Linux文件权限

> 在Linux中，文件的权限可以针对用户、用户组、其他用户分别设置权限，权限包括`r` `w` `x`，分别为读、写、执行。除此之外，还可以对文件设置特殊权限`set-uid`、`group-uid`或`sticky`。

特殊权限

> SUID、SGID、Sticky三种权限组合对应三位二进制数，一个十进制数。SUID为`100=4`,SGID为`010=2`,sticky为`001=1`。例如某个文件的权限为5755，则这个文件具有SUID和Sticky特使权限。设置特殊权限时，也可以用数字进行设置。

SUID

> **理解**

> 通常对于一个`二进制程序`来说，当它被用户执行并成为一个进程后，这个进程的`属主`都将变成该用户。如果这个二进制文件具有SUID权限`s`并被其他用户执行成为一个进程后，这个进程的属主还是原文件的属主，具有原文件属主的权限。即当一个具有特殊权限SUID的可执行文件被执行时，相应进程的属主是程序自身的属主，而不启动者。
>
> 例如`/bin/passwd`就是具有特殊权限的二进制程序，因为这个二进制程序的属主是root用户，所以任意普通用户在执行这个二进制程序时的身份是root，才得以有权限修改密码并写入/etc/shadow文件。

> **SUID特点**

> 1. SUID仅对二进制文件有效
> 2. 特殊权限仅在执行过程中有效
> 3. 执行者需要对该二进制文件有执行权限

4. 不能使用在Shell脚本文件中，因为Shell脚本仅是由调用外部二进制程序的命令组成的文本文件

用法

```
# 添加SUID权限
chmod u+s BINFILE
# 删除SUID权限
chmod u-s BINFILE
```

SGID

理解

和SUID类似，一个二进制程序具有SGID特殊权限后，当他被其他用户执行时，这个进程具有二进制程序属组的权限。SGID特殊权限可以作用于目录，当一个目录具有SGID权限时，且`其他用户具有对这个目录的写权限`，则任何用户在该目录下创建的文件的属组都将`继承这个目录的属组`。

SGID特点

1. SUID对二进制文件有效
2. SUID可以作用于目录

用法

```
# 添加SGID权限
chmod g+s BINFILE|DIRFILE
# 删除SGID权限
chmod g-s BINFILE|DIRFILE
```

SGID权限作用于目录示例

```
groupadd -g 1024 g1
mkdir -pv /testdir/dir
chgrp g1 dir    # 修改目录属组为g1
chmod g+s dir   # 添加目录的写权限和s权限
chmod go+w dir  # 给目录属组和其他人添加写权限
su - user1
# 测试在/testdir/dir目录下创建文件
touch file0
# 新创建文件的属组是dir目的属组g1-user1
-rw-r--r--. 1 user1 g1 0 Jul 16 10:34 file0
```

sticky

> **理解**

> sticky权限作用于目录。在一个公共目录，对目录拥有写和执行的每个用户都可以创建文件，删除自己的文件，但不能删除别人的文件。
>
> 例如系统目录`/tmp`目录就是由sticky权限的目录，每个用户都可以在此目录下创建删除修改自己的文件，`尽管用户对tmp目录具有w权限`，但不能删除其他用户的文件。

> **用法**

```
# 添加sticky
chmod o+t DIRFILE
# 删除sticky
chmod o-t DIRFILE
```

> **示例**

```
# 给dir目录添加sticky权限
useradd stuser1
useradd stuser2
useradd stuser3
chmod o+tw dir
# 分别切换为stuser1、2、3用户测试创建文件、删除其他用户的文件
```

其他特殊权限

> *注意：chattr指令只能在Ext2/Ext3/Ext4文件系统上完全生效，在xfs文件系统有部分属性生效。*

> **chattr**

```
# 命令格式
chattr [ -OPTION ] [mode] files...
# 选项
-R  # 递归修改
# 设置属性的符号
+|-|=
# 属性
a # 允许给文件进行追加操作
A # 不允许更新文件的访问时间
c # 文件在磁盘上自动压缩
D # 文件同步保存到磁盘
i # 启用后，不能更改、重命名、或者删除
```

