一、权限概述

㈠ 什么是权限

权限：在计算机==系统中==，权限是指某个==计算机用户==具有==使用软件资源的权利==。

思考：计算机资源分为哪两部分？

- 硬件资源　　硬盘、CPU、内存、网卡等==物理硬件资源==
- 软件资源

  软件：操作系统(特殊的软件)、应用程序。只要不启动，这些软件就是一堆静态的==文件==，并且静静的躺在我们计算机的磁盘中。

  软件资源：Linux系统中，一切皆文件！SO，这里的软件资源就是==文件资源==。

总结：

我们今天所讲的权限，指的就是：==文件资源==所拥有的相关权限，即==文件权限==。

㈡ 权限设置目的

文件权限的设置目的：==是想让某个用户有权利操作文件==

㈢ 文件权限的分类

- **普通权限**

  用户正常情况去操作文件所具有的权限
- **高级权限**

  用户对某个文件操作有特殊需求，而普通权限不能满足，需要给文件设置高级权限
- **默认权限**

  用户在系统中创建一个文件，该文件默认都会有一个权限，该权限是默认有的

注意：

权限是==设置在文件上==的，而不是用户

二、==普通权限(重点)==

㈠ 理解普通权限rwx含义

1、读权限—r(==r==ead)

- 针对==目录==

  一个目录拥有r权限，说明可以查看该==目录里的内容==（ls命令列出）
- 针对==普通文件==

  一个普通文件拥有r权限，说明可以查看该==文件的内容==(cat/head/tail/less/more等命令查看)
- 读权限==**r**==（read）用数字表示是**==4==**

2、写权限—w(==w==rite)

- 针对==目录==

  一个目录拥有w权限，说明可以在该目录里==**创建、删除、重命名**==等操作（mkdir/touch/mv/rm等）

- 针对==普通文件==

  一个普通文件拥有w权限，说明可以==**修改**==该==文件的**内容**==（vi/vim编辑器编辑文件）

- 写权限==**w**==（write）用数字表示是==**2**==

3、执行权限—x(e==x==ecute)

- 针对==目录==

  一个目录拥有x权限，说明可以==**进入或切换到**==该目录里（cd命令）

- 针对==普通文件==

  一个普通文件拥有x权限，说明可以==**执行**==该文件（一般程序文件、脚本文件、命令都需要执行权限）

- 执行权限==**x**==（execute）用数字表示是==**1**==

4、没有权限—横杠==-==

- 没有任何权限用横杠==-==表示，数字表示是==**0**==

㈡ 理解UGO的含义

1、UGO指的是什么

UGO，指的是==**用户身份**==，每个字母代表==不同的==用户身份。

- U（the user who owns it）

  文件的==拥有者==(owner)或者==创建者==

- G（other users in the file's ==g==roup）

  在文件的所属组（默认是创建文件的用户的主组）里的用户

- O（==o==ther users ==not in== the file's group）

  既不是文件的创建者，也不在文件属组里的用户，称为其他人

注意：

除了上面ugo以外，还有一个字母==**a**==（all users）,表示==所有用户==，包含ugo

2、如何判断不同身份的用户对文件的权限

```
查看文件详细信息，包含权限信息：
[root@localhost ~]# ls -l
total 144
-rw-r--r--. 1 root root     9 Mar  2 20:38 1.sh
-rw-------. 1 root root  1651 Feb 28 11:00 anaconda-ks.cfg
drwxr-xr-x. 2 root root  4096 Mar  6 18:34 Desktop
drwxr-xr-x. 2 root root  4096 Feb 28 14:12 dir1
```

㈢ 修改文件普通权限(chmod)

1、chmod命令用法

```
chmod [选项] 文件名
常见选项：
-R, --recursive      递归更改目录和目录里文件的权限
```

2、举例说明

① 通过字母形式更改文件权限

```
u:表示文件拥有者
g:表示文件属组用户
o:表示其他人，即不是文件的创建者，也不在文件属组里
a:表示所有人
```

- 环境准备

```
[root@localhost ~]# mkdir /tmp/dir1
[root@localhost ~]# touch /tmp/dir1/file{1..5}
[root@localhost ~]# touch /tmp/test{1..3}
[root@localhost ~]# ll /tmp/ -R
```

