

linux12位权限体系

原创

GeorgeKai

2017-12-29 19:07:03

评论(0)

334人阅读

作者: Georgekai
 归档: 学习笔记
 2017/12/29

本章正题: linux9位权限体系

1.1 linux的权限

注意: root用户默认对所有文件都有rw权限, 但没有x权限

1.1.1 rwx权限的含义

r read 4
 w write 2
 x execute 对文件来说如: 命令 脚本

1.1.2 对于一个文件linux的用户分类

```
[george@georgekai ~]$ ls -l /etc/hosts
-rw-r--r--. 2 root root 179 Dec 20 05:32 /etc/hosts
```

rw- r-- r---
 所有者 属组 其他人

1.1.3 如何知道george 用户对/etc/hosts文件有什么权限?

1. 我是谁 whoami
2. 我与她什么关系 主人 家人 其他人 id george
 [george@georgekai ~]\$ id george
 uid=500(george) gid=501(kai) groups=501(kai)
3. 如果是其它人, 那么其它人对应的权限是什么 ls -l/etc/hosts

注: 文件默认的权限644 rw-r--r--

目录默认的权限 755 rwxr-xr-x

命令默认的权限 644 rwxr-xr-x 如: /bin/ls

日志默认的权限 600 rx----- 如: /var/log/secure

1.1.4 修改文件的权限 chmod

1. chmod ==> change mode
2. 用户表示方法:

u user 用户
g group 组
o other 其他人

1. 增加权限

```
chmod u+x oldboy.sh
```

2. 去掉权限

```
chmod u-x oldboy.sh
```

3. 先去掉所有权限，在增加新的权限

```
chmod u=x oldboy.sh
```

4. 给主人、家庭、其他人都增加x权限

```
chmod +x oldboy.sh chmod a+x oldboy.sh 或者 chmod ugo+x oldboy.sh
```

注：oug = a ， 什么都不写只对x和r管用（常用）

5. 递归修改目录及目录内容权限

```
chmod -R +x oldboy.sh
```

注：-R 递归修改目录及目录下的内容权限

例 1：给oldboy.sh 都增加x权限

```
[root@georgekai ~]# chmod +x oldboy.sh
[root@georgekai ~]# ls -l oldboy.sh
-rwxr-xr-x. 1 root root 0 Dec 29 09:26 oldboy.sh
```

1.1.5 更改文件的所有者 chown

例1：修改文件的所有者和属组

```
[root@georgekai ~]# chown george.george oldboy.sh
[root@georgekai ~]# ls -l oldboy.sh
-rwxr-xr-x. 1 george george 0 Dec 29 09:26 oldboy.sh
```

注：1. 属组可以用 .george 表示，也可以用 : george表示

如：chown .george oldboy.sh

2. 所有者直接用 用户名 表示

如：chown george oldboy.sh

3. chown ==> change owner

1.2 对文件来说rwx权限的含义

准备环境：

```
mv /oldboy /tmp/oldboy_bak$(date +%F)
mkdir /oldboy -p
echo "echo oldboylinux" >/oldboy/test.sh
chmod +x /oldboy/test.sh
cat /oldboy/test.sh
ls -l /oldboy/test.sh
```

1.2.1 单独测试文件的r权限

```
[george@georgekai oldboy]$ ls -l test.sh
-r--r-xr-x. 1 george george 13 Dec 29 10:50 test.sh
[george@georgekai oldboy]$ cat test.sh
hostname
pwd
```

1.2.2 单独测试文件的w权限：

```
[root@georgekai oldboy]# ls -l test.sh
--w-r-xr-x. 1 george george 17 Dec 29 10:12 test.sh
```

注：1. w修改文件的内容 只有w权限，文件的所有者vim强制修改文件的内容，会导致原内容被覆盖，但是echo可以写入追加。

2. 所以：w需要r 的配合， 也就是说对于文件来说想要用vim修改至少需要有rw权限

1.2.3 单独测试文件的x权限

```
[george@georgekai oldboy]$ ls -l test.sh
---x-r-xr-x. 1 george george 6 Dec 29 10:30 test.sh
[george@georgekai oldboy]$ /oldboy/test.sh
bash: /oldboy/test.sh: Permission denied
```

