目录：

[1、Linux正则表达式 1](#_Toc418499242)

1.1什么是正则表达式 1

1.2为什么要学会正则表达式 2

1.3容易混淆的两个注意事项 2

1.4基础正则表达式 3

1.5扩展正则表达式 5

1.6Linux中的通配符 7

2、文件的权限 [8](#_Toc418499243)

2.1文件权限概述 8

2.2文件权限的实战演示 9

2.3目录权限的实战演示 11

2.4改变权限属性命令 chmod 14

2.4.1chmod修改权限的方法 14

3、umask相关的命令17

4、setuid和setgid位22

4.1回顾Linux权限属性及权限小结 22

4.2 Linux权限中的特殊权限位 23

4.3suid的介绍 23

4.4setgid权限知识 26

4.5粘滞位相关的知识 28

1. **Linux正则表达式**

**1.1 什么是正则表达式？**

简单来说，正则表达式就是为处理大量的字符串而定义的一套规则和方法，例如：假设“@”代表oldboy，“！”代表oldgirl，echo“@”===“oldboyoldgirl”

通过定义的这些特殊符号的辅助，系统管理员就可以快速过滤，替换或输出需要的字符串。

Linux正则表达式一般以行为单位处理的。

正则表达式：

1. Linux正则表达式（适合三剑客的正则grep，awk，sed）
2. PHP，JAVA，PERL，PYTHON（Perl兼容正则PCRE）

**1.2 为什么要学会正则表达式？**

在企业工作中，我们每天做的Linux运维工作中，时刻都会面对大量带有字符串的文本配置、程序、命令输出及日志文件等，而我们经常会有迫切的需要，从大量的字符串内容中查找符合工作需要的特定字符串，这就要靠正则表达式，因此，正则表达式就是为过滤这样字符串需求而生的。

**1.3 容易混淆的两个注意事项：**

1. 正则表达式应用非常广泛，存在于各种语言中，例如：php，python，java等，但是本节课要讲的是Linux正则表达式，最常用正则表达式的命令就是grep、sed、awk，换句话说Linux三剑客要想能工作的更高效，那一定离不开正则表达式配合。
2. 正则表达式和我们常用的通配符特殊字符是有本质区别的。

正则表达式（regular expressions）实际就是一些特殊字符，赋予了它特定的含义。

通配符例子：ls \*.log这里的\*就是通配符（表示所有）

注意事项：

1. Linux正则一般以行为单位处理的。
2. alias grep=’grep –color=auto’，以grep为例。
3. 注意字符集，LC\_ALL=C
4. 我们要想玩好三剑客，首先就要掌握正则表达式。Linux里正则表达式，主要是awk，sed，grep（egrep）三剑客的正则表达式。
   1. **基础正则表达式**

元字符意义 BRE（basic regular expression），正则表达式（REGULAR EXPRESSIONS）实际就是一些特殊字符，赋予了他特定的

1）^word 搜索以word开头的。Vi ^一行的开头

2）word$ 搜索以word结尾的。 Vi $一行的结尾

3）^$ 表示空行

4).     代表且只能代表任意一个字符。

#####拓展：

点的含义（.）：

1、当前目录

2、使得文件生效相当于source

3、隐藏文件

4、任意一个字符（正则）

5)\     例\.就只代表点本身，转义符号，让有着特殊身份意义的字符，脱掉马甲，还原原型。\$。  
6)\*     例o\*重复0个或多个前面的一个字符,o oo ooooo  
7).\*    匹配所有字符。 延伸^.\*以任意多个字符开头。.\*$以任意多个字符结尾

#####拓展：

.和.\*的区别就在于：

.是匹配除了空行以外的所有内容。

.\*匹配所有内容。

8)[abc]  匹配字符集合内的任意一个字符[a-zA-Z]，[0-9]。  
9)[^abc] 匹配不包含^后的任意字符的内容。中括号里的^为取反，注意和以...开头区别。  
10) a\{n,m\} 重复n到m次，前一个重复的字符。如果用egrep/sed -r可以去掉斜线。  
    a\{n,\}   重复至少n次，前一个重复的字符。如果用egrep/sed -r可以去掉斜线。  
     a\{n\}    重复n次，前一个重复的字符。如果用egrep/sed -r可以去掉斜线。  
     a\{,m\}   ?????  
注意：egrep或sed -r过滤一般特殊字符可以不转义。

**正则表达式处理技巧：**

sed -n 's#支持正则##g'

inet addr:10.0.1.7 Bcast:10.0.1.255 Mask:255.255.255.0

处理技巧：

匹配需要的目标（获取的字符串如上文的ip）前的字符串一般用以..开头(^.\*)来匹配开头，匹配的结尾写上实际的字符，

如：“^.\*addr:”表达式就匹配“ inet addr:”，

而处理需要的目标后的内容一般在匹配的开头写上实际的字符，而结尾是用以...结尾（.\*$）来匹配。如：Bcast:.\*$部分表示匹配“Bcast:10.0.0.255 Mask:255.255.255.0”。

**1.5 扩展的正则表达式：grep -E以及egrep**  
【了解即可】  
1)+ 表示重复“一个或一个以上”前面的字符（\*是0或多个）  
2)? 表示重复“0个或一个”前面的字符  
3)| 表示同时过滤多个字符串  
4)()分组过滤，后项引用。

**grep正则实战例子：**

[root@moban lili]# vi oldboy.log ####----首先编辑一个文档，内容如下：

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blos is http://oldboy.blog.51cto.com

our site is http://www.etiantian.org

my qq nmu is 49000448.

not 4900000448.

my god,i am not oldboey.but OLDBOY!

"oldboy.log" [New] 10L, 246C written

[root@moban lili]#

[root@moban lili]# cat oldboy.log

I am oldboy teacher!

I teach linux.

I like badminton ball ,billiard ball and chinese chess!

my blos is http://oldboy.blog.51cto.com

our site is http://www.etiantian.org

my qq nmu is 49000448.

not 4900000448.

my god,i am not oldboey.but OLDBOY!

