**第七章 Centos7-文件权限管理**

**本节所讲内容：**

**7.1文件的基本权限：r w x （UGO）**

**7.2文件的特殊权限：suid sgid sticky和文件扩展权限ACL**

**7.3实战：创建一个让root都无法删除的文件**

# 7.1 文件的基本权限

## 7.1.1 权限的作用

**通过对文件设定权限可以达到以下三种访问限制权限：**

**只允许用户自己访问；**

**允许一个预先指定的用户组中的用户访问；**

**允许系统中的任何用户访问。**

## 7.1.2 查看权限

**[root@xuegod63 ~]# ll /etc/passwd**

**-rw-r--r--. 1 root root 2053 9月 19 2017 /etc/passwd**

**文件权限基本解释：**

**- rw- r-- r--. 1 root root 2053 9月 19 2017 /etc/passwd**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **-** | **rwx** | **r-x** | **r-x** | **user1** | **user1** | **time** | **FILENAME** |
| **文件类型** | **拥有者的权限** | **所属组的权限** | **其他人的权限** | **拥有者** | **属组** | **最后修改时间** | **对象** |

**其中：文件类型,可以为p、d、l、s、c、b和 –**

**p表示命名管道文件**

**d表示目录文件**

**l表示符号连接文件**

**-表示普通文件**

**s表示socket套接口文件，比如我们启用mysql时，会产生一个mysql.sock文件**

**c表示字符设备文件，例： 虚拟控制台 或tty0**

**b表示块设备文件 例： sda， cdrom**

**例：**

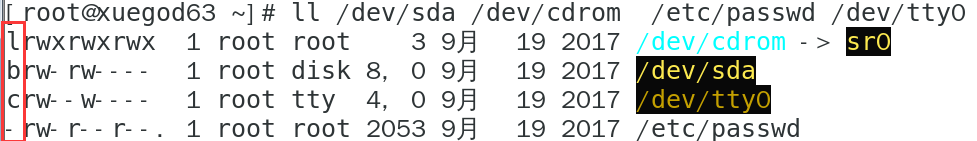
**[root@xuegod63 ~]# ll /dev/sda /dev/cdrom /etc/passwd /dev/tty0**

**lrwxrwxrwx 1 root root 3 9月 19 2017 /dev/cdrom -> sr0**

**brw-rw---- 1 root disk 8, 0 9月 19 2017 /dev/sda**

**crw--w---- 1 root tty 4, 0 9月 19 2017 /dev/tty0**

**-rw-r--r--. 1 root root 2053 9月 19 2017 /etc/passwd 如下效果图：**



**7.1.3 权限说明**

**对于文件来说：**

**r：读**

**w：写**

**x：执行**

**对于目录来说：**

**r：读（看到目录里面有什么） ls**

**w：在目录里面建文件，删除，移动 touch mkdir rm mv cp**

**x：进入 cd cat**

## 7.1.4 文件拥有者

**UGO：所有者--用户组--其它用户**

**所有者： 就是创建文件的用户，这个用户拥有对它所创建的文件的一切权限，所有者可以允许其所在的用户组可以访问所有者的文件。**

**用户组： 用户组是具有相同特征用户的逻辑集合，有时我们需要让多个用户具有相同的权限，比如查看、修改某一个文件的权限，一种方法是分别对多个用户进行文件访问授权，如果有10个用户的话，就需要授权10次，显然这种方法不太合理；另一种方法是建立一个组，让这个组具有查看、修改此文件的权限，然后将所有需要访问此文件的用户放入这个组中，那么所有用户就具有了和组一样的权限。这就是用户组。**

