驻留（粘着）在内存中。

SGID位对文件共享非常重要。启用SGID位后，你可以强制在一个共享目录下创建的新文件

都属于该目录的属组，这个组也就成为了每个用户的属组。

SGID可通过chmod命令设置。它会加到标准3位八进制值之前（组成4位八进制值），或者在

符号模式下用符号s。

如果你用的是八进制模式，你需要知道这些位的位置，如表7-6所示。

表7-6 chmod SUID、SGID和粘着位的八进制值

二进制值 八进制值 描 述

000 0 所有位都清零

001 1 粘着位置位

010 2 SGID位置位

011 3 SGID位和粘着位都置位

100 4 SUID位置位

101 5 SUID位和粘着位都置位

110 6 SUID位和SGID位都置位

111 7 所有位都置位

因此，要创建一个共享目录，使目录里的新文件都能沿用目录的属组，只需将该目录的SGID

位置位。

$ mkdir testdir

$ ls -l

drwxrwxr-x 2 rich rich 4096 Sep 20 23:12 testdir/

$ chgrp shared testdir

$ chmod g+s testdir

$ ls -l

drwxrwsr-x 2 rich shared 4096 Sep 20 23:12 testdir/

$ umask 002

$ cd testdir

$ touch testfile

$ ls -l

total 0

-rw-rw-r-- 1 rich shared 0 Sep 20 23:13 testfile

$

首先，用mkdir命令来创建希望共享的目录。然后通过chgrp命令将目录的默认属组改为包

含所有需要共享文件的用户的组（你必须是该组的成员）。最后，将目录的SGID位置位，以保证

目录中新建文件都用shared作为默认属组。

为了让这个环境能正常工作，所有组成员都需把他们的umask值设置成文件对属组成员可

写。在前面的例子中，umask改成了002，所以文件对属组是可写的。

做完了这些，组成员就能到共享目录下创建新文件了。跟期望的一样，新文件会沿用目录的

属组，而不是用户的默认属组。现在shared组的所有用户都能访问这个文件

1. 共享文件