

第10章 权限和所有者



本章内容

10.1 权限设置

10.2 更改文件和目录所有者



10.1 权限设置

- 为了有效地控制用户对文件和目录的访问，必须对其设置权限以实现安全控制，本节主要讲述如何设置权限以及更改文件和目录的所有权。

文件和目录权限简介

- 在Linux系统中，用户对一个文件或目录具有访问权限，这些访问权限决定了谁能访问，以及如何访问这些文件和目录。通过设置权限可以限制或允许以下三种用户访问：文件的用户所有者（属主）、文件的组群所有者（用户所在组的同组用户）、系统中的其它用户。

- 在Linux系统中，每一位用户都有对文件或目录的读取、写入和执行权限。第一套权限控制访问自己的文件权限，即所有者权限。第二套权限控制用户组访问其中一个用户的文件的权限。第三套权限控制其它所有用户访问一个用户的文件的权限。这三套权限赋予用户不同类型（即用户所有者、组群所有者和其它用户）的读取、写入及执行权限，这就构成了一个有9种类型的权限组。


- 同时，用户能够控制一个给定的文件和目录的访问程度。一个文件和目录可能具有读、写入及执行权限。当创建一个文件时，系统会自动地赋予文件所有者读和写的权限，这样可允许所有者显示文件内容，和修改文件。文件所有者可以将这些权限更改为任何权限。一个文件也许只有读权限，禁止任何修改。一个文件也可能只有执行权限，允许它像一个程序一样执行。

r、w、x、-字符意义

- r**（读取）：对文件而言，该用户具有读取文件内容的权限；对目录来说，该用户具有浏览目录的权限；
- w**（写入）：对文件而言，该用户具有新增、修改文件内容的权限；对目录来说，该用户具有删除、移动目录内文件的权限；
- x**（执行）：对文件而言，该用户具有执行文件的权限；对目录来说，该用户具有进入目录的权限；
- ：表示不具有该项权限。

权限字符组合举例

举例	描述
-rwx-----	用户所有者对文件具有读取、写入和执行权限
-rwxr--r--	用户所有者具有读取、写入和执行权限，其它用户则具有读取权限
-rw-rw-r-x	用户所有者和组群所有者对文件具有读取、写入权限，而其它用户只具有读取和执行权限
drwx--x--x	目录的用户所有者具有读写和进入目录权限，其它用户能进入目录，却无法读取任何数据
drwx-----	除了目录的用户所有者具有所有的权限之外，其它用户对该目录没有任何权限

- 
- 用“ls -l”命令可以显示文件的详细信息，其中包括权限，如下所示：

```
[root@rhel ~]# ls -l /root
```

```
total 272
```

```
-rw-----. 1 root root 1816 Dec  7  2012 /root/anaconda-ks.cfg
```

```
-rw-r--r--. 1 root root 44934 Dec 11  2012 /root/install.log
```

```
-rw-r--r--. 1 root root 10151 Jul  2  2012 /root/install.log.syslog
```

-rw-r--r--. 1 root root 44934 Dec 11 2012 install.log

权限

链接数量

用户

组

大小

时间

文件名

-rw-r--r--.

