

第1章 Linux系统权限介绍

1.什么是权限

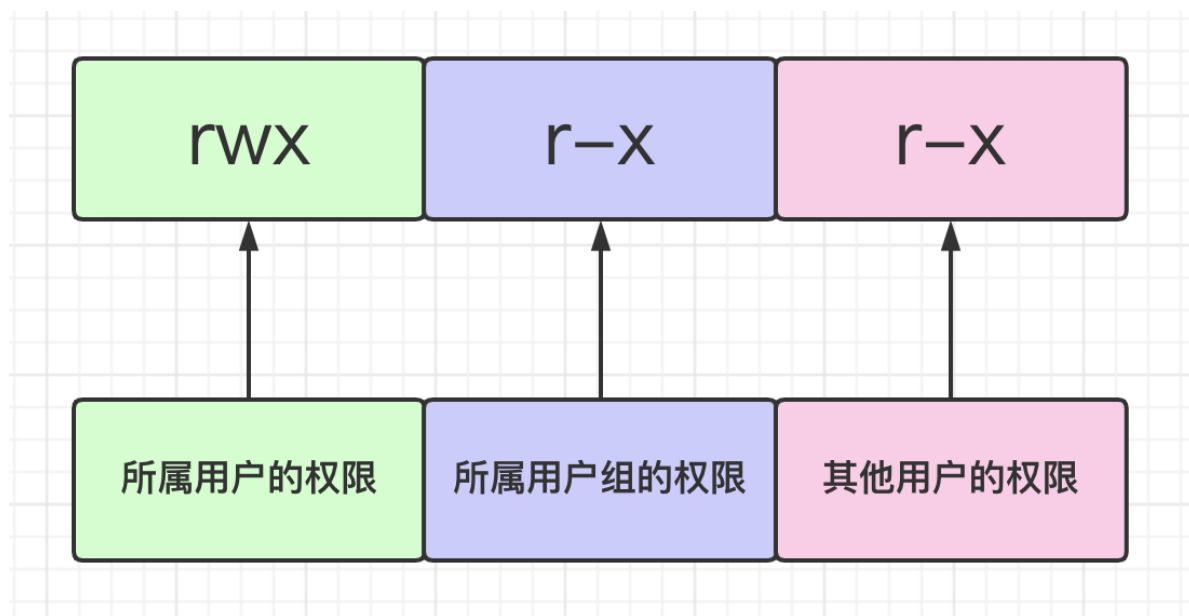
- 1 现实生活中我们每个人都有国家赋予的公民权利，国家的法律规定了哪些事我们可以做，哪些事不能做。
- 2 同样，在Linux系统中，如果我们想限制某个用户对系统的操作，就可以通过设置不同的权限达到我们想要的效果。

2.用户和用户组和权限的关系

- 1 通过前面用户管理我们知道，Linux里针对文件定义了三种角色，分别是所属用户(owner)，所属组(group)，其他用户(others)，每一种角色又对应了三种权限，分别是可读，可写，可执行。
- 2
- 3 当用户访问文件时，流程如下：
 - 1.如果是文件的所用者，则按所有者的权限进行操作。
 - 2.如果是文件的所属组，则按所属组的权限进行操作
 - 3.如果是文件不是所属用户，也不是所属组，则按其他用户的权限进行操作。

3.文件权限位说明

- 1 在Linux中，每个文件或目录都有一组共9个基础权限位，每三位字符分为一组，它们分别是所属用户权限位、用户组权限位、其他用户权限位。最终形式如“rw-r--r--”。
- 2 Linux正是通过这9个权限位来控制文件用户、用户组以及其他用户对文件的访问权限的。



