9、权限管理基本命令

前面大致的介绍了一下文件的权限信息的表示格式和含义：

drwxr-xr-x. 第一个字符表示的文件类型：

d：目录文件

l：链接文件

b：块设备文件

s：套接口文件

c：字符设备文件

p：管道文件，特殊少见的设备

-：表示普通文件

w：写

r：读

x：执行

一、更改文件或目录权限命令：chmod

①、命令名称：chmod

②、英文原意：change the permissions mode of a file

③、命令所在路径：/bin/chmod

④、执行权限：所有用户

⑤、功能描述：改变文件或目录权限

⑥、语法： chmod

【{ugoa}{+-=}{rwx}】【文件或目录】

【mode=421】【文件或目录】

-R 递归修改

注意：不是每一个Linux用户都有权限更改某个文件或目录权限，能更改文件或目录权限的只有两种用户

①、文件的所有者。我们通过ls命令查看某个文件的详细信息，可以看到该文件的所有者。

②、root用户，这不用多说，root用户是linux系统权限最大的用户。别人不能干的事，root用户都能干。

对于上面的语法 chmod 【{ugoa}{+-=}{rwx}】【文件或目录】，我们要知道ugoa分别是：u:表示所有者，g:表示所属组，o:表示其他人，a:表示所有人。

而rwx表示的意思如下：

代表字符	权限	对文件的含义	对目录的含义
r	读权限	可以查看文件内容	可以列出目录中的内容
w	写权限	可以修改文件内容	可以在目录中创建、删除文件
x	执行权限	可以执行文件	可以进入目录

对于【mode=421】【文件或目录】，这是我们将权限用数字表示，其中 r 表示4，w表示2，x表示1，分别是2的0次方，1次方，2次方。那么我们可以这样理解：具有 rwx 权限的数字就是 7，具有 rw- 权限的数字是 6，具有 r-- 权限的数字是 4。

我们常用的权限的数字模式又这几种：

644：这是文件的基本权限，代表所有者拥有读、写权限，而所属组和其他人拥有只读权限。rw-r--r--

755：这是文件的执行权限和目录的基本权限，代表所有者拥有读、写和执行权限，而所属组

和其他人拥有读和执行权限。rwxr-xr-x

777：这是最大权限。在实际的生产服务器中，要尽力避免给文件或目录赋予这样的权限，这

会造成一定的安全隐患。rwxrwxrwx

范例1：我们赋予 tmp 目录下的 tmp.log 所有者 x 的权限；赋予 所属组 w 权限，其他人 w 权限。

chmod u+x /tmp/tmp.log

chmod g+w,o+w /tmp/tmp.log

```
[root@node3 tmp]# touch tmp.log
[root@node3 tmp]# ll
总用量 0
-rw-r--r--. 1 root root 0 10月 26 22:47 tmp.log
[root@node3 tmp]# chmod u+x /tmp/tmp.log
[root@node3 tmp]# ll
总用量 0
-rwxr--r--. 1 root root 0 10月 26 22:47 tmp.log
[root@node3 tmp]# chmod g+w,o+w /tmp/tmp.log
[root@node3 tmp]# ll
总用量 0
-rwxrw-rw-. 1 root root 0 10月 26 22:47 tmp.log
[root@node3 tmp]#
```

将上面例子改为用 数字来操作，也就是说我们要给 tmp.log赋予的文件权限是 rwxrw-rw-，用数字表示是766。

chmod 766 tmp.log

```
[root@node3 tmp]# touch tmp.log
[root@node3 tmp]# ll
总用量 0
-rw-r--r--. 1 root root 0 10月 26 22:50 tmp.log
[root@node3 tmp]# chmod 766 tmp.log
[root@node3 tmp]# ll
总用量 0
-rwxrw-rw-. 1 root root 0 10月 26 22:50 tmp.log
[root@node3 tmp]#
```

我们还可以递归赋予权限，也就是加上 -R 参数给指定目录下的所有文件或目录赋予指定权限。

范例2：给 tmp 目录下所有文件和目录赋予 776 的权限

chmod -R 776 /tmp

```
[root@node3 /]# chmod -R 776 /tmp
[root@node3 /]# cd /tmp
[root@node3 tmp]# ll
总用量 0
-rwxrwxrw-. 1 root root 0 10月 26 22:50 tmp.log
[root@node3 tmp]#
```

注意知识点：对权限的正确认识，分两种对象，文件和目录

文件：

r，读：对文件的内容有读的权限，对应的命令：cat，more，less，head，tail等

w，写：可以修改文的内容，对应命令vi，vim，echo，注意对文件的写权限不能删除文件本身，要删除文件必须拥有文件的父文件夹的写权限

x，执行：表示拥有了执行的这个文件的权限，但是不是说有这个权限，文件就可以执行，文件可以执行否，前提当人是权限，但是又了权限的前提下，还必须是这个文件里的内容是正确的可以执行的代码

