权限的基本概念

Linux下一切皆文件，不同的用户对文件拥有不同的权限。在多用户计算机系统的管理中，权限是指某个特定的用户具有特定的系统资源使用权利。

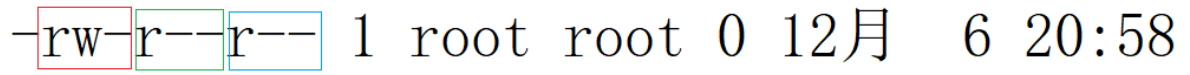
在Linux 中分别有读、写、执行权限：

	权限针对文件	权限针对目录
读r	表示可以查看文件内容	表示可以查看目录中存在的文件名称
写 w	表示可以更改文件的内容	表示是否可以删除目录中的子文件或者新建子目录(
执行 x	表示是否可以开启文件当中记录的程序,一般指二进制文件(.sh)	表示是否可以进入目录中

查看文件权限

```
root@pc:~# touch 1.txt
root@pc:~# ls -l 1.txt
-rw-r--r-- 1 root root 0 12月  6 20:58 1.txt
```

Linux中，不同用户角色创建文件默认权限不同，root用户创建文件默认权限如上图所示。



-代表文件类型是一个普通文件

红框代表文件拥有者user

绿框代表文件所属组group组内其他用户

蓝框代表其他用户other

如果是目录，文件类型为d

```
hioier@yunpc:~$ ls -l
total 4
drwxr-xr-x 2 hioier blackcat 4096 Dec  6 20:40 AA
```

文件类型

Linux一共有7种文件类型,分别如下:

-：普通文件

d：目录文件

l： 软链接（类似Windows的快捷方式）

b：block，块设备文件（例如硬盘、光驱等）

p：管道文件

c：字符设备文件

s：套接口文件/数据接口文件

文件权限设置-字母

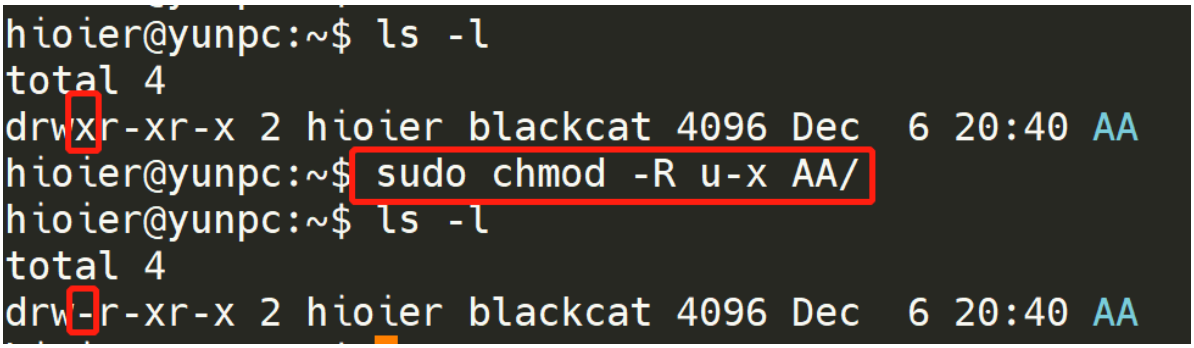
```
chmod [选项] 权限设置 文件或目录的名称

选项说明：
-R ：递归设置，针对文件夹（目录）
```

权限设置：

1：确认要给哪个身份设置权限，u、g、o、ugo(a)

2：确认是添加权限(+)、删除权限(-)还是赋予权限(=)

3：确认给这个用户针对这个文件或文件夹设置什么样的权限，r、w、x




```
sudo chmod -R ugo=rwx AA/
sudo chmod -R a=rwx AA/
```