过滤出以m为开头的行（此次命令用到的是”^”，以什么什么为开头）

[root@moban lili]# grep "^m" oldboy.log

my blos is http://oldboy.blog.51cto.com

my qq nmu is 49000448.

my god,i am not oldboey.but OLDBOY!

过滤出以m为结尾的行（此次命令用到的是”$”，以什么什么为结尾）

[root@moban lili]# grep "m$" oldboy.log

my blos is <http://oldboy.blog.51cto.com>

将文档中的空行过滤掉（此次命令用到的是”^$”，两个符号一起使用表示空行）

[root@moban lili]# grep -vn "^$" oldboy.log

1:I am oldboy teacher!

2:I teach linux.

4:I like badminton ball ,billiard ball and chinese chess!

5:my blos is http://oldboy.blog.51cto.com

6:our site is http://www.etiantian.org

7:my qq nmu is 49000448.

9:not 4900000448.

10:my god,i am not oldboey.but OLDBOY!

使用正则表达式中的“.”

[root@moban lili]# grep "oldb.y" oldboy.log

I am oldboy teacher!

my blos is <http://oldboy.blog.51cto.com>

请查找出文件中以“.”为结尾的（由于.为正则表达式符号，所以要配合\使用）（\转义符号）

[root@moban lili]# grep "\.$" oldboy.log

I teach linux.

my qq nmu is 49000448.

not 4900000448.

匹配文档中除了字母a-z的内容：

grep “[^a-z]” oldboy.log

grep拓展命令使用方法：

多选情况：

egrep ＂god|good＂ oldboy.log

使用sed命令同时取IP和子网掩码

[root@moban data]# ifconfig eth0|sed -nr 's#^.\*r:(.\*) .\*:(.\*) Ma.\*$#\1 \2#gp'

10.0.0.8 10.0.0.255

**1.6 Linux中的通配符**

**注意：通配符和正则表达式是不一样的，因此，代表的意义也是有较大区别的。通配符一般用户命令行bash环境，而Linux正则表达式用于sed,grep,awk场景。**

\* -通配符，代表任意（0到多个）字符\*\*\*\*\*

？-通配符，代表任意1个字符

；-连续不同命令的分隔符\*\*\*\*\*

# -配置文件注释\*\*\*\*\*

| -管道\*\*\*\*\*

~ -用户的家目录\*\*\*\*\*

- -上一次的目录\*\*\*\*\*

$ -变量前需要加的符号

/ -路径分隔符号

>或1> -重定向，覆盖\*\*\*\*\*

>> -追加重定向，追加。\*\*\*\*\*

< -输入重定向\*\*\*\*\*

<< -追加输入重定向。

' -单引号，不具有变量置换功能，输出时所见即所得。

'' -双引号，具有变量置换功能，解析变量后输出。不加引号相当于双引号，常用双引号。

` -tab键上面的键，反引号，两个``中间为命令，会先执行，等价$()。

{} -中间为命令区块组合或内容序列。

！ -逻辑运算中的“非”（not）

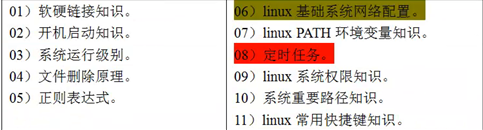
&& -当前一个指令执行成功时，执行后一个指令。

|| -当前一个指令执行失败时，执行后一个指令。

.. -上一级目录\*\*\*\*

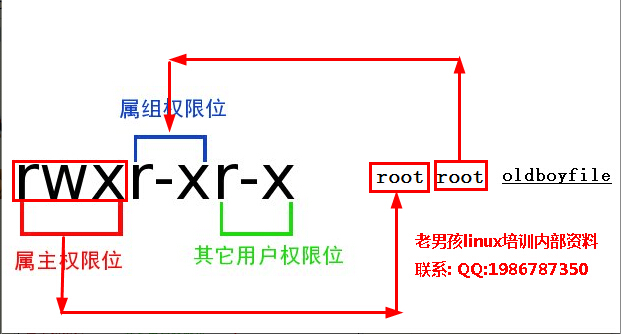
. -当前命令\*\*\*\*

第三关考试题涉及的知识点：



1. **文件的权限**

**2.1文件权限概述**



权限分为两种

1. 字符权限 rwxr-xr-x
2. 数字权限 755

R read 可读权限，对应数字4

W write 可写权限，对应数字2

X（execute，执行权限） 对应数字1

-（没有任何权限） 对应数字0

s 赋予用户执行一条命令的所有者权限

S 权限位上没有执行权限。

文件属主/用户（owner/user）：读read、写write、执行x

文件用户组（group）：读r、写w、执行x

其他用户（other）：读r、写w、执行x

如果权限位不可读、不可写、不可执行，则用-来表示。

后面还会有特殊的权限位：t T s S X +

**2.2 文件权限的实战演示**

**环境：**

groupadd incahome 一个家庭  
useradd oldboy -g incahome 主人  
useradd oldgirl -g incahome 家庭成员  
useradd test  外面的人  
id oldboy  
id oldgirl

**一个家庭（用户组）incahome：成员有oldboy与oldgirl，非家庭用户test，超级用户root**

[root@moban ~]# groupadd incahome -----##创建一个组

[root@moban ~]# useradd -g incahome oldboy -----##在这个组中创建oldboy

[root@moban ~]# useradd -g incahome oldgirl -----##在这个组中创建oldgirl

[root@moban ~]# useradd test -----##创建一个test用户

[root@moban ~]# id oldboy -----##显示oldboy信息

uid=500(oldboy) gid=500(incahome) groups=500(incahome)

[root@moban ~]# id oldgirl

uid=501(oldgirl) gid=500(incahome) groups=500(incahome)

[root@moban ~]# id test

uid=502(test) gid=502(test) groups=502(test)