**其它用户：系统内的其他所有者用户就是other用户类**

## 7.1.5 举例说明

### 7.1.5.1 常见几种文件权限组成

**- rwx --- ---：文件所有者对文件具有读取、写入和执行的权限。**

**- rwx r-- r--： 文件所有者具有读、写与执行的权限，用户组里用户及其他用户则具有读取的权限**

**- rw- rw- r-x：文件所有者与同组用户对文件具有读写的权限，而其他用户仅具有读取和执行的权限。**

**drwx--x—x： 目录所有者具有读写与进入目录的权限,其他用户近能进入该目录，却无法读取任何数据。**

**drwx------： 除了目录所有者具有完整的权限之外，其他用户对该目录完全没有任何权限。**

**举例如下：**

**每个用户都拥有自己的专属目录，通常放置/home下**

**[root@xuegod63 home]# ll /home/**

**总用量 0**

**drwx------. 3 mk mk 78 9月 19 2017 mk**

**注： [rwx------]表示目录所有者本身拥有的权限，其它用户是无法进入的。 root可以。**

**例2: 你以什么用户身份登录，那么你创建的文件或目录，自动成为该文件的所属主和组**

**[root@xuegod63 home]# su - mk**

**上一次登录：二 9月 19 12:57:21 CST 2017:0 上**

**[mk@xuegod63 ~]$**

**[mk@xuegod63 ~]$ touch a.txt**

**[mk@xuegod63 ~]$ ll a.txt**

**-rw-rw-r-- 1 mk mk 0 5月 8 20:58 a.txt**

### 7.1.5.2 更改文件的属主和属组

**改变文件的所属关系用到命令：**

**chown：可以用来改变文件(或目录)的属主**

**chgrp：可以用来改变文件（或目录）的默认属组**

**如果你要对目录进行操作，加参数 -R**

**chown**

**语法：**

**chown user:group filename 比如：chown hr:san a.txt 把文件的属主和属组改为hr,san**

**chown user filename 比如：chown san a.txt 把文件的属主改为san用户**

**chown :group filename 比如： chown :miao a.txt 把文件的属组改为miao这个组**

**chown user: filename 比如：chown san: a.txt 自动继承这个用户所有的组**

**chgrp hr filename 比如： chgrp hr f.txt**

**-R ：递归（目录下的所有内容都更改，否则只修改目录）**

**例：**

**[root@xuegod63 ~]# touch {a,b,c}.txt**

**[root@xuegod63 ~]# ll \*.txt**

**-rw-r--r-- 1 root root 0 5月 8 21:03 a.txt**

**-rw-r--r-- 1 root root 0 5月 8 21:03 b.txt**

**-rw-r--r-- 1 root root 0 5月 8 21:03 c.txt**

**[root@xuegod63 ~]# chown mk a.txt**

**[root@xuegod63 ~]# ll a.txt**

**-rw-r--r-- 1 mk root 0 5月 8 21:03 a.txt**

**[root@xuegod63 ~]# chown mk:mk a.txt**

**[root@xuegod63 ~]# ll a.txt**

**-rw-r--r-- 1 mk mk 0 5月 8 21:03 a.txt**

**[root@xuegod63 ~]# chown :root a.txt**

**[root@xuegod63 ~]# ll a.txt**

**-rw-r--r-- 1 mk root 0 5月 8 21:03 a.txt**

**互动：一个文件只有读的权限，拥有者是否可以写这个文件？**

**实验：**

**[root@xuegod63 ~]# su - mk**

**[mk@xuegod63 ~]$ touch a.txt**

**[mk@xuegod63 ~]$ ll a.txt**

**-rw-rw-r-- 1 mk mk 0 5月 8 21:07 a.txt**

**在另一个终端上，以root身份登录：**

**[root@xuegod63 ~]# chmod 000 /home/mk/a.txt #修改成000权限**

**[root@xuegod63 ~]# ll /home/mk/a.txt**

**---------- 1 mk mk 14 5月 8 21:08 /home/mk/a.txt**

**回到以mk身份登录的终端：**

**[mk@xuegod63 ~]$ vim a.txt # 写入aaa , :wq! 保存**

**在另一个终端上，以root身份登录：**

**[root@xuegod63 ~]# cat /home/mk/a.txt**

**aaaaa**

**实验结果：文件所有者一定可以写文件。 就像root可以对shadow强制写。 因shadow的拥有者是root**

## 7.1.6 修改权限

### 7.1.6.1 使用字符设定

**修改权限用的命令：chmod**

**作用：修改文件，目录的权限**

**语法：chmod [对谁操作] [操作符] [赋于什么权限] 文件名**

**对谁操作：**

**u----> 用户user，表示文件或目录的所有者**

**g---->用户组group，表示文件或目录所属的用户组**

**o---->其它用户others**

**a---->所有用户all**

**操作符:**

**+ #添加权限 ； - # 减少权限 ； = #直接给定一个权限**

**权限：r w x**

**例如下在的组合：**

|  |  |  |
| --- | --- | --- |
| **u-w** | **user** | **拥有者** |
| **g+x** | **group** | **组** |
| **o=r** | **other** | **其他人** |
| **a+x** | **all** | **所有人** |

