

权限管理

细节

一个文件主要包含下列属性，`ls -l`

```
- rwx①rwx②rwx③ user④ group⑤ date⑥ filename⑦ 111 101 101
```

- ❶ 归属用户的权限
- ❷ 归属群组的权限
- ❸ 其它用户的权限
- ❹ 归属用户
- ❺ 归属群组
- ❻ 日期信息
- ❼ 文件名称

- 对于文件夹，必须拥有它的可执行权限，才能够使用 `cd` 命令进入该文件夹；拥有可读权限，才能够使用 `ls` 命令查看该文件夹的文件列表。
- `root` 用户 拥有最高权限

可以使用 3 位的二进制数字 来描述一组权限，某一权限对应的数字为 1,则表示具有该种权限，为 0,则不具有该种权限。

使用二进制数字来描述一组权限，虽然非常直观，但是 3 组 权限需要用 9 位 数来表示，不够方便。因此我们将三组权限使用 3 位 8 进制数字 来表示。

每种权限对应的数字：

权限	r	w	x
二进制	100	010	001
八进制	4	2	1

将这 3 位 8 进制数字 相加的结果，就可以表示该组权限的具体内容，例如：

$7 = 4 + 2 + 1 = \text{rwx}$

$5 = 4 + 1 = \text{rx}$

$755 = 4 + 2 + 1, 4 + 1, 4 + 1 = \text{rwx, r-x, r-x}$

还可以使用 `a`、`u`、`g`、`o` 表示归属关系，使用 `=`、`+`、`-` 表示权限变化，使用 `r`、`w`、`x` 表示权限内容

`a` 所有用户 `u` 归属用户 `g` 归属群组 `o` 其它用户

`=` 具有权限 `+` 增加权限 `-` 去除权限

`r` 可读权限 `w` 可写权限 `x` 可执行权限

例如：

```
a+x 给所有用户增加可执行权限go-wx 将归属群组和其它用户的可写、可执行权限去掉u=rwx 归属用户具有可读、可写、可执行权限
```

chmod 权限表达式 文件|目录

更改文件的权限。权限的表达式可以使用 3 位 8 进制数字 表示，或者使用 `augo +-= rxw-s` 来表示

<code>-R</code>	递归
<code>-V</code>	显示过程
<code>-C</code>	类似“-v”，仅显示更改部分
<code>--reference=</code> 参照文件或目录	以指定文件为参照更改权限

示例：

```
chmod -R a+x pathchmod -Rv 755 path
```

chown 归属用户[:归属群组] 文件|目录

更改文件的归属用户。可以使用用户名或者用户编号

-R	递归
-v	显示过程
-c	类似“-v”，仅显示更改部分
--reference=参照文件或目录	以指定文件为参照更改权限

示例：

```
chown user:admin pathchown -R user.admin pathchown user path
```

chgrp 归属群组 文件|目录

更改文件的归属群组。可以使用群组名或者群组编号,选项同上

SUID、SGID、Sticky bit

某些情况下，需要以可执行文件归属用户的身份执行该文件，可以为该文件设置 SUID。同样，设置 SGID 能够以该文件归属群组的身份执行它。

例如：用户自行设定密码。出于安全方面的考虑， /etc/shadow 只能由 root用户 直接修改。

```
-rw----- root root /etc/shadow
```

这个时候，可以为程序 /usr/bin/passwd 设置 SUID，当普通用户执行“passwd”命令时，便能够以该程序归属用户 root 的身份修改 /etc/shadow 文件。而“passwd”程序自身带有身份验证机制，不能通过验证时拒绝执行，从而保证了安全。

```
ls -l /usr/bin/passwd-r-s--x--x root root /usr/bin/passwd
```

我们发现，归属用户的可执行权限位使用 **s**，表示 SUID。同样，归属群组的可执行权限位使用 **s**，表示 SGID。任何用户或群组都拥有“其它用户”的权限，所以不需要以 其它用户 身份执行文件，其它用户的可执行权限位便不会出现 **s**。该权限位可能出现的属性为 **t**，也就是粘着位 Sticky bit。

```
ls -ld /tmpdrwxrwxrwt root root /tmp
```

粘着位表示任何用户都可能具有写权限，但只有该归属用户或 root用户 才能够删除

SUID、SGID、Sticky bit 也可以像权限一样，使用一个八进制数表示，如下：

4	SUID
2	SGID
1	Sticky bit

通过在“chmod”命令中使用 4个八进制数 的表达式，如 4755 ，用第一位表示 SUID、SGID 或 Sticky bit，便能够为文件设置这些特殊权限。示例：

```
chmod -R 4755 path
```

lsattr [路径]

查看文件的特殊属性

-a	全部显示
-d	只显示目录

-R	递归
----	----

特殊属性包括：

a	仅供附加用途
b	不更新最后存取时间
c	压缩后存放
d	排除在转储操作之外
i	不得任意更动文件或目录
s	保密性删除文件或目录
S	即时更新文件或目录
u	预防意外删除

chattr +|-|=属性 路径

更改文件特殊属性

-R	递归
-V	显示过程