注：1. x需要r的配合，想要执行必须要知道文件里面的内容，所以用x必须有r才能执行

2. 测试执行时需要输入绝对路径，否则会被系统认为是命令

1.2.4 同时属于所有者、属组，所有者的权限生效！

```
[oldboy@oldboyedu43-lnb oldboy]$ ls -l test.sh
--w-r-xr-x 1 oldboy oldboy 17 Dec 28 14:39 test.sh
[oldboy@oldboyedu43-lnb oldboy]$ cat test.sh
cat: test.sh: Permission denied @51CTO博客
```

注：1. 属组有r权限，但是cat却不能查看

2. 如果同时属于所有者，属组、其他人，有效权限是所有者（属组和其他人的权限忽略）

文件的rwx的含义小结：

1. r 可以查看文件的内容
2. w 可以修改文件的内容 需要r权限配合
3. x 可以运行/执行 脚本/命令 需要r权限的配合

1.3 对于目录来说rwx权限的含义

r 查看目录里面内容 ls

w 可以在目录中创建、删除、重命名文件

x 表示是否可以进入到目录中 cd

1.3.1 单独测试目录的r权限

```
[george@georgekai oldboy]$ ls -ld oldboydir/
dr--r-xr-x. 2 george george 4096 Dec 29 11:16 oldboydir/
[george@georgekai oldboy]$ ls -l oldboydir/
ls: cannot access oldboydir/10.txt: Permission denied
ls: cannot access oldboydir/8.txt: Permission denied
```

注：1. 目录r权限，可以查看目录里面的文件名（文件名存放在文件所在目录的block中），与权限没关系。但文件内容的属性看不到（属性存放在文件所在目录的Inode中）。

2. 对于r权限需要x权限配合

3. x权限是否能查看/修改目录中文件的属性信息

1.3.2 单独测试目录的w权限

```
[george@georgekai oldboy]$ ls -ld oldboydir/
d-w-r-xr-x. 3 george george 4096 Dec 29 11:40 oldboydir/
[george@georgekai oldboy]$ touch oldboydir/kai.txt
touch: cannot touch `oldboydir/kai.txt': Permission denied
```

注：1. 对于w权限需要x权限配合，否则无法进行，创建、删除、重命名文件

1.3.3 单独测试目录的x权限

```
[george@georgekai oldboy]$ ls -ld oldboydir/
d--x-r-xr-x. 3 george george 4096 Dec 29 11:40 oldboydir/
[george@georgekai oldboy]$ cd oldboydir/
[george@georgekai oldboydir]$ ls
ls: cannot open directory .: Permission denied
```