文件类型

U权限

G权限

O权限

SElinux

文字设定法设置权限

- 通过文字设定法更改权限需要使用**chmod**命令，在一个命令行中可给出多个权限方式，其间用逗号隔开。

chmod的命令语法：

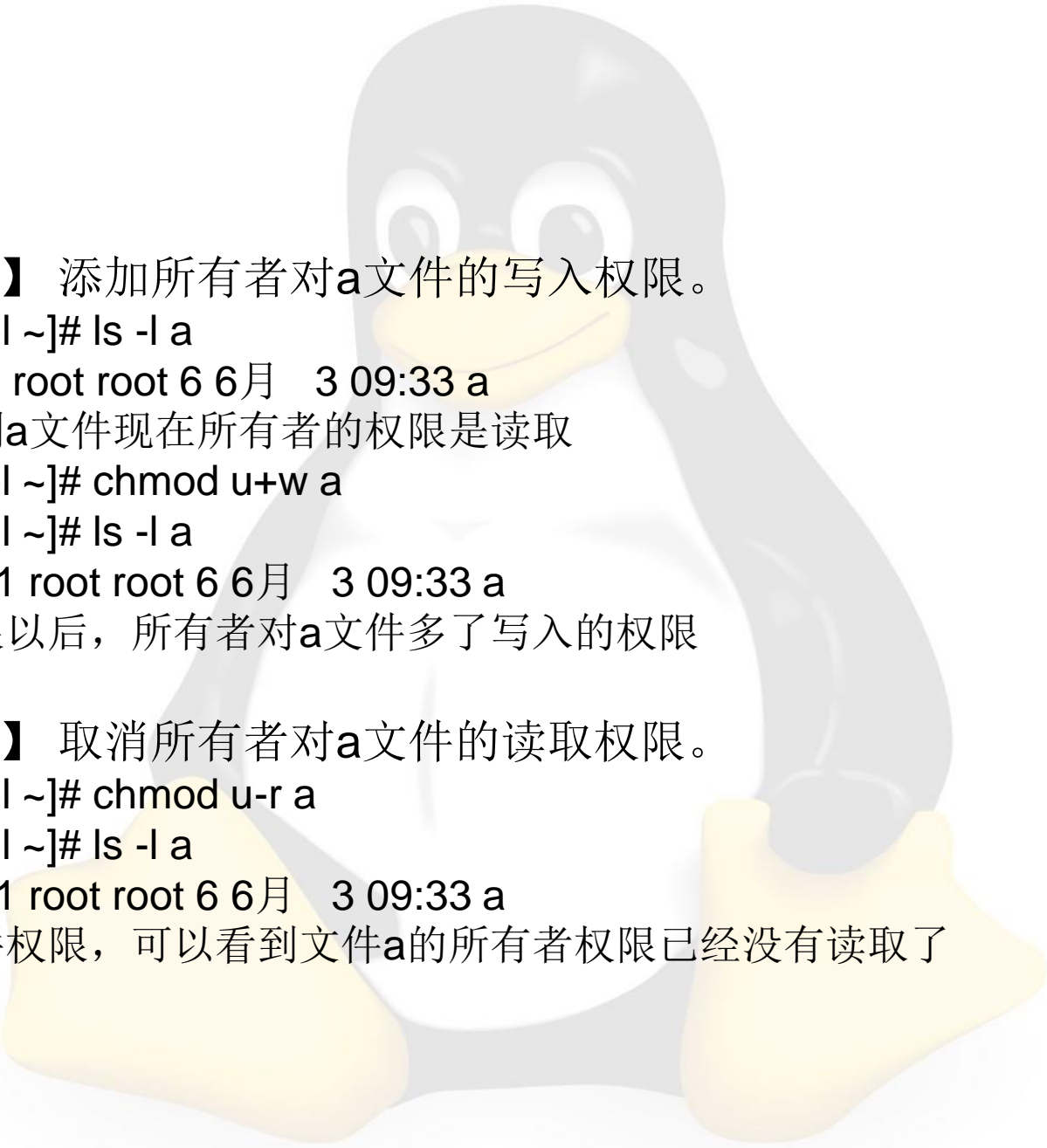
chmod [操作对象] [操作符号] [权限] [文件|目录]

操作对象

选项	选项含义
u	表示用户所有者，即文件或目录的所有者
g	表示组群所有者，即与文件的用户所有者有相同组群 GID 的所有用户
o	表示其它用户
a	表示所有用户，它是系统默认值

操作符号

选项	选项含义
+	添加某个权限
-	取消某个权限
=	赋予给定权限并取消原先权限（如果有的话）



【例10.1】 添加所有者对a文件的写入权限。

```
[root@rhel ~]# ls -l a
```

```
-r--r--r--. 1 root root 6 6月  3 09:33 a
```

//可以看到a文件现在所有者的权限是读取

```
[root@rhel ~]# chmod u+w a
```

```
[root@rhel ~]# ls -l a
```

```
-rw-r--r--. 1 root root 6 6月  3 09:33 a
```

//更改权限以后，所有者对a文件多了写入的权限

【例10.2】 取消所有者对a文件的读取权限。

```
[root@rhel ~]# chmod u-r a
```

```
[root@rhel ~]# ls -l a
```

```
--w-r--r--. 1 root root 6 6月  3 09:33 a
```

