CentOS 丛书目录 — 系统管理 — 网络服务 — 应用部署

文件权限及设置命令

内容提要

1. 理解文件权限及其分配
2. 学会查看文件和目录的权限
3. 掌握权限的文字表示法和数值表示法
4. 学会使用chmod命令设置权限
5. 学会使用chown命令修改属主和组

文件权限

文件权限（File Permission）定义了对某文件或目录的访问限制。 每个文件或目录都有9个基本权限位控制其读、写、执行。基本权限位和另外3个可以影响可执行程序运行的3个特殊权限位一起构成了文件访问模式（File Access Mode）。

三种基本权限

三种基本权限分别为：读、写、执行，说明如下：

代表字符	权限	对文件的含义	对目录的含义
r	读权限	可以读文件的内容	可以列出目录中的文件列表
w	写权限	可以修改、删节文件	可以在该目录中创建、删除文件
x	执行权限	可以执行该文件	可以使用cd命令进入该目录

1. 目录上只有执行权限，表示可以进入或穿越他进入更深层次的子目录
2. 目录上只有执行权限，要访问该目录下的有读权限的文件，必须知道文件名才可以访问
3. 目录上只有执行权限，不能列出目录列表也不能删除该目录
4. 目录上执行权限和读权限的组合，表示可以进入目录并列出目录列表
5. 目录上执行权限和写权限的组合，表示可以在目录中创建、删除和重命名文件

分配三种基本权限

在Linux中，将使用系统资源的人员分为四类：超级用户、文件或目录的属主、属主的同组人和世界上的其他人员。由于超级用户具有操作Linux系统的一切权限，所以不用指定超级用户对文件和目录的访问权限。对于其他三类用户都要指定对文件和目录的访问权限，对每一类用户都需分配三种基本的权限。

- 文件属主的权限：用于限制文件或目录的创建者
- 文件所属组的权限：用于限制文件或目录所属组的成员
- 其他用户的权限：用于限制既不是属主又不是所属组的能访问该文件或目录的其他人员

在 Linux 中通过给三类用户分配三种基本权限，就产生了文件或目录的9个基本权限位。

查看文件和目录的权限

可以使用带l参数的ls命令查看文件或目录的权限，例如：

```
$  ls -l
lrwxrwxrwx   1 osmond osmond      26 2006-05-19 13:40 Examples -> /usr/share/example-content
drwx------   2 osmond osmond      48 2006-07-01 04:14 Mail
-rw-r--r--   1 osmond osmond 9418746 2006-04-05 09:13 webmin-1.270.tar.gz
-rw-r--r--   1 osmond osmond   97582 2005-10-25 21:00 zsync_0.4.2-1_i386.deb
```

每一行显示一个文件或目录的信息，这些信息包括文件的类型（1位）、文件的权限(9位)、文件的属主（第3列）、文件的所属组（第4列），还有文件的大小以及创建时间和文件名。

描述文件权限的9个字母可分成三组，三个字母一组。这三组分别代表：文件属主的权限、文件所属组的权限、其他用户的权限。每组中的三个栏位分别表示了读取权限（r）、写入权限（w）、执行权限（x）或没有相应的权限（-）。

通常将由ls –l命令输出的第一列称为文件或目录的权限字符串。下面列出了几个权限字符串的说明。

字符串	八进制数值	说明
-rw——-	600	只有属主才有读取和写入的权限。
-rw-r--r--	644	只有属主才有读取和写入的权限；同组人和其他人只有读取的权限。
-rwx——	700	只有属主才有读取、写入、和执行的权限。
-rwxr-xr-x	755	属主有读取、写入、和执行的权限；同组人和其他人只有读取和执行的权限。
-rwx--x--x	711	属主有读取、写入、和执行权限；同组人和其他人只有执行权限。
-rw-rw-rw-	666	每个人都能够读取和写入文件。
-rwxrwxrwx	777	每个人都能够读取、写入、和执行。
drwx——	700	只有属主能在目录中读取、写入。
drwxr-xr-x	755	每个人都能够读取目录，但是其中的内容却只能被属主改变。