4.rwx权限的含义

字母权限	数字权限	权限含义
r (read)	4	读取权限
w (write)	2	写入权限
x (execut)	1	执行权限
-	0	没有权限

在Linux里，默认创建文件和目录的权限不一样，大家可以尝试用数字表达法说一下。

第2章 Linux文件及目录权限

1.文件权限和目录权限区别

虽然文件和目录的权限都是rwx三种，但是在Linux系统中对文件和目录的权限是有不同含义和区别的。

权限	对文件的作用	对目录的作用
r 读取权限	具有读取，浏览文件内容的权限	具有浏览目录及其子目录的权限
w 写入权限	具有增加，修改，删除文件内容的权限	具有增加，删除或修改目录内文件的权限
x 执行权限	具有执行文件的权限	具有进入目录的权限

2.文件权限实验

涉及到的命令：

```
1 | chmod [ugoa] [-+=] file
```

实验流程：

```
1 | 使用普通用户测试以下权限的文件：
2 | ---
3 | r--
4 | rw-
5 | r-x
6 | -w-
7 | -wx
8 | --x
9 | rwx
```

实验过程：--- 权限测试

```
1 | #1.root用户去掉文件的其他用户可读权限
2 | [root@linux dir]# chmod o-r file.txt
3 | [root@linux dir]# ll
4 | 总用量 4
5 | -rw-r----- 1 root root 21 3月 28 21:27 file.txt
6 |
7 | #2.切换到普通用户查看文件权限
```

```

8 [root@linux dir]# su - oldya
9 [oldya@linux ~]$ cd /opt/dir
10 [oldya@linux dir]$ ll
11 总用量 4
12 -rw-r----- 1 root root 21 3月 28 21:27 file.txt
13
14 #3.普通用户测试是否可读可写
15 [oldya@linux dir]$ cat file.txt #不能查看因为没有r权限
16 cat: file.txt: 权限不够
17
18 #4.测试是否可以写入内容 #不能修改因为没有w权限
19 [oldya@linux dir]$ echo 123 > file.txt
20 -bash: file.txt: 权限不够

```

实验过程：r- 权限测试

```

1 #1.root用户创建文件
2 [root@linux ~]# mkdir /opt/dir -p
3 [root@linux ~]# cd /opt/dir
4 [root@linux dir]# echo hello oldboy > file.txt
5 [root@linux dir]# ll
6 总用量 4
7 -rw-r--r-- 1 root root 13 3月 28 18:40 file.txt
8 [root@linux dir]# cat file.txt
9 hello oldboy
10
11 #2.切换到普通用户测试是否可读
12 [root@linux dir]# su - oldboy
13 [oldboy@linux ~]$ cd /opt/dir
14 [oldboy@linux dir]$ ll
15 总用量 4
16 -rw-r--r-- 1 root root 13 3月 28 18:40 file.txt
17
18 #3.测试是否可以查看
19 [oldboy@linux dir]$ cat file.txt #因为有r权限，所以可以查看
20 hello oldboy
21
22 #3.普通用户测试是否可修改
23 [oldboy@linux dir]$ vim file.txt #因为没有w权限，所以不能修改
24
25 #4.普通用户测试是否可以删除
26 [oldya@linux dir]$ rm -rf file.txt #这里不能删除是因为上层目录没有w权限
27 rm: 无法删除"file.txt": 权限不够

```

实验过程：rw- 权限测试

```

1 #1.root用户创建测试文件添加rw权限
2 [root@linux ~]# cd /opt/dir
3 [root@linux dir]# chmod o+rw file.txt
4 [root@linux dir]# ll
5 总用量 4
6 -rw-r--rw- 1 root root 21 3月 28 21:27 file.txt
7
8 #2.切换到普通用户查看文件属性
9 [root@linux dir]# su - oldya
10 [oldya@linux ~]$ cd /opt/dir