//查看文件权限，可以看到文件a的所有者权限已经没有读取了



【例10.3】 重新分配同组用户对a文件有写入的权限。

```
[root@rhel ~]# chmod g=w a
```

```
[root@rhel ~]# ls -l a
```

```
--w--w-r--. 1 root root 6 6月  3 09:33 a
```

//可以看到，同组用户原先的权限没有了，现在重新分配的是写入权限

【例10.4】 更改a文件权限，添加所有者为读取、写入权限，同组用户为读取权限，其他用户读取、写入和执行的权限。

```
[root@rhel ~]# chmod u+rw,g+r,o+rw a
```

```
[root@rhel ~]# ls -l a
```

```
-rw-rw-rwx. 1 root root 6 6月  3 09:33 a
```

【例10.5】 取消所有用户的读取、写入和执行权限。

```
[root@rhel ~]# chmod a-rwx a
```

```
[root@rhel ~]# ls -l a
```

```
-----. 1 root root 6 6月  3 09:33 a
```

数字设定法设置权限

- 文件和目录的权限表中用r、w、x这三个字符来为用户所有者、组群所有者和其它用户设置权限。有时候，字符似乎过于麻烦，因此还有另外一种方法是以数字来表示权限，而且仅需3个数字。
- 使用数字设定法更改文件权限，首先必须了解数字表示的含义：0表示没有权限，1表示可执行权限，2表示写入权限，4表示读取权限，然后将其相加。

权限字符转换为数字

- 所有数字属性的格式应该是三个0~7的数，其顺序是u、g、o。
 - r: 对应数值4;
 - w: 对应数值2;
 - x: 对应数值1;
 - : 对应数值0。

权限字符转换为数字举例

- rwx-----: 用数字表示为700;
- rwxr--r--: 用数字表示为744;
- rw-rw-r-x: 用数字表示为665;
- drwx--x--x: 用数字表示为711;
- drwx-----: 用数字表示为700。

数字设定法设置权限命令

命令语法：

chmod [n1n2n3] [文件|目录]

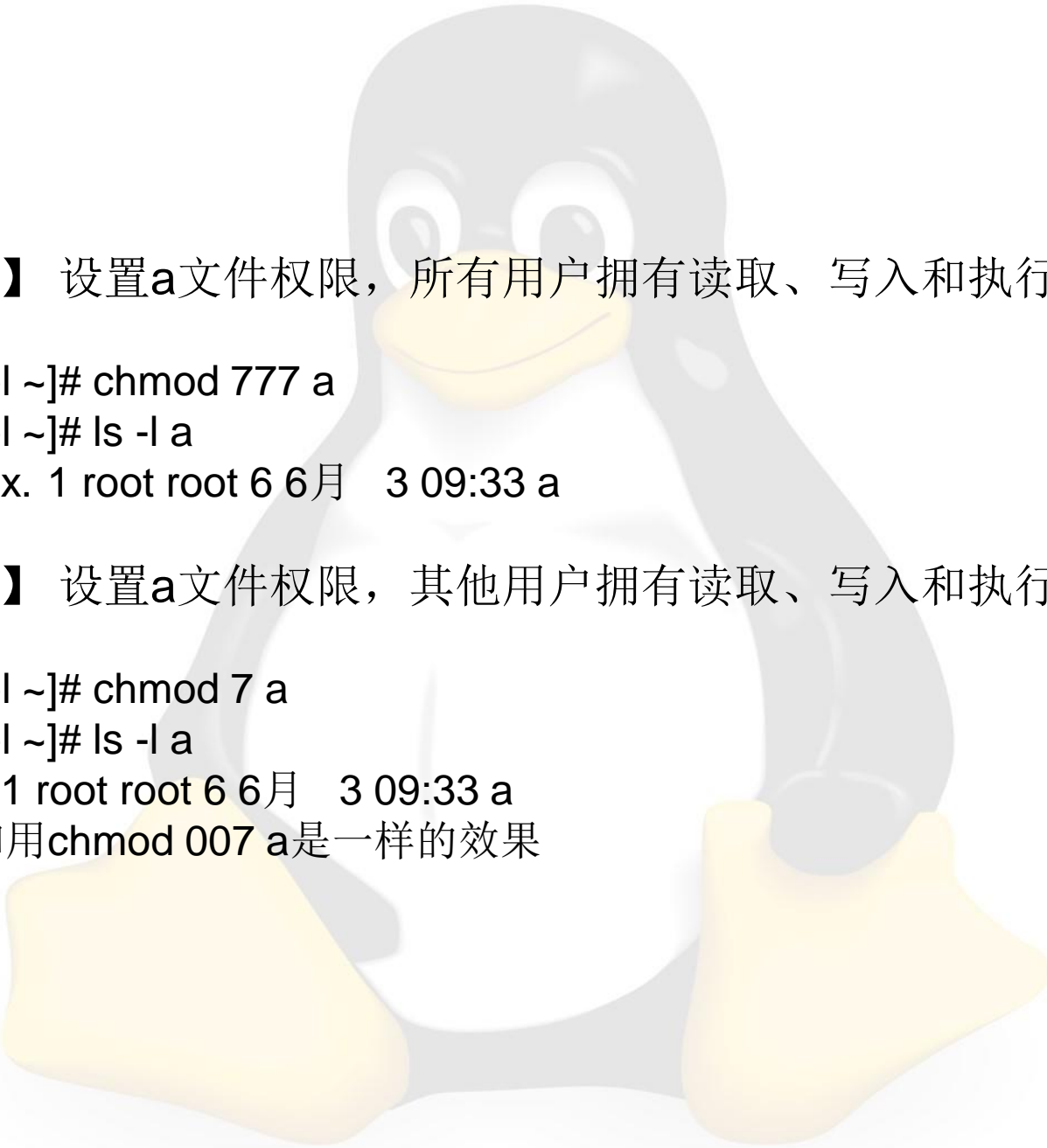
n1表示用户所有者的权限，**n2**表示组群所有者的权限，**n3**表示其它用户的权限。