**拓展：同时删除两个用户：**

[root@moban ~]# echo "oldboy oldgirl"

oldboy oldgirl

[root@moban ~]# echo "oldboyoldgirl"

oldboyoldgirl

[root@moban ~]# echo "oldboy\noldgirl"

oldboy\noldgirl

[root@moban ~]# echo -e "oldboy\noldgirl"

oldboy

oldgirl

[root@moban ~]# echo -e "oldboy\noldgirl"|sed -r 's#(.\*)#userdel \1#g'

userdel oldboy

userdel oldgirl

[root@moban ~]# echo -e "oldboy\noldgirl"|sed -r 's#(.\*)#userdel \1#g'|bash

**测试环境开始：**

[root@moban /]# mkdir /test/oldboy/ -p

[root@moban /]# cd /test/oldboy/

[root@moban oldboy]# vi test.sh

echo "I am oldboy linux"

~

"test.sh" [New] 1L, 25C written

[root@moban oldboy]# sh test.sh

I am oldboy linux

打开四个CRT窗口，循序依次为（root、oldboy、oldgirl、test）

[root@moban oldboy]# cd /test/oldboy

[root@moban oldboy]# ls

test.sh

[root@moban oldboy]# ls -l test.sh

-rw-r--r--. 1 root root 25 Oct 1 07:59 test.sh

[root@moban oldboy]# chown oldboy.incahome test.sh ----##将文件的用户更改为oldboy

[root@moban oldboy]# ls -l test.sh

-rw-r--r--. 1 oldboy incahome 25 Oct 1 07:59 test.sh

文件更改完权限之后，oldboy为这个文件的用户，incahome为用户组。

[root@moban oldboy]# chmod 765 test.sh ----##此时将文件权限修改

[root@moban oldboy]# ls -l test.sh

-rwxrw-r-x. 1 oldboy incahome 28 Oct 1 08:15 test.sh

Oldboy用户可读、写、执行；oldgirl用户可读、写；test用户可读、执行。

[root@moban oldboy]# chmod 331 test.sh

[root@moban oldboy]# ls -l test.sh

--wx-wx--x. 1 oldboy incahome 28 Oct 1 08:15 test.sh

将文件权限再次修改，此时，oldboy用户可写；oldgirl用户可写；test用户无操作权限。

####提示：当用户或组在没有读的权限是，是无法执行的。

[root@moban oldboy]# chmod 777 test.sh

[root@moban oldboy]# ls -l test.sh

-rwxrwxrwx. 1 oldboy incahome 28 Oct 1 08:15 test.sh

再次将文件权限修改，此时，所有用户均有读、写、执行的权限。当所有用户都是777权限时，oldboy、oldgirl、test是无法删除test.sh文件的，删除文件（修改文件名等）的权限时受父目录的权限控制。

**总结测试结论：**

Linux普通文件的读、写、执行权限说明：

1、可读r：表示具有读取\阅读文件内容的权限；

2、可写w：表示具有新增、修改文件内容的权限；

（如果没有r读，那么vi无法编辑，强制编辑的话会覆盖数据，但echo可以追加）；

3、执行x：表示具有执行文件的权限

3.1 文件本身要能够执行

3.2 普通用户同时还需要具备r读的权限。

3.3 任何用户、组有x权限，root就可以执行。

#####---提示：删除文件（修改文件名等）的权限是受父目录的权限控制，和文件本身权限无关。

**2.3 目录权限的实战演示**

**环境：与上面一样，具体内容如下：**

[root@moban oldboy]# mkdir test

[root@moban oldboy]# ls -ld test/

drwxr-xr-x. 2 root root 4096 Oct 1 12:47 test/

[root@moban oldboy]# touch test/a

[root@moban oldboy]# touch test/b

[root@moban oldboy]# chown oldboy.incahome test/

[root@moban oldboy]# ls -ld test/

drwxr-xr-x. 2 oldboy incahome 4096 Oct 1 12:48 test/

此时oldboy可对目录进行读、写、执行；oldgirl可对目录进行读、执行；test可对目录进行读、执行。

[root@moban oldboy]# chmod 365 test/ ----此时将目录权限修改为365

[root@moban oldboy]# ls -ld test/

d-wxrw-r-x. 2 oldboy incahome 4096 10月 1 12:56 test/ ###----此时oldboy对目录的操作为可写、执行；oldgirl对目录可进行读和写；test对目录可进行读和执行。

[root@moban oldboy]# ls test/

a b c e

此时将CRT用户切换到oldboy下：

[oldboy@moban ~]$ whoami

Oldboy

[oldboy@moban test]$ cd ..