注：1. 是能进入cd到目录中，但无法查看

目录的rwx的含义小结：

1. r 可以查看目录下的内容 需要x的配合
2. w 可以创建、删除、重命名文件 需要x的配合
3. x 可以cd到目录下 需要rw的配合，否则没什么卵用

1.4 特别需要注意：删除一个文件，需要对文件所在的目录拥有wx权限。

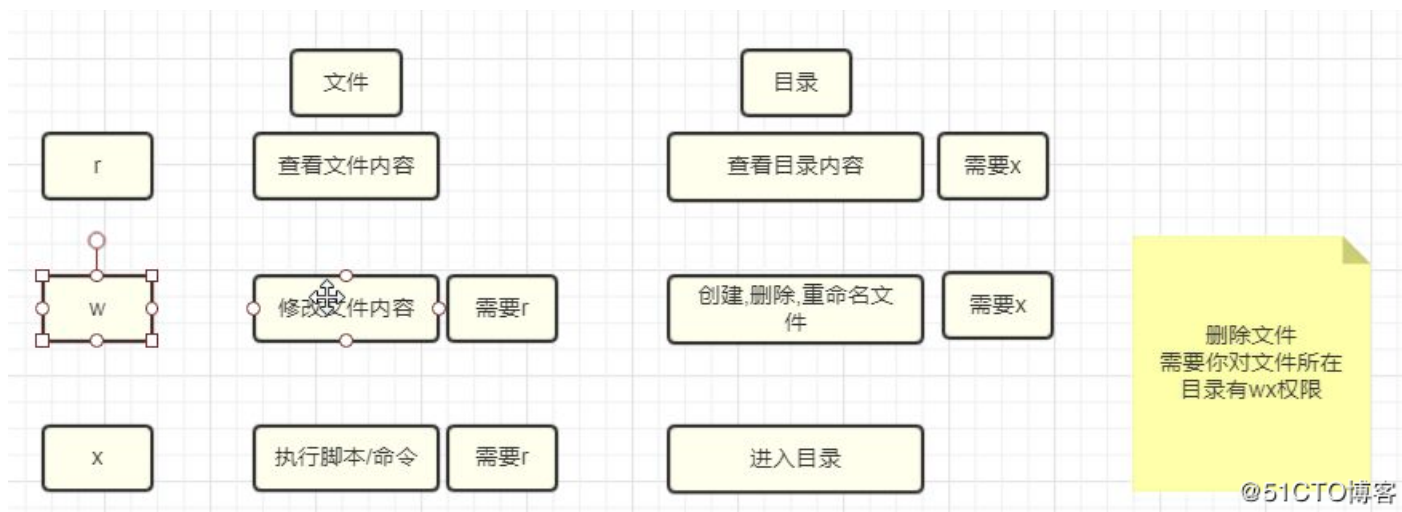
文件： 1. Inode 中存放的是文件的属性，如权限，block的位置

2. block 中存放的是文件的数据

目录： 1. Inode 中存放的是目录的属性，如权限，block的位置

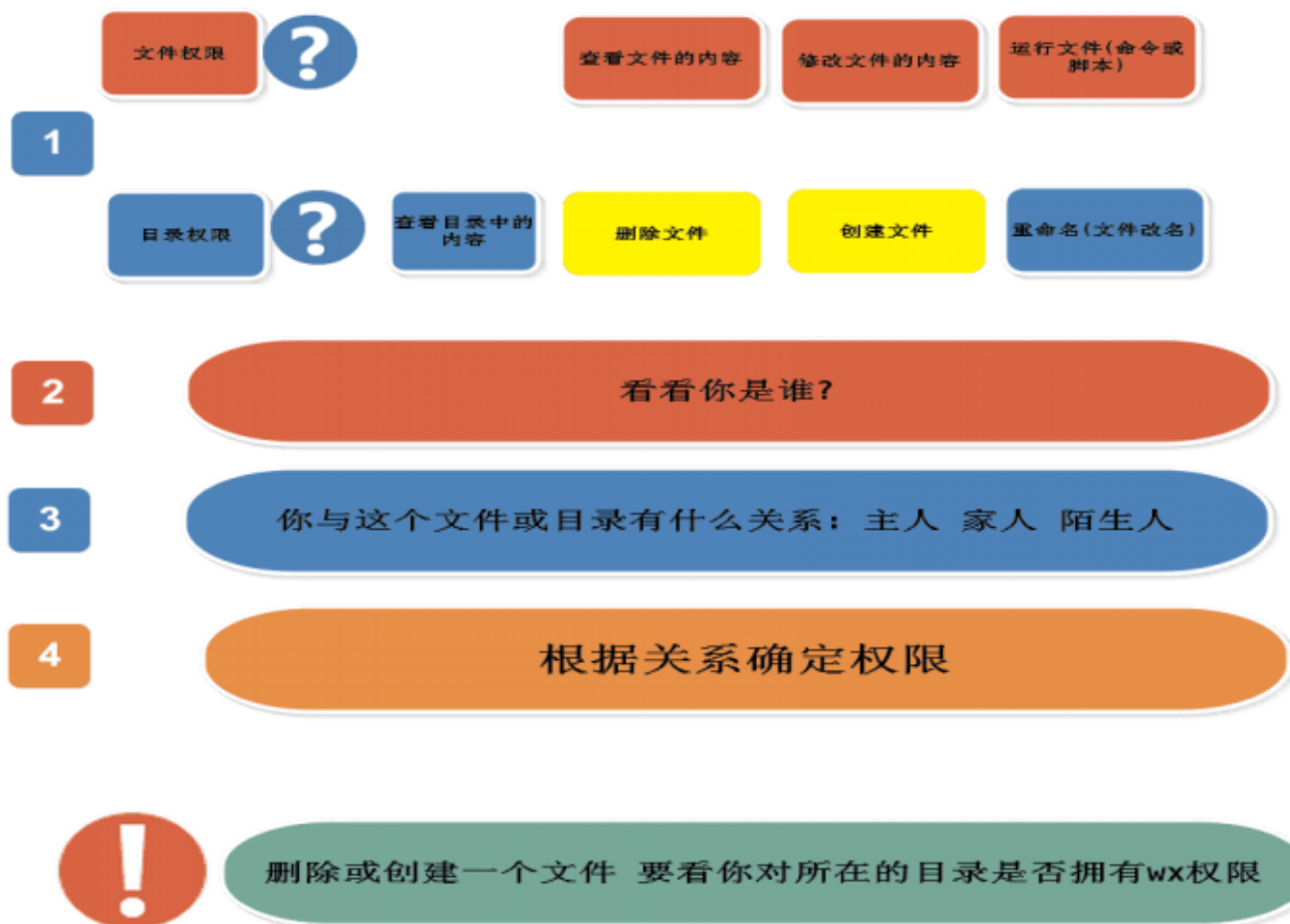
2. block 中存放的是目录下的文件名和文件名对应的Inode，vim 加目录名可查看

