linux 基础

## facl

文件系统访问控制列表

tom :

tom tom基本组

jerry:

jerry jerry基本组

tom 创建一个文件jerry也可以rw ,如果jerry不属于tom基本组,那么就无法rw.那么tom只能将rw权限给o, 也可以使用SUID但是这种方法不安全.不建议,如过给o rw权限 ,那么不仅仅是jerry 可以rw,其他所有人都可以rw

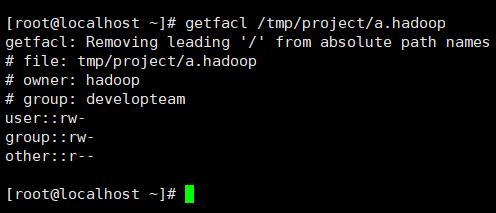
基于linux系统安全模型,这样给o加rw,文件的风险大大增加,位了避免这种现象,FACL排上用处.

FACL File System Access Control List 文件系统访问控制列表

**利用文件的扩展属性保存了额外的控制权限**.

**getfacl**

获取文件的访问控制列表



**setfacl**

**-m 设定**

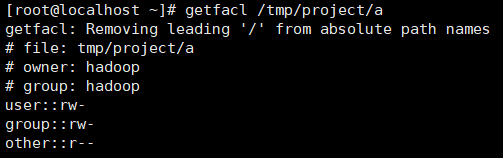
**u:name/UID:perm file**

**g:name/GID:perm file**

**-x 取消**

**u:name file**

**g:name file**



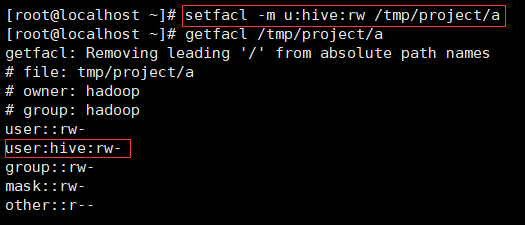
hive 用户想对a文件w 但是没有权限



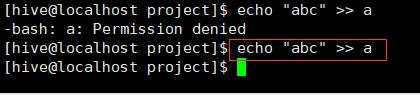
### 设定

**设定用户**

setfacl -m u:hive:rw /tmp/project/a

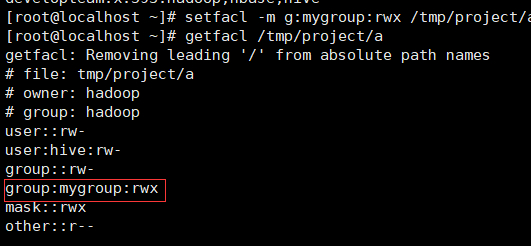


hive用户就可以对a文件w



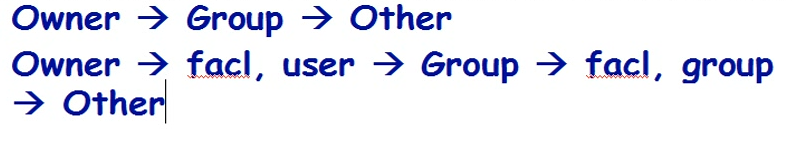
**设定组**

setfacl -m g:mygroup:rwx /tmp/project/a



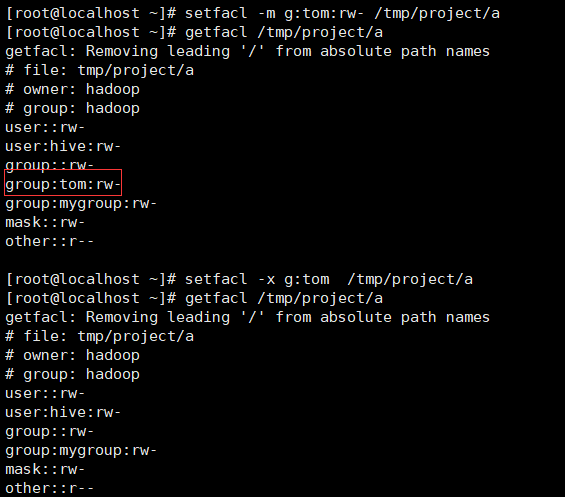
如果有了facl那么文件权限的应用次序为

**owner - facl,user - group - facl,grop -other**

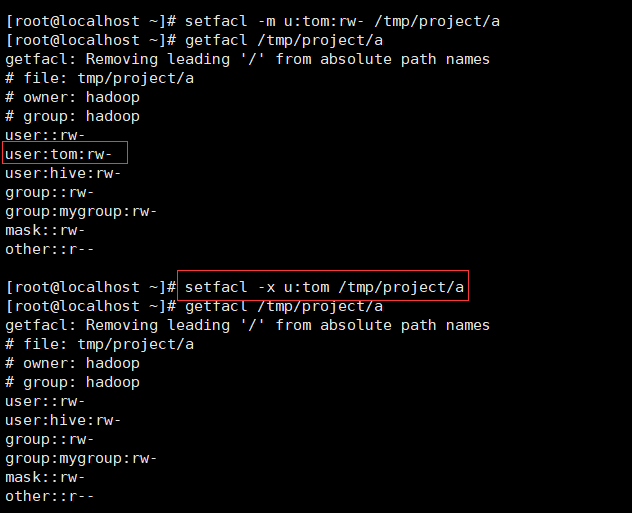


