组管理和权限管理

2022年3月18日 20:53

1. Linux组基本介绍

- Linux中每个用户属于一个组，不能独立于组以外。所以在Linux中每个文件存在组的概念：
  1. 所有者
  2. 所在组
  3. 其他组
  4. 改变用户所在组

2. 文件/目录所有者

一般为文件的创建者，谁创建了该文件，就自然的称为该文件的所有者。

- 查看文件所有者：ls -ahl （a是all，h是human，l是long）

2) 应用实例：创建一个组 police,再创建一个用户 tom,将 tom 放在 police 组 ,然后使用 tom 来创建一个文件 ok.txt，看看情况如何

```
[root@hadoop1 home]# groupadd police
[root@hadoop1 home]# useradd -g police tom
[root@hadoop1 home]# passwd tom
更改用户 tom 的密码 。
新的 密码：
无效的密码： WAY 过短
无效的密码： 过于简单
重新输入新的 密码：
passwd： 所有的身份验证令牌已经成功更新。
[root@hadoop1 home]#
```

```
[tom@hadoop1 ~]$ pwd
/home/tom
[tom@hadoop1 ~]$ touch ok.txt
[tom@hadoop1 ~]$ ls -ahl
总用量 28K
drwx------. 4 tom  police 4.0K 3月  18 18:51 .
drwxr-xr-x. 9 root root   4.0K 3月  18 18:50 ..
-rw-r--r--. 1 tom  police   18 5月  11 2016 .bash_logout
-rw-r--r--. 1 tom  police  176 5月  11 2016 .bash_profile
-rw-r--r--. 1 tom  police  124 5月  11 2016 .bashrc
drwxr-xr-x. 2 tom  police 4.0K 11月 12 2010 .gnome2
drwxr-xr-x. 4 tom  police 4.0K 3月  17 21:07 .mozilla
-rw-r--r--. 1 tom  police    0 3月  18 18:51 ok.txt
[tom@hadoop1 ~]$
```

这里就是显示ok.txt文件的所有者是tom

tom是所有者，police是所在组

- 修改文件所有者：chown（用户名）（文件名） 是change owner的意思

```
[root@amos home]# touch apple.txt
[root@amos home]# ll
总用量 4
drwx------. 17 admin admin  4096 3月   3 19:35 admin
-rw-r--r--.  1 root  root      0 3月  19 17:28 apple.txt
drwxr-xr-x.  4 root  root     25 3月   6 17:15 laug
drwx------.  5 tom   police  140 3月  19 17:19 tom
[root@amos home]# chown tom apple.txt
[root@amos home]# ll
总用量 4
drwx------. 17 admin admin  4096 3月   3 19:35 admin
-rw-r--r--.  1 tom   root      0 3月  19 17:28 apple.txt
drwxr-xr-x.  4 root  root     25 3月   6 17:15 laug
drwx------.  5 tom   police  140 3月  19 17:19 tom
```

（只改变了所有者，没有改变组信息）

3. 文件/目录所在组

当某个用户创建了一个文件之后，默认这个文件所在组就是该用户所在的组

- 查看文件所有者：ls -ahl
- 修改文件所在组：chgrp（组名）（文件名）

```
[root@amos home]# chgrp police apple.txt
[root@amos home]# ll
总用量 4
drwx------. 17 admin admin  4096 3月   3 19:35 admin
-rw-r--r--.  1 tom   police    0 3月  19 17:28 apple.txt
drwxr-xr-x.  4 root  root     25 3月   6 17:15 laug
drwx------.  5 tom   police  140 3月  19 17:19 tom
```