【例10.6】 设置a文件权限，所有者拥有读取、写入和执行的权限。

```
[root@rhel ~]# ls -l a
-r--r--r--. 1 root root 6 6月  3 09:33 a
[root@rhel ~]# chmod 700 a
[root@rhel ~]# ls -l a
-rwx-----. 1 root root 6 6月  3 09:33 a
```

【例10.7】 设置a文件权限，所有者拥有读取，同组用户有读取、写入和执行的权限。

```
[root@rhel ~]# chmod 470 a
[root@rhel ~]# ls -l a
-r--rwx---. 1 root root 6 6月  3 09:33 a
```



【例10.8】 设置a文件权限，所有用户拥有读取、写入和执行的权限。

```
[root@rhel ~]# chmod 777 a  
[root@rhel ~]# ls -l a  
-rwxrwxrwx. 1 root root 6 6月  3 09:33 a
```

【例10.9】 设置a文件权限，其他用户拥有读取、写入和执行的权限。

```
[root@rhel ~]# chmod 7 a  
[root@rhel ~]# ls -l a  
-----rwx. 1 root root 6 6月  3 09:33 a  
//在这里和用chmod 007 a是一样的效果
```

【例10.10】 设置/home/user目录连同他的子文件夹的权限为777。

```
[root@rhel ~]# mkdir /home/user
```

```
[root@rhel ~]# touch /home/user/abc
```

```
[root@rhel ~]# chmod -R 777 /home/user
```

//表示将整个/home/user目录及其中的文件和子目录的权限都设置读取、写入和执行

```
[root@rhel ~]# ls -l /home|grep user
```

```
drwxrwxrwx. 2 root  root  4096 6月  3 09:44 user
```

```
[root@rhel ~]# ls -l /home/user/abc
```

```
-rwxrwxrwx. 1 root root 0 6月  3 09:44 /home/user/abc
```

特殊权限简介

- 除了基本权限之外，还有三个特殊的权限。用户如果没有特殊的需求，一般是不需要启用特殊权限，避免出现安全方面的隐患。

特殊权限有以下3种类型。

(1) SUID

对一个可执行文件，不是以发起者身份来获取资源，而是以可执行文件的用户所有者身份来执行；对一个目录无影响。

(2) SGID

对一个可执行文件，不是以发起者身份来获取资源，而是以可执行文件的组群所有者身份来执行；对一个目录，在该目录中创建的任意新文件的所属组与该目录的所属组相同。

(3) Sticky

对一个可执行文件无影响；对目录设置Sticky之后，尽管其它用户有写权限，也必须由文件所有者执行删除和移动等操作。

文字设定法设置特殊权限

- 通过文字设定法更改特殊权限需要使用 chmod 命令，chmod 的命令格式如下：

权限	命令	模式
SUID	chmod u+s	s=x+SUID S=--+SUID
SGID	chmod g+s	s=x+SGID S=--+SGID
Sticky	chmod o+t	t=x+Sticky T=--+Sticky