### 取消

setfacl -x g:tom /tmp/project/a



[root@localhost ~]# setfacl -x u:tom /tmp/project/a



**+** 具有扩展属性



### 为目录设定默认访问控制列表

**setfacl**

**-m 设定**

**d:u:name/UID:perm file**

**d:g:name/GID:perm file**

**-x 取消**

**u:name file**

**g:name file**

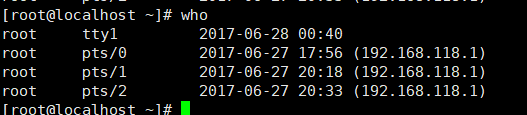
子目录就自动继承父目录的访问控制列表

## 终端类型

### who

whoami 显示当前登录用户

who 显示系统用户有哪些



Linux终端类型:

1. console 控制台 直接连到物理机

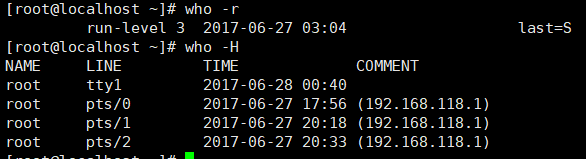
pty 物理终端 = console (VGA)

2. tty# 虚拟控制台 通常在本地 (VGA)

3. ttys 串行终端

4. pts/# 伪终端 通常是通过其他软件远程链接 如xshell

who -r 显示当前运行级别



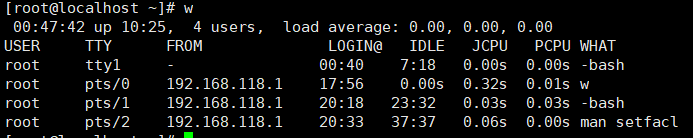
who | grep “hadoop”

每隔5秒,就来查看hadoop是否已经登录,如登录显示登录,并退出.

sleep 休眠

### w

显示比who更为详细

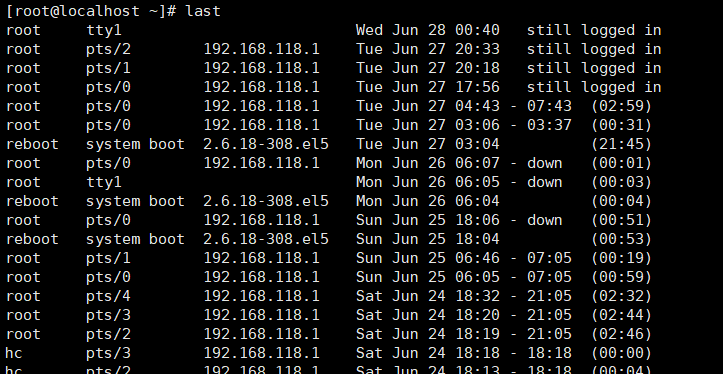


### last

显示用户登录日志

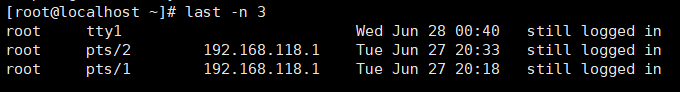
内容是在/var/log/wtmp ,记录了用户登录历史和系统重启历史

last 默认全部显示



last -n #

显示 # 次



### lastb

显示 var/log/btmp 文件的内容

显示用户错误登录尝试



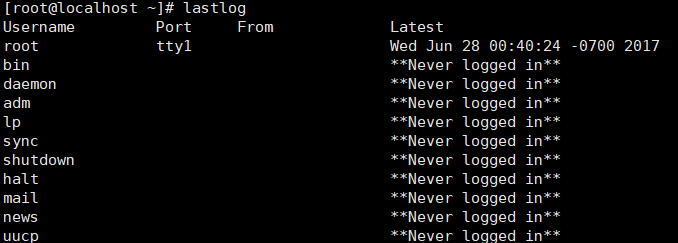
**lastb -n #**

显示前几次登录

黑客离开系统干的最后一件事情就是抹去btmp文件的登录信息.

