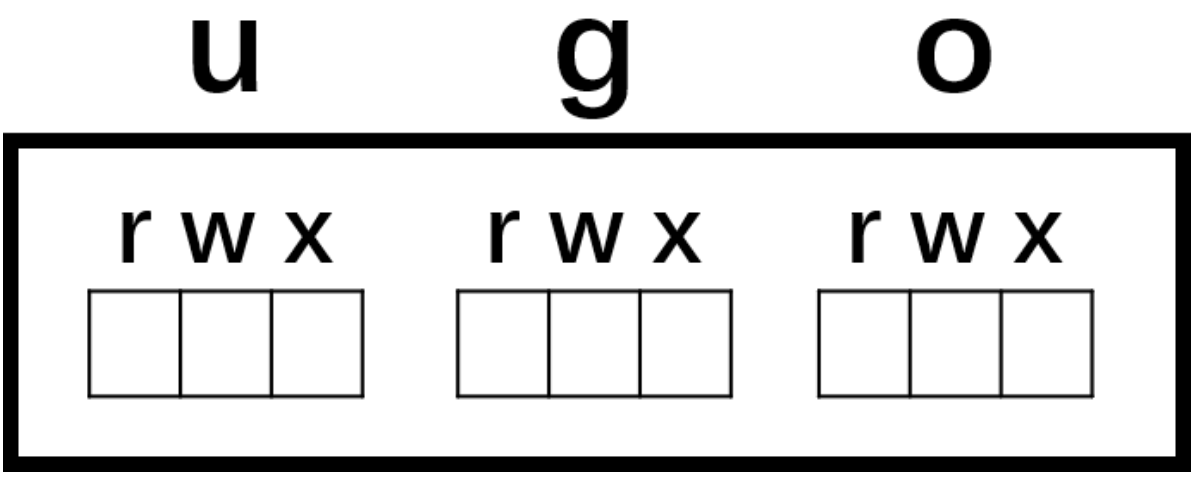
基本权限 UGO

文件权限设置：可以赋予某个用户或组，能够以何种方式，访问某个文件



UGO设置基本权限(r、w、x)

权限对象：

属主：u

属组：g

其他人：o

权限类型:

读：r　　4

写：w　　2

执行：x　　1

rwx 7

rw- 6

r-- 4

764

rwxrw-r--

权限管理

- 更改文件的属主、属组

```
[root@xwz ~]# chown centos:hr file1      # 改属主、属组
[root@xwz ~]# chown centos file1         # 只改属主
[root@xwz ~]# chown :hr file1            # 只改属组
```

```
[root@xwz ~]# chgrp it file1              # 改文件属性组
[root@xwz ~]# chgrp -R it dir1            # 改文件属性组
```

- 更改文件权限

使用符号更改

	对象	赋值符	权限类型	
	u	+	r	
chmod	g	-	w	file1
	o	=	x	
	a			

```
[root@xwz ~]# chmod u+x file1             # 属主增加执行
[root@xwz ~]# chmod a=rwx file1           # 所有人等于读写执行
[root@xwz ~]# chmod a=- file1             # 所有人没有权限
[root@xwz ~]# chmod ug=rw,o=r file1       # 属主属组等于读写，其他人只读
[root@xwz ~]# ll file1                    # 显示结果
```

使用数字

```
[root@xwz ~]# chmod 644 file1
[root@xwz ~]# ll file1
```

chown 改变某个文件的属主

chmod 改变某个文件的访问模式

- r、w、x权限对文件和目录的意义

权限	对文件的影响	对目录的影响
r(读取)	可以读取文件的内容	可以列出目录的内容(文件名)，可以使用ls命令
w(写入)	可以更改文件的内容	可以创建或删除目录中的任一文件，可以使用touch、rm命令
x(可执行)	可以作为命令执行文件	可以访问目录的内容(取决于目录中文件的权限)，可以使用cd命令

```
[root@xwz ~]# mkdir /dir10
[root@xwz ~]# touch /dir10/file1
[root@xwz ~]# chmod 777 /dir10/file1
[root@xwz ~]# ll -d /dir10/
drwxr-xr-x. 2 root root 19 9月   4 11:44 /dir10/
[root@xwz ~]# ll /dir10/file1
-rwxrwxrwx. 1 root root 0 9月   4 11:44 /dir10/file1
[root@xwz ~]# su centos
[centos@xwz root]$ cat /dir10/file1
[centos@xwz root]$ rm -rf /dir10/file1
rm: 无法删除"/dir10/file1": 权限不够
```

对目录有w权限

```
[root@xwz ~]# chmod 777 /dir10/
[root@xwz ~]# chmod 000 /dir10/file1
[root@xwz ~]# ll -d /dir10/
drwxrwxrwx. 2 root root 19 9月   4 11:44 /dir10/
[root@xwz ~]# ll /dir10/file1
----------. 1 root root 0 9月   4 11:44 /dir10/file1
[root@xwz ~]# su centos
[centos@xwz root]$ cat /dir10/file1
cat: /dir10/file1: 权限不够
[centos@xwz root]$ rm -rf /dir10/file1
```