- 使用字母形式修改文件权限

```
[root@localhost tmp]# pwd
/tmp
[root@localhost tmp]# ll test1
-rw-r--r--. 1 root root 0 Mar  6 20:45 test1
```

```
[root@localhost tmp]# chmod u+x test1
[root@localhost tmp]# ll test1
-rwxr--r--. 1 root root 0 Mar  6 20:45 test1
[root@localhost tmp]# chmod g+w test1
[root@localhost tmp]# ll test1
-rwxrw-r--. 1 root root 0 Mar  6 20:45 test1
[root@localhost tmp]# chmod o-r test1
[root@localhost tmp]# ll test1
-rwxrw----. 1 root root 0 Mar  6 20:45 test1

[root@localhost tmp]# ll test2
-rw-r--r--. 1 root root 0 Mar  6 20:45 test2
[root@localhost tmp]# chmod a+x test2
[root@localhost tmp]# ll test2
-rwxr-xr-x. 1 root root 0 Mar  6 20:45 test2

[root@localhost tmp]# ll test3
-rw-r--r--. 1 root root 0 Mar  6 20:45 test3
[root@localhost tmp]# chmod u+x,g+w,o-r test3
[root@localhost tmp]# ll test3
-rwxrw----. 1 root root 0 Mar  6 20:45 test3

[root@localhost tmp]# chmod u=rw,g=rx,o+r test3
[root@localhost tmp]# ll test3
-rw-r-xr--. 1 root root 0 Mar  6 20:45 test3

修改目录的权限：
[root@localhost tmp]# ll -d dir1/
drwxr-xr-x. 2 root root 4096 Mar  6 20:45 dir1/
[root@localhost tmp]# ll dir1/
total 0
-rw-r--r--. 1 root root 0 Mar  6 20:45 file1
-rw-r--r--. 1 root root 0 Mar  6 20:45 file2
-rw-r--r--. 1 root root 0 Mar  6 20:45 file3
-rw-r--r--. 1 root root 0 Mar  6 20:45 file4
-rw-r--r--. 1 root root 0 Mar  6 20:45 file5

1.只修改目录本身的权限
[root@localhost tmp]# chmod g+w dir1/
[root@localhost tmp]# ll -d dir1/
drwxrwxr-x. 2 root root 4096 Mar  6 20:45 dir1/
[root@localhost tmp]# ll dir1/
total 0
-rw-r--r--. 1 root root 0 Mar  6 20:45 file1
-rw-r--r--. 1 root root 0 Mar  6 20:45 file2
-rw-r--r--. 1 root root 0 Mar  6 20:45 file3
-rw-r--r--. 1 root root 0 Mar  6 20:45 file4
-rw-r--r--. 1 root root 0 Mar  6 20:45 file5
说明：目录下面文件的权限并没有修改

2.修改目录以及目录里所有文件的权限（递归修改），使用-R参数
[root@localhost tmp]# chmod -R o+w dir1/
[root@localhost tmp]# ll -d dir1/
drwxrwxrwx. 2 root root 4096 Mar  6 20:45 dir1/
[root@localhost tmp]# ll dir1/
total 0
-rw-r--rw-. 1 root root 0 Mar  6 20:45 file1
-rw-r--rw-. 1 root root 0 Mar  6 20:45 file2
```

```
-rw-r--rw-. 1 root root 0 Mar  6 20:45 file3
-rw-r--rw-. 1 root root 0 Mar  6 20:45 file4
-rw-r--rw-. 1 root root 0 Mar  6 20:45 file5
```

② 通过数字形式更改文件权限

- 学会用数字表示权限

```
字母和数字对应关系：
r--4
w--2
x--1
---0

rw-  r-x  r--   用数字表示就是654
rwx  rw-  ---   用数字表示就是760
755 用字母表示就是rwx r-x r-x
644 用字母表示就是rw- r-- r--
```

