第8讲：用户管理

1 Linux安全模型

**资源分派：**

**Authentication：认证,验证用户身份**

**Authorization：授权,不同的用户设置不同权限**

**Accouting|Audition：审计**

**当用户登录时，系统会自动分配令牌token,包括用户标识和组成员等信息**

1.1 用户

**Linux中每个用户是通过User Id （UID）来唯一标识的。**

**管理员：root, 0**

**普通用户：1-60000 自动分配**

**系统用户：**

**1-499 （CentOS 7以前）**

**1-999 （CentOS7以后）**

**对守护进程获取资源进行权限分配**

**登录用户：**

**500+ （CentOS7以前）**

**1000+（CentOS7以后）**

**给用户进行交互式登录使用**

1.2 用户组

**Linux中可以将一个或多个用户加入用户组中，用户组是通过Group ID（GID） 来唯一标识的。**

**管理员组：root, 0**

**普通组： 系统组：1-499（CentOS 7以前）, 1-999（CentOS7以后）, 对守护进程获取资源进行权限分配**

**普通组：500+（CentOS 7以前）, 1000+（CentOS7以后）, 给用户使用**

1.3 用户和组的关系

**users**

**用户的主要组(primary group)：**

**用户必须属于一个且只有一个（primary）主组，默认创建用户时会自动创建**

**和用户名同名的组，做为用户的主要组，由于此组中只有一个用户，又称为私有组**

**用户的附加组(supplementary group)： 一个用户可以属于零个或多个辅助组**

**范例**

[root@centos8 ~]#id postfix

uid=89(postfix) gid=89(postfix) groups=89(postfix),12(mail)

1.3 安全上下文

**Linux安全上下文Context：运行中的程序，即进程 (process)，以进程发起者的身份运行，进程所能够访问资源的权限取决于进程的运行者的身份**

**比如：root 身份运行/bin/cat /etc/shadow和 wang的身份运行/bin/cat /etc/shadow ，得到的结果是不同的，shadow能否能被访问是由运行者的身份决定，非程序本身**

范例:

[wang@centos8 ~]$cat /etc/shadow

cat: /etc/shadow: Permission denied

[root@centos8 ~]#cat /etc/shadow

root:$6$zsrWEC56PrKifAEz$hylCuGySe.H6l6O2MRvbtqy/VZgnZbau.y57dE85.YHq03MTJVV4UvQ

VIDcYA1IJzbgpWE0vTU.BtPHLbNBNn0:18246:0:99999:7:::

bin:\*:18027:0:99999:7:::

daemon:\*:18027:0:99999:7:::

adm:\*:18027:0:99999:7:::

lp:\*:18027:0:99999:7:::

审计

enchanced security Linux

selinux

用户 进程 资源

2 用户和组的配置文件

2.1 用户和组的主要配置文件

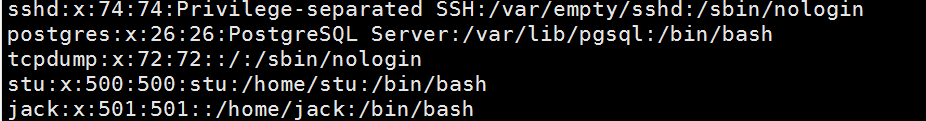
**/etc/passwd：用户及其属性信息(名称、UID、主组ID等）**

**/etc/shadow：用户密码及其相关属性 sha512**

**/etc/group：组及其属性信息**

**/etc/gshadow：组密码及其相关属性**

2.2 passwd文件格式



login name：登录用名（wang）

passwd：密码 (x)

UID：用户身份编号 (1000)

GID：登录默认所在组编号 (1000) ，initial login group

GECOS：用户全名或注释

home directory：用户主目录 (/home/wang)

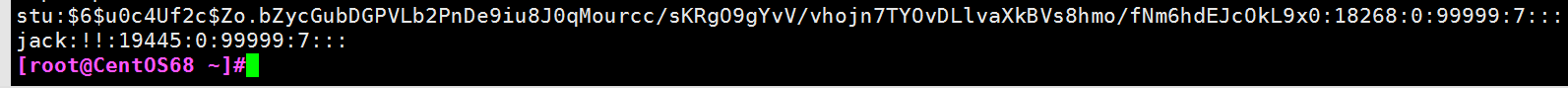
shell：用户默认使用shell (/bin/bash)

**加密方式**

**authconfig --test | grep hashing**

**/etc/sysconfig/authconfig 这个文件是authconfig命令的配置文件**

2.3 shadow文件格式



**1 登录用名**

**2 用户密码:一般用sha512加密**

**3 从1970年1月1日起到密码最近一次被更改的时间**

**4 密码再过几天可以被变更（0表示随时可被变更）**

**5 密码再过几天必须被变更（99999表示永不过期）**

**6 密码过期前几天系统提醒用户（默认为一周）**

**7 密码过期几天后帐号会被锁定**

**8 从1970年1月1日算起，多少天后帐号失效**

**9 最后一栏预留**

**更改密码加密算法：**

#authconfig --passalgo=sha256 --update

**密码的安全策略**

**足够长 7**

**使用数字、大写字母、小写字母及特殊字符中至少3种**

**使用随机密码**

**定期更换,不要使用最近曾经使用过的密码**

2.4 group文件格式

**群组名称：就是群组名称**

**群组密码：通常不需要设定，密码是被记录在 /etc/gshadow**

**GID：就是群组的 ID**

**以当前组为附加组的用户列表(分隔符为逗号)**

**A list of supplementary groups which the user is also a member of**

2.5 gshadow文件格式

**群组名称：就是群的名称**

**群组密码：**

**组管理员列表：组管理员的列表，更改组密码和成员**

**以当前组为附加组的用户列表：多个用户间用逗号分隔**

2.6 文件操作

vipw和vigr

pwck和grpck

3 用户和组管理命令

**用户管理命令**

**useradd**

**passwd**

**usermod**

**userdel**

**组帐号维护命令**

**groupadd**

**groupmod**

**groupdel**

3.1 useradd用户创建

**useradd 命令可以创建新的Linux用户**

**格式**

**useradd [options] LOGIN**

**常见选项：**

-u UID 1044

-o 配合-u 选项，不检查UID的唯一性

-g GID 指明用户所属基本组，可为组名，也可以GID

-c "COMMENT“ 用户的注释信息

-d HOME\_DIR 以指定的路径(不存在)为家目录

-s SHELL 指明用户的默认shell程序，可用列表在/etc/shells文件中

-G GROUP1[,GROUP2,...] 为用户指明附加组，组须事先存在

~~-N 不创建私用组做主组，使用users组做主组~~

-r 创建系统用户 CentOS 6之前: ID<500，CentOS 7以后: ID<1000

~~-m 创建家目录，用于系统用户~~

~~-M 不创建家目录，用于非系统用户~~

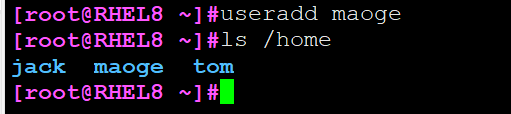
#useradd -u 1066 -c “注释” -d /home/jack -G danny jack

范例:

useradd -r -u 48 -g apache -s /sbin/nologin -d /var/www -c "Apache" apache

useradd 命令默认值设定由/etc/default/useradd定义

创建一个普通用户 maoge



[root@centos8 ~]#cat /etc/default/useradd

# useradd defaults file

GROUP=100

HOME=/home

INACTIVE=-1

EXPIRE=

SHELL=/bin/bash

SKEL=/etc/skel

CREATE\_MAIL\_SPOOL=yes

**创建用户参考的三个东东，就是用户产生的模板信息**

**/etc/default/useradd 文件,用户ID等规定**

**/etc/skel/ 主目录里面的以点开头的东东**

**/etc/login.defs密码相关参数**

/etc/default/useradd 文件内容

|  |  |
| --- | --- |
| /etc/default/useradd 文件内容 | |
| **参数** | **含义** |
| GR0UP=100 | Linux 中默认用户组有两种机制：一种是私有用户组机制，系统会创建一个和用户名相同的用户组作为用户的初始组；另一种是公共用户组机制，系统用 GID 是 100 的用户组作为所有新建用户的初始组。目前我们采用的是私有用户组机制。  如果useradd没有指定组，并且/etc/login.defs中的USERGROUPS\_ENAB为no或者useradd使用了-N选项时，此时该参数生效。创建用户时使用此组ID |
| HOME=/home | 指的是用户主目录的默认位置，所有新建用户的主目录默认都在 /home/下，新建的 用户的主目录就为 /home/用户名/。 |
| INACTIVE=-1 | 指的是密码过期后的宽限天数，也就是 /etc/shadow 文件的第七个字段。这里默认值是 -1，代表所有新建立的用户密码永远不会失效。 |
| EXPIRE= | 表示密码失效时间，也就是 /etc/shadow 文件的第八个字段。默认值是空，代表所有新建用户没有失效时间，永久有效。 |
| SHELL=/bin/bash | 表示所有新建立的用户默认 Shell 都是 /bin/bash。 |
| SKEL=/etc/skel | 在创建一个新用户后，你会发现，该用户主目录并不是空目录，而是有 .bash\_profile、.bashrc 等文件，这些文件都是从 /etc/skel 目录中自动复制过来的。因此，更改 /etc/skel 目录下的内容就可以改变新建用户默认主目录中的配置文件信息。 |
| CREATE\_MAIL\_SPOOL=yes | 指的是给新建用户建立邮箱，默认是创建。也就是说，对于所有的新建用户，系统都会新建一个邮箱，放在 /var/spool/mail/ 目录下，和用户名相同。例如，lamp1 的邮箱位于 /var/spool/mail/lamp1。 |

3.2 命令getent

命令介绍

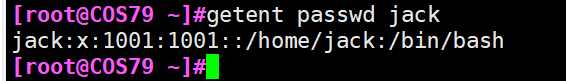
getent 命令用来查看系统的数据库(标准配置文件)中的相关记录。即使这些数据库不是在本地，比如 ldap 或者 nis 中的数据库，也可以使用 getent 查看。

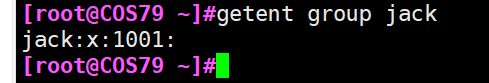
支持的数据库：

ahosts，ahostsv4 ，ahostsv6， aliases ，ethers ，**group**， gshadow， hosts， netgroup， networks， **passwd**， protocols， rpc ，services， shadow

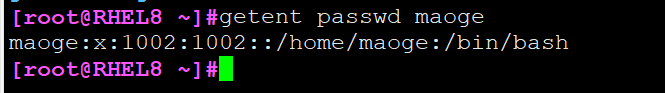
#getent passwd 用户名

例 查看用户jack信息





查看用户maoge信息



**/etc/login.defs 文件**

**/etc/default/useradd 文件**

**上面相关的4个文件写了信息进去**

**创建用户的主目录**

**有关shell的配置文件从/etc/skel 目录copy到用户的主目录**

3.4 passwd设置用户密码

命令格式

# passwd [选项] [用户名]

**<==所有人均可使用改自己的密码，只有root才可以使用选项和带用户名称**

选项：

--stdin ：从标准输入接收用户密码，可以通过自前一个管线的数据，作为密码输入，对 shell script 有帮助！实现减少脚本运行期间减少交互。

如将tom用户的密码修改为123

#echo “123” | passwd --stdin tom

-l ：是 Lock 的意思，会将 /etc/shadow 第二栏最前面加上 ! 使密码失效；

-u ：与 -l 相对，是 Unlock 的意思！

下面几个选项与/etc/shadow文件后面的密码属性字段相关

-S ：列出密码相关参数，亦即 shadow 档案内的大部分信息。

-n ：后面接天数，shadow 的第 4 字段，多久不可修改密码天数

-x ：后面接天数，shadow 的第 5 字段，多久内必须要更动密码

-w ：后面接天数，shadow 的第 6 字段，密码过期前的警告天数

-i ：后面接天数，shadow 的第 7 字段，密码失效天数

说明

1 这个命令是所有用户都可以用来修改自己的密码，root可以修改所有用户的密码

**对于root而言,重设用户密码时一定要记得带上用户名,如果不带用户名,是修改root自己的密码**

2 **只有root可以强行设置简单密码，会提示，但是也就是提示而已**

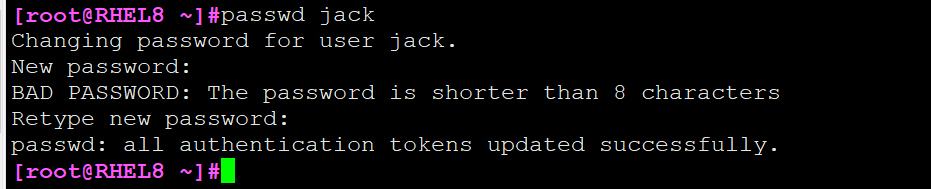
3 对于普通用户而言，密码是有设置复杂程度要求，像123或者**123%abcd**这种密码是无法通过的， 三种组合，8位长度，尽量不要连续数字和字母