> **lsattr**

```
# 查看文件的属性
lasttr FILENAME
```

访问控制列表

> 访问控制列表(Access Control List)，ACL可以具体针对某一个用户、某一个组、某一个文件进行细分的权限设置。ACL

> 的使用需要文件系统的支持。

> **查看文件系统是否支持ACL**

```
tune2fs -l /dev/sdaN
```

> **setfacl**

```
# 格式
setfacl [-bkndRLPvh] [{-m|-x} acl_spec] [{-M|-X} acl_file] file...
setfacl --restore=file
# 选项
-m      # 设定acl权限
-x      # 删除acl权限
-b      # 彻底删除所有acl权限
-k      # 移除默认acl权限
-R      # 递归设置
-M      # 从文件中读取acl配置，-M acl.txt FILENAME
# 特殊用法
#  默认在data目录下创建的文件也具有和data一样的acl权限
setfacl -m d:xuejw0:rwx data
# 复制文件1的acl权限给文件2
getfacl FILENAME1 | setfacl --set-file=- FILENAME2
```

```
# 查看文件的acl权限信息
getfacl FILENAME
# 备份acl
getfacl FILENAME > acl.tct
```

> *注意：复制文件时，不会将源文件的acl属性复制。cp命令加-p选项可以连同acl属性一同复制。*

> **facl中的mask**

> mask权限值是权限定点，所有其他权限不能高于mask的值。

```
# 设定文件的mask权限
getfacl -m mask::rw FILENAME
```

> *注意：如果设定mask权限之后，再次设定facl权限，可能会将mask的权限调大。*

chown|chgrp|chmod

chown|chgrp

```
chown
选项：
    -R  # 递归操作目录下的所有文件
    --reference=/path/filename  #  取filename的权限对另一个文件设置
格式：
    chown OPTION USER:GROUP FILE
    chown OPTION USER.GROUP FILE
    chown OPTION USER FILE
示例：
    chown user1: file0      # 将file0的属主改为user1，属组改为user1的属组
    chown :user1 file0      # 将file0的属组改为user1组
```

```
chgrp
# 普通用户可以将属组改为此用户所属于的组时有权限执行
选项：
-R      # 递归对文件和目录进行操作
```

chmod

```
1、修改某类用户的权限：u，g，o，a(a表示所有用户)
chmod u=rwx  file1
chmod u=rw file1
chmod o=rx file1
chmod o=rwx,u=rwx file1
chmod  uo=rwx  file1
2、修改某类用户的某些位权限
chmod  u-wx file,......
chmod  u+x,g-x file,......
chmod  -x file,......          #将三类用户的执行权限都去掉
3、权限位X
# 一般配合 -R 选项使用，对目录下子目录加x执行权限，但不对文件加执行权限，当文件本来就有执行权限时，X 权限位会给文件加上执行权限
# 还可以使用数字法对权限进行修改
```

关于权限应注意的问题

> 1. 对于文件的删除权限，取决于用户对其父目录写权限。
> 2. 对目录而言，有执行权限才能进入该目录。
> 3. 对于目录，如果没有读权限，则无法看到目录内的文件列表，但可以查看目录下的文件的内容。
> 4. 目录的权限是一个基本权限。

umask

> umask是控制默认新建文件和目录的默认权限分配的。root的默认umask是022,普通用户的默认umask是002。在Linux中，新创建的文件默认是没有执行权限，因此新建文件的权限是666，新创建目录的权限的777，尽管新创建文件没有执行权限，但依然对于系统的安全来说是一个非常大的隐患。这时候就需要umask(遮罩码)将一些权限去掉。

> **umask计算**

```
# 把二进制格式的默认最大权限与umask相对应的位上，umask位上为1对应的位上去掉换成0位
0666   000 110 110 110 # 文件默认最大权限
0022   000 000 010 010 # umask值
0644   000 110 100 100 # 默认权限
```

> 对于目录：777 - umask = 目录默认权限
> 对于文件：666 - umask = 文件默认权限 (然后对结果奇数加一，偶数不变)

> umask命令

```
umask -S     # 以模式化，而不以数字化显示
umask -p     # 执行结果可被调用
umask XXXX   # 设置umask，但命令设置是临时生效，可以配置到环境配置文件中
```