对目录有w权限，可以在目录中创建新文件，可以删除文件夹中的文件(跟文件权限无关)

对文件x权限小心给予

ACL设置基本权限(r,w,x)

UGO设置基本权限：只能一个用户，一个组和其他人

ACL设置基本权限：r、w、x

ACL基本用法

```
setfacl
常用选项：
-m ：添加acl设定参数
-x ：删除acl设定参数
-b ：移除所有的ACL设定参数
-R ：递归添加acl设定参数
-d ：添加默认acl设定参数（目录）
删除用户权限：setacl -x u:username filename
删除组权限：setacl -x g:groupname filename
删除整个acl权限：setacl -b filename
```

设置

```
[root@xwz ~]# ll file1
-rw-r--r--. 1 centos it 0 9月   4 11:03 file1
[root@xwz ~]# getfacl file1
# file: file1
# owner: centos
# group: it
user::rw-
group::r--
other::r--

[root@xwz ~]# setfacl -m u:centos:rw file1      # 增加用户权限
[root@xwz ~]# setfacl -m u:user05:- file1       # 增加用户权限
[root@xwz ~]# setfacl -m o::rw file1            # 修改其他人权限
```

查看/删除

```
[root@xwz ~]# ll file1
```

```
-rw-rw-rw-+ 1 centos it 0 9月   4 11:03 file1
[root@xwz ~]# getfacl file1
# file: file1
# owner: centos
# group: it
user::rw-
user:centos:rw-
user:user05:---
group::r--
mask::rw-
other::rw-

[root@xwz ~]# setfacl -m g:hr:r file1       # 增加组权限
[root@xwz ~]# setfacl -x g:hr file1         # 删除组权限
[root@xwz ~]# setfacl -b file1              # 删除所有acl权限
```

```
[root@xwz ~]# man setfacl
[root@xwz ~]# getfacl file1 |setfacl --set-file=- file2 # 复制file1的acl给file2
```

mask

用于临时降低用户或组(除属主和其他人)的权限

mask决定了他们的最高权限

建议：为了方便管理文件权限，其他人的权限置为空

```
[root@xwz ~]# setfacl -m o::- file1
[root@xwz ~]# setfacl -m m::--- file1
[root@xwz ~]# getfacl file1
# file: file1
# owner: centos
# group: it
user::rw-
group::r--          #effective:---
mask::---
other::---
```

default

一般针对目录，默认权限独立于该目录本身的权限，规定了在该目录中创建的文件的默认ACL权限。

default可以指定在目录中创建出的新文件的acl权限

要求：希望centos能够对`/home`以及以后在`/home`下新建的文件有读、写、执行权限

```
[root@xwz ~]# setfacl -m u:centos:rwx /home
[root@xwz ~]# setfacl -m d:u:centos:rwx /home
[root@xwz ~]# getfacl /home
getfacl: Removing leading '/' from absolute path names
# file: home
# owner: root
# group: root
user::rwx
user:centos:rwx
```

```
group::r-x
mask::rwx
other::r-x
default:user::rwx
default:user:centos:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```

特殊权限

文件的特殊权限包括：SUID 4、SGID 2、SBIT 1

suid：借出程序所有者的权限

s：程序所属主有x权限

S：程序所属主没有x权限

SUID权限仅对二进制程序有效

仅在本程序中拥有改权限

属主拥有s权限，即可将自己的权限暂时借给其他人使用

```
[root@xwz ~]# ll /root/file2
-rw-r--rw-. 1 root root 0 9月   4 13:48 /root/file2
[root@xwz ~]# su - centos
上一次登录：日 9月  8 18:31:49 CST 2019pts/1 上
[centos@xwz ~]$ cat /root/file2
cat: /root/file2: 权限不够
[centos@xwz ~]$ ll /root/file2
ls: 无法访问/root/file2: 权限不够
```

系统会检查进程的所有者，根据所有者设置的权限来确定是否对文件有权限

```
[root@xwz ~]# ll /etc/shadow
----------. 1 root root 1760 9月   5 11:12 /etc/shadow
# 普通用户依旧是可以修改密码
[root@xwz ~]# ll /usr/bin/passwd
-rwsr-xr-x. 1 root root 27832 6月  10 2014 /usr/bin/passwd
```

SGID：借出用户组的权限

二进制程序有效

执行者拥有x权限

执行过程中暂时拥有用户组权限

高级权限的类型

s：程序所属主有x权限

S：程序所属主没有x权限

SBIT权限：用来做共享目录
当属主拥有x权限时，用小写的字母t表示，当属主没有x权限时，用大写字母T权限表示1
只针对目录有效
用户在此目录中创建文件时，只有root用户和自己可以删除该文件，其他用户是不可以修改此文件
典型例子/tmp这个目录
[zhangsan@localhost tmp]$ ll -d /tmp/
drwxrwxrwt. 9 root root 4096 9月 13 16:08 /tmp/

suid　　4　　# 使用文件所有者身份执行文件<针对文件>

sgid　2　　# 新建文件继承目录属组<针对目录>

sticky　1　　# 文件只能由文件拥有者，root,文件夹拥有者删除<针对目录>

- 设置或者修改特殊权限