**密码例子：**

**168@hkmp这个密码是可以的**

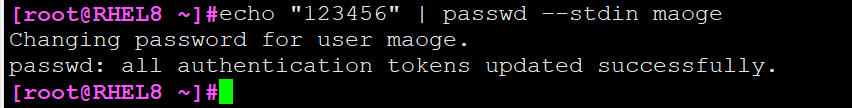
范例

创建jack用户密码为abcdef



下面这种方法创建的密码在命令历史记录中是明文的，一般在批量创建用户时使用或者初始账户要求时使用，后面一般都会设置用户下次登录时必须更改密码

创建maoge用户密码为123456



3.5 usermod修改用户

功能：命令可以修改用户属性

格式：

usermod [OPTION] login

常见选项：

-u UID: 新UID

-g GID: 新主组

-G GROUP1[,GROUP2,...[,GROUPN]]]：新附加组，原来的附加组将会被覆盖；若保留原有，则要同时使用-a选项

-a, --append append the user to the supplemental GROUPS mentioned by the -G option without removing he user from other groups

#usermod –aG 组列表 用户名

-s SHELL：新的默认SHELL

-c 'COMMENT'：新的注释信息

-d HOME: 新家目录不会自动创建；若要创建新家目录并移动原家数据，同时使用-m选项

-l login\_name: 新的名字

-L: lock指定用户,在/etc/shadow 密码栏的增加 !

-U: unlock指定用户,将 /etc/shadow 密码栏的 ! 拿掉

-e YYYY-MM-DD: 指明用户账号过期日期

-f INACTIVE: 设定非活动期限

范例:

1 修改rose用户的主目录

#usermod –d /tmp/rose rose

2 将用户加入到组，将jack用户加入到maoge组

#usermod -aG maoge jack

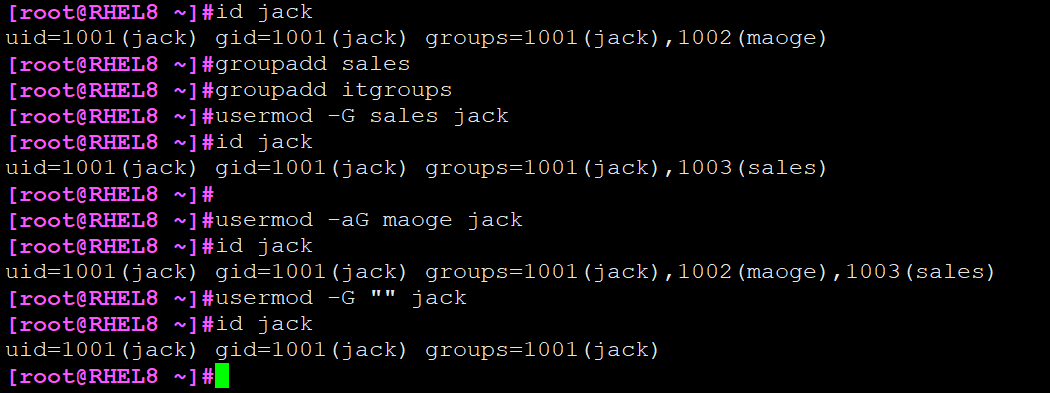
-a表示append的意思

3 将用户的附加组清掉

usermod -G “” jack

4将用户jack的主要组修改为sales

#usermod -g sales jack



PS：主要组是不能清除的，这个组是必须存在的哈

3.6 chage命令

chage 可以修改用户密码策略

#chage [选项] 用户名

常见选项：

-d LAST\_DAY ，实质改第3栏，如果把这一栏改成0，就意味着用户下次登录必须修改密码

-m --mindays MIN\_DAYS

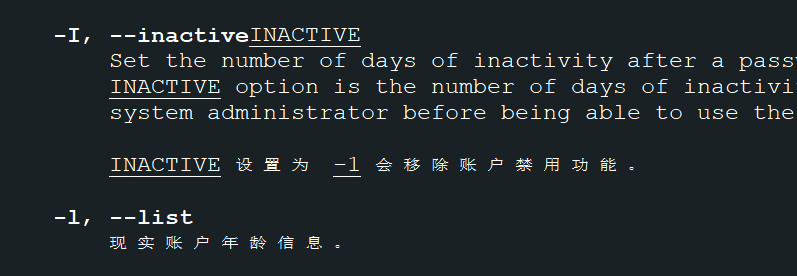
-M --maxdays MAX\_DAYS

-W --warndays WARN\_DAYS

-I --inactive INACTIVE 密码过期后的宽限期 (大写字母i)

-E --expiredate EXPIRE\_DATE 用户的有效期

-l 显示密码策略



chage 有一个功能很不错喔！如果你想要让『使用者在下一次（批量创建用户时的第一次登录）登入时， 强制他们一定要更改密码后才能够使用系统资源』，可以利用的方法来处理的！

范例

创建用户tom

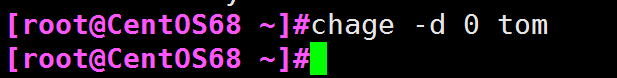
#useradd tom

设置密码123

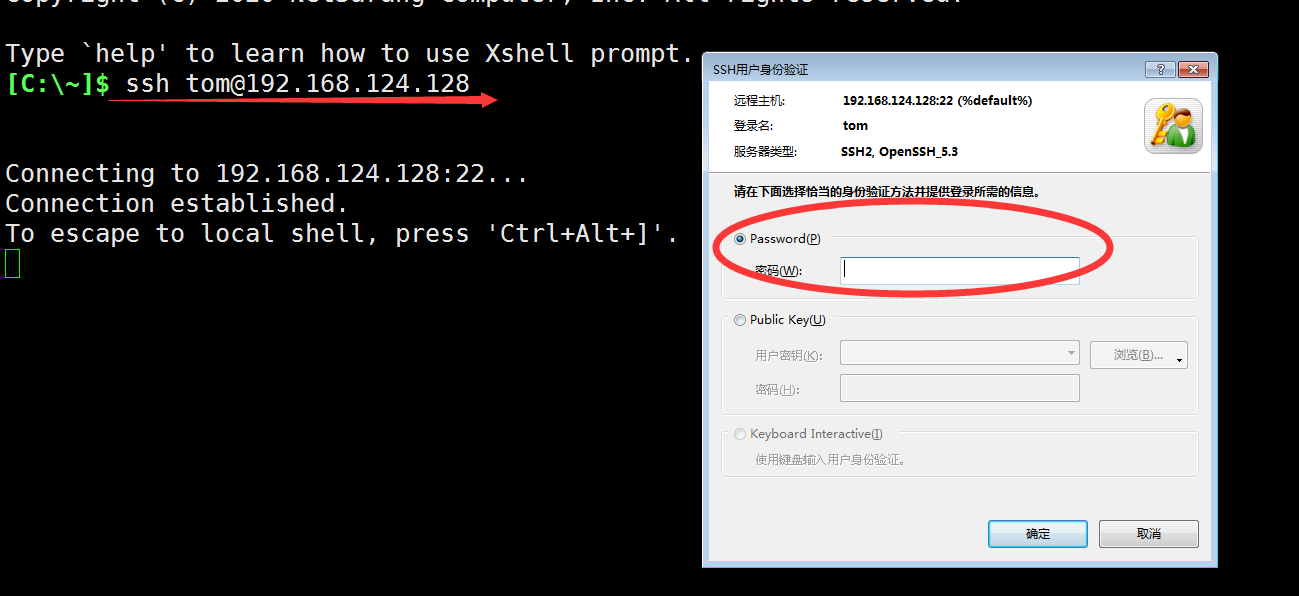
#echo "123" | passwd --stdin tom

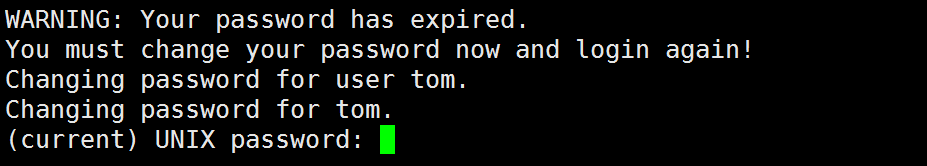
将第3栏改成0

#chage –d 0 tom



测试



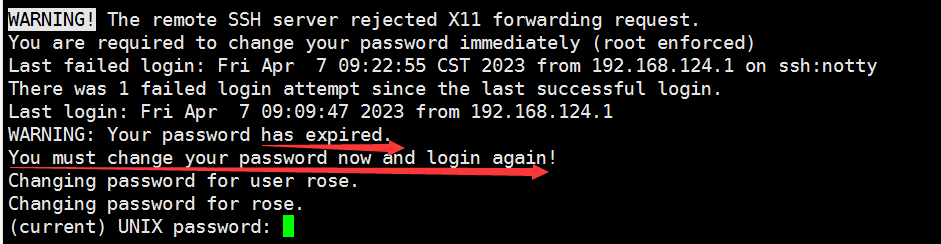


#下一次登录强制重设密码

[root@centos8 ~]#chage -d 0 rose

#chage –d 0 用户名 强制用户下次登录时修改密码,这个实际是修改第3位

这种一般用在用户初步创建下发密码



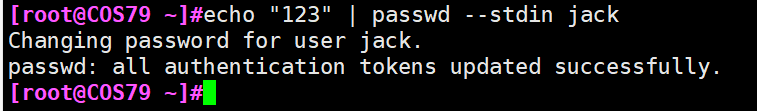
**passwd --stdin的使用**

说明：只能root才能使用

这种用法主要是用脚本在大量创建用户时使用

这种主要是明文的，所以后面一般都要加上让用户下次登录强行让用户修改密码，这个密码对于root去设置用户密码时没有复杂程度要求

下面就是利用--stdin选项将jack用户密码修改成123



3.7 userdel 删除用户

userdel [选项] 用户名

选项

-f --force 此选项强制删除用户账户，甚至用户仍然在登录状态。它也强制 userdel 删除用户的主目录和邮箱，即使其它用户也使用同一个主目录或邮箱不属于指定的用户。如果 /etc/login.defs中的 USERGROUPS\_ENAB 定义为yes，并且如果有一个和用户同名的组，也会删除此组，即使它仍然是别的用户的主组。

注意：此选项危险，可能会破坏系统的稳定性。

-r, --remove 用户主目录中的文件将随用户主目录和用户邮箱一起删除。在其它文件系统中的文件必须手动搜索并删除。 邮箱在 login.defs 文件中的 MAIL\_DIR 变量中定义。

这两个选项慎用,会删除用户的数据

#useradd passwd usermod chage userdel

**一般建议锁掉账号**

4 组的管理命令

1 groupadd命令

1 创建组

组名字最长不可以超过32个字符

**范例一：新建一个群组，名称为 group1**

[root@study ~]# groupadd group1

[root@study ~]# grep group1 /etc/group /etc/gshadow

/etc/group:group1:x:1503:

/etc/gshadow:group1:!::