文件权限设置-数字

权限	对应数字	意义
r	4	可读
w	2	可写
x	1	可执行
-	0	没有权限

```
chmod 777 1.txt
```

```
root@pc:~# ls -l 1.txt
-rw-r--r-- 1 root root 0 12月   6 20:58 1.txt
root@pc:~# chmod 777 1.txt
root@pc:~# ls -l 1.txt
-rwxrwxrwx 1 root root 0 12月   6 20:58 1.txt
```

文件拥有者和所属组设置

拥有者设置

```
chown [选项] 新文件拥有者名称 文件名称
选项说明：
-R ：代表递归修改，主要针对文件夹
chown blackcat 1.txt
```

```
root@pc:/home/hioier# ls -l
total 12
-rw-rw-r-- 1 hioier hioier    0 12月   6 21:29 1.txt
drwxr-xr-x 2 hioier hioier 4096 11月  26 09:48 Desktop
drwxr-xr-x 2 hioier hioier 4096 11月  26 09:48 Documents
drwxr-xr-x 2 hioier hioier 4096 11月  26 09:48 Downloads
root@pc:/home/hioier# chown blackcat 1.txt
root@pc:/home/hioier# ls -l
total 12
-rw-rw-r-- 1 blackcat hioier    0 12月   6 21:29 1.txt
drwxr-xr-x 2 hioier   hioier 4096 11月  26 09:48 Desktop
drwxr-xr-x 2 hioier   hioier 4096 11月  26 09:48 Documents
drwxr-xr-x 2 hioier   hioier 4096 11月  26 09:48 Downloads
root@pc:/home/hioier#
```

所属组设置

```
chgrp [选项] 新文件所属组名称 文件名称
选项说明：
-R ：代表递归修改，主要针对文件夹
```

```
root@pc:/home/hioier# chgrp blackcat 1.txt
root@pc:/home/hioier# ls -l
total 12
-rw-rw-r-- 1 blackcat blackcat    0 12月   6 21:29 1.txt
drwxr-xr-x 2 hioier   hioier   4096 11月  26 09:48 Desktop
drwxr-xr-x 2 hioier   hioier   4096 11月  26 09:48 Documents
drwxr-xr-x 2 hioier   hioier   4096 11月  26 09:48 Downloads
```

同时修改拥有者和所属组

```
chown [选项] 文件拥有者名称:文件所属组名称 文件名称
或
chown [选项] 文件拥有者名称.文件所属组名称 文件名称
选项说明:
-R : 代表递归修改，主要针对文件夹
```

```
root@pc:/home/hioier# chown hioier.hioier 1.txt
root@pc:/home/hioier# ls -l
total 12
-rw-rw-r-- 1 hioier hioier    0 12月   6 21:29 1.txt
drwxr-xr-x 2 hioier hioier 4096 11月  26 09:48 Desktop
drwxr-xr-x 2 hioier hioier 4096 11月  26 09:48 Documents
drwxr-xr-x 2 hioier hioier 4096 11月  26 09:48 Downloads
```

特殊权限

冒险位SETUID（针对二进制文件）

作用：为了让一般使用者临时具有该文件所属主/组的执行权限。

例如：/usr/bin/passwd在执行它的时候需要去修改/etc/passwd和/etc/shadow等文件，这些文件除了root外，其他用户都没有写权限，但是又为了能让普通用户修改自己的密码，那么该如何操作?

去除S位权限

```
chmod u-s /usr/bin/passwd
或者
chmod 0755 /usr/bin/passwd
```

添加S位权限

```
chmod u+s /usr/bin/passwd
或者
chmod 4755 /usr/bin/passwd
```

强制位SETGID（针对目录）

作用：如果一个目录有强制位，那么任何用户在该目录里所创建的文件属组都会继承该目录的属组。

去除S位权限

```
chmod g-s 目录名
```

添加S位权限

```
chmod g+s 目录名
或
chmod 2xxx 目录名
```

粘附位T（针对目录）

作用：只允许文件的创建者和root用户删除文件（防止误删除权限位）

去除粘附位

```
chmod -R o-t /share
或
chmod -R 0777 /share
```

添加粘附位

```
chmod -R o+t /share
或
chmod -R 1777 /share
```

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

umask

umask表示创建文件时的默认权限（即创建文件时不需要设置而天生的权限）

我们创建一个普通文件最高权限666，而创建一个文件夹其最高权限777。

实际文件权限 = 最高权限 - umask的值

获取用户umask值

```
umask

0022

注：0022中第一位0代表特殊权限位，可以不设置。
umask的默认值，在root和普通用户下是不一样的，分别是022和002
```

