link: null
title: 珠峰架构师成长计划
description: -rw-r--r--
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=95 sentences=22, words=513

1. 文件权限

```
-rw-r--r--
```

- 文件类型
  - - 文件
  - d 目录
  - l 软链接文件
- 所有者 所属组 其它人
- r读 w写 x执行

chmod [选项] 模式 文件名

- 选项
  - -R 递归
- 模式
  - [ugoa][+-=][rwx]
- 权限数字
  - r 4
  - w 2
  - x 1

```
chmod 000 1.txt
chmod u+w 1.txt
chmod g+x 1.txt
chmod o+r 1.txt

chmod g+x,o+x 1.txt

chmod u-w 1.txt

chmod u=rwx 1.txt

chmod 777 1.txt
```

- 对文件来说最高权限是 x
- 对目录来讲最高权限是 w,只有读权限没有意义，对目录有了写权限，里面可以做任何事情

权限 含义 示例 r 读取文件内容 cat more head tail w 编辑、新增、修改文件内容,不能删除文件,除非对目录有写权限 vi echo x 可执行

权限 含义 示例 r 可以查看目录下的文件名 ls w 具有修改目录结构的权限。如新建、删除和重命名此目录下的文件和目录 touch rm mv cp x 进入目录 cd

```
useradd zf1
passwd zf1

cd /home/zf1
mkdir folder
touch folder/1.txt 默认755
chmod 750 folder
chmod 640 folder/1.txt
chmod 754 folder
chmod 755 folder
chmod 644 folder/1.txt
chmod 646 folder/1.txt
chmod 757 folder
```

- chown 用户名 文件名
- 如果想让一个用户拥有7权限，可以把这个文件的所有者改成这个用户名

```
chmod 755 folder
chown zf1 folder
```

- chgrp 组名 文件名
- 创建用户名的时候会为它创建一个所属组

```
chgrp zf1 folder
chown root:root folder
```

2.默认权限

- 查看默认权限
- 0022
  - 第一位0 文件特殊权限
  - 022 文件默认权限
- 默认权限就是文件一创建后就拥有的权限
- 文件默认不能建立可执行文件，必须手工赋予执行权限
- 文件默认权限最大为 666
- 默认权限需要换算成字母再相减
- 建立文件之后的默认权限，为666减去umask值

666 - 022 = 744

- 目录默认权限最大为777
- 建立目录之后的默认权限，为777减去umask值

777 - 022 = 755

临时修改

```
umask 0002
```

永久修改

```
vi /etc/profile
```

3. ACL

- 一个文件或文件夹只能有一个所有者和所属组，无法适应某些应用场景
- 访问控制(Access Control List，ACL)就是为特定的用户和组分配特定权限
- dumpe2fs 命令是查询指定分区详细文件系统信息的命令
    - -h 仅显示块中的信息，而不显示磁盘块组的详细信息

```
dumpe2fs -h /dev/sda1

Default mount options:    user_xattr acl
```

重新挂载根分区，并挂载加入ACL权限

```
mount -o remount,acl /dev/sda1
```

```
vi /etc/fstab
UUID /  ext4 defaults,acl  1 1
mount -o remount /dev/sda1
```

- setfacl 选项 文件名

选项 含义 -m 设定ACL权限 -x 删除指定的ACL权限 -b 删除所有的ACL权限 -d 设定默认ACL权限 -k 删除默认的ACL权限 -R 递归设定ACL权限

- 给用户 guest赋予rx权限
- 格式 "u:用户名:权限"

```
useradd guest
mkdir /home/guest/folder
useradd teacher
groupadd students
mkdir folder
chown teacher:students /home/guest/folder
chmod 770 /home/guest/folder
```

3.2.2 设置最大权限

- mask是最大有效权限，如果给用户赋予了ACL权限，是需要和mask的权限做与运算后才能得到真正权限
- 如果mask值是777，那么任何数与它相与，得到的是权限本身
- 如果允许自定义ACL，又不想让它超出最大权限

```
setfacl -m m:r folder
[root@localhost someone]# getfacl folder
# file: folder
# owner: teach
# group: stus
user::rwx
user:someone:r-x        #effective:r--
group::rwx            #effective:r--
mask::r--
other::---
```

setfacl -x u:用户名 文件名

```
setfacl -x u:someone folder
```

setfacl -b 文件名

```
setfacl -b folder
```

setfacl -x g:组名 文件名

```
setfacl -x g:students folder
```

- 向下一级一级传递权限
- 父目录设定ACL权限的时候，所有的子文件和子目录也会拥有相同的ACL权限
- 递归仅能赋给目录不能赋给文件

```
setfacl -m u:someone:rx folder
setfacl -m u:someone:rx -R folder
```

- 默认ACL权限是指如果给父目录设置了默认ACL权限，那么父目录里所有新建的子文件都会继承父目录的ACL权限
- setfacl -m d:u:用户名:权限 目录名

```
setfacl -m d:u:someone:rw folder
```