把权限设为 666 会允许每个人对文件或目录都有读取和写入的权限。把权限设为 777 允许每个人都有读取、写入和执行的权利。这些权限可能会允许对机密文件的篡改，因此，一般来说，使用这类设置是不明智的。

特殊权限位

三个特殊权限位是在可执行程序运行时影响操作权限的。他们分别是SUID位、SGID位和sticky-bit位。

特殊权限	说明
SUID	当一个设置了SUID 位的可执行文件被执行时，该文件将以所有者的身份运行，也就是说无论谁来执行这个文件，他都有文件所有者的特权。任意存取该文件拥有者能使用的全部系统资源。如果所有者是 root 的话，那么执行人就有超级用户的特权了。
SGID	当一个设置了SGID 位的可执行文件运行时，该文件将具有所属组的特权，任意存取整个组所能使用的系统资源；若一个目录设置了SGID，则所有被复制到这个目录下的文件，其所属的组都会被重设为和这个目录一样，除非在复制文件时加上-p（preserve，保留文件属性）的参数，才能保留原来所属的群组设置。
sticky-bit	对一个文件设置了sticky-bit之后，尽管其他用户有写权限，也必须由属主执行删除、移动等操作；对一个目录设置了sticky-bit之后，存放在该目录的文件仅准许其属主执行删除、移动等操作。

由于特殊权限会拥有一些"特权"，因而用户若无特殊需要，不应该去打开这些权限，避免安全方面出现严重漏洞、甚至摧毁系统。

一个设置了SUID的典型例子是passwd程序，它允许普通用户改变自己的口令，这是通过改变/etc/shadow文件的口令字段实现的。然而系统管理员决不允许普通用户拥有直接改变/etc/shadow文件的权利，因为这绝对不是个好主意。解决方法是将passwd程序设置SUID，当passwd程序被执行的时将拥有超级用户的权限，而passwd程序运行结束又回到普通用户的权限。下面显示passwd程序的权限。

```
$  ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 27132 2006-07-11 20:51 /usr/bin/passwd
```

一个设置了sticky-bit的典型例子是系统临时文件目录/tmp，这避免了不守法的用户存心搞鬼，恣意乱删其他用户放置

的文件。下面显示/tmp目录的权限。

```
$  ls -ld /tmp
drwxrwxrwt 4 root root 168 2007-03-17 10:59 /tmp
```

从上面的显示可以看出：SUID是占用属主的x位置来表示；SGID是占用组的x位置来表示；Sticky-bit是占用其他人的x位置来表示的。在表示上有大小写之分，假若同时设置执行权限和SUID、SGID与Sticky，权限标识字符是小写的形式，例如：

```
-rwsr-sr-t     1 root     root       2007-03-17 21:47  showme
```

倘若关闭执行的权限，则标识字符会变成大写，例如：

```
-rwSr-Sr-T     1 root     root       2007-03-17 21:47  showme
```

由于三种特殊权限通常是作用在可执行程序的，所以S和T很少见。

权限设置命令

chmod

系统管理员和文件的属主可以根据需要来更改文件的权限。更改文件和目录的操作权限使用 chmod [http://www.proxyserve.net/index.php?q=aHR0cDovL21hbi5jeC9jaG1vZA%3D%3D] 命令进行，有两种设置方法：文字设定法和数值设定法。

文字设定法

chmod命令的文字设定法格式是：

```
格式1：chmod [-R]{[ugoa][+-=][rwxst]} <文件名或目录名>
格式2：chmod [-R]{[ugoa][+-=][ugo]} <文件名或目录名>
```

说明：

- -R：若操作对象是目录，则递归地对该目录下所有的子目录实施权限设置
- { }内的内容称为一个模式，可以指定多个模式，多个模式之间用逗号间隔
  - 第一个选项表示要赋予权限的用户
    - u 表示属主（user）
    - g 表示所属组用户（group）
    - o 表示其他用户（other）
    - a 表示所有用户（all）
  - 第二个选项表示要进行的操作
    - + 表示增加权限
    - - 表示删除权限
    - = 表示分配权限，同时将原有权限删除
  - 第三个选项是要分配的权限
    - r 表示允许读取
    - w 表示允许写入