4. 其他组

除文件的所有者和所在组的用户外，系统的其他用户都是文件的其他组

5. 文件属性

- ls -l中显示的内容如图所示

1. 在Linux中第一个字符代表这个文件是目录、文件或链接文件等等。
   - 当为[d]则是目录
   - 当为[-]则是文件；
   - 若是[l]则表示为**软链接**文档(link file)；
   - 若是[b]则表示为装置文件里面的可供储存的接口设备(可随机存取装置)；
   - 若是[c]则表示为装置文件里面的串行端口设备，例如键盘、鼠标(一次性读取装置)。
2. 接下来的字符中，以三个为一组，且均为『rwx』 的三个参数的组合。[r]代表可读(read)、[w]代表可写(write)、[x]代表可执行(execute)。要注意的是，这三个权限的位置不会改变，如果没有权限，就会出现减号[-]而已。 每个文件的属性由左边第一部分的10个字符来确定（如下图）。

5.1 rwx权限详解(文件和目录的权限)

- rwx作用到文件：
  - r：read，可读。读取查看。
  - w：write，可以修改。但不代表可以删除该文件。删除一个文件的前提条件是对该文件所在的目录有写权限，才能删除该文件。
  - x：execute，可执行。可以被执行。
- rwx作用到目录：
  - r：可以读取，ls查看目录内容。
  - w：可以修改，目录内创建+删除+重命名目录。
  - x：可执行，可以进入该目录。

5.2 修改权限

目录（文件夹）的权限是最大的，如果你连目录都进不去，即使目录下的文件你能访问也是访问不了这个文件的

5.2.1 chgrp：更改文件属组

- chgrp [-R] 属组名 文件名
  - -R：递归更改文件属组，就是在更改某个目录文件的属组时，如果加上-R的参数，那么该目录下的所有文件的属组都会更改。

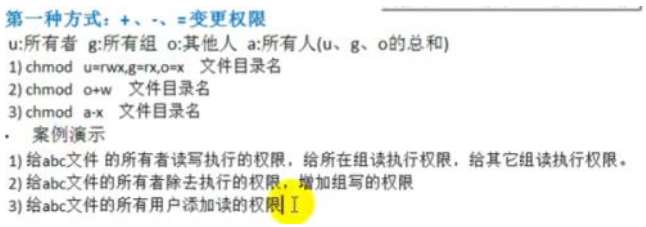
5.2.2 chown：更改文件属主，也可以同时更改文件属组

- chgrp [-R] 属组名 文件名
  - -R：递归更改文件属组，就是在更改某个目录文件的属组时，如果加上-R的参数，那么该目录下的所有文件的属组都会更改。

语法:

- chgrp [-R] 属主名（新） 文件名
- chown [-R] 属主名（新）：属组名（新） 文件名

5.2.3 chmod：更改文件9个属性



- Linux文件属性有两种设置方法，一种是数字，一种是符号。
- Linux文件的基本权限就有九个，分别是owner/group/others三种身份各有自己的read/write/execute权限。
- 我们可以使用数字来代表各个权限，各权限的分数对照表如下：
  - r:4（对文件夹来说可以让别人在进入之后使用ll看有多少内容；对文件来说就是能否阅读（进入）此文件）

    - w:2（对文件夹来说可以增删文件夹中的文件；对文件来说就是可以修改文件内容）
    - x:1（对文件夹来说只有文件夹带x才能让别的用户进入此文件夹，但是ll无权限；对文件来说就是可否执行此文件，只有x无法进入文件）
- 每种身份(owner/group/others)各自的三个权限(r/w/x)分数是需要累加的，例如当权限为：[-rwxrwx---] 分数则是：
    - owner = rwx = 4+2+1 = 7
    - group = rwx = 4+2+1 = 7
    - others= --- = 0+0+0 = 0

5.2.3.1 使用数字修改权限

- chmod [-R] xyz 文件或目录
    - xyz : 就是刚刚提到的数字类型的权限属性，为 rwx 属性数值的相加。
    - -R : 进行递归(recursive)的持续变更，亦即连同次目录下的所有文件都会变更
    - 举例来说，如果要将.bashrc这个文件所有的权限都设定启用，那么命令如下：

```
[root@www ~]# ls -al .bashrc
-rw-r--r--  1 root root 395 Jul  4 11:45 .bashrc
[root@www ~]# chmod 777 .bashrc
[root@www ~]# ls -al .bashrc
-rwxrwxrwx  1 root root 395 Jul  4 11:45 .bashrc
```