【例10.11】 添加a文件的特殊权限为SUID。

```
[root@rhel ~]# ls -l a
-----. 1 root root 6 6月  3 09:33 a
[root@rhel ~]# chmod u+s a
[root@rhel ~]# ls -l a
---S-----. 1 root root 6 6月  3 09:33 a
```

【例10.12】 添加a文件的特殊权限为SGID。

```
[root@rhel ~]# chmod g+s a
[root@rhel ~]# ls -l a
---S--S---. 1 root root 6 6月  3 09:33 a
```

【例10.13】 添加a文件的特殊权限为Sticky。

```
[root@rhel ~]# chmod o+t a
[root@rhel ~]# ls -l a
---S--S--T. 1 root root 6 6月  3 09:33 a
```

数字设定法设置特殊权限

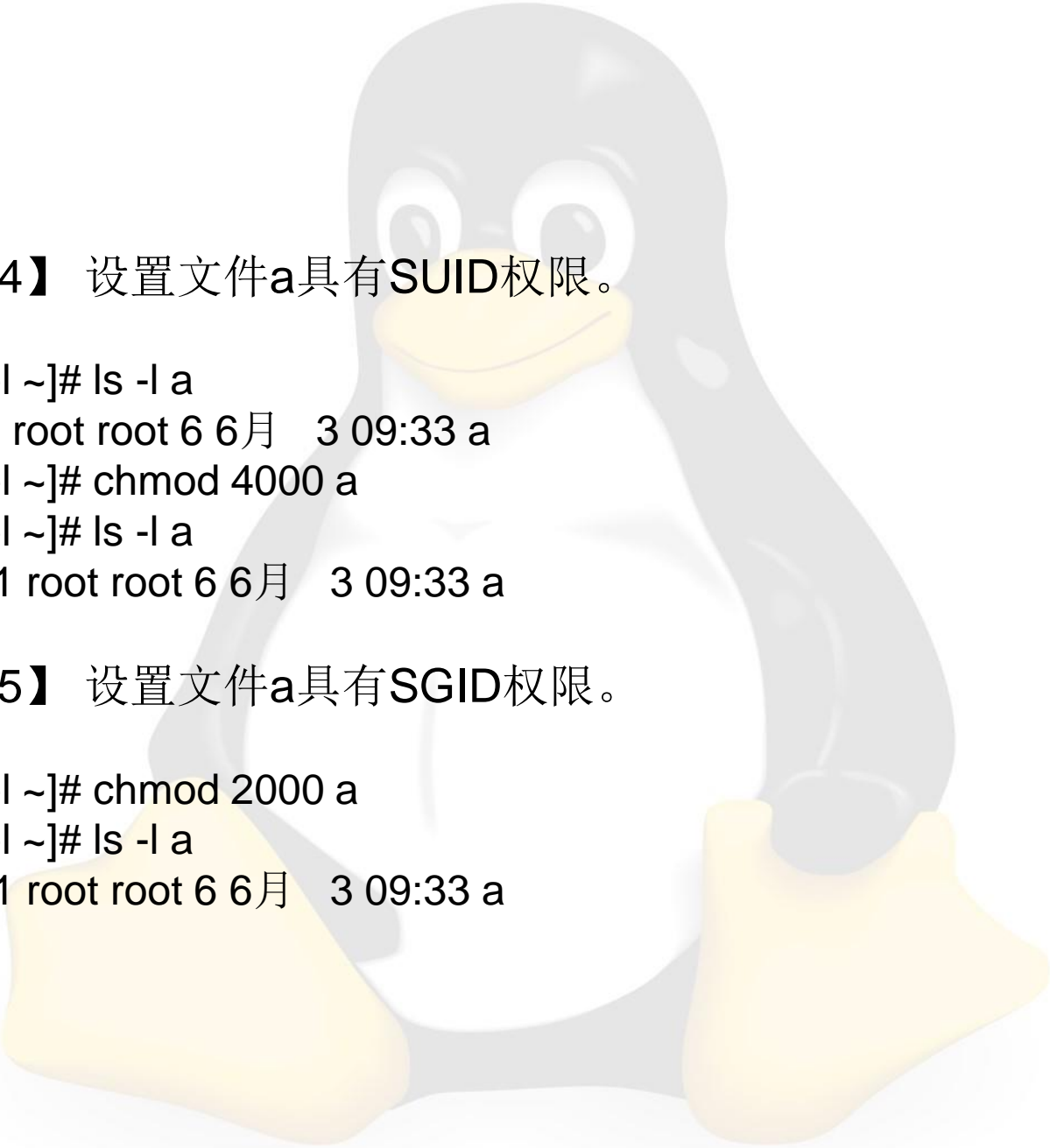
- 如果要设置特殊权限，就必须使用四位数字才能表示。

特殊权限的对应数值如下表示。

SUID: 对应数值4;

SGID: 对应数值2;

Sticky: 对应数值1。



【例10.14】 设置文件a具有SUID权限。

```
[root@rhel ~]# ls -l a
-r--r--r--. 1 root root 6 6月  3 09:33 a
[root@rhel ~]# chmod 4000 a
[root@rhel ~]# ls -l a
---S-----. 1 root root 6 6月  3 09:33 a
```

【例10.15】 设置文件a具有SGID权限。

```
[root@rhel ~]# chmod 2000 a
[root@rhel ~]# ls -l a
-----S---. 1 root root 6 6月  3 09:33 a
```



【例10.16】 设置文件a具有Sticky权限。

```
[root@rhel ~]# chmod 1000 a  
[root@rhel ~]# ls -l a  
-----T. 1 root root 6 6月  3 09:33 a
```

【例10.17】 设置文件a具有SUID，SGID和Sticky权限。

```
[root@rhel ~]# chmod 7000 a  
[root@rhel ~]# ls -l a  
---S--S--T. 1 root root 6 6月  3 09:33 a
```