目录：

r，读：对目录，是可以查看文件夹里的内容，对应命令是ls

w，写：可以修改文件里的内容，也就是可以复制，删除，新建，移动文件夹里的子目录和文件，对应的命令cp，touch，mv，rm...命令

x，执行：文件夹显示是不能运行的，对文件夹设置执行权限，表示可以进入到文件夹里边去，对应的命令cd

总结：我们要删除一个文件，不是对文件本身有w权限就可以，必须要拥有这个文件的父文件夹具有w的权限才可以！

二、改变文件或目录所有者命令：chown

①、命令名称：chown

②、英文原意：change file ownership

③、命令所在路径：/bin/chown

④、执行权限：所有用户

⑤、功能描述：改变文件或目录的所有者

⑥、语法： chown 【用户】【文件或目录】

注意：能更改文件或目录的所有者用户是 root

这里我们通过useradd【用户名】命令创建用户，然后通过passwd【用户名】输入密码，这两个命令后面会将。

我们通过这两个命令创建 vae 用户

```
[root@Node1 tmp]# useradd vae
[root@Node1 tmp]# passwd vae
更改用户 vae 的密码 。
新的 密码：
无效的密码： WAY 过短
无效的密码： 过于简单
重新输入新的 密码：
passwd： 所有的身份验证令牌已经成功更新。
[root@Node1 tmp]#
```

然后我们将tmp.log的所有者更改为 vae 用户：chown vae tmp.log

```
[root@node3 tmp]# ll
总用量 0
-rwxrwxrw-. 1 root root 0 10月 26 22:50 tmp.log
[root@node3 tmp]# chown vae tmp.log
[root@node3 tmp]# ll
总用量 0
-rwxrwxrw-. 1 vae root 0 10月 26 22:50 tmp.log
[root@node3 tmp]#
```

首先tmp.log所有者是root

我们通过此命令更改所有者为 vae 用户

所有者已经发生改变

三、改变文件或目录所属组命令：chgrp

①、命令名称：chgrp

②、英文原意：change file group ownership

③、命令所在路径：/bin/chown

④、执行权限：所有用户

⑤、功能描述：改变文件或目录的所属组

⑥、语法： **chgrp【用户组】【文件或目录】**

注意：能更改文件或目录的所有者用户是 root

四、显示、设置文件的缺省权限命令：umask

①、命令名称：umask

②、英文原意：the user file-creation mask

③、命令所在路径：shell 内置命令

④、执行权限：所有用户

⑤、功能描述：显示、设置文件的缺省权限

⑥、语法： umask 【-S】

-S 以rwx形式显示新建文件的缺省权限

注意：可能大家不太明白这个命令的意思，我们分别执行umask和 umask -S ，如下：

```
[root@node3 tmp]# umask
0022
[root@node3 tmp]# umask -S
u=rwx,g=rx,o=rx
[root@node3 tmp]#
```

其中umask 执行显示结果是 0022,第一个0表示特殊权限，后面我们会单独进行讲解有哪几种特殊权限。022表示权限的掩码值，我们用7 7 7 减去 0 2 2得到755（**这里的减，不是单纯的对应位置上的数相减，一会儿我们详细说说掩码值是怎么转成权限值的**），表示的就是下面通过加上-S输出的rwxr-xr-x，这个值用数字表示就是755.

这个意思说明创建一个文件目录的默认权限所有者为rwx,所属组为rx,其他人为rx。也就是说创建一个新文件夹默认权限为 rwxr-xr-x

注意这里创建目录和创建一个文件会有点不一样，来验证一下创建文件：

```
[root@node3 tmp]# umask -S
u=rwx,g=rx,o=rx
[root@node3 tmp]# touch a.txt
[root@node3 tmp]# ll a.txt
-rw-r--r--. 1 root root 0 10月 27 21:10 a.txt
[root@node3 tmp]#
```

我们发现使用touch命令创建了一个文件a.txt，然后发现权限并不是rwxr-xr-x（755），而是rw-r--r--（644）。对比发现少了三个x，也就是少了可执行权限。这是为什么呢？

这是因为在Linux系统中，所有新创建的文件都是没有可执行权限的。这是出于Linux系统的一种自我保护，因为类似的病毒木马程序都是具有可执行权限的。所以在Linux系统中，新创建的文件是没有可执行权限的。对于新建的文件最高的权限666

了解了如何查看默认的权限后，我们如何设置默认权限呢？

比如我们想将新创建的文件权限设置为rwxr-xr--，也就是754。

我们用777减去754得到023。也就是通过执行 umask 023 来完成默认权限设置。

```
[root@node3 tmp]# umask -S
u=rwx,g=rx,o=rx
[root@node3 tmp]# umask 023 设置默认文件的权限为754
[root@node3 tmp]# umask -S
u=rwx,g=rx,o=r
[root@node3 tmp]#
```

掩码值的计算方法：

1、将默认最大权限（目录777，文件666）和umask值都转换为2进制

2、对umask取反

3、将默认权限和umask取反后的值做与运算

4、将得到的二进制值再转换8进制，即为权限，

例1： umask 为022

6 6 6 umask 0 2 2

110 110 110 000 010 010 # 转成二进制

111 101 101 # umask取反的值

110 110 110 与 #第二步，默认权限和umask取反后的值做与运算

111 101 101 # umask取反的值

110 100 100

6 4 4 #转成8进制