- 使用数字形式修改文件权限

```
# chmod 644 file1
# chmod 700 file2
# chmod -R 755 dir1
```

总结：

1. 用户是否可以删除目录里的文件，看的是目录的权限！！！
2. 对于正常能够操作的目录来说，应该默认具备r-x

3、课堂练习

1. 创建5个用户user01~user05和一个admin组
2. 将user01~user03用户加入到admin组里
3. user01用户在其家目录里创建file1~file3三个文件
4. user02用户编辑/home/user01/file1文件的内容：good good study,day day up!
5. user05用户往/home/user01/file1文件里追加内容：I known
6. user04用删除/home/user01家目录的所有文件

三、高级权限(了解)

㈠ 高级权限有哪些

1、冒险位(SETUID)

- 冒险位，指文件操作者(用户)==临时拥有==文件==拥有者==的权限
- 冒险位，一般针对的是==命令==或者==脚本文件==
- 冒险位，用字母表示是==s或S==；数字表示是==4==
- 冒险位的设置：`chmod u+s 文件名`或者`chmod 4xxx 文件名`

2、强制位(SETGID)

- 强制位，一般针对的是==目录==

  如果一个目录拥有强制位，那么==任何用户==在该目录里所创建的任何文件的==属组==都会继承==该目录的属组==。

- 强制位，用字母表示是==s或S==；数字表示是==2==
- 强制位的设置：`chmod g+s 文件名`或者`chmod 2xxx 文件名`

3、==粘滞位(STICKY)==

- 粘滞位，一般针对的是==公共目录==

  如果一个公共目录拥有粘滞位，那么该目录下的文件，只有==root==和==文件的创建==者可以删除，其他人只能自己管理自己。（A用户不能删除B用户创建的文件）

- 粘滞位，用字母表示是==t或T==；数字表示是==1==
- 粘滞位的设置：`chmod o+t 文件名`或者`chmod 1xxx 文件名`

㈡ 高级权限设置

1、冒险位举例

需求：
给一个vim命令设置冒险位，目的是任何人拿vim去修改文件可以临时获得文件拥有者的权限

```
[root@localhost tmp]# which vim
/usr/bin/vim
[root@localhost tmp]# ll /usr/bin/vim
-rwxr-xr-x. 1 root root 2324712 Dec 22  2016 /usr/bin/vim
[root@localhost tmp]# chmod u+s /usr/bin/vim
或者
[root@localhost tmp]# chmod 4755 /usr/bin/vim
[root@localhost tmp]# ll /usr/bin/vim
-rwsr-xr-x. 1 root root 2324712 Dec 22  2016 /usr/bin/vim
测试验证，普通用户使用vim修改一个本没有权限修改的文件：
[root@localhost tmp]# ll /etc/passwd
-rw-r--r--. 1 root root 1650 Mar  5 20:39 /etc/passwd
[root@localhost tmp]# su - user01
[user01@localhost ~]$ vim /etc/passwd
验证是否可以修改成功，如果可以，说明user01用户临时拥有了/etc/passwd文件拥有者的权限
```

2、强制位举例

需求：
给目录dir2设置一个强制位，测试是否任何人在该目录里创建的文件属组都是该目录的属组

```
[root@localhost tmp]# ll -d dir2
drwxr-xr-x. 2 root root 4096 Mar  6 13:42 dir2
给dir2增加强制位：
[root@localhost tmp]# chmod g+s dir2
给dir2目录设置权限，让其他人可以写
[root@localhost tmp]# chmod o+w dir2
[root@localhost tmp]# ll -d dir2
drwxr-srwx. 2 root root 4096 Mar  6 13:42 dir2
```

```
测试普通用户user01在dir2目录里创建文件的属组是否是dir2的属组
[root@localhost tmp]# su - user01
[user01@localhost ~]$ touch /tmp/dir2/file1
[user01@localhost ~]$ ll /tmp/dir2/file1
-rw-rw-r--. 1 user01 root 0 Mar  6 13:44 /tmp/dir2/file1
```

3、==粘滞位举例==

```
需求：
在创建一个公共目录/tmp/dir3,要求所有人都可以在该公共目录里创建、删除文件；但是只能自己管理自己，不能删除别人的文件

[root@localhost ~]# mkdir /tmp/dir3
[root@localhost ~]# chmod 777 /tmp/dir3
[root@localhost ~]# chmod o+t /tmp/dir3
或者一步到位：
[root@localhost ~]# chmod 1777 /tmp/dir3
[root@localhost ~]# ll -d /tmp/dir3
drwxrwxrwt. 2 root root 4096 Mar  6 13:52 /tmp/dir3
测试验证：
自己完成
```