10.2 更改文件和目录所有者

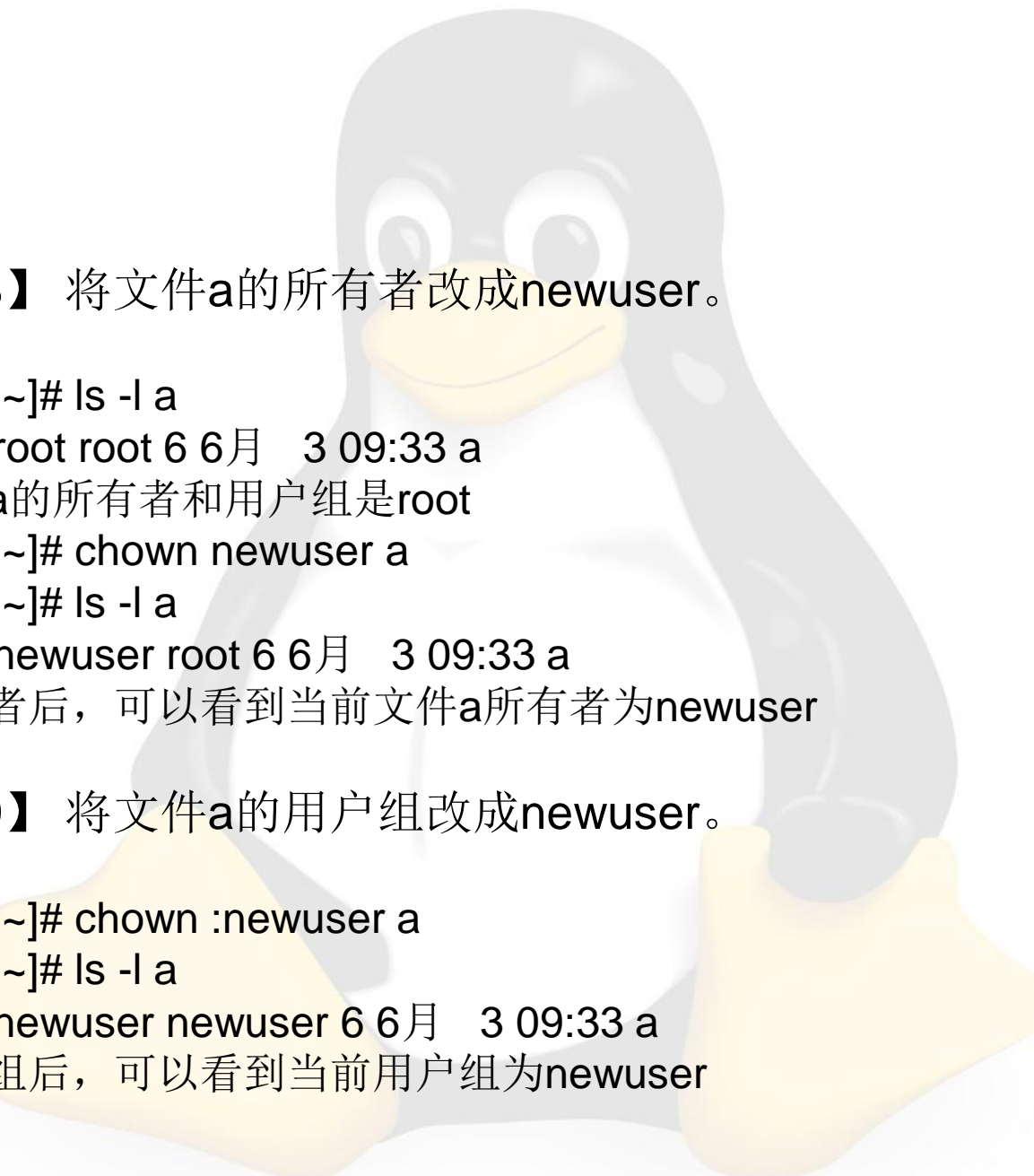
- 文件和目录的创建者默认就是该文件和目录的所有者，他们对该文件和目录具有任何权限，可以进行任何操作。他们也可以将所有者转交给别的用户，使别的用户对该文件和目录具有任何操作权限。文件和目录的所有者及所属用户组也能修改，可以通过命令来修改。

- 使用**chown**命令可以更改文件和目录的用户所有者和组群所有者。

命令语法：

chown [选项] [用户.组群] [文件|目录]

chown [选项] [用户:组群] [文件|目录]