### lastlog

显示每一个用户上次登录成功的信息



**lastlog -u username**

lastlog -u hive



### basename

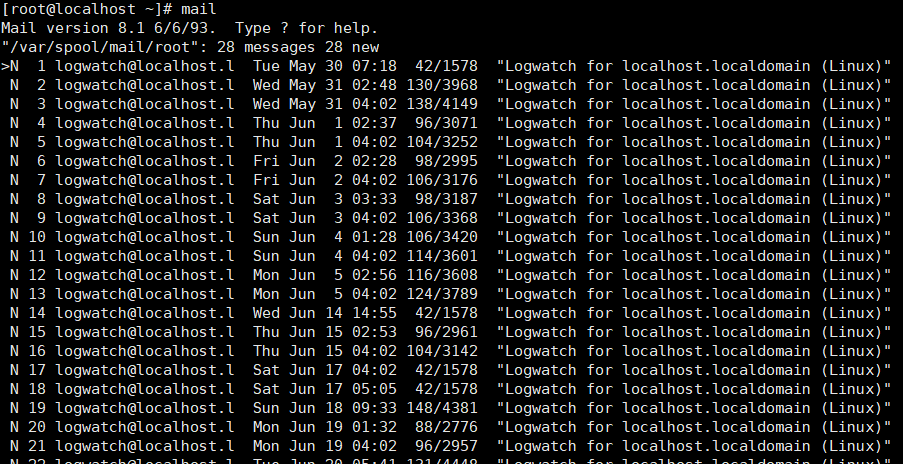
直接取得文件名



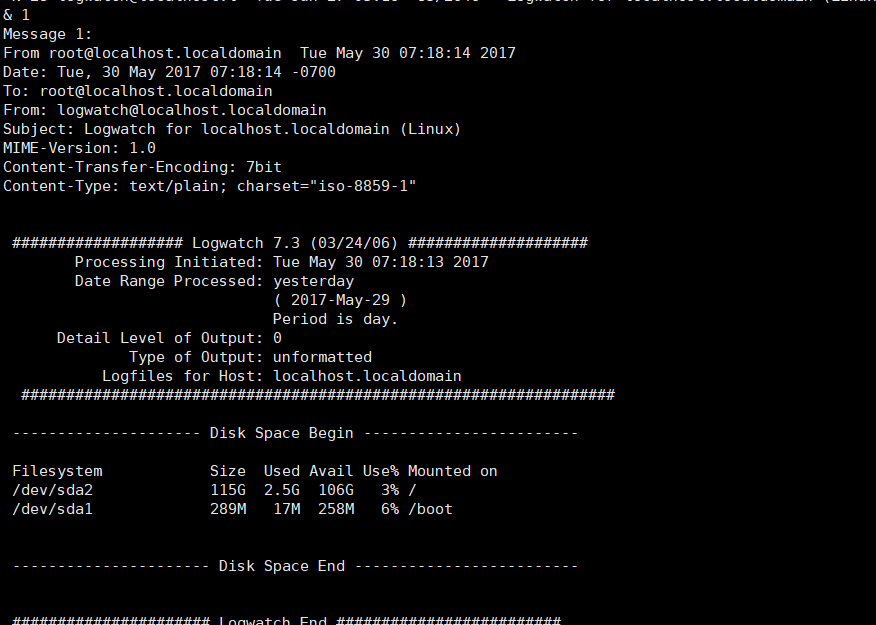
basename $0 取得执行脚本的路径

$0 执行脚本的脚本路径及名称

### mail



选择邮件编号 打开邮件



d 删除

r 回复的

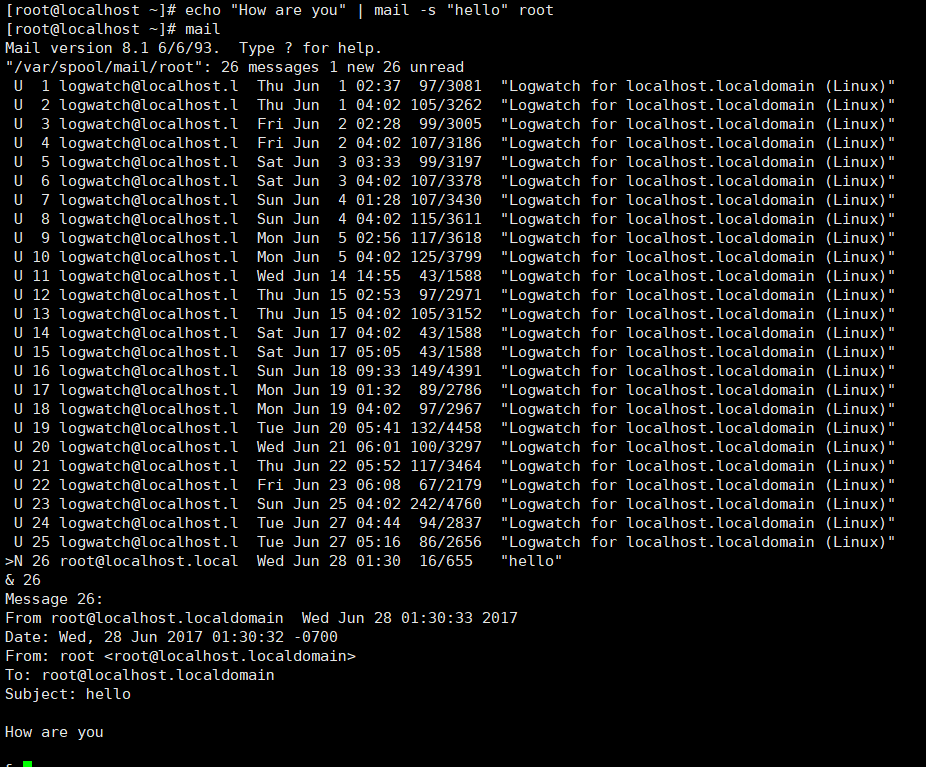
看完后就放到 mbox目录下

#### 发邮件

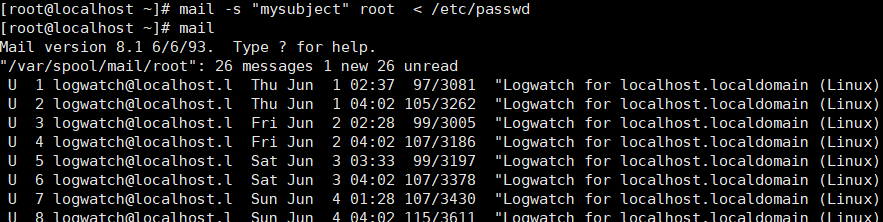
**mail -s**

给root发邮件

echo “How are you” | mail -s “hello” root



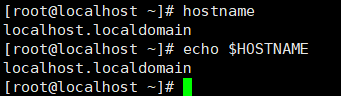
mail -s “mysubject” root < /etc/passwd



### hostname

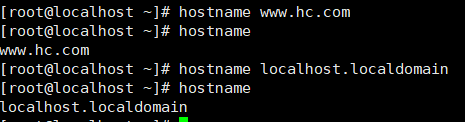
显示用户主机名

hostname == echo $HOSTNAME 环境变量名



可以修改主机名

hostname 主机名称



### 生成随机数

随机数生成器

**/dev/random**

**熵池** 保留的是你敲键盘的时间间隔,(cpu中断时间间隔)这个是真正随机 ,如果取空了,会让你敲键盘

linux 服务器将cpu中断时间间隔记录下来保存到熵池,取完了,等待更多的中断响应间隔

所以这个一旦用完就会阻塞进程.

random 更安全.

**/dev/urandom**

如果熵池取空了,就会使用软件的方式模拟随机数. 不会阻塞进程.



写一个脚本 生成10个随机数,并找出最大值