[oldboy@moban oldboy]$ cd test/

[oldboy@moban test]$ ll

ls: 无法打开目录.: 权限不够

[oldboy@moban test]$ ls

ls: 无法打开目录.: 权限不够

[oldboy@moban test]$ touch d

[oldboy@moban test]$ mv a g

#####以上操作说明oldboy对目录内可进行写、执行的操作权限。#####

此时将CRT切换到oldgirl用户下：

[oldgirl@moban oldboy]$ whoami

oldgirl

[oldgirl@moban oldboy]$ cd test/

-bash: cd: test/: 权限不够

[oldgirl@moban oldboy]$ ls test/

ls: 无法访问test/e: 权限不够

ls: 无法访问test/c: 权限不够

ls: 无法访问test/b: 权限不够

ls: 无法访问test/g: 权限不够

ls: 无法访问test/d: 权限不够

b c d e g

[oldgirl@moban oldboy]$ touch test/a

touch: 无法创建"test/a": 权限不够

[oldgirl@moban oldboy]$ mv test/b test/a

mv: 正在访问"test/a": 权限不够

#####以上操作说明oldgirl对目录内只可进行读的操作权限。#####

此时将CRT用户切到test用户下：

[test@moban oldboy]$ whoami

test

[test@moban oldboy]$ cd test/

[test@moban test]$ ls

b c d e g

[test@moban test]$ ls -l

总用量 0

-rw-r--r--. 1 oldboy incahome 0 10月 1 13:30 b

-rw-r--r--. 1 root root 0 10月 1 13:30 c

-rw-r--r--. 1 oldboy incahome 0 10月 1 13:32 d

-rw-r--r--. 1 root root 0 10月 1 13:30 e

-rw-r--r--. 1 oldboy incahome 0 10月 1 12:56 g

[test@moban test]$ touch a

touch: 无法创建"a": 权限不够

[test@moban test]$ mv b a

mv: 无法将"b" 移动至"a": 权限不够

#####以上操作说明test对目录内可进行读、执行的操作权限。#####

总结测试结论：Linux目录权限的读、写、执行权限说明：

1、可读r：表示具有可读\阅读文件内容的权限。

即ls dir，但是不能进入到目录里，即无法cd dir

测试环境总结的R结论：

1.1 如果没有x（执行）权限，ls列表时可以看到所有文件名，但是会提示无权访问目录下文件。

1.2 如果ls –l列表，所有的属性会带有问号，也会提示会提示无权访问目录下文件，但是可以看到所有文件名。

2、可写w：表示具有增加、删除或修改目录内文件名（一般指文件名）的权限。

（需要X权限的配合，没有x权限配合无法touch、rm文件。）

3、可执行x：表示具有进入目录的权限；

（例如cd dir）但是没有r无法列表，没有w无法创建和删除文件。

**Linux文件权限和目录权限的对比：**

r：

1. 对于文件可进行读取文件内容。 ####（cat）
2. 对于目录可进行列表。 ####（ls）

w:

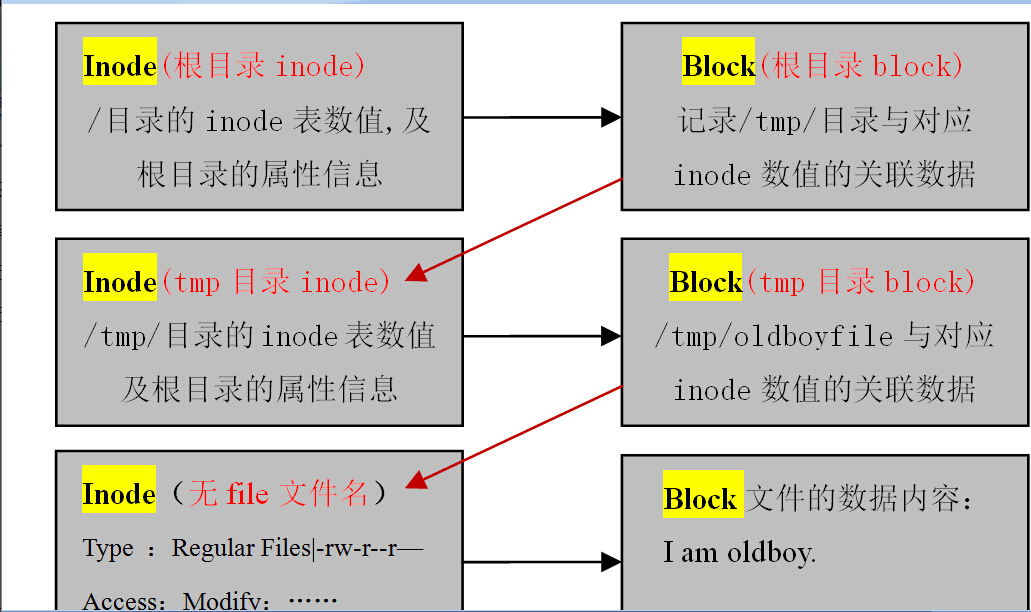
1、对于文件可进行修改文件内容。 ####vi、echo

2、对于目录可进行增加、修改文件。 ####需要x配合

x:

1、对于文件可进行执行操作。 ####需要有r的配合

2、对于目录可进行切换到目录中。 ####cd命令



**2.4 改变权限属性命令 chmod**

**2.4.1 chmod修改权限的方法：**

**1、使用权限数字设置权限的命令格式如下：**

[root@moban oldboy]# touch lifen.txt

[root@moban oldboy]# ll

总用量 0

-rw-r--r--. 1 root root 0 10月 1 14:29 lifen.txt

[root@moban oldboy]# chmod 777 lifen.txt

[root@moban oldboy]# ll

总用量 0

-rwxrwxrwx. 1 root root 0 10月 1 14:29 lifen.txt

[root@moban oldboy]# chmod 655 lifen.txt

[root@moban oldboy]# ll

总用量 0

-rw-r-xr-x. 1 root root 0 10月 1 14:29 lifen.txt

**数字与权限之间的关系如下表：**

|  |  |
| --- | --- |
| **八进制数字** | **权限** |
| **0** | **---** |
| **1** | **--x** |
| **2** | **-w-** |
| **3** | **-wx** |
| **4** | **r--** |
| **5** | **r-x** |
| **6** | **rw-** |
| **7** | **rwx** |

**2、使用权限字符设置权限的命令格式如下：**

chmod [用户类型] [+|-|=] [权限字符] 文件名、目录名，用表格表示如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **chmod** | 用户类型 | 操作字符 | 权限字符 | **文件或目录** |
| u（user） | +（加入） | r |
| g（group） |  | - |
| o（others） | -（减去） | w |
| a（all） | =（全部） | x |