修改umask值（一般不要更改）

临时修改

```
umask 002

777 - 002 = 775
```

永久修改

```
vim ~/.bashrc
1：在文件末尾添加umask 002
2：保存退出
3：新开终端生效
```

ACL权限

ACL，是 Access Control List（访问控制列表）的缩写，在 Linux 系统中， ACL 可实现对单一用户或者某个组设定访问文件的权限，ACL优势就是让权限控制更加的精准。

安装acl

```
apt install acl
```

获取ACL权限

```
getfacl 文件或目录名称
```

设置ACL权限

```
setfacl [选项] 文件或目录名称

选项说明：
-m ： 修改acl策略
-x ： 去掉某个用户或者某个组的权限
-b ： 删除所有的acl策略
-d ： 该目录下新建的文件和目录都会继承acl策略，但已存在的没有
-R ： 该目录下已存在的文件和目录都会继承acl策略，但新建的没有
mask ：指的是用户或群组能拥有的最大ACL权限，也就是说，给用户或群组设定的ACL权限不能超过mask规定的权限范围，超出部分做无效处理。
```

```
示例-给用户增加acl权限：
setfacl -m u:hioier:rw 1.txt

示例-给用户删除acl权限：
setfacl -x u:hioier 1.txt
```

```
root@yunpc:/home/hioier# setfacl -m u:hioier:rw 1.txt
root@yunpc:/home/hioier# getfacl 1.txt
# file: 1.txt
# owner: root
# group: root
user::rw-
user:hioier:rw-
group::r--
mask::rw-
other::r--
```

```
root@yunpc:/home/hioier# setfacl -x u:hioier 1.txt
root@yunpc:/home/hioier# getfacl 1.txt
# file: 1.txt
# owner: root
# group: root
user::rw-
group::r--
mask::r--
other::r--
```

```
示例-给用户组增加acl权限：
setfacl -m g:blackcat:rw 2.txt

示例-给用户组删除acl权限：
setfacl -x g:blackcat 2.txt
```

```
root@yunpc:/home/hioier# setfacl -m g:blackcat:rw 2.txt
root@yunpc:/home/hioier# getfacl 2.txt
# file: 2.txt
# owner: root
# group: root
user::rw-
group::r--
group:blackcat:rw-
mask::rw-
other::r--
```

```
root@yunpc:/home/hioier# setfacl -x g:blackcat 2.txt
root@yunpc:/home/hioier# getfacl 2.txt
# file: 2.txt
# owner: root
# group: root
user::rw-
group::r--
mask::r--
other::r--
```

```
示例-删除所有权限：
setfacl -b 1.txt
```

mask权限设置：

```
setfacl -m m::r 1.txt
```

```
root@yunpc:/home/hioier# setfacl -m m::r 1.txt
root@yunpc:/home/hioier# getfacl 1.txt
# file: 1.txt
# owner: root
# group: root
user::rw-
user:hioier:rw-                 #effective:r--
group::r--
mask::r--
other::r--
```

目录递归授权

```
root@yunpc:/home/hioier# setfacl -Rm u:hioier:rwx AA/
root@yunpc:/home/hioier# getfacl AA/
# file: AA/
# owner: root
# group: root
user::rwx
user:hioier:rwx
group::r-x
mask::rwx
other::r-x
```

```
root@yunpc:/home/hioier# setfacl -dm u:hioier:rwx BB/
root@yunpc:/home/hioier# getfacl BB
# file: BB
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:user:hioier:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```