```

```

11 [oldya@linux dir]$ ll
12 总用量 4
13 -rw-r--rw- 1 root root 21 3月 28 21:27 file.txt
14
15 #3.测试是否可以读取文件
16 [oldya@linux dir]$ cat file.txt      #可以读取，因为有r权限
17 echo "hello oldboy!"
18
19 #4.测试是否可以修改文件
20 [oldya@linux dir]$ cat file.txt      #可以修改，因为有w权限
21 echo "hello oldboy!"
22 [oldya@linux dir]$ vim file.txt
23 [oldya@linux dir]$ cat file.txt
24 echo "hello oldboy!"
25 echo "v2"
26
27 #4.测试是否可以删除
28 [oldya@linux dir]$ rm -rf file.txt   #不能删除，因为上层目录没有w权限
29 rm: 无法删除"file.txt": 权限不够

```

实验过程：r-x权限测试

```

1  #1.使用root用户创建具有执行权限的文件并测试
2  [root@linux dir]# chmod 645 file.txt
3  [root@linux dir]# ll
4  总用量 4
5  -rw-r--r-x 1 654 root 34 3月 29 08:37 file.txt
6  [root@linux dir]# cat file.txt
7  echo "hello szoldboy!"
8  [root@linux dir]# bash file.txt
9  hello szoldboy!
10 [root@linux dir]# ./file.txt
11 hello szoldboy!
12
13 #2.使用普通用户切换并测试
14 [root@linux dir]# su - oldya
15 [oldya@linux ~]$ cd /opt/dir
16 [oldya@linux dir]$ ./file.txt      #可以执行因为有x权限
17 hello szoldboy!
18 [oldya@linux dir]$ bash file.txt   #可以执行因为有x权限
19 hello szoldboy!

```

实验过程：-w- 权限测试

```

1  #1.root用户取消文件的r权限，添加w权限
2  [root@linux dir]# chmod o-r file.txt
3  [root@linux dir]# chmod o+w file.txt
4  [root@linux dir]# ll
5  总用量 4
6  -rw-r---w- 1 root root 31 3月 29 08:32 file.txt
7
8  #2.切换到普通用户查看文件属性
9  [root@linux dir]# su - oldya
10 [oldya@linux ~]$ cd /opt/dir
11 [oldya@linux dir]$ ll
12 总用量 4

```

```

13 -rw-r---w- 1 root root 31 3月 29 08:32 file.txt
14
15 #3.测试是否可以读取
16 [oldya@linux dir]$ cat file.txt      #不能查看因为没有r权限
17 cat: file.txt: 权限不够
18
19 #4.测试是否可以修改
20 [oldya@linux dir]$ echo "v3" >> file.txt    #可以直接使用echo写入，因为有w权限
21
22 #5.测试是否可以读取
23 [oldya@linux dir]$ cat file.txt          #不能读取，因为没有r权限
24 cat: file.txt: 权限不够
25
26 #6.测试是否可以删除
27 [oldya@linux dir]$ rm -rf file.txt        #不能删除，因为上层目录没有w权限
28 rm: 无法删除"file.txt": 权限不够

```

实验过程：-wx权限测试

```

1  #1.root用户创建wx权限的文件
2  [root@linux dir]# chmod o=wx file.txt
3  [root@linux dir]# ll
4  总用量 4
5  -rw-r---wx 1 654 root 23 3月 29 15:46 file.txt
6
7  #2.切换到普通用户查看文件信息
8  [root@linux dir]# su - oldya
9  [oldya@linux ~]$ cd /opt/dir
10 [oldya@linux dir]$ ll
11 总用量 4
12 -rw-r---wx 1 654 root 23 3月 29 15:46 file.txt
13
14 #3.测试读取文件
15 [oldya@linux dir]$ cat file.txt      #不能查看，因为没有r权限
16 cat: file.txt: 权限不够
17
18 #4.测试修改权限
19 [oldya@linux dir]$ echo 'echo v2' >> file.txt    #可以修改，因为有w权限
20
21 #5.测试读取权限
22 [oldya@linux dir]$ bash file.txt      #因为没有r权限，所以即使有x权限也不能
    执行
23 bash: file.txt: 权限不够
24 [oldya@linux dir]$ ./file.txt
25 bash: ./file.txt: 权限不够