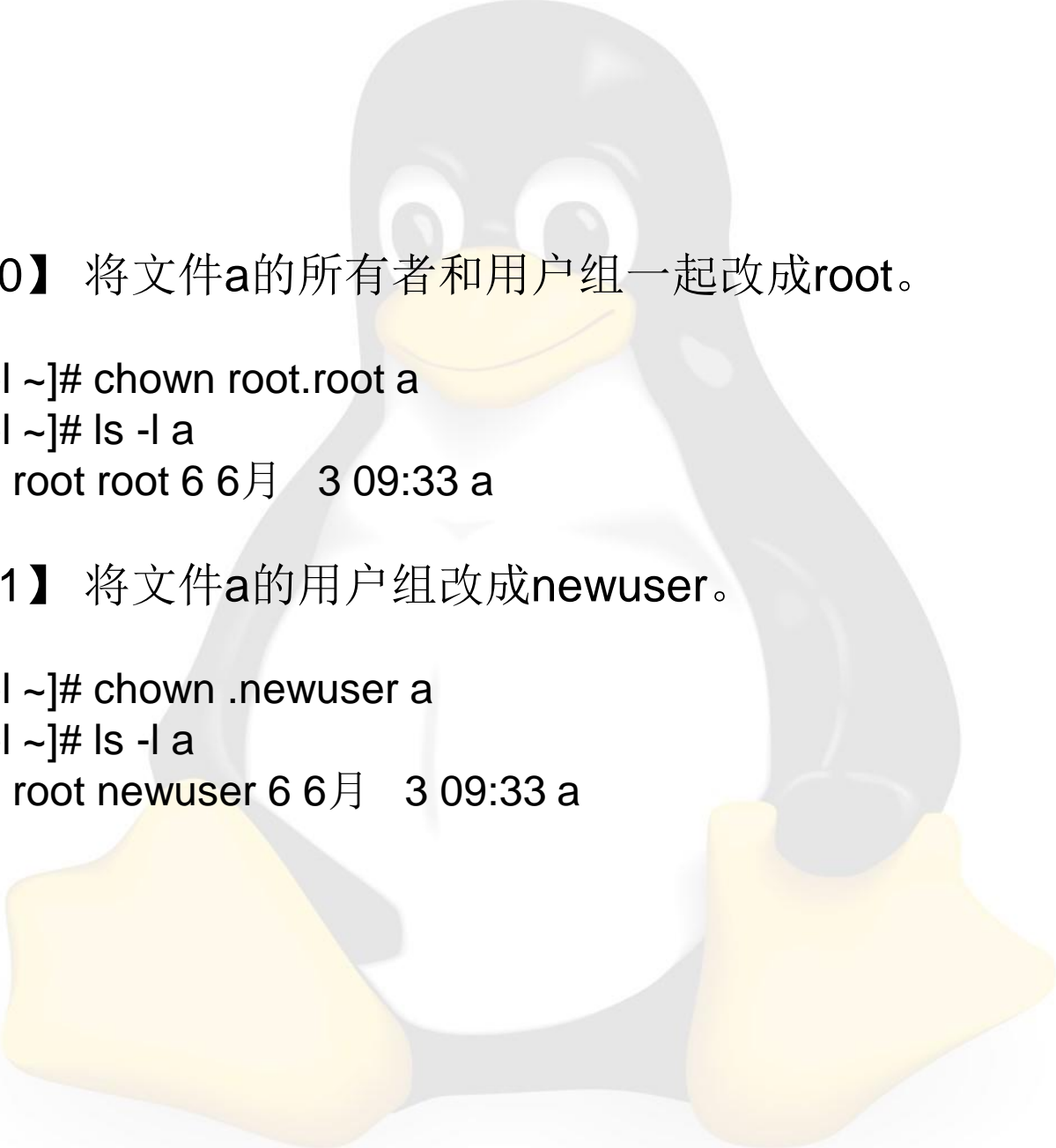


【例10.18】 将文件a的所有者改成newuser。

```
[root@rhel ~]# ls -l a
-r--r--r--. 1 root root 6 6月  3 09:33 a
//目前文件a的所有者和用户组是root
[root@rhel ~]# chown newuser a
[root@rhel ~]# ls -l a
-r--r--r--. 1 newuser root 6 6月  3 09:33 a
//更改所有者后，可以看到当前文件a所有者为newuser
```

【例10.19】 将文件a的用户组改成newuser。

```
[root@rhel ~]# chown :newuser a
[root@rhel ~]# ls -l a
-r--r--r--. 1 newuser newuser 6 6月  3 09:33 a
//更改用户组后，可以看到当前用户组为newuser
```



【例10.20】 将文件a的所有者和用户组一起改成root。

```
[root@rhel ~]# chown root.root a  
[root@rhel ~]# ls -l a  
-r--r--r--. 1 root root 6 6月  3 09:33 a
```

【例10.21】 将文件a的用户组改成newuser。

```
[root@rhel ~]# chown .newuser a  
[root@rhel ~]# ls -l a  
-r--r--r--. 1 root newuser 6 6月  3 09:33 a
```


【例10.22】 将目录/root/b连同它的下级文件/root/b/ccc的所有者和用户组一起更改为newuser。

```
[root@rhel ~]# ls -l /root|grep b
drwxr-xr-x. 2 root root    4096 6月  3 09:51 b
[root@rhel ~]# ls -l /root/b/ccc
-rw-r--r--. 1 root root 0 6月  3 09:51 /root/b/ccc
//查看目录/root/b和文件/root/b/cc所有者和用户组，当前为root
[root@rhel ~]# chown -R newuser.newuser /root/b
[root@rhel ~]# ls -l /root|grep b
drwxr-xr-x. 2 newuser newuser 4096 6月  3 09:51 b
[root@rhel ~]# ls -l /root/b/ccc
-rw-r--r--. 1 newuser newuser 0 6月  3 09:51 /root/b/ccc
//查看目录/root/b和文件/root/b/cc所有者和用户组，当前为newuser
```