1.4.1 总结：文件和目录的权限

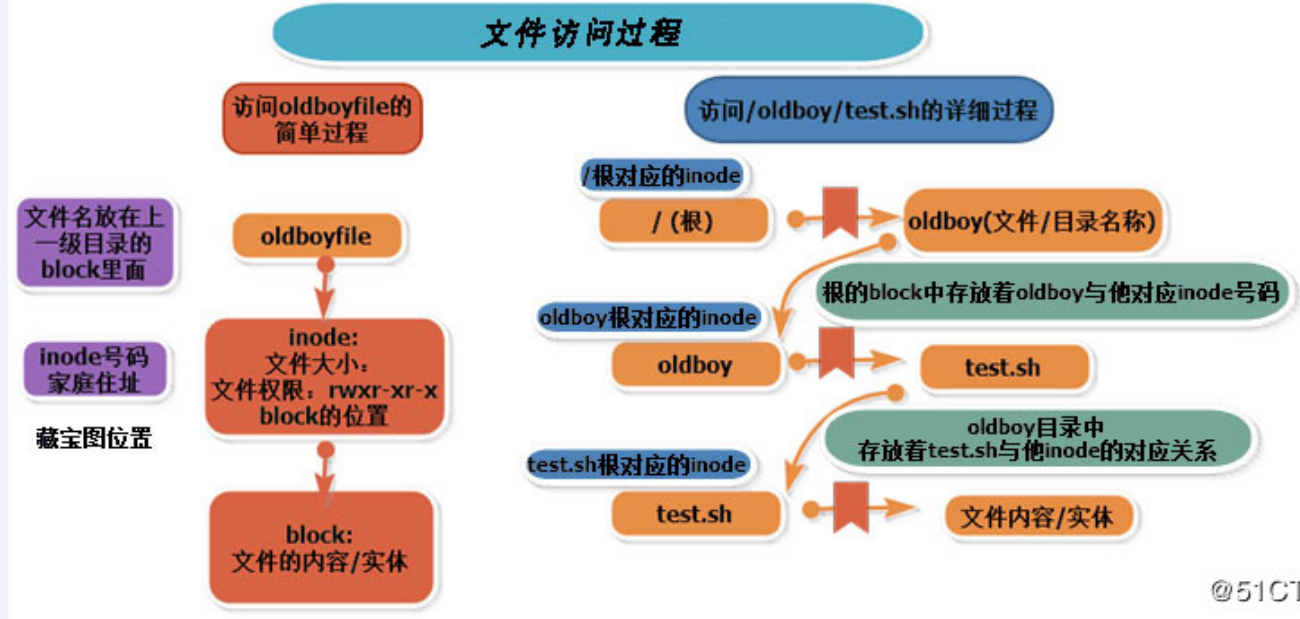


1.4.2 rm /etc/hosts 为什么权限不足?

```
[george@georgekai oldboy]$ rm /etc/hosts
rm: remove write-protected regular file `/etc/hosts'? y
rm: cannot remove `/etc/hosts': Permission denied
```



1.4.3 查看/oldboy/test.sh的过程：（是为了解决permission denied权限拒绝）



勤奋努力，善于总结！
每天学习一点点！

小伙伴们可以关注我的微信公众号：linux运维菜鸟之旅，更新比51cto慢一些，不过要方便许多



关注“中国电信天津网厅”公众号，首次绑定可免费领2G流量，为你的学习提供流量！



版权声明：原创作品，如需转载，请注明出处。否则将追究法律责任