# 群组的 GID 也是会由 500 以上最大 GID+1 来决定！

### 2 gpasswd组成员管理

[root@study ~]# gpasswd [-A user1,...] [-M user3,...] groupname

[root@study ~]# gpasswd [-rR] groupname

选项与参数：

**-M ：将某些账号加入这个组当中！这个是重新设置组成员**

a=123

a=a&”123” >>

[someone@study ~]$ gpasswd [-ad] user groupname

选项与参数：

-a ：将**某位**使用者加入到 groupname 这个群组当中！

**-d ：将某位使用者移除出 groupname 这个群组当中。**

范例

将stu,tom,jack,rose 4个用户设置为sales组的成员,相当于给这个组成员重新赋值

[root@CentOS68 ~]#gpasswd -M tom,jack,rose,stu sales

组成员查看

[root@CentOS68 ~]#groupmems -g sales -l

向一个组里面添加新成员进来，追加的意思

[root@CentOS68 ~]#gpasswd -a user1 sales

Adding user user1 to group sales

从一个组里面删除组成员

#gpasswd -d user1 sales

修改组成员最直接还有一种办法就是直接修改/etc/group第四位上面

3 groups

**groups 查看用户属于哪几个组以及初始组是哪一个,第一个就是初始组**

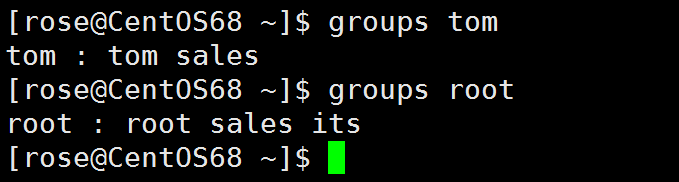
**格式**

**NAME**

**groups - print the groups a user is in**

**SYNOPSIS**

**groups [OPTION]... [USERNAME]...**



4 newgrp

**功能：临时更改初始组(primary group)**

**newgrp - log in to a new group**

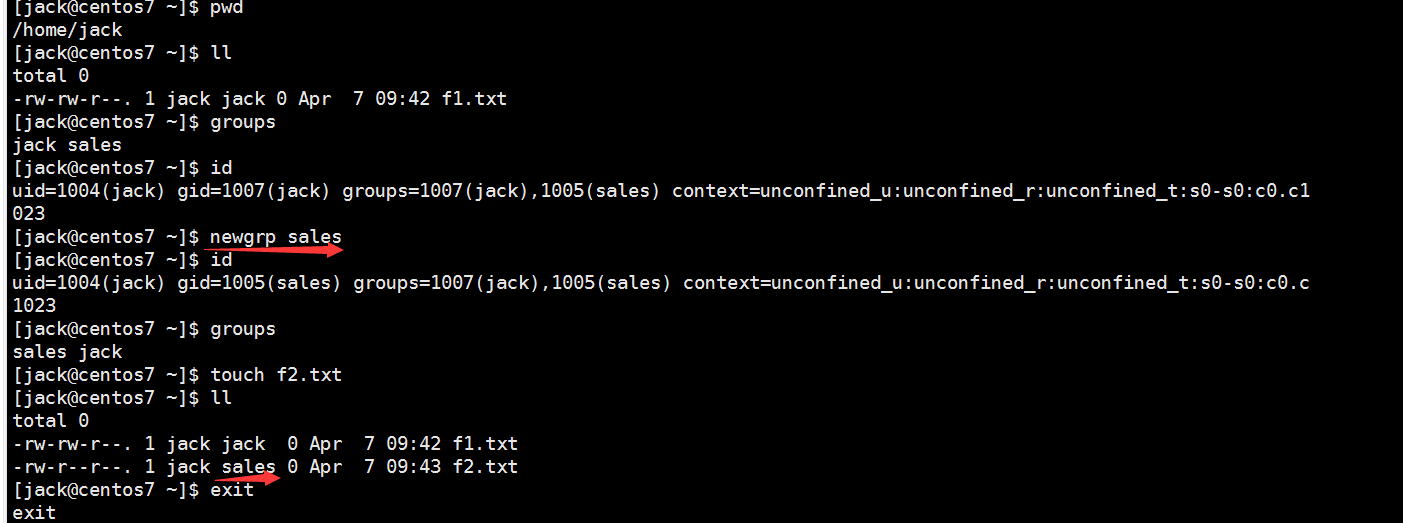
**The newgrp command is used to change the current group ID during a login session**

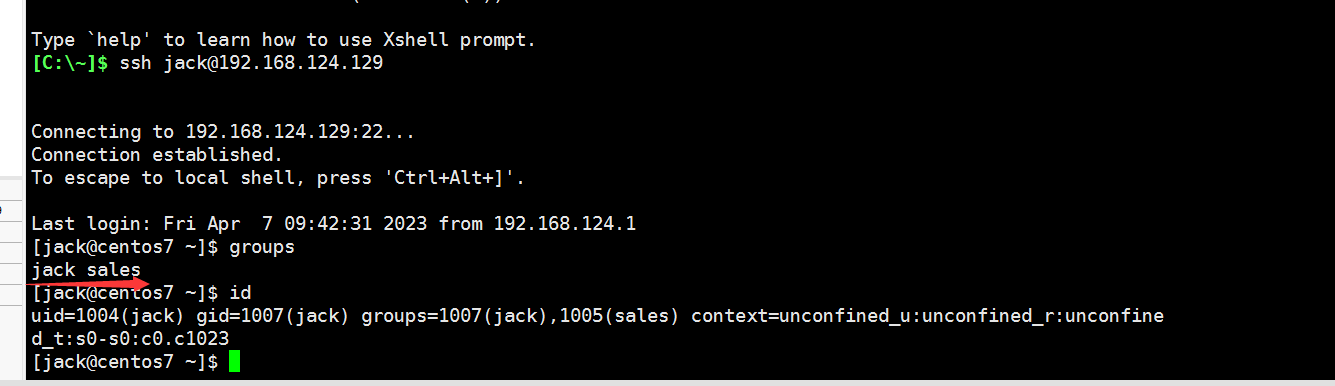
**SYNOPSIS**

**newgrp [-] [group]**

我们知道，每个用户可以属于一个初始组（用户是这个组的初始用户），也可以属于多个附加组（用户是这个组的附加用户）。既然用户可以属于这么多用户组，那么用户在创建文件后，默认生效的组身份是哪个呢？  
当然是初始用户组的组身份生效，因为初始组是用户一旦登陆就获得的组身份。也就是说，用户的有效组默认是初始组，因此所创建文件的属组是用户的初始组。那么，既然用户属于多个用户组，能不能改变用户的初始组呢？使用命令 newgrp 就可以。

**修改后用户注销就会失效，如果要永久修改需要usermod命令修改**





5 groupmems

管理用户主要组的成员

groupmems [options] [action]

常见选项：

-g, --group groupname 更改为指定组 (只有root)

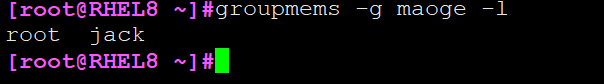
actions:

-a, --add username 指定用户加入组

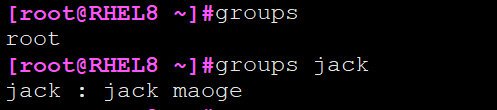
-d, --delete username 从组中删除用户

-p, --purge 从组中清除所有成员

-l, --list 显示组成员列表



groups 可查看用户组关系



6 几个辅助命令

id finger w whoami last

5 切换用户或以其他用户身份执行命令

**su: 即switch user,命令可以切换用户身份，并且以指定用户的身份执行命令**

普通用户登录系统，要做管理员工作时候再切换到root身份上去操作，操作完了之后退出

**切换用户的方式：**

su UserName：**非登录式切换**，即不会读取目标用户的配置文件，不改变当前工作目录

su - UserName：**登录式切换**，会读取目标用户的配置文件，切换至家目录，完全切换

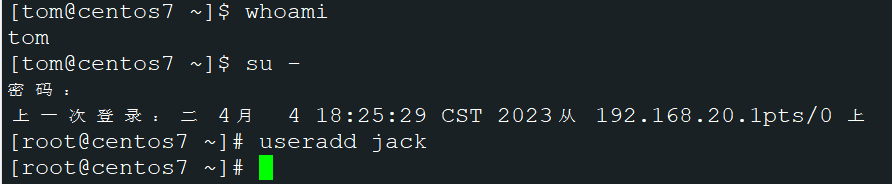
说明：root su至其他用户无须密码；非root用户切换时需要密码

#su -

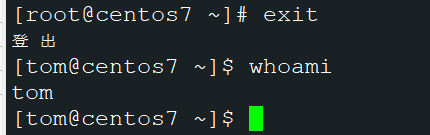
#su 后面没有用户名,则表示切换到root

范例

以普通用户登录系统后需要以root身份去执行命令时



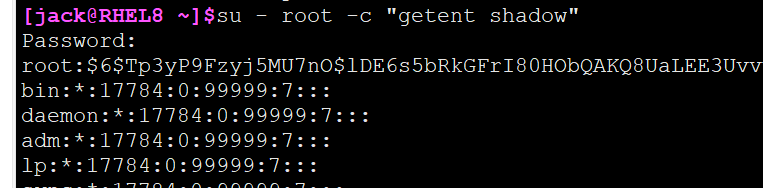
退出root用exit



换个身份执行命令：

su [-] UserName -c 'COMMAND'

范例：



6 sudo用户授权管理

**root用户密码忘记的情况下,是可以通过单用户模式进系统直接重设密码**

/etc/sudoers

# visudo命令是编辑上面这个文件，它可以检查是否写正确

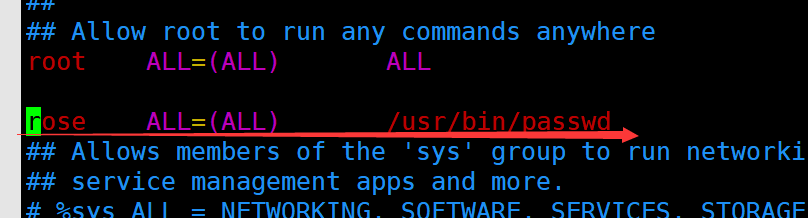
编辑文件里面的东东，有固定的格式

1 授权

在**/etc/sudoers**文件中找到如下内容行,大约在100行附近

使用者帳號 可下達指令的主機名稱=(可切換的身份) 可下達的指令

root ALL=(ALL) ALL <==這是預設值



上面这一行的四个部份的意义是：

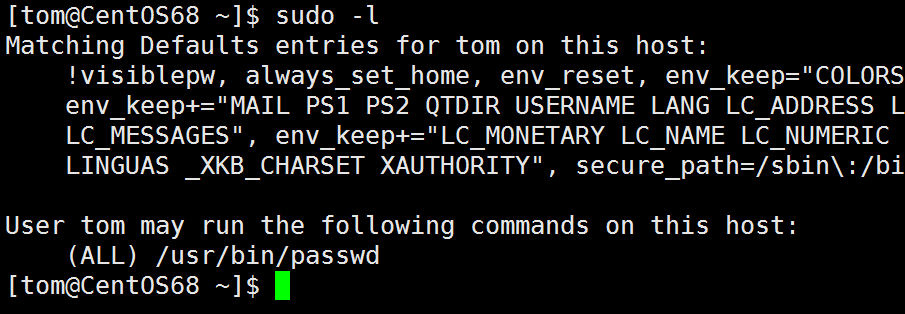
『使用者账号』：系统的哪个账号可以使用 sudo 这个指令的意思；

『可下达指令的主机名称』：这个账号可以透过 sudo 对某些主机下达指令的意思。基本上，如果针对本机的话，这个项目可以都填写为 ALL 即可。 如果你想让此用户透过 sudo 对其他网络上的特定主机下达指令时，就得要在这里规范才行。