chmod的字符式语法相对简单，对文件或目录权限的改变时，是通过比较直观的字符的形式来完成在字符式语法中，相关字母的定义；

例子：

rw-r--r--转换为rwxr-xrwx

[root@moban oldboy]# touch test.txt

[root@moban oldboy]# ll

总用量 0

-rw-r--r--. 1 root root 0 10月 1 15:00 test.txt

[root@moban oldboy]# chmod u+x,g+x,o=rwx test.txt

[root@moban oldboy]# ll

总用量 0

-rwxr-xrwx. 1 root root 0 10月 1 15:00 test.txt

**chmod小结：**

**用户或用户组定义：**

u 代表属主/用户（owner/user）

g 代表属组（group）

o 代表其他用户（other）

a 代表属主、属组和其他用户，也就是上面三个用户（或组）的所有（all）

**权限定义字母：**

r 代表读权限，用数字4表示；

w 代表写权限，用数字2表示；

x 代表执行权限，用数字1表示；

- 代表没有权限，用数字0表示；

**权限增减字符：**

+ 添加某个权限

- 取消没有权限。

= 取消其他所有权限赋予给定的权限。

**生产案例：网站的服务用户为oldboy**

防止木马入侵：

文件和目录给什么权限，安全临界点：

对于文件 给什么权限比较安全？  644 用户和组都是root

(rw-r--r--)###普通用户只可读文件  
对于目录 给什么权限比较安全？  755 用户和组都是root

（rwx-r-xr-x）###普通用户可列表，可切换到目录，但不能修改与删除文件。

#####特殊参数：-R，如果给目录定义权限的话，那么目录下的目录和文件也将变更权限。#####

[root@moban ~]# cd /oldboy/

[root@moban oldboy]# ll

总用量 8

drwxr-xr-x. 7 root root 4096 10月 1 20:10 test

-rw-r--r--. 1 root root 19 10月 1 19:14 test.sh

[root@moban oldboy]# cd ..

[root@moban /]# chmod -R 000 oldboy/

[root@moban /]# ll -d oldboy/

d---------. 3 root root 4096 10月 1 20:09 oldboy/

[root@moban /]# cd oldboy/

[root@moban oldboy]# ll

总用量 8

d---------. 7 root root 4096 10月 1 20:10 test

----------. 1 root root 19 10月 1 19:14 test.sh

1. **umask相关的命令**

**控制文件和目录权限的一个命令。**

**########此节摘自老男孩老师博文#########**

umask是通过八进制的数值来定义用户创建文件或目录的默认权限。  
umask 数值表示的是禁止的权限。

创建文件默认最大权限为666 （-rw-rw-rw-），默认创建的文件没有可执行权限x位。

对于文件来说，umask的设置是在假定文件拥有八进制666的权限上进行的，文件的权限就是666减去umask(umask的各个位数字也不能大于6，如，077就不符合条件)的掩码数值；重点在接下来的内容，如果umask的部分位或全部位为奇数，那么，在对应为奇数的文件权限位计算结果分别再加1就是最终文件权限值。

创建目录默认最大权限777（-rwx-rwx-rwx）,默认创建的目录属主是有x权限，允许用户进入。

对于目录来说，umask的设置是在假定文件拥有八进制777权限上进行，目录八进制权限777减去umask的掩码数值。

**文件权限的一般计算方法：**

默认文件权限计算方法

1）假设umask值为：022（所有位为偶数）

6 6 6 ==>文件的起始权限值

0 2 2 - ==>umask的值

---------

6 4 4

2）假设umask值为：045（其他用户组位为奇数）

6 6 6 ==>文件的起始权限值

0 4 5 - ==>umask的值

---------

6 2 1 ==>计算出来的权限。由于umask的最后一位数字是5，所以，在其他用户组位再加1。

0 0 1 +

---------

622 ==>真实文件权限

默认目录权限计算方法

7 7 7 ==>目录的起始权限值

0 2 2 - ==>umask的值

---------

7 5 5

规范成图表如下：



实例一：umask所有位全为偶数时，多数读者对这个例子无疑问

[root@oldboy oldboy]# umask

0022 #→umask当前数值

[root@oldboy oldboy]# umask 044 #→更改为044

[root@oldboy oldboy]# umask

0044

[root@oldboy oldboy]# mkdir umask\_test #→建目录测试

[root@oldboy oldboy]# ls -ld umask\_test

drwx-wx-wx 2 root root 4096 Nov 12 19:21 umask\_test #→对应数字权限为733,是不是符合上面的计算方法？

[root@oldboy oldboy]# touch umask\_test.txt

[root@oldboy oldboy]# ls -l umask\_test.txt

-rw--w--w- 1 root root 0 Nov 12 19:21 umask\_test.txt #→对应数字权限为622,是不是符合上面的计算方法？

实例二：umask值的部分或全部位为奇数时，这个是读者疑问最大的

当umask值的其他属组位为奇数时

[root@oldboy oldboy]# umask 0023

[root@oldboy oldboy]# mkdir dir

[root@oldboy oldboy]# touch file

[root@oldboy oldboy]# ls -l

总计 4

drwxr-xr-- 2 root root 4096 11-15 01:04 dir #→对应数字权限为754

-rw-r--r-- 1 root root 0 11-15 01:04 file #→对应数字权限为644

提示：根据前面的计算方法，当umask为0023时，dir的权限应该是754,而file的权限应该为643，但是由于umask的其他组位为奇数，因此最终权限为其他组位加1，即643加001（对应实践结果644）。注意：umask为偶数的位不要加1。

实例三：umask值的所有位为奇数时

[root@oldboy oldboy]# umask 0551

[root@oldboy oldboy]# umask

0551

umask 为0551 根据掩码方法计算：目录权限为226，文件权限115，而实际文件权限为226（umask的三个权限位都是奇数，所以，每个位分别加1就是正确的权限）

[root@oldboy oldboy]# mkdir dir5

[root@oldboy oldboy]# touch file5

[root@oldboy oldboy]# ls -l

总计 4

d-w--w-rw- 2 root root 4096 11-15 01:27 dir5 #→目录对应数字权限为226

--w--w-rw- 1 root root 0 11-15 01:27 file5 #→目录对应数字权限为226

再来一例验证下：

[root@oldboy oldboy]# umask 0333

[root@oldboy oldboy]# umask

0333

umask 为0333 根据掩码方法计算：目录权限为444，文件权限333，而实际文件权限为444（umask的三个位都是奇数，所以，每个位分别加1就是正确的权限）