**例：chmod修改权限**

**[root@xuegod63 ~]# touch 1.txt**

**[root@xuegod63 ~]# ll 1.txt**

**-rw-r--r-- 1 root root 0 5月 8 21:20 1.txt**

**[root@xuegod63 ~]#**

**[root@xuegod63 ~]# chmod u-w 1.txt**

**[root@xuegod63 ~]# ll 1.txt**

**-r--r--r-- 1 root root 0 5月 8 21:20 1.txt**

**[root@xuegod63 ~]# chmod g+x 1.txt**

**[root@xuegod63 ~]# ll 1.txt**

**-r--r-xr-- 1 root root 0 5月 8 21:20 1.txt**

**[root@xuegod63 ~]# chmod a+x 1.txt # 给shell脚本加一个可执行权限**

**[root@xuegod63 ~]# ll 1.txt**

**-r-xr-xr-x 1 root root 0 5月 8 21:20 1.txt**

**[root@xuegod63 ~]#**

**[root@xuegod63 ~]# chmod a=rwx 1.txt**

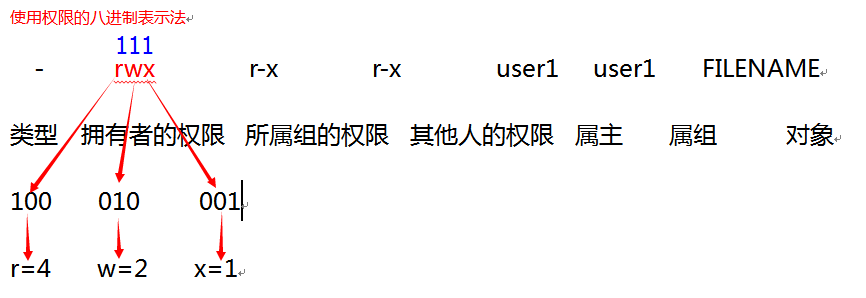
**[root@xuegod63 ~]# ll 1.txt**

**-rwxrwxrwx 1 root root 0 5月 8 21:20 1.txt**

### 7.1.6.2 使用八进制（0-7）数字表示权限法

|  |  |  |  |
| --- | --- | --- | --- |
| **权限** | **二进制值** | **八进制值** | **描述** |
| **---** | **000** | **0** | **没有任何权限** |
| **--x** | **001** | **1** | **只有执行权限** |
| **-w-** | **010** | **2** | **只有写入权限** |
| **-wx** | **011** | **3** | **有写入和执行权限** |
| **r--** | **100** | **4** | **只有读取权限** |
| **r-x** | **101** | **5** | **有读取和执行权限** |
| **rw-** | **110** | **6** | **有读取和写入权限** |
| **rwx** | **111** | **7** | **有全部权限** |

**例1：**

****

**例1：**

**互动：rw- 的值是多少？ 答： 4+2=6**

**rwx r-x r-x 的值是多少？ 答： rwx=4+2+1=7 ; r-x=4+1=5 rwx r-x r-x=7 5 5**

**语法：**

**chmod 755 文件或文件夹名字**

**chmod a=rwx b.txt 等于 chmod 777 b.txt**

**例：**

**[root@xuegod63 ~]# touch dd.txt**

**[root@xuegod63 ~]# ll dd.txt**

**-rw-r--r-- 1 root root 0 5月 8 21:40 dd.txt**

**[root@xuegod63 ~]# chmod 755 dd.txt**

**[root@xuegod63 ~]# ll dd.txt**

**-rwxr-xr-x 1 root root 0 5月 8 21:40 dd.txt**

**[root@xuegod63 ~]# chmod 700 dd.txt**

**[root@xuegod63 ~]# ll dd.txt**

**-rwx------ 1 root root 0 5月 8 21:40 dd.txt**

## 7.1.7 权限对文件和目录的影响

**有三种权限可以应用：读取，写入与执行，这些权限对访问文件和目录的影响如下：**

|  |  |  |
| --- | --- | --- |
| **权限** | **对文件的影响** | **对目录的影响** |
| **r(读取)** | **可以读取文件的内容** | **可以列出目录的内容（文件名）** |
| **w(写入)** | **可以更改文件的内容** | **可以创建或删除目录中的任意文件** |
| **x(执行)** | **可以作为命令执行文件** | **可以访问目录的内容（取决于目录中文件的权限）** |

