

8. Linux文件和目录（中）

笔记本： 优秀笔记

创建时间： 2018/12/21 16:04

更新时间： 2018/12/21 16:05

作者： 306798658@qq.com

chmod文件或目录权限

- 之前学习的ls -l命令时候，学习了列出每个位置都是什么。

```
[root@aming1 ~]# ls -l 111.txt
-rw-r--r--. 1 root root 1008 5月 13 22:26 111.txt
```

- 开头除了第一位是文件类型外，后面九位都是权限位。
- 在inode使用量前面11位。第一位表示文件类型，后面九位表示文件权限，最后一个(.)是受制于selinux。如果关闭selinux，新建的文件或目录就会没有(.)
- 在九位权限位置里，前三位代表所有者权限，中间三位是所属组权限，后三位是其他用户权限。
- 其中rwx

r 读 数字用4代表

w 写 数字用2代表

x 执行 数字用1代表

- 权限一般也用数字表示，比如764，就是所有者 (u) rwx,所属组 (g) rw-, 其他用户 (o) r--

```
r+w+x=7
rwxrwxrwx=777
```

修改权限命令: chmod

chmod 更改权限 change mode 的缩写

- 语法: chmod 777 文件/目录名字

```
[root@aming1 ~]# ls -l 111.txt
```

```
-rw-r--r--. 1 root root 1008 5月 13 22:26 111.txt[root@aming1 ~]# chmod 777 111.txt
[root@aming1 ~]# ls -l 111.txt
-rwxrwxrwx. 1 root root 1008 5月 13 22:26 111.txt
```

- 或者: `chmod u=rwx,g=rw,o=r 文件/目录名字`

```
[root@aming1 ~]# ls -l 111.txt
-rw-r--r--. 1 root root 1008 5月 13 22:26 111.txt
[root@aming1 ~]# chmod 777 111.txt
[root@aming1 ~]# ls -l 111.txt
-rwxrwxrwx. 1 root root 1008 5月 13 22:26 111.txt
[root@aming1 ~]# chmod u=rwx,g=rw,o=r 111.txt
[root@aming1 ~]# ls -l 111.txt
-rwxrw-r--. 1 root root 1008 5月 13 22:26 111.txt
```

- `chmod -R`

操作只对目录或文件本身生效。如需目录及子文件子目录全部生效需要加-R

加上-R 命令可以同时更改目录及目录下所有的权限。

```
[root@aminglinux-01 ~]# chmod 700 /tmp/aming1/
[root@aminglinux-01 ~]# ls -l /tmp/
总用量 236
-rwxr-xr-x. 1 root root 117656 8月 9 09:19 111
drwx----- 3 root root 17 8月 9 09:23 aming1
drwxr-xr-x. 4 root root 31 8月 9 09:34 aming2

[root@aminglinux-01 ~]# ls -l /tmp/aming1/
总用量 0
drwxr-xr-x. 3 root root 30 8月 9 09:40 111
[root@aminglinux-01 ~]# chmod -R 777 /tmp/aming1/
[root@aminglinux-01 ~]# ls -l /tmp/aming1/
总用量 0
drwxrwxrwx. 3 root root 30 8月 9 09:40 111
[root@aminglinux-01 ~]# ls -l /tmp/
总用量 236
-rwxr-xr-x. 1 root root 117656 8月 9 09:19 111
drwxrwxrwx. 3 root root 17 8月 9 09:23 aming1
drwxr-xr-x. 4 root root 31 8月 9 09:34 aming2
```

- 还有一个用法是 `chmod a+x 文件/目录名字`

意思是u, g, o同时加上x权限。

a-x,就是全部去掉x权限。

也是可以，u+w, u-x, g+w, g-w等等都可以用。

```
[root@aminglinux-01 ~]# chmod 700 /tmp/aming1/

[root@aminglinux-01 ~]# ls -l /tmp/
总用量 236

drwx-----. 3 root root 17 8月 9 09:23 aming1


[root@aminglinux-01 ~]# chmod a+x /tmp/aming1/
[root@aminglinux-01 ~]# ls -l /tmp/
总用量 236

drwx--x--x. 3 root root 17 8月 9 09:23 aming1


[root@aminglinux-01 ~]# chmod a-x /tmp/aming1/
[root@aminglinux-01 ~]# ls -l /tmp/
总用量 236

drw-----. 3 root root 17 8月 9 09:23 aming1

[root@aminglinux-01 ~]#
```

chown 更改所有者和所属组

- chown = change owner 更改所有者
- chown 不仅可以更改所有者，也可以更改所属组。
- cat /etc/passwd 来查看系统用户都有哪些。
- 用法：chown 所有者:所属组 文件名/目录