```
chmod u+s file
chmod g+s dir
chmod o+t dir
chmod 4777 file
chmod 7777 file
chmod 2770 dir
chmod 3770 dir
```

`/tmp`文件夹是1777权限，否则会导致程序不能正常运行

大写的高级权限为表示普通权限没有`x`

小写的高级权限为表示普通权限有`x`

chattr

```
[root@xwz ~]# lsattr file2
---------------- file2
[root@xwz ~]# chattr +a file2
[root@xwz ~]# lsattr file2
-----a---------- file2
[root@xwz ~]# man chattr
----------
ATTRIBUTES(属性)
       当修改设置了'A'属性的文件时,它的atime记录不会改变.
       这可以在笔记本电脑系统中避免某些磁盘I/O处理.

       设置了`a'属性的文件只能在添加模式下打开用于写入.  只有超级用户可以设置或清除该属性.

       设置了`c'属性的文件在磁盘上由内核自动进行压缩处理.
       从该文件读取时返回的是未压缩的数据.
       对该文件的一次写入会在保存它们到磁盘之前进行数据压缩.

       设置了`d'属性的文件不能对其运行 dump(8) 程序进行备份.

       设置了`i'属性的文件不能进行修改:你既不能删除它,
       也不能给它重新命名,你不能对该文件创建链接,              而且也不能对该文件写入任何数据.
```

```
        只有超级用户可以设置或清除该属性.

        当删除设置了`s'属性的文件时,将对其数据块清零 并写回到磁盘上.

        当修改设置了`S'属性的文件时,                                修改会同步写入到磁盘上;
这与应用
        到文件子系统上的`sync'挂载选项有相同的效果.

        当删除设置了`u'属性的文件时, 将会保存其内容.  这使得用户可以请求恢复被删除的文件.
----------------
```

进程umask

进程 新建文件、目录的默认权限会收到umask的影响，umask表示要减掉得到权限

shell (vim,touch) 新文件或目录权限

vsftpd 新文件或目录权限

samba 新文件或目录权限

useradd 用户HOME

```
[root@xwz ~]# type -a umask
umask 是 shell 内嵌
umask 是 /usr/bin/umask
[root@xwz ~]# help umask
umask: umask [-p] [-S] [模式]
    显示或设定文件模式掩码。

    设定用户文件创建掩码为 MODE 模式。如果省略了 MODE，则
    打印当前掩码的值。

    如果MODE 模式以数字开头，则被当作八进制数解析；否则是一个
    chmod(1) 可接收的符号模式串。

    选项：
      -p    如果省略 MDOE 模式，以可重用为输入的格式输入
      -S    以符号形式输出，否则以八进制数格式输出

    退出状态：
    返回成功，除非使用了无效的 MODE 模式或者选项。
```

示例1:在shell进程中创建文件

```
[root@xwz ~]# umask          # 查看当前用户的umask权限
0022
[root@xwz ~]# umask -S       # 查看最终有的权限
u=rwx,g=rx,o=rx
[root@xwz ~]# touch file1
[root@xwz ~]# mkdir dir1
[root@xwz ~]# ll -d dir1/ file1
drwxr-xr-x. 2 root root 6 9月   9 09:29 dir1/
-rw-r--r--. 1 root root 0 9月   9 09:29 file1
```

示例2:修改shell umask值(临时)

```
[root@xwz ~]# umask 0000
[root@xwz ~]# mkdir dir2
[root@xwz ~]# touch file2
[root@xwz ~]# ll -d file2 dir2
drwxrwxrwx. 2 root root 6 9月   9 09:31 dir2
-rw-rw-rw-. 1 root root 0 9月   9 09:31 file2
```

示例3:修改shell umask值(永久 建议别改)

```
[root@xwz ~]# vim /etc/profile
--------------
 59 if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ];
then
 60     umask 002
 61 else
 62     umask 022
 63 fi
---------------
[root@xwz ~]# source /etc/profile       # 立即在当前shell中生效
```

示例4:通过umask决定新建用户`HOME`目录的权限

```
[root@xwz ~]# vim /etc/login.defs
-----------------
 61 # The permission mask is initialized to this value. If not specified,
 62 # the permission mask will be initialized to 022.
 63 UMASK           077
------------------
```