『(可切换的身份)』：这个账号可以切换成什么身份来下达后续的指令，预设 root 可以切换成任何人；

『可下达的指令』：可用该身份下达什么指令？这个指令请务必使用绝对路径撰写。 预设 root 可以切换任何身份且进行任何指令之意。

#sudo -l 查看获得的授权



2 如何执行被授权的指令

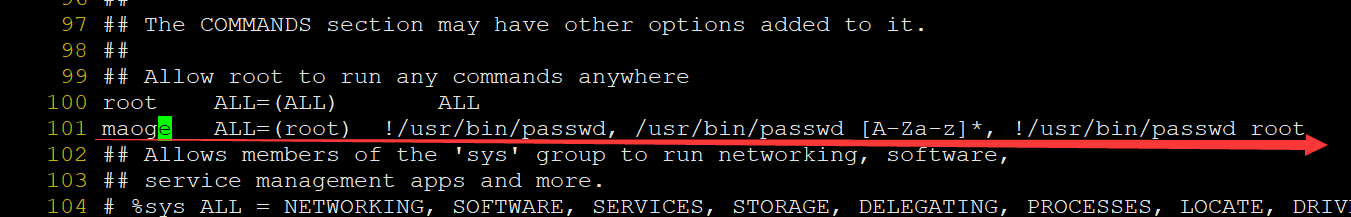
范例：授权给一个用户单独修改用户密码

在文件中添加如下行，

用户名 **ALL=(root) !/usr/bin/passwd, /usr/bin/passwd [A-Za-z]\*, !/usr/bin/passwd root**

rose **ALL=(root) !/usr/bin/passwd, /usr/bin/passwd [A-Za-z]\*, !/usr/bin/passwd root**

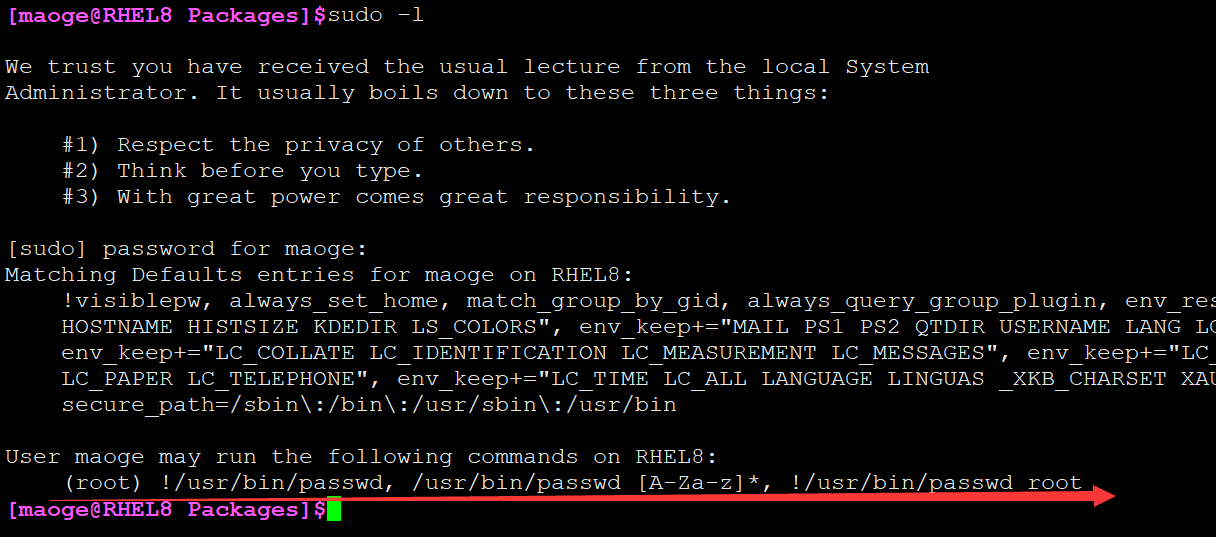
提示：命令要使用绝对路径



在设定值中加上惊叹号『 ! 』代表『不可执行』的意思。因此上面这一行会变成：可以执行『 passwd 任意字符』，但是『 passwd 』与『 passwd root 』这两个指令例外！ 如此一来 myuser1 就无法改变 root 的密码了！这样这位使用者可以具有 root 的能力帮助你修改其他用户的密码， 而且也不能随意改变 root 的密码！很有用处的！

查看自己获得的sudo命令

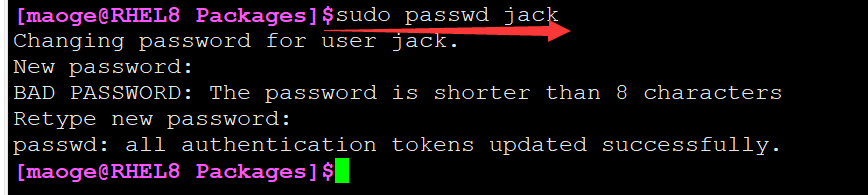
#sudo -l

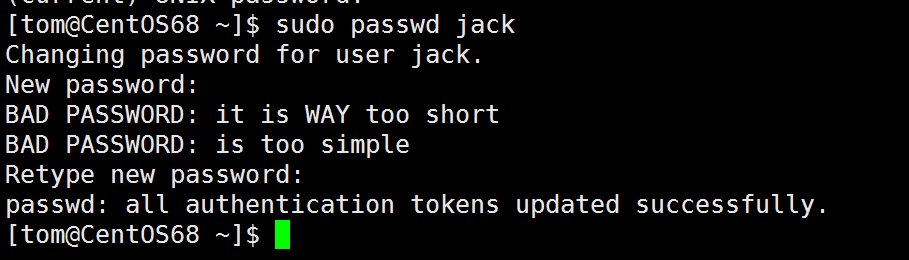


3 执行sudo获得的命令

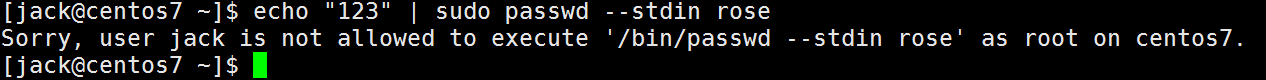
语法：#sudo 指令

将jack用户密码修改





下面这个--stdin还是不能使用的，注意



3 Linux 中 root 与 sudo 的用法与区别

Linux 下面有两个概念可能大家接触的比较多，一个是 sudo 命令，还有一个是 root 账户。Sudo 命令可以以最高权限执行命令，而 root 账户下所有命令都有最高权限，也就是相当于所有命令都默认加了 sudo。

那么 sudo 和 root 的区别到底是什么呢，为什么我们建议使用 sudo 而不是直接使用 root 账户？在本教程中，您将了解 root 访问、sudo 命令、如何使用 sudo 运行命令以及 sudo 访问和 root 之间的区别。

什么是 root？

root 是指 Linux 等类 Unix 系统中的超级用户帐户。它是用于系统管理的系统上具有最高访问权限的特权帐户。此根/超级用户帐户的用户标识符 (UID) 为零，无论帐户名称如何。

root 用户拥有整个系统的完全权限（root 特权）。它可以做诸如修改系统的核心部分、升级系统、更改系统配置以及启动、停止和重新启动所有正在运行的系统服务之类的事情。

以 root 身份登录（使用 su -）时，终端命令提示符符号从

**在某些系统（如 Ubuntu）上，root 用户默认被锁定**。（备注：搬瓦工并不会锁定，但是腾讯云默认会锁定）。

什么是 Sudo？

sudo (superuser do) 命令是一个命令行实用程序，它允许用户以 root 或其他用户身份执行命令。它提供了一种有效的方式来授予某些用户适当的权限以使用特定的系统命令或以 root 用户身份运行脚本。

虽然有点类似于 su 命令，但 [sudo](http://mp.weixin.qq.com/s?__biz=MzI0MDQ4MTM5NQ==&mid=2247512892&idx=2&sn=9c86c0c7b3def46250ce2805cfc359b6&chksm=e918d620de6f5f360912b33c1384e4c516060308d31c61cae98d775d181c4432ab7fe499b57e&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 的不同之处在于它默认需要用户的密码进行身份验证，而不是 su 需要的目标用户的密码。Sudo 也不会产生 root shell；相反，它以提升的权限运行程序或命令，不像 su，它产生一个 root shell。

使用 sudo，系统管理员可以执行以下操作：

* 授予用户或用户组以提升或 root 权限运行某些命令的能力。
* 查看每个使用 sudo 的用户的用户 ID 的日志。
* 控制用户可以在主机系统上使用什么命令。

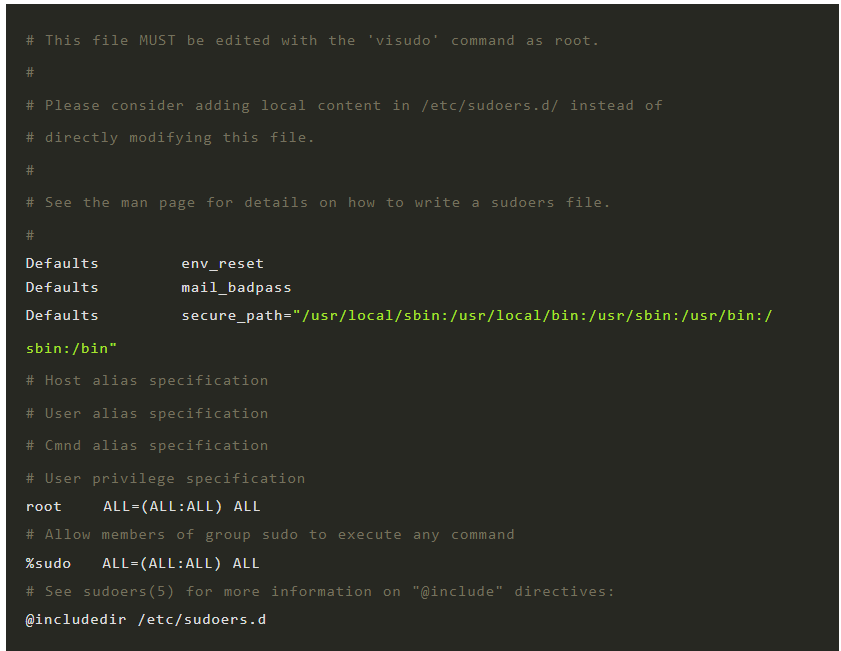
Sudo 会记录在 /var/log/auth.log 文件中执行的所有命令和参数的日志，可以在出现故障时进行分析。

sudoers 文件

sudo 使用默认的 sudoers 安全策略，并保留一个特殊的配置文件 /etc/sudoers。该文件可用于控制访问权限和密码提示超时。

注意：您必须具有提升的权限才能查看 sudoers 文件

打开 /etc/sudoers 文件；它应该是这样的：



其中这行：

root         ALL=(ALL:ALL)ALL

意味着 root 用户拥有无限的权限并且能够在系统上运行任何命令。

%sudo ALL=(ALL:ALL)ALL

允许组 sudo 的所有成员执行任何命令。

注意：sudoers 文件中的 ‘%’ 代表一个组，而不是注释。

从 /etc/sudoers 文件的第一行可以看出：

# This file MUST be edited with the 'visudo' command as root

不要尝试直接编辑 sudoers 文件。使用具有 root 权限的 visudo 命令。

使用 sudo 运行命令很简单，只需要在命令前面加上 sudo 即可：

$ sudo command

一般来说会提示输入密码，输入密码后回车即可。

$ sudo command  
[sudo]  password for user:

Sudo 对比 Root

最小权限原则是一种信息和计算机安全概念，它认为授予程序和用户执行任务所需的最少或最低限度的权限。

以 root 用户登录后，输入到终端的每一条命令都以系统最高权限运行，违反了最小权限原则。像 rm 这样的简单命令可用于删除核心根目录或文件，而不会在意外时提示用户。例如，如果您尝试使用以下命令删除 /etc 之类的根目录：

$ rm -rf /etc

当您以普通用户身份登录时，您将被拒绝许可。当以 root 身份登录时，不会显示任何提示，并且整个文件夹将被删除 – 这很可能会破坏您的系统，因为运行系统所需的特殊配置文件存储在 /etc 目录中。您也可能最终错误地格式化磁盘，并且系统不会提示您。

此缺陷还扩展到以 root 身份运行代码或应用程序；应用程序中的一个小错误可能会删除一些系统文件，因为该应用程序是在最高权限下运行的。

Sudo 提供细粒度的访问控制。它仅向需要它的特定程序授予提升的权限。您知道哪个程序以提升的权限运行，而不是使用 root shell（以 root 权限运行每个命令）。

Sudo 也可以配置为以另一个用户身份运行命令，指定允许哪些用户和组使用 sudo 运行命令，或者通过编辑 sudoers 文件设置以 root 权限运行程序的超时。