[root@oldboy oldboy]# mkdir dir3

[root@oldboy oldboy]# touch file3

[root@oldboy oldboy]# ls -l|grep 3

dr--r--r-- 2 root root 4096 11-15 01:30 dir3 #→目录对应数字权限为444

-r--r--r-- 1 root root 0 11-15 01:30 file3 #→目录对应数字权限为444

2）通过8进制字母符号计算（记忆此法也可）

拿上面的实例3验证。

[root@oldboy oldboy]# umask 551

umask值为551,对应的权限为-r-xr-x--x，即所有的文件和目录都取消权限中的-r-xr-x--x权限。

文件权限：【-rw-rw-rw-】-【-r-xr-x--x】=【--w--w--rw-】=226（文件的x位上没有x，就不用取消了）

目录权限：【-rwxrwxrwx】-【-r-xr-x--x】=【--w--w--rw-】=226（目录的x位上有x，直接取消即可）

继续拿前面的例子验证：

[root@oldboy oldboy]# umask 333

umask值为333,对应的权限为-wx-wx-wx，即所有的文件和目录都取消权限中的-wx-wx-wx权限。

文件权限：【-rw-rw-rw-】-【-wx-wx-wx】=【-r--r--r--】=444（文件的x位上没有x，就不用取消了）

目录权限：【-rwxrwxrwx】-【-wx-wx-wx】=【-r--r--r--】=444（目录的x位上有x，直接取消即可）

**综合2.2文件、目录权限实例来进行的对umask的小结：**

1、超级管理员的umask默认为0022；incahome组中oldboy与oldgirl的用户umask默认为0022；其他用户test的umask值为0002。

2、若想umask默认值为0022的条件有两点：

（1）用户与用户组的名字不能相同。

（2）UID的默认值要小于199。

**查看umask权限的路径：**

[root@moban ~]# sed -n '65,69p' /etc/bashrc

if [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then

umask 002

else

umask 022

fi

1. **setuid和setgid位**

**4.1 回顾Linux权限属性及权限小结**

ls –l 结果中的第二列-rwxr-xr-x. 用户 用户组 时间 文件名

r read 读

w write 写

x 执行

**对于文件来讲：**

r 读取文件内容

w 修改文件内容，删除文件和文件的w无关。

x 执行权限，文件本身就要可以执行，需要r配合。

**对于目录来讲：**

r 浏览目录里内容的权限，需要x配合。

w 创建、删除目录内容文件的权限，和文件本身无关。

x 进入目录的权限。

**umask：控制Linux系统默认权限的一个命令。**

File 644

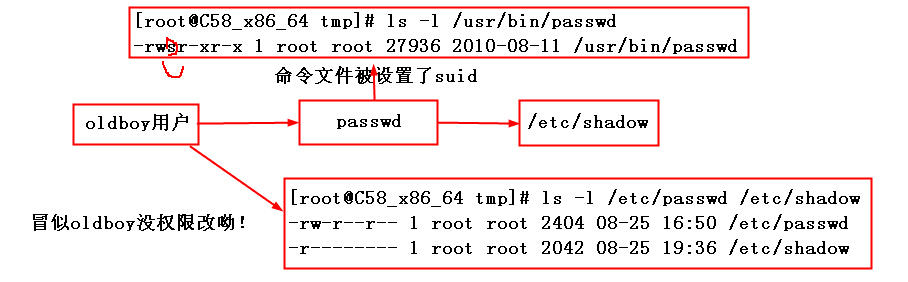
Dir 755

**改变文件及目录的默认权限：**

Umask 044 033

**4.2 Linux权限中的特殊权限位**

Linux系统基本权限位默认为9位权限，实际是12位权限位，额外3位权限位： suid、sgid、sticky



[root@moban ~]# ls -l `which passwd`

-rwsr-xr-x. 1 root root 30768 2月 22 2012 /usr/bin/passwd

此时在用户权限位上x的位置上有一个s，这个s就是setuid权限位。（当执行权限位有x时，为s，如果没有x，那么就会变成S）

setuid目录的作用：

当普通用户执行这条命令的时候，拥有这个命令对应属主（root）的权限。（仅限于在执行命令的过程中）

**特殊权限位基本说明：**

**Linux系统基本权限位为9位权限，但还有额外3位权限位，共12位权限。**

suid s(x) 4 用户对应的权限位

sgid s(x) 2 用户组对应的权限位

sticky t(x) 1 其他用户对应的权限位

设置特殊权限位方法：chmod u+s，g+s，o+t 命令/目录

**4.3** **suid的介绍**

设置特殊权限位数字方法：chmod 7755 oldboy.txt

suid设置：  
[root@oldboy new]# chmod 4644 f1  
[root@oldboy new]# ls -l f1  
-rwSr--r-- 1 root root 0 Aug  9 07:11 f1  
[root@oldboy new]# chmod +x f1  
[root@oldboy new]# ls -l f1     
-rwsr-xr-x 1 root root 0 Aug  9 07:11 f1  
sgid设置：  
[root@oldboy new]# chmod g+s f1  
[root@oldboy new]# ls -l f1      
-rwsr-sr-x 1 root root 0 Aug  9 07:11 f1

设置粘滞位：  
[root@oldboy new]# chmod o+t f1  
[root@oldboy new]# ls -l f1      
-rwsr-sr-t 1 root root 0 Aug  9 07:11 f1  
[root@oldboy new]# ls -l f1     
-rwSr-Sr-T 1 root root 0 Aug  9 07:11 f1