㈢ 总结

- 高级权限分类
  - 冒险位——>针对命令 s/S 4 chmod u+s 命令文件
  - 强制位——>针对目录 s/S 2 chmod g+s 目录
  - ==粘滞位==———>针对公共目录 t/T 1 chmod o+t 公共目录
- 高级权限设置

```
chmod 4xxx  文件名
chmod 2xxx   目录名
chmod 1777   公共目录
```

四、默认权限(了解)

㈠ 什么是文件的默认权限

所谓文件的默认权限(遮罩权限)，是指用户创建文件后，==文件天生==就有的权限，不需要设置。

㈡ 文件默认权限由谁控制

文件默认权限由一个叫做**==umask==**的东西来控制。

㈢ umask如何控制文件默认权限

1、临时控制

- 什么是临时控制?

> 临时控制，指的是用命令`umask`临时设置，只在当前终端当前进程中生效。

```
查看当前用户的umask：
[root@localhost ~]# umask
0022
[root@localhost ~]# su - user01
[user01@localhost ~]$ umask
0002
注意：
1. 管理员和普通用户的umask不同，就表示管理员和普通用户创建的文件的默认权限不同！
2. 第1位数字表示高级权限；后面3位数字表示普通权限
```

- 如何临时设置用户的umask?

  写在前面：

  Linux系统中，默认创建目录的最大权限是==0777==；文件的最大权限是==0666==

  777

  666　rw-rw-rw-

```
[root@localhost ~]# umask 0007  临时设置root用户的umask为0007
问：umask=0007，那么在当前终端上root用户所创建目录和普通文件的权限分别是什么呢？
计算过程如下：
umask=文件的最大权限-文件的默认权限
目录：
0007=0777-目录的默认权限
目录的默认权限=0777-0007=0770=rwxrwx---      770

普通文件：
0007=0666-普通文件的默认权限
普通文件的默认权限=0666-0007=0660=rw-rw----

说明：
1.权限用数字表示时没有负数，所以最小就是0
2.默认权限规则遵循Linux系统中权限最小化原则

$ stu1   umask 0003
dir:
默认权限=目录最大权限-umask=777-003=774    rwxrwxr--
file:
默认权限=文件最大权限-umask=666-003=663  rw- rw- -wx 实际：  664 rw-rw-r--
```

2、永久控制

- 什么是永久设置?

> 永久设置，指的是通过修改配置文件设置，对用户的所有终端所有进程生效

- 修改哪个配置文件呢?

  1. 相关配置文件介绍

全局配置文件（针对所有用户所有进程）
/etc/profile
系统和用户的环境变量信息，当用户第一次登录时,该文件被读取
/etc/bashrc
每个运行的bash信息（系统别名、函数及默认权限的定义），当bash被打开时,该文件被读取

局部配置文件（针对某个特定用户以及用户的所有进程）
~/.bashrc
当前用户的bash信息，当用户登录和每次打开新的shell时该文件被读取
~/.bash_profile
当前用户的环境变量，当用户登录时，该文件被读取
~/.bash_history
保存当前用户历史命令的文件
~/.bash_logout
当用户退出bash或者终端时，会首先执行该文件里的代码，然后再退出