```

实验过程：--x权限测试

```

1 #1.使用root创建只有x权限，没有rw权限的文件
2 [root@linux dir]# chmod 641 file.txt
3 [root@linux dir]# ll
4 总用量 4
5 -rw-r----x 1 654 root 23 3月 29 15:46 file.txt
6
7 [root@linux dir]# su - oldya
8 [oldya@linux ~]$ cd /opt/dir
9 [oldya@linux dir]$ ./file.txt #因为没有r权限，所以即使有x权限也不能执行
10 bash: ./file.txt: 权限不够
11 [oldya@linux dir]$ bash file.txt #因为没有r权限，所以即使有x权限也不能执行
12 bash: file.txt: 权限不够

```

3.总结-文件权限需要注意的知识点

对于r读取权限需要注意的地方：

- 1 如果没有可读r的配合，那么使用vim编辑文件时会提示无法编辑(但可强制编辑覆盖文件)，可以使用echo等命令进行重定向或追加内容到文件。

对于w写入权限需要注意的地方：

- 1 删除、移动或创建文件等的权限是受父目录(上一级目录)的权限控制的(因为文件名没有存放在Inode里，而是在上级目录的block里存放着的，若修改上级目录的block(删除文件本体)，当然会受到上级目录的Inode的权限控制)，与文件本身的权限无关，因此，文件本身的可写w权限，与文件是否能被删除和改名无关。

对于x执行权限需要注意的地方：

- 1 首先文件的本身要能够执行(文件为命令或脚本)。
- 2 如果是普通用户，同时还需要具备可读r的权限才能执行文件。
- 3 而root用户只要有可执行x的权限就能执行文件。

4.目录权限实验

实验流程：

- 1 分别创建一下权限的目录进行测试：
- 2 r--
- 3 rw-
- 4 r-x
- 5 -w-
- 6 -wx
- 7 --x
- 8 rwx

实验步骤：r-- 权限测试

- 1 #1.root用户修改目录权限只有r权限
- 2 [root@linux opt]# chmod 754 dir
- 3 [root@linux opt]# ls -ld dir
- 4 drwxr-xr-- 2 oldboy oldboy 22 3月 29 17:29 dir
- 5

```

6 #2.切换到普通用户
7 [root@linux opt]# su - oldya
8
9 #3.测试可否进入目录
10 [oldya@linux ~]$ cd /opt/dir          #这里进入不了目录是因为没有x权限
11 -bash: cd: /opt/dir: 权限不够
12
13 #4.测试是否可以浏览目录下文件
14 [oldya@linux ~]$ ll /opt/dir
15 ls: 无法访问/opt/dir/file.txt: 权限不够      #这里显示不正常因为没有x权限的配合
16 总用量 0
17 -????????? ? ? ? ?      ? file.txt

```

实验步骤: rw- 权限测试

```

1 #1.创建rw-权限的目录
2 [root@linux opt]# chmod o=rw dir
3 [root@linux opt]# ll -d dir
4 drwxr-xrw- 2 oldboy oldboy 6 3月 29 19:37 dir
5
6 #2.切换到普通用户
7 [root@linux opt]# su - oldya
8
9 #3.测试可否切换到目录
10 [oldya@linux ~]$ cd /opt/dir
11 -bash: cd: /opt/dir: 权限不够
12
13 #4.测试可否浏览目录下文件
14 [oldya@linux ~]$ ll /opt/dir          #虽然可以看到文件名，但是没有x权
    限，文件名显示有问题
15 ls: 无法访问/opt/dir/file.txt: 权限不够
16 总用量 0
17 -????????? ? ? ? ?      ? file.txt
18
19 #5.测试可否查看文件内容
20 [oldya@linux ~]$ cat /opt/dir/file.txt
21 cat: /opt/dir/file.txt: 权限不够
22
23 #6.测试可否删除目录下文件
24 [oldya@linux ~]$ rm -rf /opt/dir/file.txt
25 rm: 无法删除"/opt/dir/file.txt": 权限不够