## 7.1.8 补码

**为什么我们创建的文件的权限是644呢？**

**我们创建文件的默认权限是怎么来的？**

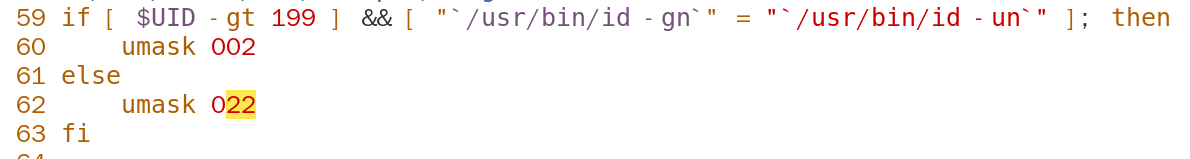
**umask命令允许你设定文件创建时的缺省模式，对应每一类用户(文件属主、同组用户、其他用户)存在一个相应的umask值中的数字**

**文件默认权限＝666 ，目录默认权限＝777**

**我们一般在/etc/profile、$ [HOME]/.bash\_profile或$[HOME]/.profile中设置umask值。**

**永久生效，编辑用户的配置文件vim .bash\_profile**

**[root@xuegod63 ~]# vim /etc/profile**



**注： UID大于199 且用户的组名和用户名一样，那么 umask值为002，否则为022.**

**注： -gt 在shell中表示大于； id -g 显示用户组ID ，id -gn显示组名。**

**临时生效： umask 权限补码**

**[root@xuegod63 ~]# umask 044**

**[root@xuegod63 ~]# touch ss.txt**

**[root@xuegod63 ~]# ll ss.txt**

**-rw--w--w- 1 root root 0 5月 8 21:47 ss.txt**

**权限的算法：一般情况是：目录默认权限-umask 值**

**666-022=644**

**777-022=755**

**#这是一个好的记忆方法，但不严谨。**

**互动：umask掩码为033 创建普通文件后，权限是什么？**

**互动：umask掩码为033 创建普通文件后，权限是什么？ 666-033=633 ( rw- -wx -wx) ?**

**例：[root@xuegod63 ~]# umask 033**

**[root@xuegod63 ~]# touch k.txt**

**[root@xuegod63 ~]# ll k.txt**

**-rw-r--r-- 1 root root 0 5月 8 22:00 k.txt**

**答：结果为： 644**

**权限科学的计算方法：**

**1、将默认权限（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**

**例2： umask 为033 结果为： 644**

**6 6 6 umask 0 3 3**

**110 110 110 000 011 011 # 转成二进制**

**111 100 100 # umask取反的值**

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

**111 100 100 # umask取反的值**

**110 100 100**

**6 4 4 #转成8进制**

# 7.2 文件的特殊权限：suid sgid sticky和文件扩展权限ACL

**其实文件与目录设置不止这些，还有所谓的特殊权限。由于特殊权限会拥有一些“特权”.**

**特殊权限：**

**7.2.1 文件的特殊权限：suid sgid sticky**

**1、SUID（set uid设置用户ID）：限定：只能设置在二进制可执行程序上面。对目录设置无效**

**功能：程序运行时的权限从执行者变更成程序所有者的权限**

**2、SGID：限定：既可以给二进制可执行程序设置，也可以对目录设置**

**功能：在设置了SGID权限的目录下建立文件时，新创建的文件的所属组会，继承上级目录的所属组**

**3、Stickybit：粘滞位权限是针对目录的，对文件无效，也叫防删除位**

**这3个特殊权限对应的数值为**

|  |  |  |
| --- | --- | --- |
| **SUID** | **SGID** | **Stickybit** |
| **u+s或u=4** | **g+s或g=2** | **o+t或o=1** |

**SUID属性一般用在可执行文件上，当用户执行该文件时，会临时拥有该执行文件的所有者权限。使用”ls -l” 或者”ll” 命令浏览文件时，如果可执行文件所有者权限的第三位是一个小写的”s”，就表明该执行文件拥有SUID属性。比如/usr/bin/passwd文件**