下面修改一下新建文件的所有者。

```
[root@aminglinux-01 ~]# touch /tmp/lunyu.log
[root@aminglinux-01 ~]# ls -l /tmp/lunyu.log
-rw-r--r--. 1 root root 0 8月 9 21:45 /tmp/lunyu.log
[root@aminglinux-01 ~]# chown linyu /tmp/lunyu.log
[root@aminglinux-01 ~]# ls -l /tmp/lunyu.log
-rw-r--r--. 1 linyu root 0 8月 9 21:45 /tmp/lunyu.log
[root@aminglinux-01 ~]#
```

- 更改所属组命令：chgrp，用法同上。但是chown既可以更改所有者也可以同时或者单独更改所属组
- chown 同时或者单独更改所属组

```
[root@aminglinux-01 ~]# chown linyu:games /tmp/lunyu.log
[root@aminglinux-01 ~]# ls -l /tmp/lunyu.log
-rw-r--r--. 1 linyu games 0 8月 9 21:45 /tmp/lunyu.log

[root@aminglinux-01 ~]# chown :root /tmp/lunyu.log
[root@aminglinux-01 ~]# ls -l /tmp/lunyu.log
-rw-r--r--. 1 linyu root 0 8月 9 21:45 /tmp/lunyu.log
```

- chown -R 命令

同样可以一次性更改目录及子目录和文件的所有者和所属组。

umask

linux在创建一个文件或者目录时候都会有一个默认的权限。这个是怎么来的呢。运行一下umask

```
[root@aminglinux-01 ~]# umask0022
```

这个是root用户下的umask值，通过这个值可以确定文件以及目录默认是什么权限。

首先创建一个目录和文件。然后修改一下啊umask值，在创建一个目录和文件。然后对比一下看有什么联系。

```
[root@aminglinux-01 ~]# touch 123.txt[root@aminglinux-01 ~]# mkdir aaa[root@aminglinux-01 ~]#
ls -l
总用量 8
-rw-r--r--. 1 root root 0 8月 9 22:08 123.txtdrwxr-xr-x. 2 root root 6 8月 9 22:08 aaa
```

在0022时候，文件默认是：644，目录默认是：755

```
[root@aminglinux-01 ~]# umask 002
[root@aminglinux-01 ~]# umask
0002
```

```
[root@aminglinux-01 ~]# touch 456.txt[root@aminglinux-01 ~]# mkdir bbb[root@aminglinux-01 ~]#
ls -l
```

总用量 8

```
-rw-r--r--. 1 root root 0 8月 9 22:08 123.txt-rw-rw-r--. 1 root root 0 8月 9 22:14
456.txt-drw-r-xr-x. 2 root root 6 8月 9 22:08 aaadrwxrwxr-x. 2 root root 6 8月 9 22:14
bbb[root@aminglinux-01 ~]#
```

在0002时候，文件默认是：664，目录默认是：775

- 得出答案是：

在0022时候，文件默认是：644，目录默认是：755

在0002时候，文件默认是：664，目录默认是：775

首先目录，必须会有x权限，因为想查看这个目录，首先都要进入查看，所以有x权限。所以默认权限最大的话就是7.文件不需要必须有x权限，所以默认6就可以满足。

由此规律得出：目录：777-022=755
777-002=775
文件：666-022=644
666-002=664

- 这个规律合乎逻辑但是不一定是准确的。
- 如果我们设置umask 为 003。按照之前的算法666-003=663。实际上经过测试是664

```
[root@aminglinux-01 ~]# umask 003
[root@aminglinux-01 ~]# touch 888.txt
[root@aminglinux-01 ~]# ls -l 888.txt
-rw-rw-r--. 1 root root 0 8月 9 22:27 888.txt
[root@aminglinux-01 ~]#
```

- 那么之前的算法就是有漏洞的。
- 正确的算法是：

文件 666 (rw-rw-rw-) - 003(-----wx)=rw-rw-r-- 也就是664

目录 777 (rwxrwxrwx) - 002(-----w-)=rwxrwxr-x 也就是775