```

实验步骤: r-x 权限测试

```

1 #1.更改目录属性为oldboy
2 [root@linux opt]# chown -R oldboy:oldboy /opt/dir
3 [root@linux opt]# ll -d dir
4 drwxr-xr-x 2 oldboy oldboy 22 3月 29 17:29 dir
5
6 #2.切换到普通用户
7 [root@linux opt]# su - oldya
8
9 #3.测试能否进入目录
10 [oldya@linux ~]$ cd /opt/dir          #这里之所以能进入目录，是因为有r权限和x权限
11
12 #3.测试能否查看目录下文件

```

```

13 [oldya@linux dir]$ ll                                     #这里之所以能看到文件，是因为有r权限
14 总用量 4
15 -rw-r--r-- 1 oldboy oldboy 10 3月 29 17:29 file.txt
16
17 #4.测试能否创建文件
18 [oldya@linux dir]$ touch oldya.txt #这里创建不了文件是因为没有w权限
19 touch: 无法创建"oldya.txt": 权限不够
20
21 #5.测试能否删除文件
22 [oldya@linux dir]$ rm -rf file.txt #这里删除不了是因为没有w权限
23 rm: 无法删除"file.txt": 权限不够

```

实验步骤: -w- 权限测试

```

1 #1.使用root用户修改目录权限为-w-
2 [root@linux opt]# chmod o=w dir
3 [root@linux opt]# ll -d dir
4 drwxr-x-w- 2 oldboy oldboy 22 3月 29 20:32 dir
5
6 #2.使用普通用户登陆
7 [root@linux opt]# su - oldya
8
9 #3.测试可否切换目录
10 [oldya@linux ~]$ cd /opt/dir #切换不了目录因为没有x权限
11 -bash: cd: /opt/dir: 权限不够
12
13 #4.测试可否查看目录下文件列表
14 [oldya@linux ~]$ ll /opt/dir #查看不了因为没有r权限
15 ls: 无法打开目录/opt/dir: 权限不够
16
17 #5.测试可否创建文件
18 [oldya@linux ~]$ touch /opt/dir/file2.txt #虽然有w权限，但是没有x权限
19 配合依然不能创建文件
20 touch: 无法创建"/opt/dir/file2.txt": 权限不够
21
22 #6.测试可否读取文件内容 #查看不了文件内容因为没有r权限
23 [oldya@linux ~]$ cat /opt/dir/file.txt
24 cat: /opt/dir/file.txt: 权限不够
25
26 #7.测试可否删除文件 #删除不了因为没有w权限
27 [oldya@linux ~]$ rm -rf /opt/dir/file.txt
28 rm: 无法删除"/opt/dir/file.txt": 权限不够

```

实验步骤: -wx 权限测试

```

1 #1.使用root用户修改目录权限为-wx
2 [root@linux opt]# chmod o=wx dir
3 [root@linux opt]# ls -ld dir
4 drwxr-x-wx 2 oldboy oldboy 22 3月 29 17:29 dir
5
6 #2.使用普通用户登陆
7 [root@linux opt]# su - oldya
8
9 #3.普通用户测试能否切换目录
10 [oldya@linux ~]$ cd /opt/dir #这里可以登陆是因为目录具备x权限
11 [oldya@linux dir]$

```



```

12
13 #4.普通用户测试能否浏览文件内容
14 [oldya@linux dir]$ ls -l .           #这里看不到是因为没有r权限
15 ls: 无法打开目录.: 权限不够
16
17 #5.普通用户测试能否修改文件内容
18 [oldya@linux dir]$ echo v3 > file.txt      #这里不能写入是因为没有w权限
19 -bash: file.txt: 权限不够
20
21 #6.普通用户测试能否删除文件
22 [oldya@linux dir]$ rm -rf file.txt         #这里可以删除是因为具备w权限