#!/bin/bash

#

declare -i MAX=0

for I in {1..10} ; do

MYRANDOM=$RANDOM

if [ $I -le 9 ] ;then

echo "$MYRANDOM,"

else

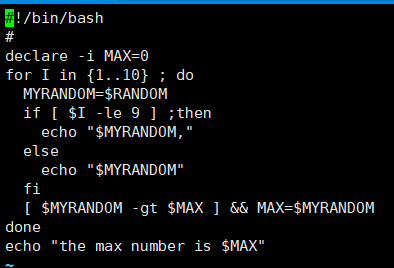
echo "$MYRANDOM"

fi

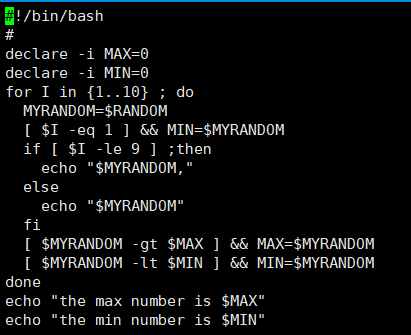
[ $MYRANDOM -gt $MAX ] && MAX=$MYRANDOM

done

echo "the max number is $MAX"



最小数



#!/bin/bash

#

declare -i MAX=0

declare -i MIN=0

for I in {1..10} ; do

MYRANDOM=$RANDOM

[ $I -eq 1 ] && MIN=$MYRANDOM

if [ $I -le 9 ] ;then

echo "$MYRANDOM,"

else

echo "$MYRANDOM"

fi

[ $MYRANDOM -gt $MAX ] && MAX=$MYRANDOM

[ $MYRANDOM -lt $MIN ] && MIN=$MYRANDOM

done

echo "the max number is $MAX"

echo "the min number is $MIN"

### case 语句

控制结构

顺序结构

选择结构

循环结构

选择结构

if condition ;then

statement

fi

if conditon ;then

statement

elif

statement

else

statement

fi

case $1 in

value1)

…

;;

value2)

…

;;

value3)

…

;;

esac

