Linux基础

文件查找

**grep** ，**egrep** ，**fgrep** 文本查找，查找文本中的字符串的

# 文件查找

locate , find

## locate

非实时查找而且还是模糊匹配，根据全系统文件数据库进行的，

速度快，用的不是很多

## find

**实时查找**

**精确查找**

遍历指定目录中的所有文件完成查找，速度慢，

支持根据 文件名，文件类型，正则等查找条件，

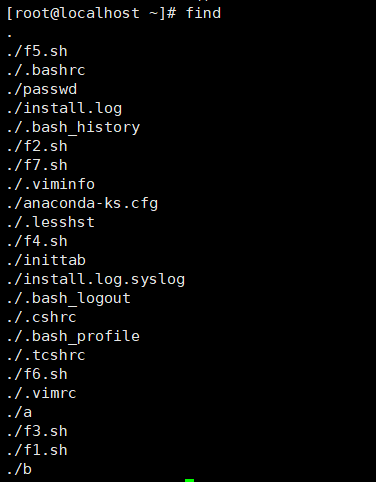
### 用法

**find 路径 查找条件 查找到以后的处理动作**

路径模式默认为当前目录

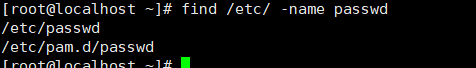
查找条件默认为当前目录下的所有文件

处理动作默认为标准输出显示



### 匹配条件

**-name**  对文件名做精确匹配，区分大小写



文件名通配

**\***  匹配任意长度的任意字符

**?** 匹配任意单个字符

**[ ]**  匹配指定范围内的单个字符

[abc] [a-z] [A-Z] [0-9] [a-zA-Z] [0-9a-zA-Z]

[:space:] 表示空白字符

[[:space:]] 匹配有空白字符的文件

[:punct:] 所有的标点符号

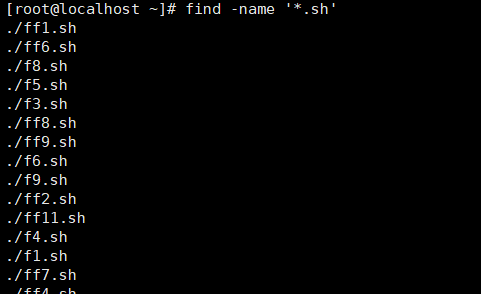
[:lower:] 所有的小写字母

[:upper:] 所有的大写字母

[:alpha:] 所有的大小写字母

[:digit:] 数字

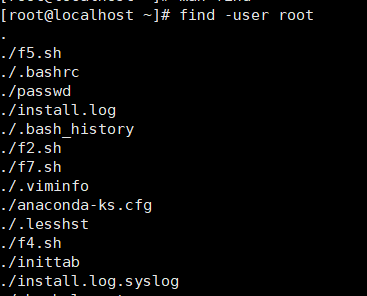
[:alnum:] 大小写字母和数字



**-iname** 文件名不区分大小写

**-regex pattern** 根据模式来查找

**-user** 根据用户查找



**-group** 根据组名查找

**-uid** 根据用户id查找

**-gid** 根据组id查找

**-nouser** 没有属主的文件

**-nogroup** 没有属组的文件

**-type** 根据文件类型查找

f 普通文件

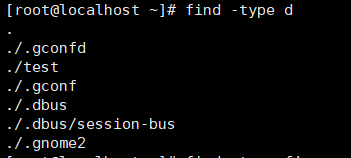
d 目录

c 字符设备

p 管道文件

s 套接字文件

l 软连接文件



**-size** 根据文件大小查找,默认k

k kb

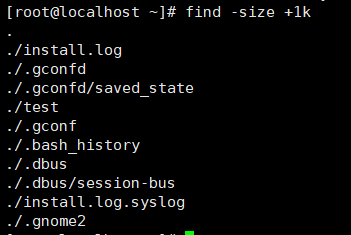
M

G

+10k 大于10k

-10k 小于10k

10k 等于10k



### 组合条件

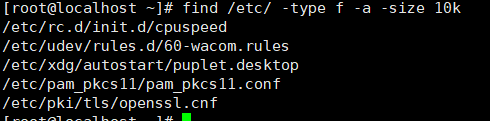
默认是与

**-a** 与

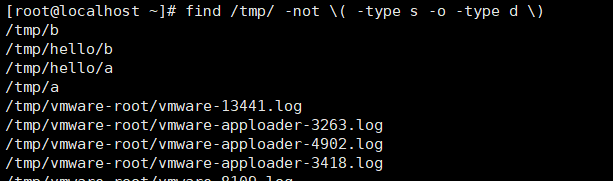
**-o** 或

**-not** 非

查找类型是普通文件并且大小为10kb的文件



find /tmp/ -not -type s -a -not -type d == find /tmp/ -not \( -type s -o -type d \)