**执行此命令后，显示有setuid特殊权限的命令如下：**

[root@moban ~]# find / -type f -perm 4755|xargs ls –l

-rwsr-xr-x. 1 root root 77336 10月 15 2014 /bin/mount

-rwsr-xr-x. 1 root root 38200 7月 24 01:55 /bin/ping

-rwsr-xr-x. 1 root root 36488 7月 24 01:55 /bin/ping6

-rwsr-xr-x. 1 root root 34904 10月 15 2014 /bin/su

-rwsr-xr-x. 1 root root 53472 10月 15 2014 /bin/umount

-rwsr-xr-x. 1 root root 10272 10月 15 2014 /sbin/pam\_timestamp\_check

-rwsr-xr-x. 1 root root 34840 10月 15 2014 /sbin/unix\_chkpwd

-rwsr-xr-x. 1 root root 54496 2月 20 2015 /usr/bin/at

-rwsr-xr-x. 1 root root 66352 4月 7 17:52 /usr/bin/chage

-rwsr-xr-x. 1 root root 51784 3月 30 2015 /usr/bin/crontab

-rwsr-xr-x. 1 root root 71480 4月 7 17:52 /usr/bin/gpasswd

-rwsr-xr-x. 1 root root 36144 4月 7 17:52 /usr/bin/newgrp

-rwsr-xr-x. 1 root root 30768 2月 22 2012 /usr/bin/passwd

-rwsr-xr-x. 1 root root 22544 3月 17 2015 /usr/bin/pkexec

**suid小结：针对命令和二进制程序的**

1. 用户（属主）权限位上x的位置上有一个s，这个s就是setuid权限位。

当执行权限位有x时，为小s，如果没有x，那么就会变成大S

1. suid作用是让普通用户可以以root（或属主）的用户角色运行只有root（或属主）才能运行的程序或命令，或程序命令对应本来没有权限操作的文件等。
2. 问题：希望oldboy用户能够删除本来无权删除的文件。
3. sudo给oldboy授权rm。
4. 给rm命令设置suid。
5. 设置上级目录权限。
6. suid修改的是执行的命令passwd，而不是处理的目标文件/etc/shadow。

[root@moban oldboy]# touch lifen.sh

[root@moban oldboy]# ll

总用量 0

-rw-r--r--. 1 root root 0 10月 1 18:39 lifen.sh

[root@moban oldboy]# chmod u+s lifen.sh

[root@moban oldboy]# ll lifen.sh

-rwSr--r--. 1 root root 0 10月 1 18:39 lifen.sh

1. 仅对二进制命令和程序有效，不能用在shell等类似脚本文件上。
2. 二进制命令程序需要有可执行权限x配合。
3. suid权限仅在程序命令执行过程中有效。
4. 执行suid命令的任意系统用户都可以获得该命令程序在执行期间对应的拥有者的所有权限。
5. suid是一把双刃剑，是一个比较危险的功能，对系统安全有一点的威胁。系统suid的无用的功能取消suid权力（安全优化）

如何查找系统中设置了suid的命令

[root@moban ~]# find / -type f -perm 4755|xargs ls –l

-rwsr-xr-x. 1 root root 77336 10月 15 2014 /bin/mount

-rwsr-xr-x. 1 root root 38200 7月 24 01:55 /bin/ping

-rwsr-xr-x. 1 root root 36488 7月 24 01:55 /bin/ping6

-rwsr-xr-x. 1 root root 34904 10月 15 2014 /bin/su

**setuid基于命令实战实例应用**

现让一个普通用户oldboy拥有root用户才能拥有的rm删除权限，除了使用su或sudo临时切换到root身份操作外，还可以使用如下命令：

[root@moban oldboy]# cd /home

[root@moban home]# touch oldboytest.txt

[root@moban home]# ls -l oldboytest.txt

-rw-r--r--. 1 root root 0 10月 1 18:55 oldboytest.txt

[root@moban home]# su - oldboy

[oldboy@moban ~]$ cd /home/

[oldboy@moban home]$ rm -f oldboytest.txt

rm: 无法删除"oldboytest.txt": 权限不够

此时该本次主角setuid与setgid出场了。

[root@moban ~]# ls -l /bin/rm

-rwxr-xr-x. 1 root root 57440 10月 15 2014 /bin/rm

[root@moban ~]#

[root@moban ~]# chmod 4755 /bin/rm

[root@moban ~]# ls -l /bin/rm

-rwsr-xr-x. 1 root root 57440 10月 15 2014 /bin/rm

[root@moban ~]# su - oldboy

[oldboy@moban ~]$ cd /home/

[oldboy@moban home]$ ls -l oldboytest.txt

-rw-r--r--. 1 root root 0 10月 1 18:55 oldboytest.txt

[oldboy@moban home]$ rm -f oldboytest.txt

[oldboy@moban home]$ ls -l oldboytest.txt

ls: 无法访问oldboytest.txt: 没有那个文件或目录

如上内容所示，设置了rm的setuid位后，就让普通用户在rm指令上有了root才能享有的删除权力。

通过这个例子，可以理解setuid与setgid位的应用了，如同前面所说，让普通用户超越本身的能力，让普通用户能执行只有root才能执行的命令。在这一点，要和su和sudo区分开来。

**4.4 setgid权限知识**

sgid知识总结介绍：

1. 与suid不同的是，sgid即可以针对文件也可以针对目录设置。
2. Sgid是针对用户组权限位修改的。

**对于文件来说，sgid的功能如下：**

1. sgid仅对二进制命令程序有效。
2. 二进制命令或程序需要有可执行权限x
3. 执行程序的任意用户可以获得该命令程序执行期间所属组的权限。

**对于目录：sgid的功能如下：**

1） Linux默认情况所有用户创建文件，默认用户和组都是自身。

2） sgid可以让用户在此目录下创建文件和目录，具有和此目录相同的用户组设置。

setgid位主要用在目录中，为某个目录设置了setgid位以后，在该目录中新创建的文件具有该目录的所有者权限，而不是创建该文件的用户的默认所有者。这样，使得在多个用户之间共享一个目录中的文件变得简单。提示：用八进制数2000表示setgid权限位。