```

实验步骤: --x 权限测试

```

1 #1.使用root用户创建一个没有r权限的目录
2 [root@linux opt]# chmod 751 dir
3 [root@linux opt]# ll -d dir
4 drwxr-x--x 2 oldboy oldboy 22 3月 29 17:29 dir
5
6 #2.切换到普通用户测试可否进入目录
7 [oldya@linux ~]$ cd /opt/dir      #这里可以进入目录是因为有x权限
8
9 #3.测试可以查看目录下文件          #这里看不到文件是因为没有r权限
10 [oldya@linux dir]$ ls -l
11 ls: 无法打开目录.: 权限不够
12
13 #4.测试是否可以编辑文件
14 [oldya@linux dir]$ echo v2 > file.txt      #这里不能修改是因为没有w权限
15 -bash: file.txt: 权限不够
16
17 #5.测试是否可以删除文件
18 [oldya@linux dir]$ rm -rf file.txt         #这里删除不了是因为没有w权限
19 rm: 无法删除"file.txt": 权限不够

```

5.总结-目录权限需要注意的知识点

对于r读取权限需要注意的地方:

- 1 具备r读取权限可执行ls dir命令查看目录下的内容。
- 2 但是如果如果没有可执行x权限的配合, 则不能切换到指定目录里, 即无法执行cd dir命令, 而且在执行ls命令查看列表时, 虽然可以看到所有的文件名, 但是会提示无权访问目录下的文件。
- 3 执行ls -l命令查看列表目录下的文件属性时, 所有文件对应的属性都会带有问号, 也会提示无权访问目录下的文件, 但是依然可以看到所有的文件名。

对于w写入权限需要注意的地方:

- 1 如果没有可执行x权限的配合, 即使有w权限, 也无法删除或创建文件。

对于x执行权限需要注意的地方:

- 1 可以执行cd dir 命令切换到目录下, 但是如果如果没有r就无法列表目录下的文件及子目录。

第3章 Linux权限命令

1.chmod 更改文件或目录的权限

命令说明：

- 1 | `chmod`命令是用来改变文件或目录权限的命令。
- 2 | 但只有文件的属主和超级用户`root`才能够执行这个命令。

命令格式：

`chmod`支持两种修改权限的模式一种是字母表达，一种是数字表达

- 1 | `chmod [ugoa] [-+=] [rwx] [文件或目录]`
- 2 | `chmod [7|6|5|4|3|2|1] [文件或目录]`

用户位说明:

- | | | |
|---|----------------|-----------------------|
| 1 | <code>u</code> | 所属用户 |
| 2 | <code>g</code> | 所属组 |
| 3 | <code>o</code> | 其他用户 |
| 4 | <code>a</code> | 代表所有 <code>ugo</code> |

操作字符说明：

- | | | |
|---|----------------|------------------|
| 1 | <code>-</code> | 取消权限 |
| 2 | <code>+</code> | 添加权限 |
| 3 | <code>=</code> | 取消所有权限，然后赋予给定的权限 |

关键参数：

- 1 | `-R` 递归处理指定目录及其子目录下的所有文件

命令实践：

- 1 | `chmod 755 /opt/dir`
- 2 | `chmod u+x /opt/dir`
- 3 | `chmod g-x /opt/dir`
- 4 | `chmod o=rwx /opt/dir`
- 5 | `chmod a+x /opt/dir`

2.chown 更改文件或目录的所属用户及用户组

命令语法：

- 1 | `chown [选项] [用户:用户组] [文件或目录]`

常用写法：

- 1 | `#同时更改用户和用户组`
- 2 | `chown oldboy:oldboy /opt/oldboy`
- 3 | `chown oldboy.oldboy /opt/oldboy`
- 4 |
- 5 | `#这种写法也是同时修改用户和组`

```
6 chown oldboy: /opt/oldboy
7 chown oldboy. /opt/oldboy
8
9 #只修改用户组
10 chown :oldboy /opt/oldboy
11 chown .oldboy /opt/oldboy
12
13 #递归修改目录及子目录的权限
14 chown -R oldboy:oldboy /opt/oldboy
```