### 根据时间戳查找

**天**

**-mtime** 修改时间

**-ctime**  创建时间

**-atime** 访问时间

**+#** +5 离现在为止至少5天，5天以前访问过

**-#** -5 最近5天之内访问过的

**#** 5 离现在5天的时候访问过

**分**

**-mmin**

**-cmin**

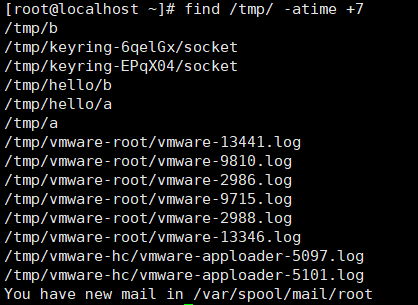
**-amin**

**+#**

**-#**

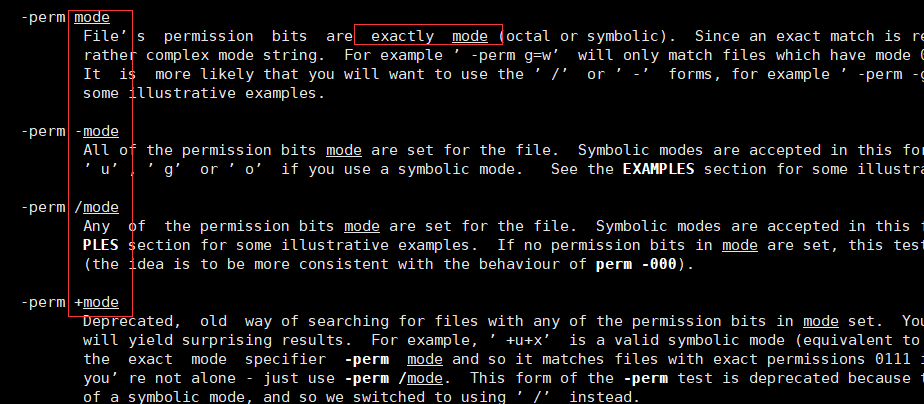
**#**

find /tmp/ -atime +7 已经7天没有访问过的文件



### 根据权限查找

**-perm**



**+mode**

过时

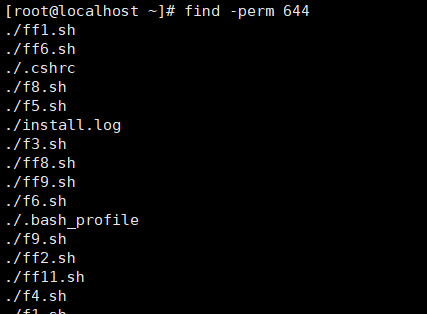
**-mode**

精确匹配，才符合条件显示

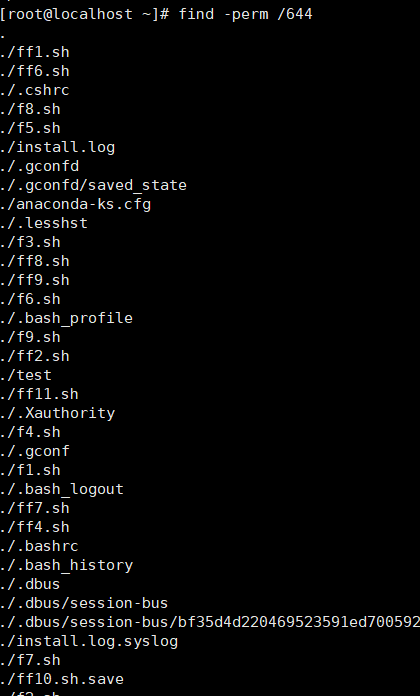
**/mode**

任意一位包含mode就满足条件

find -perm 644 权限精确匹配



