

Linux 文件基本属性

Linux系统是一种典型的多用户系统，不同的用户处于不同的地位，拥有不同的权限。为了保护系统的安全性，Linux系统对不同的用户访问同一文件（包括目录文件）的权限做了不同的规定。

在Linux中我们可以使用ll或者ls -l命令来显示一个文件的属性以及文件所属的用户和组，如：

```
[root@www /]# ls -l
total 64
dr-xr-xr-x  2 root root 4096 Dec 14  2012 bin
dr-xr-xr-x  4 root root 4096 Apr 19  2012 boot
.....
```

实例中，bin文件的第一个属性用"d"表示。"d"在Linux中代表该文件是一个目录文件。

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)。

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。其中，[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）。

文件 类型	属主 权限			属组 权限			其他用户 权限		
0	1	2	3	4	5	6	7	8	9
d	rwX			r-X			r-X		
目录 文件	读	写	执行	读	写	执行	读	写	执行

从左至右用0-9这些数字来表示。

第0位确定文件类型，第1-3位确定属主（该文件的所有者）拥有该文件的权限。

第4-6位确定属组（所有者的同组用户）拥有该文件的权限，第7-9位确定其他用户拥有该文件的权限。

其中，第1、4、7位表示读权限，如果用"r"字符表示，则有读权限，如果用 "-" 字符表示，则没有读权限；

第2、5、8位表示写权限，如果用"w"字符表示，则有写权限，如果用 "-" 字符表示没有写权限；第3、6、9位表示可执行权限，如果用"x"字符表示，则有执行权限，如果用 "-" 字符表示，则没有执行权限。

Linux文件属主和属组

```
[root@www /]# ls -l
total 64
drwxr-xr-x 2 root  root  4096 Feb 15 14:46 cron
drwxr-xr-x 3 mysql mysql 4096 Apr 21  2014 mysql
.....
```

对于文件来说，它都有一个特定的所有者，也就是对该文件具有所有权的用户。

同时，在Linux系统中，用户是按组分类的，一个用户属于一个或多个组。

文件所有者以外的用户又可以分为文件所有者的同组用户和其他用户。

因此，Linux系统按文件所有者、文件所有者同组用户和其他用户来规定了不同的文件访问权限。

在以上实例中，mysql 文件是一个目录文件，属主和属组都为 mysql，属主有可读、可写、可执行的权限；与属主同组的其他用户有可读和可执行的权限；其他用户也有可读和可执行的权限。

对于 root 用户来说，一般情况下，文件的权限对其不起作用。

更改文件属性

1、chgrp：更改文件属组

语法：

```
chgrp [-R] 属组名 文件名
```

参数选项

- -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。

2、chown：更改文件属主，也可以同时更改文件属组

语法：

```
chown [-R] 属主名 文件名
chown [-R] 属主名:属组名 文件名
```

进入 /root 目录 (~) 将install.log的拥有者改为bin这个账号：

```
[root@www ~] cd ~
[root@www ~]# chown bin install.log
[root@www ~]# ls -l
-rw-r--r-- 1 bin  users 68495 Jun 25 08:53 install.log
```

将install.log的拥有者与群组改回为root：

```
[root@www ~]# chown root:root install.log
[root@www ~]# ls -l
-rw-r--r-- 1 root root 68495 Jun 25 08:53 install.log
```

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。

先复习一下刚刚上面提到的数据：文件的权限字符为：『-rwxrwxrwx』，这九个权限是三个三个一组的！其中，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
- w:2
- x:1

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

所以等一下我们设定权限的变更时，该文件的权限数字就是770啦！变更权限的指令chmod的语法是这样的：

```
chmod [-R] xyz 文件或目录
```

选项与参数：

- 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
```

那如果要将权限变成 -rwxr-xr-- 呢？那么权限的分数就成为 [4+2+1][4+0+1][4+0+0]=754。

符号类型改变文件权限

还有一个改变权限的方法，从之前的介绍中我们可以发现，基本上就九个权限分别是：

- (1)user

- (2)group
- (3)others

那么我们就可以使用 **u, g, o** 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			

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
```

[← Linux 远程登录](#)

Linux 文件与目录管理 [→](#)

 **1 篇笔记**

 **写笔记**

 每个文件的属性由左边第一部分的10个字符（如上的“dr-xr-xr-x”）来确定。我们把十个字符拆开看：

- 10位字符表示：
- 0位：确定文件类型
- 1-3位：确定该文件的所有者对文件的权限 **owner**
- 4-6位：确定所有者的同组用户拥有该文件的权限 **group**
- 7-9位：确定其他用户拥有该文件的权限 **others**
- 第一个字符：代表这个文件的类型，是目录、文件，还是一个链接等等
- [d] 目录
- [-] 文件

- ● **[l]** 链接文档(link file)
- ● **[b]** 可供储存的接口设备(可随机存取装置)
- ● **[c]** 串行端口设备,例如键盘、鼠标(一次性读取装置)
- ● **接下来的字符**: 以三个一组分成三组,用 r、w、x 三个参数的组合表示,位置不会改变
- ● **[r]** 代表可读(read)
- ● **[w]** 代表可写(write)
- ● **[x]** 代表可执行(execute)
- ● **[-]** 没有权限

blockpunk 3个月前 (12-10)