2. 如何永久设置用户的umask？

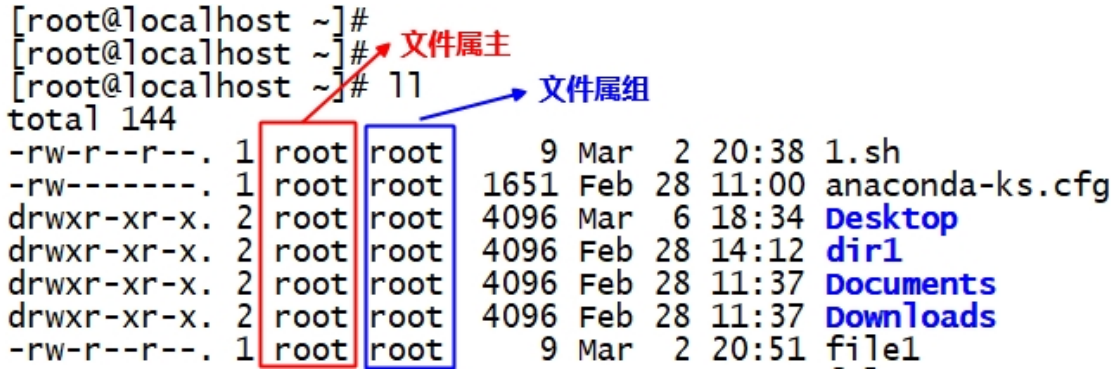
1.针对所有用户生效
vim /etc/bashrc
在该文件的最后增加以下内容：
umask 0007

重新读取该配置文件让其立马生效
source /etc/bashrc
或者
. /etc/bashrc

2.针对某个用户生效
比如，只针对user01用户生效
[user01@localhost ~]$ vim ~/.bashrc
在该文件的最后增加以下内容：
umask 0007

五、==文件的属主和属组==

(一) 如何查看文件的属主和属组



(二) ==如何修改文件的属主和属组==

1、chown命令修改

`chown`命令既可以修改文件的属主，也可以修改文件的属组。

```
只修改文件的属主
# chown 用户名  文件名
修改文件的属主和属组
# chown 用户名.组名  文件名
# chown 用户名:组名  文件名
# chown 用户名.     文件名 //没有指定组名，默认是用户的主组
只修改文件的属组
# chown .组名 文件名
# chown :组名 文件名

可以加-R选项，表示递归修改
```

2、chgrp命令修改

`chgrp`命令只能修改文件的属组。

```
# chgrp 组名 文件名
```

六、ACL访问控制策略(扩展)

㈠ ACL能做什么

1. ACL访问控制策略可以作为前面所讲==权限的补充==，==更加细==的来控制文件的权限
2. ACL策略可以==只针对某个用户==在文件上有相应权限
3. ACL策略也可以==只针对多个用户或者一个组==里的所有用户在文件上有相应权限

㈡ 如何设置文件的ACL策略

1、设置ACL策略(setfacl)

```
常用选项：
-m  修改或者设置ACL策略
-R 递归授权，对目录下已存在的目录或文件有acl策略，但新建的文件没有
-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 默认ACL策略，只针对目录，该目录下新建的目录和文件都会继承acl策略
mask: 定义除其他人和所有者外的最大权限
```

==重点掌握：==

```
setfacl -m u:用户:rwx  /home/redhat/file1   给单个用户单独加权限
setfacl -m g:组名:rwx  /home/redhat/file1   给单个组单独加权限

setfacl -x u:用户 /home/redhat/file1          去掉某个用户的权限
setfacl -x g:组名 /home/redhat/file1              去掉某个组的acl策略
setfacl -b /home/redhat/file1                     删除文件上所有的acl策略

# setfacl -m u:user01:rw file1       针对于单个用户给可读可写权限
# setfacl -m g:sysadmin:rw file1     针对于单个组给可读可写权限
```

2、查看ACL策略(getfacl)

```
# getfacl 文件名
```

㈢ 课堂练习

1. 删除当前系统中的所有普通用户后，再创建5个用户stu1~stu5和一个组admin
2. 将stu1~stu3加入到admin组中
3. stu1用户在自己的家目录里创建3个文件file1~file3
4. 只允许stu4用户修改file1文件
5. 同时允许admin组的所有成员修改file2文件的内容

七、综合练习

1. 使用普通用户stu1登录系统，并在/u01/STU1目录下创建一个文件zhangsan，内容为：我是某某某，我要努力学习！我一定能行的！加油！（I am jack, I want to study hard,I can do it,come on）
2. 使用stu2用户登录系统，并修改stu1用户刚刚创建的文件zhangsan，增加内容：我要和你挑战（I want to challenge you）！并在相同的目录下创建一个自己的文件lisi，内容同上
3. stu3用户同时可以查看stu1和stu2两个用户的文件，但是不能做任何修改