因此，**不建议使用 root shell 运行命令**，因为您破坏系统的机会要高得多。如果您需要更高权限或 root 权限来运行命令，请**使用 sudo 确保只有该命令以 root 权限运行。**

7 实验安排

1. 创建用户，tom,jack,rose，stu几个用户，均设置密码为123

useradd tom

passwd tom

#echo “123” | passwd --stdin tom

弱密码

**打开一个终端，用上面的tom用户登录测试**

1. 创建组：组名字sales

groupadd sales

1. 利用gpasswd将上面三个用户加入到这个组里面去

#gpasswd –M tom,rose,jack sales

#gpasswd -a stu sales

# groupmems -g sales -l

1. 利用

su - tom将用户身份切换一下，

touch一个tom.txt文件，

利用groups观察一下

再用newgrp修改初始组为sales,

再touch一个文件file1.txt

再观察文件所属组

利用ls –l 查看文件所属组是哪个？然后回到root用户

6．将rose用户设置为下次登录时必须更改密码，用rose用户登录检查效果。

chage –d 0 rose

7 利用 --stdin为jack设置密码 abc%1234

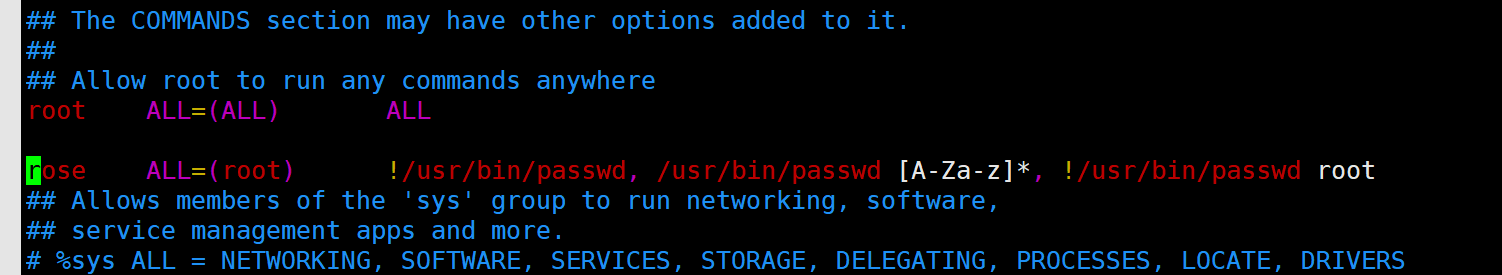
#echo “abc%1234” | passwd --stdin jack

8 将tom用户授权可以修改系统中除了root以外的用户的密码

tom ALL=(root) !/usr/bin/passwd, /usr/bin/passwd [A-Za-z]\*, !/usr/bin/passwd root

参考下面这个文件格式

#vim /etc/sudoers



改好之后用 :w!保存

然后再打开个终端利用tom登录，将jack密码改成123456

$ sudo -l

$sudo passwd jack

## 附root密码忘记的解决办法

### Linux 6 忘记密码解决方法

Linux系统忘记root密码的情况该怎么办呢？重新安装系统吗？当然不用！进入单用户模式更改一下root密码即可。

步骤如下：

这个操作是需要我们在机器前，且重启服务器，在服务器启动界面出现3秒倒计时内按任意一个键，进入如图1所示。

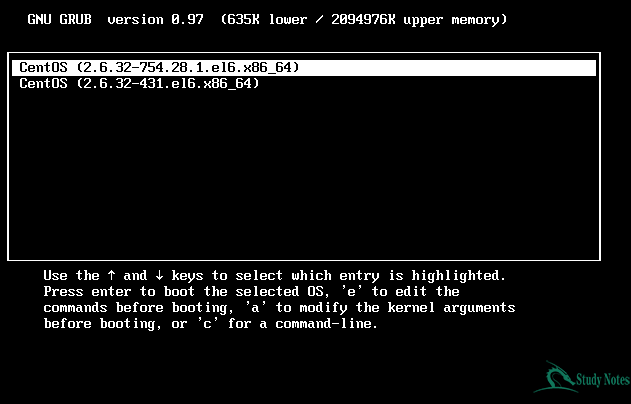


图1

然后输入“e”进入下一步，如图2所示。

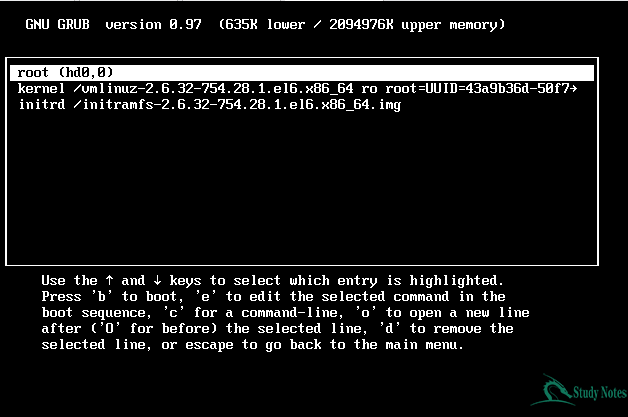


图2

然后我们选择第二条信息，Kernel开始的这行，然后再按一下“e”字母键进入如图3所示。

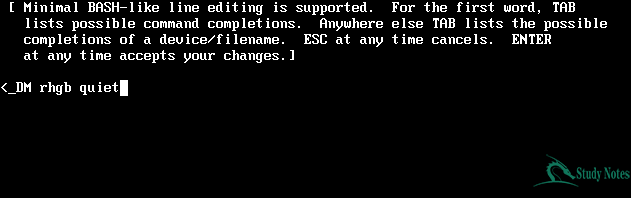


    图3

在最后一行信息中可以看到我们光标是在最后的，这也是默认的光标位置，这条信息其实是很长，可以左右移动光标查看，但我们这次的操作是需要在这行信息的末尾处输入数字1的内容即可，如图4所示。

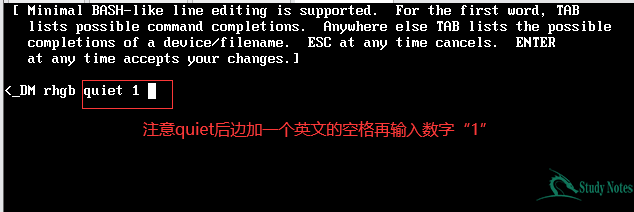


图4

然后再按一下回车键返回如图5所示。

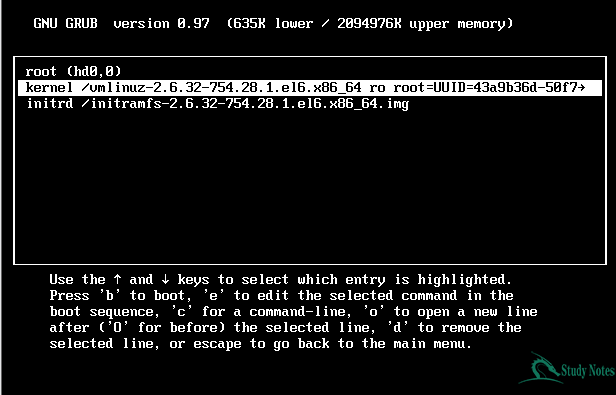


图5

然后再按一下“b”键启动系统，系统启动成功进入单用户模式如图6所示。

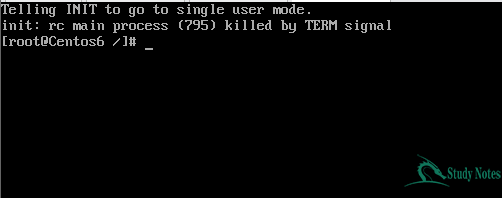


图6

这时我们就可以像正常进入系统那样使用passwd命令来修改root用户的密码，如图7所示。

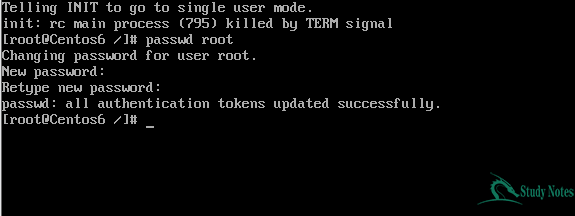


图7

成功修改密码后输入reboot命令重启系统即可使用新设置的密码登录root用户。忘记其他用户密码只需在passwd命令指定用户即可。

### Linux 7 忘记密码解决方法

其实在centos7的版本不能再叫单用户模式了，关于这个单用户在centos6以及之前的版本是有这种运行级别的概念。它有0至6一共有7个级别，0表示关机，1表示单用户，2至4都是命令行下的多用户模式。5表示图形操作界面，6表示重启，这些都是6以及之前版本的概念，7的话就不存在这些概念了。为什么我会把7叫但用户模式呢？先运行一条命令：ls -l /usr/lib/systemd/system/runlevel\*target 如图8所示。

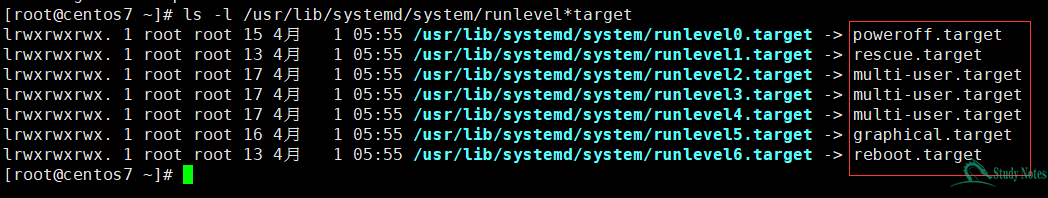


图8

敲了这个命令后呢可以看到黄色方框的是6以及之前版本的运行级别从0至6分别对应红色方框7版本的相关。而rescue,target就是现在要使用的模式，而它对应的则是6以及一下版本的单用户运行级别，所以我叫了但用户模式，但它与单用户模式还是有其别的。