****

**互动： 普通用户mk，没有对shadow文件写入的权限， 但是mk用户使用passwd修改自己密码时，可以修改shadow文件中的内容，这是什么原因？**

**[root@xuegod63 ~]# ll /etc/shadow**

**----------. 1 root root 1179 9月 19 2017 /etc/shadow**

**[root@xuegod63 ~]# su - mk**

**上一次登录：二 5月 8 21:07:24 CST 2018pts/0 上**

**[mk@xuegod63 ~]$ passwd**

**更改用户 mk 的密码 。**

**为 mk 更改 STRESS 密码。**

**（当前）UNIX 密码：123456**

**新的 密码：Xuegod\*666**

**重新输入新的 密码：Xuegod\*666**

**passwd：所有的身份验证令牌已经成功更新。**

**[root@xuegod63 ~]# vim /etc/shadow #查看shadow文件已经被mk用户修改成功。**

**因为mk用户执行passwd命令时，权限会提升成root用户，所以可以修改成功。**

**例2：**

**[root@xuegod63 ~]# useradd mk**

**[root@xuegod63 ~]# su - mk**

**[root@xuegod63 ~]$less /etc/shadow #看不到内容**

**[root@xuegod63 ~]# su - root**

**[root@xuegod63 ~]# chmod u+s /usr/bin/less #切换到root，给一个suid权限**

**[root@xuegod63 ~]# su - mk**

**[mk@xuegod63 ~]$ less /etc/shadow #看到**

**查看u+s后的效果：**

**[root@xuegod63 ~]# ll /usr/bin/less**

**-rwsr-xr-x 1 root root 154536 Sep 26 2011 /usr/bin/less**

**[root@xuegod63 ~]# ps -axu | grep less**

**root 43407 0.0 0.0 110260 980 pts/0 S+ 22:30 0:00 less /etc/shadow**

**另外：**

**[root@xuegod63 ~]# chmod 4755 /usr/bin/less # 等同于 chmod u+s /usr/bin/less**

**SGID：**

**限定：既可以给二进制可执行程序设置，也可以给目录设置。**

**功能：在设置了SGID权限的目录下建立文件时，新创建的文件的所属组会继承上级目录的权限。**

**[root@xuegod63 ~]# mkdir test**

**[root@xuegod63 ~]# ll -d test**

**drwxr-xr-x 2 root root 4096 Jan 24 20:14 test**

**[root@xuegod63 ~]# chmod g+s test**

**[root@xuegod63 ~]# !ll**

**ll -d test**

**drwxr-sr-x 2 root root 4096 Jan 24 20:14 test**

**测试：sgid效果**

**[root@xuegod63 ~]# chown :bin test/**

**[root@xuegod63 ~]# touch test/a.txt**

**[root@xuegod63 ~]# ll !$**

**ll test/a.txt**

**-rw-r--r-- 1 root bin 0 Jan 24 20:15 test/a.txt**

**Stickybit**

**限定：只作用于目录**

**功能：目录下创建的文件只有root、文件创建者、目录所有者才能删除。**

**例： 系统中的tmp目录就是这样**

**[root@xuegod63 ~]# ll -d /tmp/**

**drwxrwxrwt. 11 root root 4096 Jan 24 19:41 /tmp/**

**用法：**

**chmod o+t /tmp/test/**

**7.2.1 文件扩展权限ACL**

**扩展ACL ：access control list**

**例：设置用户mk对文件a.txt拥有的rwx权限 ，mk不属于a.txt的所属主和组，mk是other。怎么做？**

**[root@xuegod63 ~]# touch /tmp/a.txt**

**[root@xuegod63 ~]# getfacl /tmp/a.txt**

**getfacl: Removing leading '/' from absolute path names**

**# file: tmp/a.txt**

**# owner: root**

**# group: root**

**user::rw-**

**group::r--**

**other::r--**

**#setfacl -m u:mk:rwx /tmp/a.txt u ： 设置某个用户拥有的权限**

**[root@xuegod63 ~]# getfacl /tmp/a.txt**

**getfacl: Removing leading '/' from absolute path names**

**...