    - x 表示允许执行
    - s 表示设置 SUID 或 SGID
    - t 表示设置 sticky-bit
- 两种格式的不同
    - 格式1的用法是对文件或目录设置r、w、x、s、t等权限
    - 格式2的用法是参考当前u、g、o的权限设置其它用户类的权限

下面给出几个使用格式1的例子：

```
$  touch testfile1
$  ll
-rw-r--r-- 1 osmond osmond 0 2007-03-17 15:27 testfile1
# 取消组用户和其他用户对文件的读取权限
$  chmod go-r testfile1
$  ll
-rw------- 1 osmond osmond 0 2007-03-17 15:27 testfile1
# 对文件的属主添加执行权限
$  chmod u+x testfile1
$  ll
-rwx------ 1 osmond osmond 0 2007-03-17 15:27 testfile1
# 对文件的属主取消执行权限同时添加组用户和其他用户对文件的读取权限
$  chmod u-x,go+r testfile1
$  ll
-rw-r--r-- 1 osmond osmond 0 2007-03-17 15:27 testfile1
# 对文件添加SUID 和 SGID设置的同时添加执行权限
$  chmod ug+xs testfile1
$  ll
-rwsr-sr-- 1 osmond osmond 0 2007-03-17 15:27 testfile1
# 对文件添加 sticky-bit 设置的同时添加执行权限
$  chmod +xt testfile1
$  ll
-rwsr-sr-t 1 osmond osmond 0 2007-03-17 15:27 testfile1
```

下面给出几个使用格式2的例子：

```
$  touch testfile1
$  ll
-rw-r--r-- 1 osmond osmond 0 2007-03-17 15:00 testfile1
# 将组权限设置为与属主相同
$  chmod g=u testfile1
$  ll
-rw-rw-r-- 1 osmond osmond 0 2007-03-17 15:00 testfile1
# 对其他人添加属主具有的权限
$  chmod o+u testfile1
$  ll
-rw-rw-rw- 1 osmond osmond 0 2007-03-17 15:00 testfile1
# 对其他人取消属主具有的权限
$  chmod o-u testfile1
$  ll
-rw-rw---- 1 osmond osmond 0 2007-03-17 15:00 testfile1
```

数值设定法

chmod 命令的数值设定法格式是：

```
chmod [-R] [n0]n1n2n3 <文件名或目录名>
```

说明：

- -R：若操作对象是目录，则递归地对该目录下所有的子目录实施权限设置
- 其中n1代表属主的权限，n2代表组用户的权限，n3代表其他用户的权限，这三个选项都是8进制数字

权限			数值表示		说明
读	写	执行	二进制	八进制	
-	-	-	000	0	没有权限
-	-	x	001	1	允许执行
-	w	-	010	2	允许写入
-	w	x	011	3	允许执行和写入
r	-	-	100	4	允许读取
r	-	x	101	5	允许执行和读取
r	w	-	110	6	允许写入和读取
r	w	x	111	7	允许执行写入和读取

- 其中 n0 是设置特殊权限的8进制数字，当不设置特殊权限时 n0 可以省略。

权限			数值表示		说明
SUID	SGID	sticky	二进制	八进制	
-	-	-	000	0	不设置特殊权限
-	-	t	001	1	只设置sticky
-	s	-	010	2	只设置SGID
-	s	t	011	3	只设置SGID和sticky
s	-	-	100	4	只设置SUID
s	-	t	101	5	只设置SUID和sticky
s	s	-	110	6	只设置SUID和SGID
s	s	t	111	7	同时设置三种特殊权限