要修改密码同样也是得需要重启然后在服务器启动界面出现3秒倒计时内如图9所示界面后按一下“e”键。

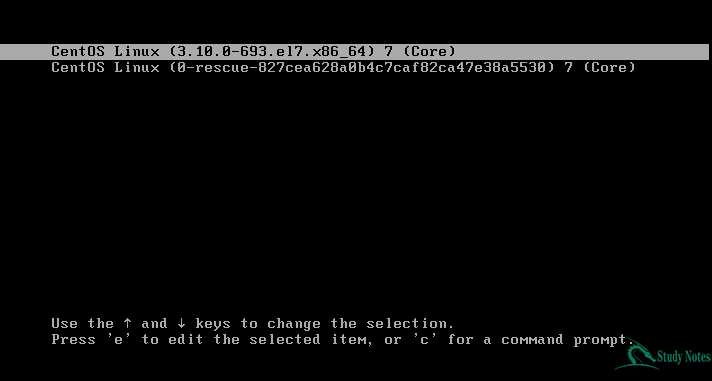


图9

按下“e”键后进入如图10界面。

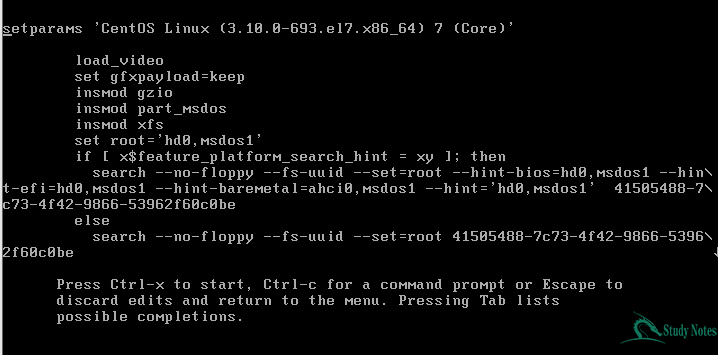


图10

可以看到很多信息，使用上下左右可以移动查看信息，这里直接使用向下移动光标至linux16 /vmlinuz-3.10\*\*\*\*\*这一行，如图11所示。

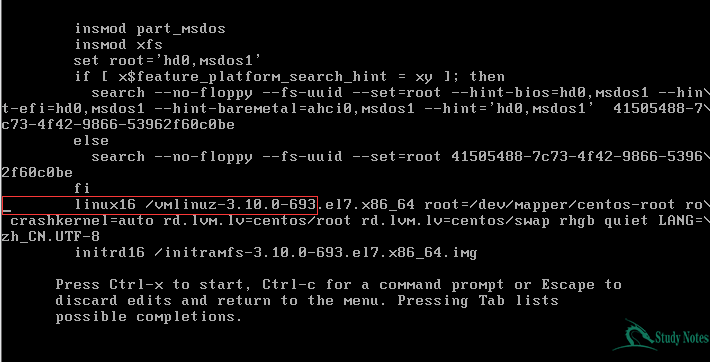


图11

然后向右移动光标到ro\这个位置后边，插入：rw init=/sysroot/bin/sh\_  如图12所示。



图12

切记sh\_的下划线后边不要有空格，不然会报错，修改完成直接按Ctrl+X启动系统，稍微等一段时间启动成功后如图13所示。

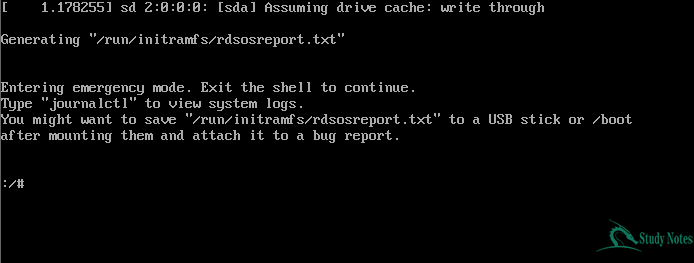


图13

这是我们进入的也相当于windows 的PE那样，我们需要切换到本机的本身系统中去，输入 chroot  /sysroot切换进去如图14所示。

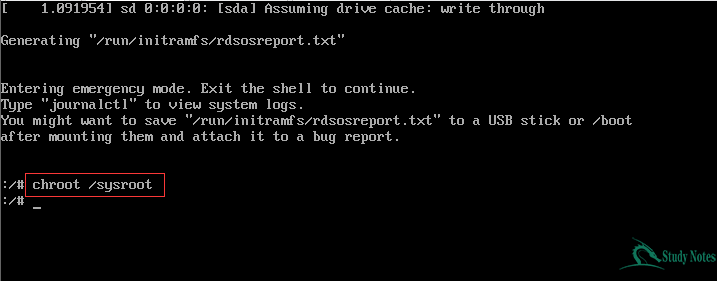


图14

这时候我们可以直接像在进入系统那样是有passwd命令来修改root用户密码，如图15所示。

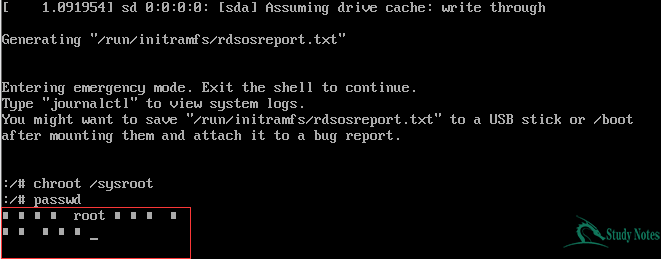


图15

这是会出现乱码，这是因为系统安装时选择的是中文而在linux中这个模式下不支持中文，其实这乱码就是一些提示，可以忽略直接输入密码，如果不习惯可以先切换一下语言，在操作修改密码，Ctrl+C结束修改密码命令，然后输入LANG=en,再修改root用户密码，如图16所示。



图16

修改完成后记得输入最后一条命令，如果忘记输入密码则修改不成功，输入：touch /.autorelabel 如图17所示。

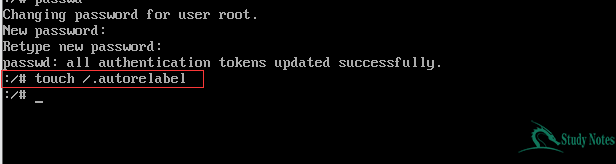


图17

创建.autorelabel文件后，这里的reboot是没效果的，首先输入“exit”命令退出当前状态再输入“reboot”即可重启，如图18所示。



图18

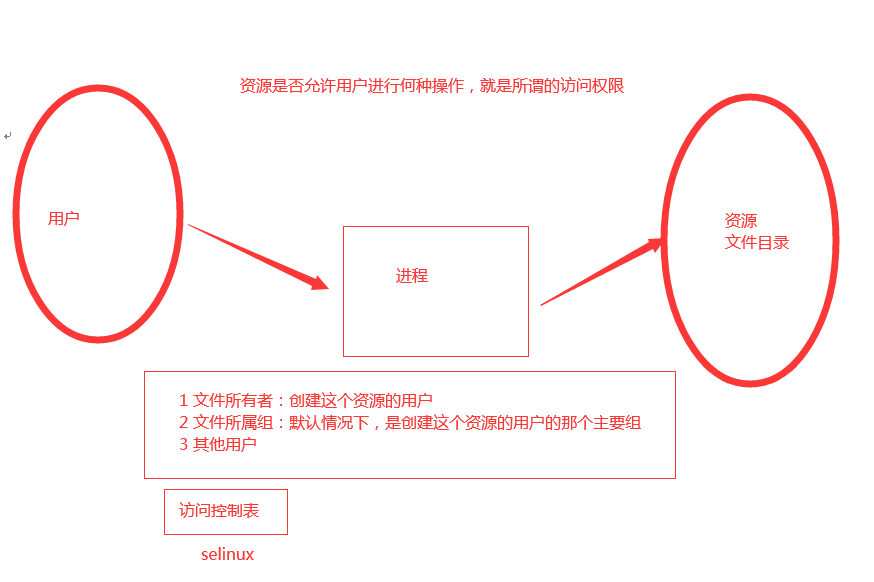
重启成功后我们就可与使用刚修改的密码登录系统了，忘记其他用户密码需要在“passwd”命令指定相关用户即可。

第9讲 文件目录权限管理

上次内容回顾用户管理

root用户可以访问/etc/inittab cat nano vim

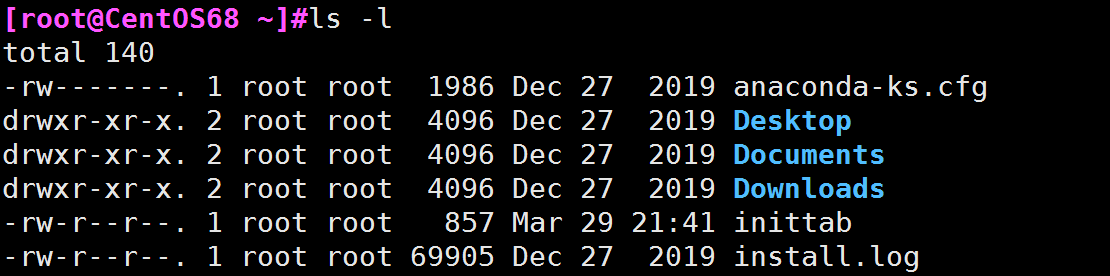
root一定使用vim编辑器才可以访问/etc/inittab

selinux

1 文件目录权限查看

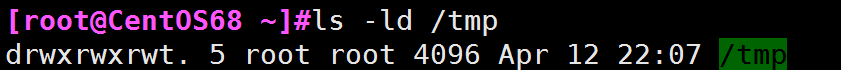
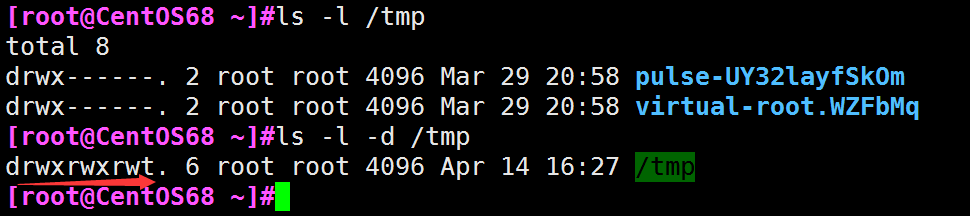
1 ls 查看文件权限

#ls -l 查看目录中的对象权限使用



2 查看目录权限

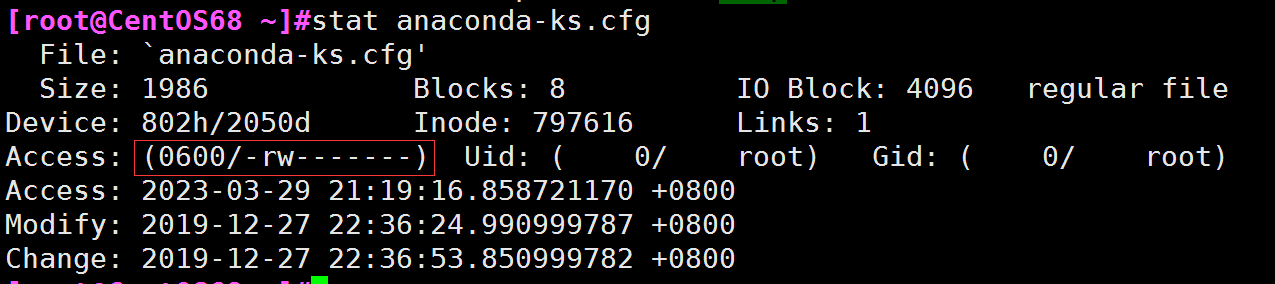
#ls -ld **一定要带这个选项d**



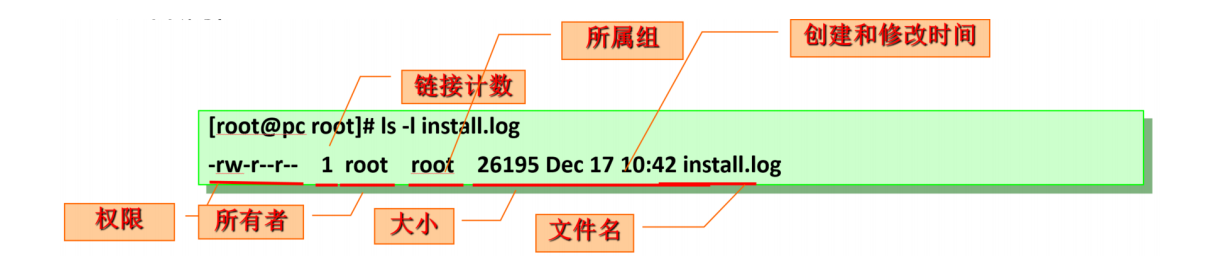
### 2 stat命令的使用

功能： 查看文件或者目录权限

**#stat主要用于查看目录本身权限和数字权限比较方便**



权限位说明



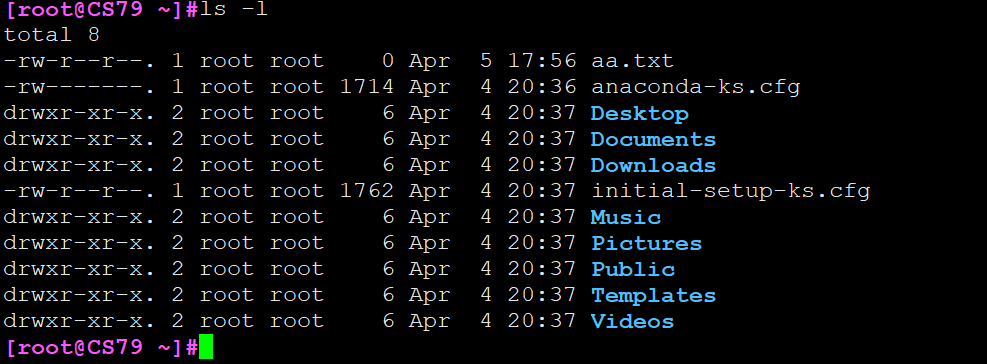
权限位共9位

用户类型3类

所有者 所有组 其他用户

一类用户权限占3位

**有权限就有字符表示**，**没有这个对应权限这个位置不是空，而是 –**



2 改变文件所有者组 chown

chown 命令可以修改文件的属主，也可以修改文件属组

格式

chown [OPTION]... [OWNER][:[GROUP]] FILE...

chown [OPTION]... --reference=RFILE FILE...