4. sudo权限

- root把本来只有超级管理员可以使用的命令赋予普通用户来使用
- sudo操作的对象是系统命令
- 通过 visudo可以由超级用户赋权
- 实际修改的是 /etc/sudoers文件
- 命令必须写绝对路径

```
root    ALL=(ALL) ALL
用户名 被管理主机地址=(可使用的身份) 授权命令(绝对路径)

somethone ALL=(root) /usr/sbin/useradd
```

```
%wheel ALL=(ALL) ALL
%组名 被管理主机地址(IP)=(可使用的身份) 授权命令(绝对路径)
```

```
sudo -l  查看目前的sudo权限
```

6. 特殊权限

- set User ID
- 只有可以执行的二进制程序才能设置SUID权限
- 命令执行者要对该程序有x(执行)权限
- 命令执行者在执行该程序的时候身份会变成文件的所有者
- SetUID权限只在该程序执行过程中有效，也就是说身份改变在程序执行过程中有效
- 文件属主的x权限,用s代替.表示被设置了SUID,如果属主位没有x权限,会显示为大写S,表示有故障(权限无效)

字母表示法

```
chmod u+s 文件
chmod u-s 文件
```

数字表示法,在普通三位数字权限位之前,用4代表添加的SUID位 chmod 4755 文件 添加 SUID权限到二进制程序文件(添加到目录无意义)

```
chmod 4755 文件
chmod 0xxx 可以删除文件的 SUID(无法删除目录的SUID)
```

```
chmod 4755 hello.sh

ll /etc/shadow
ll /user/bin/passwd
```

```
chmod 4755 文件名
chmod u+s 文件名 给所有者加suid权限
chmod g+s 文件名 给所属组加suid权限
chmod o+s 文件名 给其它人加suid权限
```

```
chmod 0755 文件名
chmod u-x 文件名 给所有者减去suid权限
```

- 关键目录应该严格控制写权限
- 用户的密码要正确设置
- 不能轻易给文件赋SetUID权限

```
chmod 4755 /bin/vi
/bin/vi /etc/shadow
```

Set Group ID

字母表示法

```
chmod g+s 文件

chmod g-s 文件
```

数字表示法

```
chmod 2755  文件/目录 添加SGID到目录或文件
在普通数字权限位前,用2代表添加SGID位
chmod 0755 文件/目录 删除文件的SGID,(目录不受影响)
chmod 755 文件/目录 同上
```

- 只有可执行的二进制程序才能设置SGID权限
- 命令执行者要对该程序拥有x(执行)权限
- 命令执行在执行程序的时候，执行者的组身份升级为该程序文件的所属组
- setGID权限同样只在该程序执行过程中有效，也就是说组身份改变只在程序执行过程中有效

```
chmod 4755 /bin/cat
chmod 0755 /bin/cat
chmod u+s /bin/cat
chmod u-s /bin/cat

cat /root/uid.txt
```

- 因为 /usr/bin/locate是可执行二进制程序，可以赋予SGID
- 执行用户对 /usr/bin/locate 命令拥有执行权限
- 执行 /usr/bin/locate 命令时组身份会升级为slocate组，而slocate组对/var/lib/mlocate/mlocate.db文件拥有读权限，所以普通用户也可以使用locate命令
- 当命令执行结束，普通用户的组身份切换回自己的组身份
- 可以针对文件或者目录设置GID权限
- 普通用户必须对此目录拥有 r和 x权限，才能进入此目录
- 普通用户在此目录中的组会变成此目录的所属组
- 如果普通用户对此目录拥有 w权限时，新建的文件的默认属组是这个目录的所属组

```
mkdir test3

chmod 2755 test3
chmod g+s test3
ll -d test3
chmod 777 test3

touch 1.txt
ll -h 1.txt

取消 SetGID
chmod 0755  文件名
chmod g-s 文件名
```

- 粘着位目前只针对目录有效
- 普通用户对该目录拥有w和x权限，即普通用户可以在此目录拥有写入权限
- 如果没有粘着位，因为普通用户拥有w权限，所以可以删除此目录下的所有文件，包括其它用户的文件。但一旦赋予了粘着位，除了root之外，普通用户就算拥有了w权限，也只能删除自己建立的文件，不能删除其它用户的文件
- 必须给其它人赋完整权限

```
chmod 1755  目录名
chmod o+t 目录名
```

```
chmod 0755  目录名
chmod o-t 目录名

ll -d /tmp
```

- chattr [+-=] [选项] 文件或目录名

选项 文件 目录 +i 不允许对文件进行删除、改名，也不能添加和修改数据 只能修改目录下文件的数据，但不允许建立和删除文件 +a 只能在文件中增加属性，但不能删除也不能修改数据 只允许在目录中建立和修改文件，不能删除

```
chattr +i folder
lsattr folder
```