实例：

[oldboy@oldboy ~]$ sudo su -  
Welcome to oldboy training,26..  
[root@oldboy ~]# mkdir /home/admins  
[root@oldboy ~]# groupadd adminuser  
[root@oldboy ~]# ls -ld /home/admins/  
drwxr-xr-x 2 root root 4096 10月 23 17:11 /home/admins/  
[root@oldboy ~]# chown .adminuser /home/admins/  
[root@oldboy ~]# ls -ld /home/admins/            
drwxr-xr-x 2 root adminuser 4096 10月 23 17:11 /home/admins/  
[root@oldboy ~]# touch /home/admins/ddd  
[root@oldboy ~]# ll /home/admins/  
总用量 0  
-rw-r--r-- 1 root root 0 10月 23 17:12 ddd  
[root@oldboy ~]# chmod 2755 /home/admins/  
[root@oldboy ~]# ls -ld /home/admins/      
drwxr-sr-x 2 root adminuser 4096 10月 23 17:12 /home/admins/  
[root@oldboy ~]# touch /home/admins/msn  
[root@oldboy ~]# ll /home/admins/            
总用量 0  
-rw-r--r-- 1 root root      0 10月 23 17:12 ddd  
-rw-r--r-- 1 root adminuser 0 10月 23 17:13 msn

**4.5** **粘滞位相关的知识**

例子：

查看粘滞位信息如下所示：

[root@moban ~]# ls -ld /tmp/

drwxrwxrwt. 6 root root 4096 10月 2 01:38 /tmp/

粘滞位概念：

粘滞位的功能现在也很少用了，不过对于像/tmp目录这样的，是整个系统临时文件存放地，还是有点意义。一个目录即使他的所有权限都开放rwxrwxrwx，如果是设置了粘滞位，除非目录的属主和root用户有权限删除它，除此之外其他用户不能删除这个目录。用途一般是把一个文件夹的权限都打开，然后来共享文件，像/tmp目录一样，方便带来的安全隐患，生产环境我们一般不这样使用！

**特殊权限对应的数字小结：**

1、Suid： 4000 权限字符 s （S），用户为上的x位上设置。

授权方法： chmod 4755 oldboy.txt 或chmod u+s test

2、Sgid： 2000 权限字符s（S），用户组位的x为上设置。

授权方法：chmod 2755 oldboy.txt或chmod g+s test

3、粘滞位： 1000 权限字符t（T），其他用户位的x位上设置。

授权方法 chmod 1777 /tmp 或 chmod o+t /tmp

如果对应位有x，则出神入化权限表现为小写，否则表现为大写。

tmp经典的粘滞位目录案例，特点，谁都有写权限，因此安全成问题。常常是木马第一手跳板地点。

chown命令：

chown 用户 文件或目录 #<===仅仅授权用户。

chown ：组 文件或目录 #<===仅仅授权组。等同于“chgrp组 文件或目录”

Chown 用户：组 文件或目录 #<===表示授权用户和组

强调：

1. 其中的冒号“：”可以用点号“.”替代。
2. 要授权的用户和组名，必须是Linux系统里实际存在的。

[root@oldboy oldboy]# ls -l test.txt  
-rwxrwxrwx 1 root root 23 10月 20 19:09 test.txt  
[root@oldboy oldboy]# chown oldboy test.txt  
[root@oldboy oldboy]# ls -l test.txt         
-rwxrwxrwx 1 oldboy root 23 10月 20 19:09 test.txt  
[root@oldboy oldboy]# chown :oldgirl test.txt   
chown: 无效的组: ":oldgirl"  
[root@oldboy oldboy]# chown :incahome test.txt  
[root@oldboy oldboy]# ls -l test.txt            
-rwxrwxrwx 1 oldboy incahome 23 10月 20 19:09 test.txt  
[root@oldboy oldboy]# chown root.root test.txt  
[root@oldboy oldboy]# ls -l test.txt            
-rwxrwxrwx 1 root root 23 10月 20 19:09 test.txt

**改变文件的属组的命令chgrp （change group）**

语法：chgrp oldboy fff

特殊权限位需谨记一点：

[root@moban tmp]# ll

总用量 8

-rw-r--r--. 1 oldgirl incahome 0 10月 2 02:27 gongquanle.txt

-rw-rw-r--. 1 test test 0 10月 2 02:27 ldk.haha

-rw-r--r--. 1 oldboy incahome 0 10月 2 02:25 lifen.sh

drwx------. 2 root root 4096 10月 1 21:26 ssh-NqVyXR4683

drwx------. 2 root root 4096 10月 2 01:38 ssh-vEKmeb4950

[root@moban tmp]# userdel -r oldboy

[root@moban tmp]# userdel -r test

[root@moban tmp]# userdel -r oldgirl

[root@moban tmp]# groupdel incahome

[root@moban tmp]# ll

总用量 8

-rw-r--r--. 1 501 500 0 10月 2 02:27 gongquanle.txt

-rw-rw-r--. 1 502 502 0 10月 2 02:27 ldk.haha

-rw-r--r--. 1 500 500 0 10月 2 02:25 lifen.sh

drwx------. 2 root root 4096 10月 1 21:26 ssh-NqVyXR4683

drwx------. 2 root root 4096 10月 2 02:29 ssh-xzmzZS5290

当删除创建文件的用户和组的时候，文件信息的后面显示：用户和组全为数字表示。若想恢复就创建一个和创建是一样的组、用户。

**章节重点总结：**

主要内容小结：

1. Linux的9位权限及rwx字符的作用、对应数字及对应的用户和用户组图解。
2. Linux文件的权限说明。（rwx对于文件来说代表什么意思）
3. Linux命令的权限说明。（rwx对于目录来说代表什么意思）
4. 企业生产场景网站命令设置权限的原则。

（单台服务器网站权限控制方案upload服务器）

（集群服务器网站权限控制方案（上传、WEB、静态服务））

1. umask的作用以及和文件、命令的对应的默认权限关系。（特殊权限）
2. suid是什么、作用，passwd命令案例，rm。做了小结了。
3. sgid是什么、作用，文件locate案例。命令用户组案例。
4. 粘滞位是什么、作用，/tmp目录案例。
5. 以数字及字符方式更改权限，chmod命令。
6. 更改文件用户和组，chown命令。
7. 更改用户组的，chgrp。。####要多学，不然该被开发取代了。
8. chattr，lsattr更改文件属性。
9. setfacl知识不需要了解。