用法说明：

OWNER 只修改所有者

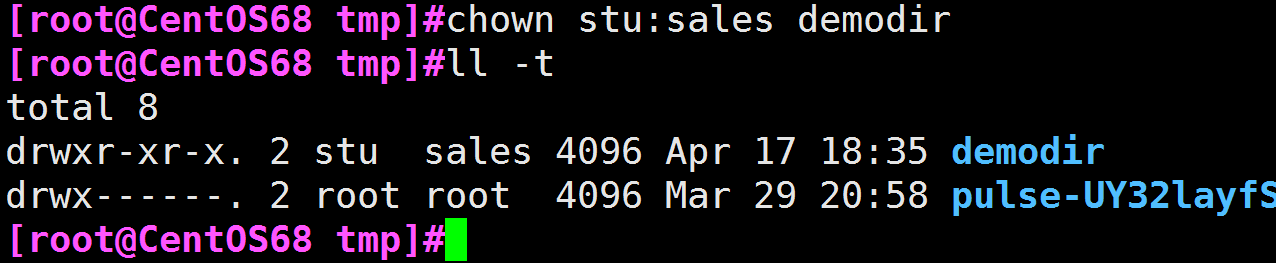
OWNER:GROUP 同时修改所有者和属组

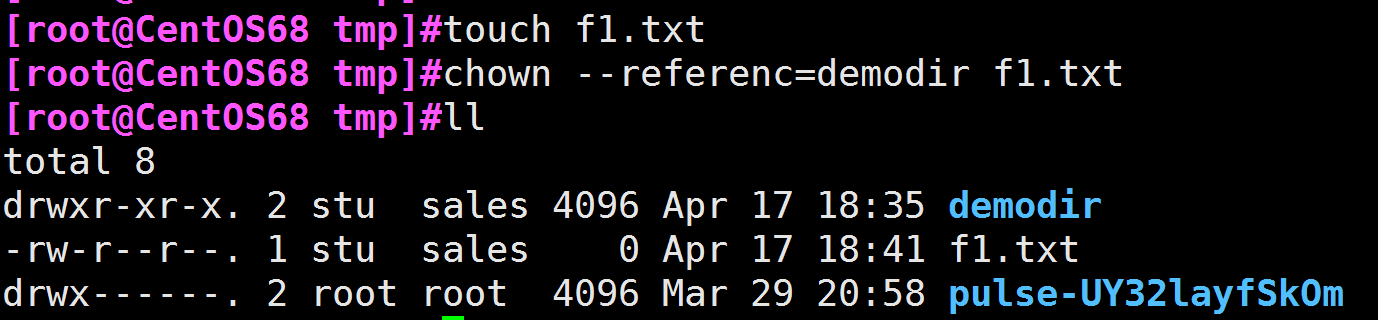
:GROUP 只修改属组，冒号也可用 . 替换

-R: 递归

--reference=RFILE 参考指定的的属性，来修改

范例:





**设置文件的属组信息chgrp ，但这个**命令可以只修改文件的属组

3 权限的含义

**文件的权限主要针对三类用户进行定义**

owner 属主, u

group 属组, g

other 其他, o

上面三个合在一起 a

**每个文件针对每类访问者都定义了三种基本权限**

r Readable

w Writable

x eXcutable

**对文件的权限：**

r 可使用文件查看类工具获取其内容

w 可修改其内容

x 可以把此文件提请内核启动为一个进程

**对目录的权限：**

r 可以使用ls查看此目录中文件列表

w **可在此目录中创建文件，也可删除此目录中的文件**

x 可以使用ls -l查看此目录中文件元数据（须配合r），可以cd进入此目录

**文件能否被删除的权限是在文件的父目录上来控制的**

**上述权限对于root而言，是无效的**

4 chmod文件权限设置

1 字符设置

**通过权限字母和操作符表达式的方法来修改或者设置权限**

用户类型字符

u 所有者

g 组

o 其它

a 所有人

权限

r w x

运算方法

用户类型+权限

用户类型-权限

重新设定 用户类型=新权限

**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**

#chmod a+x demo.sh

其中，“用户类型”可用以下字母中的任一个或它们的组合来表示需要设置权限的部分。

u：表示对文件所有者设置权限。

g：表示对文件所有者相同组的所有用户设置权限。

o：表示对其他用户设置权限。

a：表示对所有用户设置权限。

紧跟在用户类型后面的是操作符，这 3 个符号的意义如下：

+：添加某个权限。

-：取消某个权限。

=：赋予给定权限并取消其他所有权限。

而“权限字符”可使用r、w、x的组合。

#chmod u+x aa.txt

#chmod o+x account.sh

#chmod o-x account.sh

用户类型可以连写，中间不需要任何分隔符号

#chmod ug+x,o=r account.sh

#chmod a+x account.sh

重新设置权限

#chmod u=rwx,g=rw,o=r account.sh

**范例**

创建一个脚本文件**sh1.sh**,输出hello world,给这个可执行权限,脚本内容如下

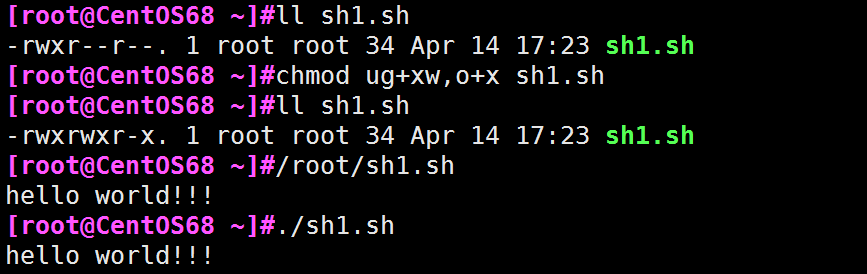
#vim sh1.sh

文件内容如下两行



创建完成后，对所有用户增加对sh1.sh可执行权限

#chmod a+x sh1.sh



脚本执行需要给可执行权限

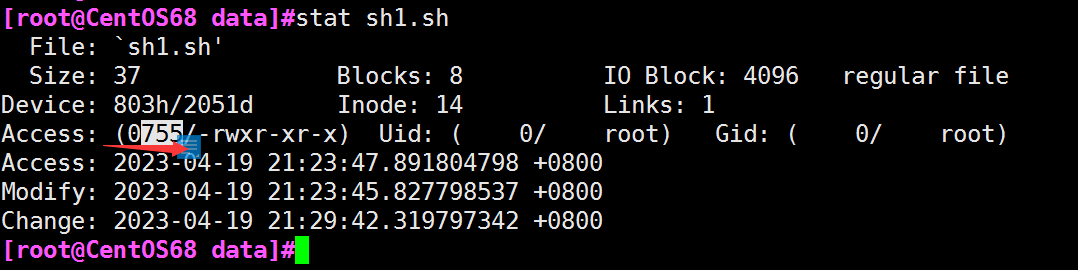
以绝对路径即fullname或者./脚本名来执行

2数字方法重新设置权限

**原理：**

**权限三类用户共占9栏，每三栏代表一类用户，对于每类用户而言，有权限的位置用二进制数1表示，没有则用0表示，然后将这3位二进制数转成对应的8进制数即得到当前对应的这类用户权限数值结果，最后要将三类用户的都利用相同的方法转成8进制数。**

数值权限查看stat命令可以查看



上面这个栗子sh1.sh的字符权限如下

|  |  |  |  |
| --- | --- | --- | --- |
| 用户类型 | 所有者 | 所有组 | 其它用户 |
| 字符权限 | r w x | r - x | r - x |
| 二进制数 | 1 1 1 | 1 0 1 | 1 0 1 |
| 八进制数 | 4+2+1=7 | 4+0+1=5 | 4+0+1=5 |