find -perm /644



find -perm -644 文件权限完全包含此mode

-744

-755 等等

find -perm -001 查找其他用户有执行权限的文件

find -perm -022 查找组有写权限**并且**其他也有写权限

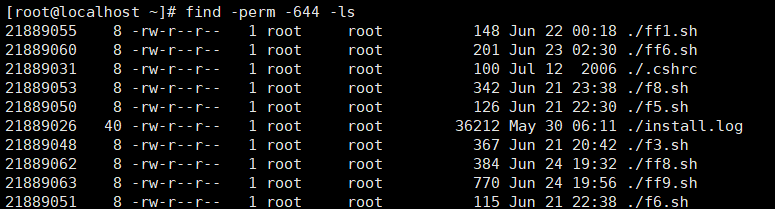
find -perm /022 查找组有写权限**或者**其他也有写权限

find -perm -007 其他用户有rwx的权限文件

### 动作

**-print** 显示

**-ls** 类似ls -l 显示每个文件的详细信息



**-ok command {} \; {} 代表找的文件**

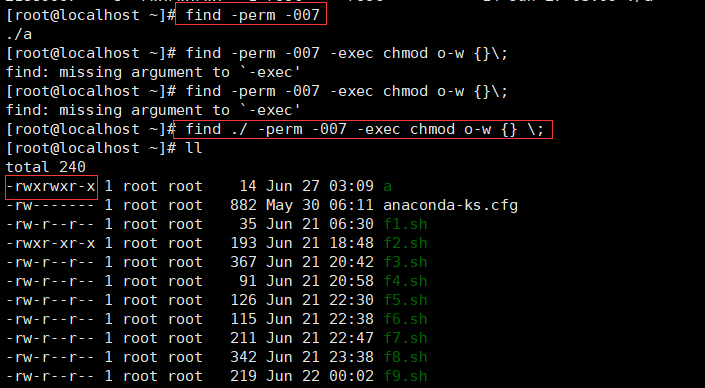
**或者**

**-exec**

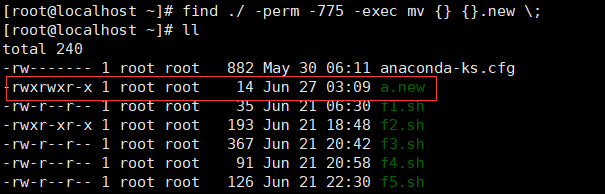
-ok 每一个操作需要用户确认，-exec 不需要用户确认

find ./ -perm -007 -exec chmod o-w {} \;