5.2.3.2使用符号修改权限

- 我们就可以使用 u(user), g(group), o(others) 来代表三种身份的权限!
- 此外， a 则代表 all，即全部的身份。读写的权限可以写成 r, w, x，也就是可以使用下表的方式来看

chmod	u	+(加入)	r	文件或目录
	g	-(除去)	w	
	o	=(设定)	x	
	a			

- 举例说明

如果我们需要将文件权限设置为 -rwxr-xr--，可以使用 `chmod u=rwx,g=rx,o=r 文件名` 来设定

```
#  touch test1    // 创建 test1 文件
# ls -al test1    // 查看 test1 默认权限
-rw-r--r-- 1 root root 0 Nov 15 10:32 test1
# chmod u=rwx,g=rx,o=r  test1    // 修改 test1 权限
# ls -al test1
-rwxr-xr-- 1 root root 0 Nov 15 10:32 test1
```

而如果是要将权限去掉而不改变其他已存在的权限呢？例如要拿掉全部人的可执行权限，则：

```
#  chmod  a-x test1
# ls -al test1
-rw-r--r-- 1 root root 0 Nov 15 10:32 test1
```

实战练习

police ， bandit

jack, jerry: 警察

xh, xq: 土匪

(1) 创建组

bash> groupadd police

bash> groupadd bandit

(2) 创建用户

(3) jack 创建一个文件，自己可以读写，本组人可以读，其它组没人任何权限

```
[root@hadoop1 kkk]#
[root@hadoop1 kkk]# useradd -g police jack
[root@hadoop1 kkk]# useradd -g police jerry
[root@hadoop1 kkk]# useradd -g bandit xh
[root@hadoop1 kkk]# useradd -g bandit xq
[root@hadoop1 kkk]#
[root@hadoop1 kkk]#
[root@hadoop1 kkk]#
[root@hadoop1 kkk]# passwd jack
更改用户 jack 的密码 。
新的 密码：
无效的密码： WAY 过短
```

(3) jack 创建一个文件，自己可以读写，本组人可以读，其它组没人任何权限

```
[jack@hadoop1 ~]$ chmod 640 jack01.txt
[jack@hadoop1 ~]$ ls -l
总用量 4
-rw-r-----. 1 jack police 6 3月  18 21:20 jack01.txt
[jack@hadoop1 ~]$
```

(4) jack 修改该文件，让其它组人可以读，本组人可以读写

```
[jack@hadoop1 ~]$
[jack@hadoop1 ~]$ chmod o=r,g=rw jack01.txt
[jack@hadoop1 ~]$ ls -l
总用量 4
-rw-rw-r--. 1 jack police 6 3月  18 21:20 jack01.txt
[jack@hadoop1 ~]$
```

给其它组的用户读r
给自己所在组的用户
rw,读写权限

(5) xh 投靠 警察，看看是否可以读写.

先用 root 修改 xh 的组 ：

```
[root@hadoop1 kkk]#
[root@hadoop1 kkk]#
[root@hadoop1 kkk]# usermod -g police xh
[root@hadoop1 kkk]#
```

使用 jack 给他的家目录 /home/jack 的所在组一个 rx 的权限

```
rwx------. 5 zwj   wudang  4096 3月  18 16:3
jack@hadoop1 home]$ chmod g=rx jack/
jack@hadoop1 home]$ ls -l
总用量 92
```

xh 需要重新注销在到 jack 目录就可以操作 jack 的文件

```
[xh@hadoop1 jack]$ ls -l
总用量 4
-rw-rw-r--. 1 jack police 6 3月  18 21:20 jack01.txt
[xh@hadoop1 jack]$ vim jack01.txt
[xh@hadoop1 jack]$
```

xh可以对jack01.txt进行读修