**

**user::rw-**

**user:mk:rwx**

**[mk@xuegod63 ~]$ vim /tmp/a.txt**

**[mk@xuegod63 ~]$ ll /tmp/a.txt**

**-rw-rwxr--+ 1 root root 8 5月 8 22:42 /tmp/a.txt**

**例2： 给目录加扩展权限**

**[root@xuegod63 ~]# mkdir /tmp/test**

**#setfacl -m d:u:mk:rwx /tmp/test #** **-d default 设置默认acl，对目录有效，此目录下新建的目录或文件都继承此acl权限**

**例：测试一下 -d 参数：**

**[root@xuegod63 ~]# mkdir /tmp/test**

**[root@xuegod63 ~]# setfacl -m d:u:mk:rwx /tmp/test**

**[root@xuegod63 ~]# getfacl /tmp/test/**

**getfacl: Removing leading '/' from absolute path names**

**# file: tmp/test/**

**# owner: root**

**# group: root**

**user::rwx**

**group::r-x**

**other::r-x**

**default:user::rwx**

**default:user:mk:rwx**

**default:group::r-x**

**default:mask::rwx**

**default:other::r-x**

**[root@xuegod63 ~]# touch /tmp/test/a.txt**

**[root@xuegod63 ~]# mkdir /tmp/test/data**

**[root@xuegod63 ~]# getfacl /tmp/test/a.txt #因为-d参数，所以test下所有创建的文件和目录都继承了默认的acl权限**

**getfacl: Removing leading '/' from absolute path names**

**# file: tmp/test/a.txt**

**# owner: root**

**# group: root**

**user::rw-**

**user:mk:rwx #effective:rw-**

**group::r-x #effective:r--**

**mask::rw-**

**other::r--**

**[root@xuegod63 ~]# getfacl /tmp/test/data**

**getfacl: Removing leading '/' from absolute path names**

**# file: tmp/test/data**

**# owner: root**

**# group: root**

**user::rwx**

**user:mk:rwx**

**group::r-x**

**mask::rwx**

**other::r-x**

**default:user::rwx**

**default:user:mk:rwx**

**default:group::r-x**

**default:mask::rwx**

**default:other::r-x**

**例3：给目录下所有文件都加扩展权限**

**[root@xuegod63 ~]# setfacl -R -m u:lee:rw- testdirectory/  #-R一定要在-m前面，表示目录下所有文件**

**[root@xuegod63 ~]# setfacl -x u:mk /tmp/a.txt # 去掉单个权限**

**[root@xuegod63 ~]# setfacl -b /tmp/a.txt # 去掉所有acl权限**

# 7.3实战：创建一个让root都无法删除的文件

**发现windows中 有文件删除不了，怎么办？ 使用360 强制删除，粉碎文件**

**那么在Linux下怎么办？**

**[root@xuegod63 ~]# touch hack.sh aa.sh**

**[root@xuegod63 ~]# ll hack.sh aa.sh**

**-rw-r--r-- 1 root root 0 May 24 21:29 aa.sh**

**-rw-r--r-- 1 root root 0 May 24 21:29 hack.sh**

**[root@xuegod63 ~]# rm -rf aa.sh**

**黑客使用xshell悄悄执行在后台添加attr扩展属性：（这个别让学员看到^\_^）**

**[root@xuegod63 ~]# chattr +i hack.sh**

**删除文件：**

**[root@xuegod63 ~]# rm -rf hack.sh #发现删除不了**

**为什么删除不了？**

**从REHL6 开始，新增加文件系统扩展属性：**

**命令：chattr**

**参数： a 只能追加内容 ； i 不能被修改**

**+a: 只能追加内容 如： echo aaa >> hack.sh**

**+i：即Immutable，系统不允许对这个文件进行任何的修改。如果目录具有这个属性，那么任何的进程只能修改目录之下的文件，不允许建立和删除文件。**

**注：immutable [ɪˈmju:təbl] 不可改变的 ； Append [əˈpend] 追加**

**-i ：移除i参数。 -a :移除a参数**

**解决：**

**[root@xuegod63 ~]# lsattr hack.sh**

**----i----------- hack.sh**

**[root@xuegod63 ~]# chattr -i hack.sh**

**[root@xuegod63 ~]# echo aa >> hack.sh**

**[root@xuegod63 ~]# lsattr hack.sh #查看扩展属性**

**---------------- hack.sh**

**[root@xuegod63 ~]# chat**

**chat chattr**

**[root@xuegod63 ~]# chattr +a hack.sh**

**[root@xuegod63 ~]# rm -rf hack.sh**

**rm: 无法删除"hack.sh": 不允许的操作**

**[root@xuegod63 ~]# echo aaa >> hack.sh**

**总结：**

**7.1文件的基本权限：r w x （UGO）**

**7.2文件的特殊权限：suid sgid sticky和文件扩展权限ACL**

**7.3实战：创建一个让root都无法删除的文件**