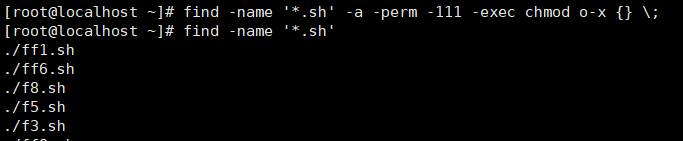
找到其他有rwx的权限并把w权限去掉

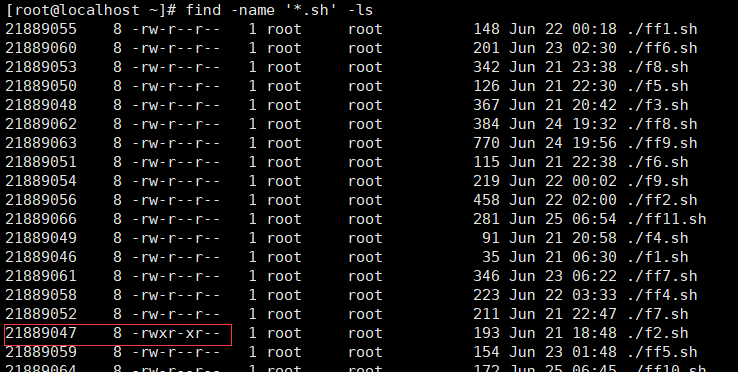


**find ./ -perm -775 -exec mv {} {}.new \;**



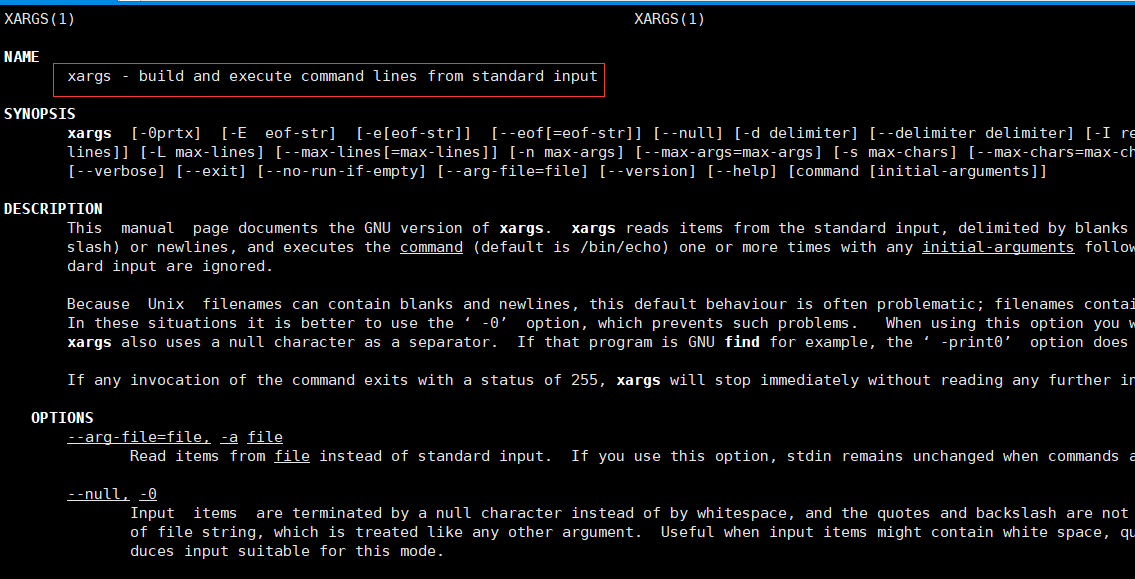
去掉.sh文件其他用户的执行权限

find -name '\*.sh' -a -perm -111 -exec chmod o-x {} \;



### xagrs

**from standard input**

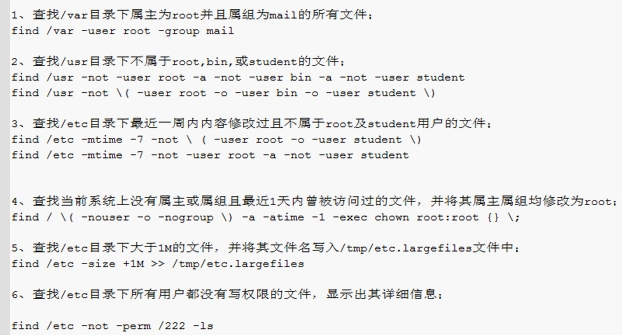


find /tmp/ -size +1M -exec echo {} >> /tmp/largfile \;

xargs不需要占位符，不需要\ ; 只需要管道

find /tmp/ -size +1M | xargs echo >> /tmp/largefile

**练习**



# 特殊权限

## SUID

运行某程序时，相应进程的属主是程序文件自身的属主，而不是启动者

添加权限

**chmod u+s file**

**chmod u-s file**

如果file有执行权限,则SUID为显示为s ,否则显示为S

**这个权限不能乱给**

特殊场景之一

passwd 命令 普通用户也能修改自己的命令,需要特殊权限,以管理员权限运行



## SGID

运行某程序时，相应进程的属组是程序文件自身的属组，而不是启动者的基本组

**chmod g+s file**

**chmod g-s file**

如果file有执行权限,则SGID为显示为s ,否则显示为S

场景之一

一个目录下的文件,一个开发团队都可以访问编写和修改.

developteam

hadoop , hbase ,hive 3个用户

1.创建3个用户

useradd hadoop

useradd hbase

useradd hive

2.创建目录 project

mkdir /tmp/project

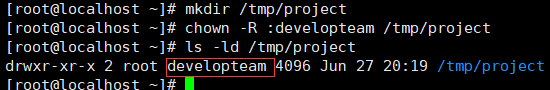
3.添加组 developteam

groupadd developteam

4.修改project目录的**属组权限**

-R 递归修改

chown -R :developteam /tmp/project



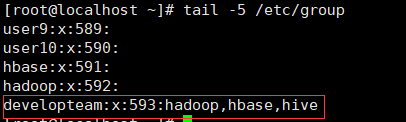
**可知 project目录对于属组developteam的权限是r-x,(5)没有写权限**