下面给出几个使用数值设置法的例子：

```
$  touch testfile2
$  ll
-rw-r--r-- 1 osmond osmond 0 2007-03-17 16:25 testfile2
# 对文件的属主设置可读、写和执行权限，所属组用户和其他用户只设置读和执行的权限，没有写的权限。
$  chmod 755 testfile2
$  ll
-rwxr-xr-x 1 osmond osmond 0 2007-03-17 16:25 testfile2
# 取消组用户和其他用户对文件users1的一切权限
$  chmod 600 testfile2
$  ll
-rw------- 1 osmond osmond 0 2007-03-17 16:25 testfile2
# 为文件设置SUID和属主的执行权限
$  chmod 4700 testfile2
$  ll
-rws------ 1 osmond osmond 0 2007-03-17 16:25 testfile2
# 同时设置SUID和SGID权限，并使组用户能读、写、执行，其他人能读和执行
$  chmod 6775 testfile2
$  ll
-rwsrwsr-x 1 osmond osmond 0 2007-03-17 16:25 testfile2
# 设置 sticky权限
```

```
$  chmod 1755 testfile2
$  ll
-rwxr-xr-t 1 osmond osmond 0 2007-03-17 16:25 testfile2
```

chown

改变文件的属主和组可以用chown命令，命令格式是：

```
chown [-R] 〈用户[:组]〉〈文件或目录〉
```

说明：

- -R：若操作对象是目录，则递归地对该目录下所有的子目录实施设置
- 要单独改变组，可以使用下面的格式：（注意：组前必须有":"）

chown [-R] <:组> <文件或目录>

例如：

```
$  touch  testfile3
$  ll
-rw-rw-r-- 1 osmond osmond 0 Dec 14 15:19 testfile3
# 切换为超级用户
$  su -
# 将文件testfile3的属主改成jason
# chown jason /home/osmond/testfile3
$  ll /home/osmond/testfile3
-rw-rw-r-- 1 jason osmond 0 Dec 14 15:19 /home/osmond/testfile3
# 将文件testfile3的组改成 users
# chown :users /home/osmond/testfile3
# ll /home/osmond/testfile3
-rw-rw-r-- 1 jason users 0 Dec 14 15:19 /home/osmond/testfile3
# 将文件testfile3的属主和组改成 osmond
# chown osmond:osmond /home/osmond/testfile3
# ll /home/osmond/testfile3
-rw-rw-r-- 1 osmond osmond 0 Dec 14 15:19 /home/osmond/testfile3
# 退出 root 的登录
# exit
$  mkdir -p testdir/dir1
$  ll
drwxrwxr-x 3 osmond osmond 4096 Dec 14 15:25 testdir
-rw-rw-r-- 1 osmond osmond    0 Dec 14 15:19 testfile3
# 切换为超级用户
$  su -
# 将 testdir 目录及其子目录下的所有文件或目录的属主和组都改成 apache
# chown -R apache:apache /home/osmond/testdir
# tree -ug /home/osmond
/home/osmond/
|-- [apache   apache  ]  testdir
|   `-- [apache   apache  ]  dir1
`-- [osmond   osmond  ]  testfile3
```

umask

用户可以使用umask命令设置文件的缺省生成掩码。缺省的生成掩码告诉系统当创建一个文件或目录时不应该赋予其哪些权限。如果用户将 umask 命令放在环境文件（.bash_profile）中，就可以控制所有新建的文件或目录的访问权限。

umask命令的格式为

```
umask [u1u2u3]
```

其中：u1表示的是不允许属主有的权限；u2表示的是不允许同组人有的权限；u3表示的是不允许其他人有的权限。

可以使用不带任何参数或带 -S 参数的umask命令查看当前的文件缺省生成掩码:

```
$  umask
0022
$  umask -S
u=rwx,g=rx,o=rx
```

例如：设置允许同组用户有写权限可以如下设置

```
$  umask 002
$  touch testfile4
$  ll testfile4
-rw-rw-r-- 1 osmond osmond 0 2007-03-17 17:15 testfile4
$  mkdir testdir1
$  ll -d testdir1
drwxrwxr-x 2 osmond osmond 48 2007-03-17 17:20 testdir1
```

这与默认的不允许允许同组用户有写权限不同

```
$  umask 022
$  touch testfile5
$  ll testfile5
-rw-r--r-- 1 osmond osmond 0 2007-03-17 17:16 testfile5
$  mkdir testdir2
$  ll -d testdir2
drwxr-xr-x 2 osmond osmond 48 2007-03-17 17:18 testdir2
```

- 显示源文件
- 登录