5. 为新添加的3个用户指定额外组

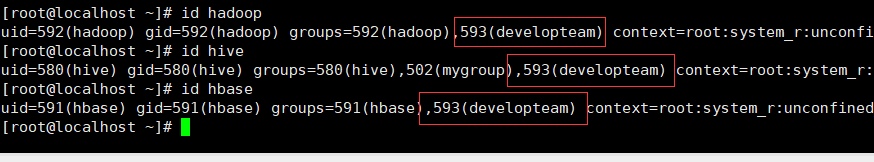
usermod -a -G developteam hadoop

-a -G 一起使用避免将将用原有的附加组覆盖,而是在尾部追加附加组





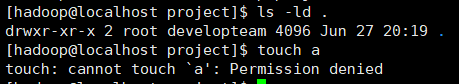
developteam 就有3个附加组



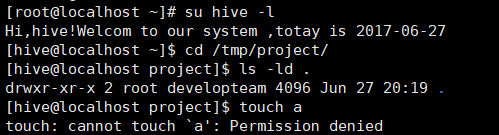
hadoop用户创建文件是没有权限的，

**project目录的属组developteam的权限是r-x,(5)没有写权限**

**所以developteam组的成员hadoop不能创建文件**

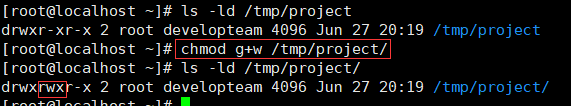


**同样developteam组的其他成员hive ，hbase也不能创建文件**

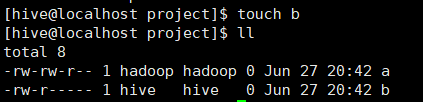


**怎么办？赋写权限**

chmod g+w /tmp/project

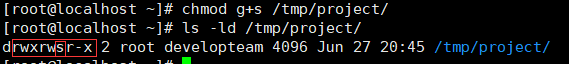


让project目录的属组developteam有写权限，那么developteam的成员hadoop,hive,hbase就有写权限了。



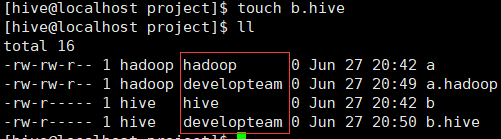
**给project目录添加SGID权限，**

chmod g+s /tmp/project/



加上SGID权限后，创建文件 a.hadoop b.hive 的属组就变成了project目录的属组权限

developteam



所以 developteam属组的成员hadoop,hive,hbase 就可以任意访问编辑这个project目录下的文件了，因为project目录的属组也是developteam.

但是问题来了，就可以任意删除别人的文件。怎么解决？ sticky来帮忙

## sticky

冒险位

应用场景

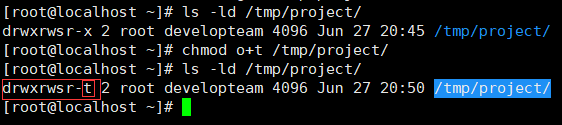
在一个公共目录，每个用户都可以创建文件，能删除自己的文件，但是不能删别人的文件。

如果file有执行权限,则sticky为显示为t ,否则显示为T

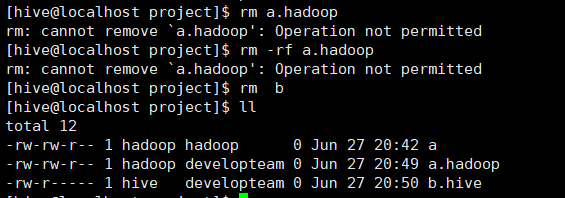
**chmod o+t file**

**chmod o-t file**

chmod o+t /tmp/project/



hive用户不论怎么删都不能删除hadoop用户的创建的文件



SUID SGID sticky 3个位进行组合

000 0 无

001 1 sticky

010 2 SGID

011 3 SGID sticky

100 4 SUID

101 5 SUID stick

110 6 SUID SGID

111 7 SUID SGID sticky

chmod 1755 file 1 sticky

chmod 2755 file 2 SGID

chmod 3755 file 3 SGID sticky

chmod 4755 file 4 SUID

chmod 5755 file 5 SUID sticky

chmod 6755 file 5 SUID SGID

chmod 7755 file 5 SUID SGID sticky

这就是为什么umask为4位 最开始一位就是特殊权限。