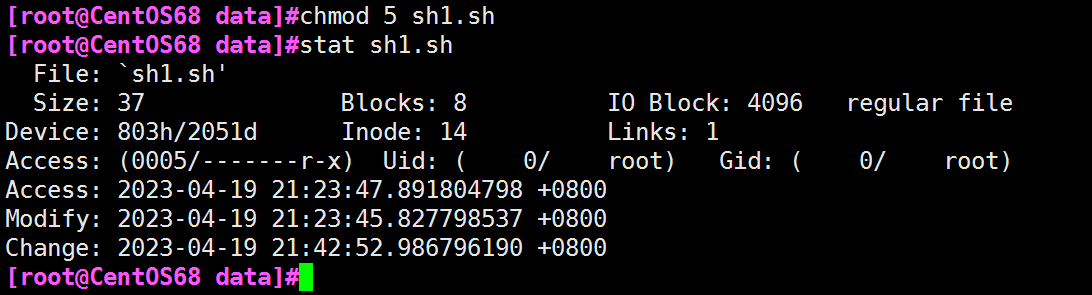
所以采用数值方法设置上面的sh1.sh的权限就是下面这样子写了

#chmod 755 sh1.sh

对于文件夹一般的权限默认是755，对于文件默认的权限一般是644，少了一个可执行权限

能不能简写？

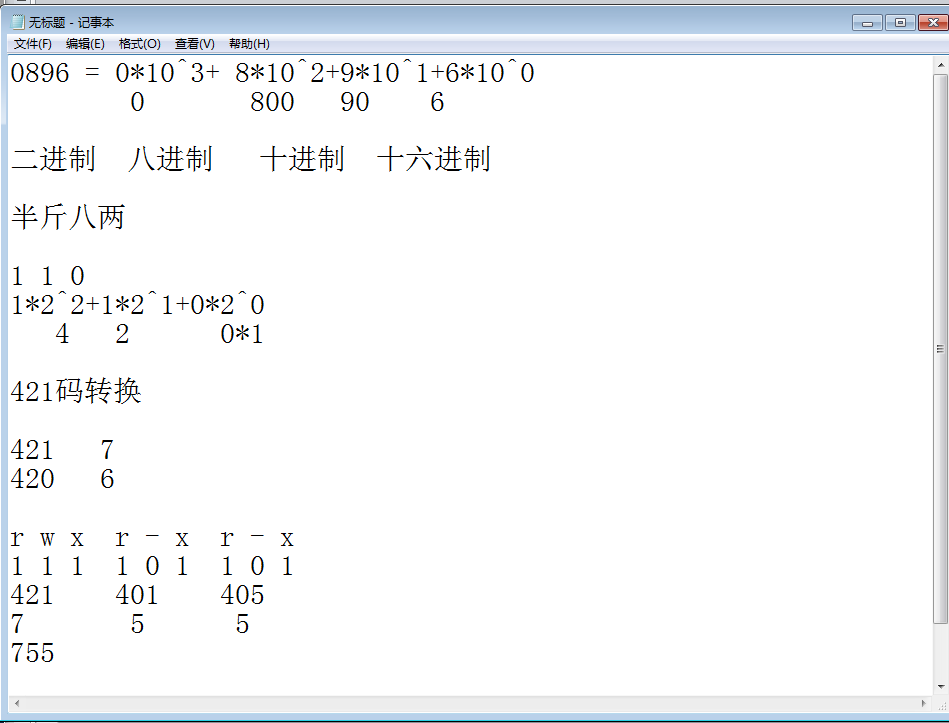
#chmod 5 sh1.sh

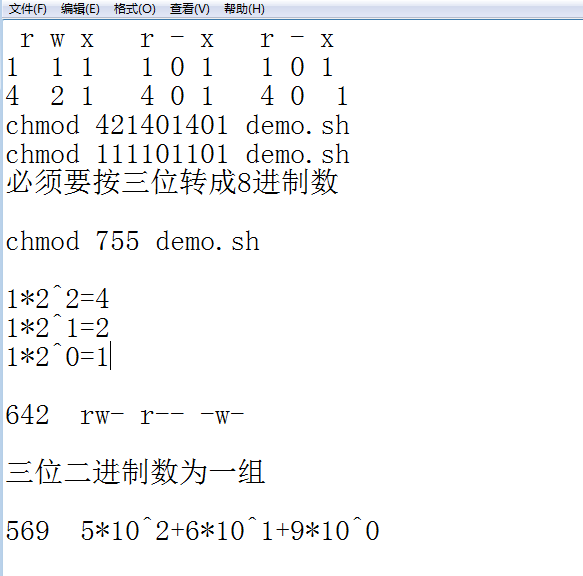


这个是重新设置，对应位置上的不能去省，省掉就是0

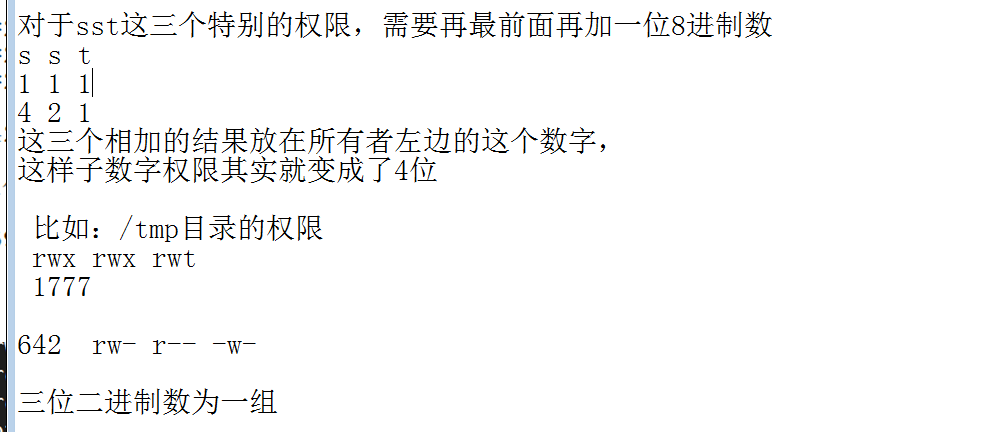
**千位数字上是sst(特殊权限，本章后面会讲)按三个二进制数来算的**

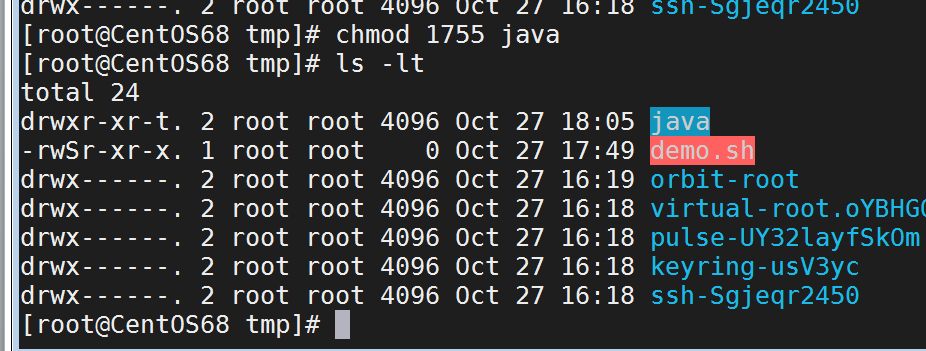
二进制的转换 421码





**特种权限用数字怎么来处理**





注意 使用数字是重新设置

数字从个位开始

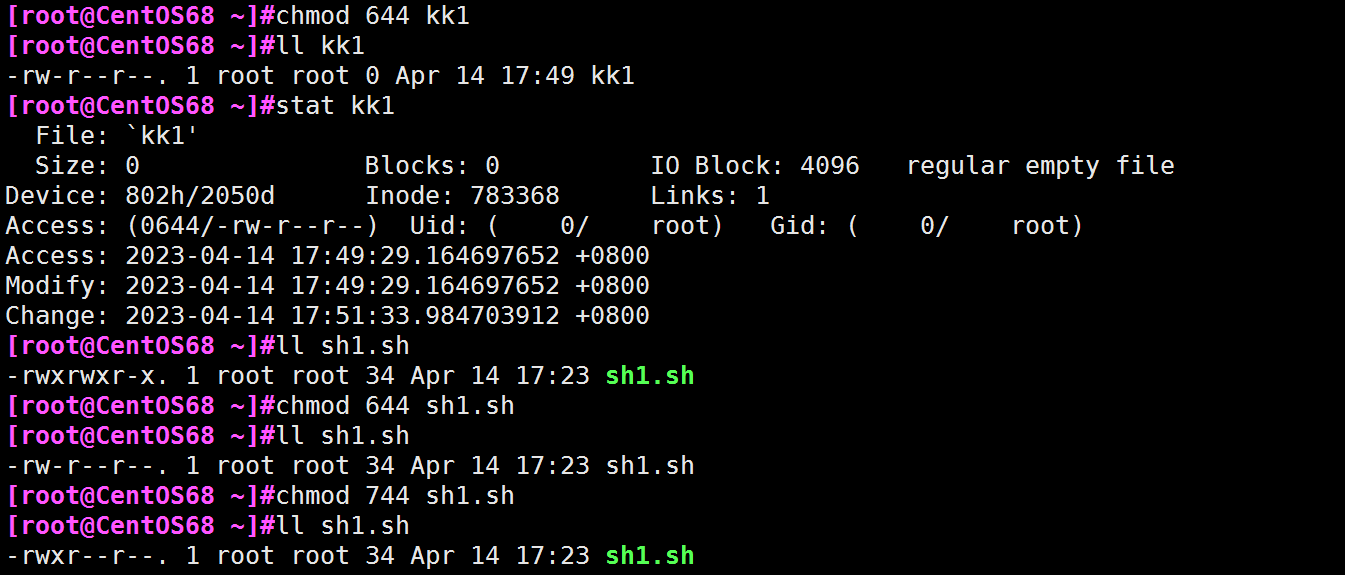
如果只写两个数字，那么就是个位和十位，而这个时候百位和千位则是0

个位代表：其它用户权限

十位代表：文件所有组的权限

百位代表：文件所有者权限

千位代表：文件目录特殊权限组合值



5 特殊权限

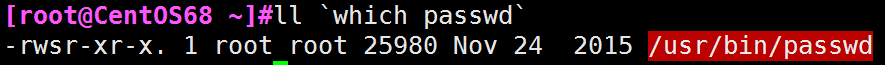
Linux文件系统上的特殊权限，前面介绍了三种常见的权限：r, w, x

还有三种特殊权限：SUID, SGID, Sticky

1 特殊权限SUID

出现的位置在所有者的那个x位上，用s表示，小写的

栗子：



安全上下文

前提：进程有属主（所有者）和属组（所属组）；文件有属主（所有者）和属组（所属组）

1. 任何一个可执行程序文件能不能启动为进程,取决用户对程序文件是否拥有执行权限

**2. 启动为进程之后，其进程的属主（所有者）为发起者,进程的属组为发起者所属的组**

3. 进程访问文件时的权限，取决于进程的**发起者**

(a) 进程的发起者，同文件的属主：则应用文件属主权限

(b) 进程的发起者，属于文件属组；则应用文件属组权限

© 应用文件“其它”权限

**二进制的可执行文件上SUID权限的功能：**

任何一个可执行程序文件能不能启动为进程：取决发起者对程序文件是否拥有执行权限

**启动为进程之后，其进程的属主(所有者)为原程序文件的属主（这是重点，也就是SUID的作用）**

SUID只对二进制或可执行程序脚本有效

SUID设置在目录上无意义

当在所有者位置出现s权限的时候，启动进程的用户会临时获得这个程序的（passwd）所有者身份root,我们修改自己密码的时候需要写入/etc/shadow文件，身份是以root去写的

SUID权限设定：

chmod u+s FILE...

chmod 4xxx FILE

数值时：

就出现4位八进制数字了（最左边的那个），也是由三个二进制数来组成，suid的s占二制数最高位

1 0 0

4

当我们只是一个文件在所有者位置上出现s这种权限时，千位是100,转了之后就是4

chmod u-s FILE...

范例#ls -l /usr/bin/passwd

-rwsr-xr-x. 1 root root 34928 May 11 2019 /usr/bin/passwd

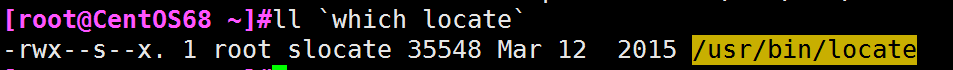
2 特殊权限SGID

出现位置，在权限第二栏组这位置x权限位变成了小s

栗子



我们如果用locate去搜索时，其实是要去读这个文件



**二进制的可执行文件上SGID权限功能：**

任何一个可执行程序文件能不能启动为进程：取决发起者对程序文件是否拥有执行权限

**启动为进程之后，其进程的属组为原程序文件的属组（SGID的作用）**

SGID权限设定：

chmod g+s FILE...

chmod 2xxx FILE

0 1 0

千位上面是2

chmod g-s FILE...

**目录上的SGID权限功能：**

默认情况下，当用户在这个目录里面去创建文件或者文件夹（目录）时，其属组（所有者）为此用户所属的主组

一旦某目录被设定了SGID，则对此目录有，写权限的用户在此目录中创建的文件所属的组为此目录的属组，通常用于创建一个协作目录

SGID权限设定：

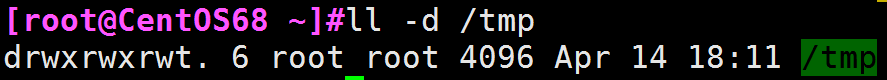
chmod g+s DIR...

chmod 2xxx DIR

chmod g-s DIR...

3 特殊权限 Sticky 位

**只能出现在目录权限上，出现位置是其他用户的x权限位置，也就是这个时候原来的x权限变成了t**



具有写权限的目录通常用户可以删除该目录中的任何文件，无论该文件的权限或所有权

在目录设置Sticky 位，**只有文件的所有者或root可以删除该文件**

sticky 设置在文件上无意义

Sticky权限设定：

chmod o+t DIR...

chmod 1xxx DIR

chmod o-t DIR...

范例

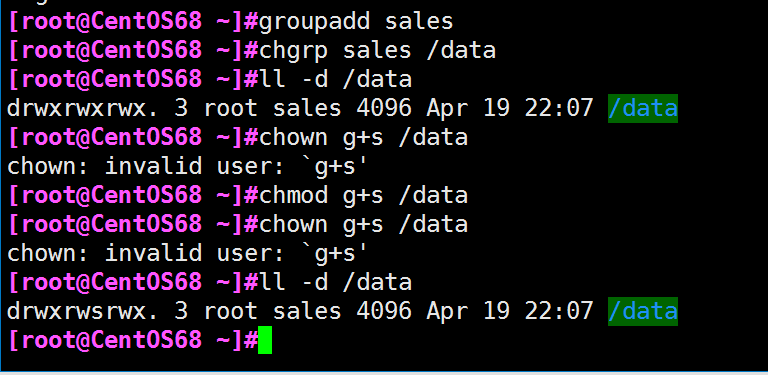
#ll -d /tmp

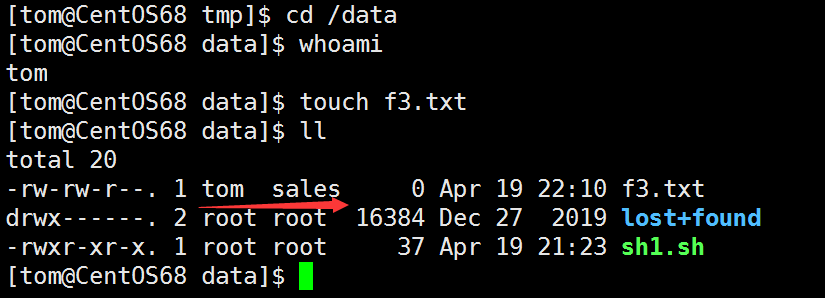
drwxrwxrwt. 15 root root 4096 Dec 12 20:16 /tmp

s s t

1. 1 1

**上面三个权限能不能同时出现在一个对象身上啊？**





6权限掩码

umask的值可以用来保留在创建文件权限

新建文件或者目录，sst都是没有的

**实现方式：**

新建文件的默认权限: 666-umask，默认把x权限取消了，如果所得结果某位存在执行（奇数）权限，则将其权限-1,偶数不变

新建目录的默认权限: 777-umask

非特权用户umask默认是 002

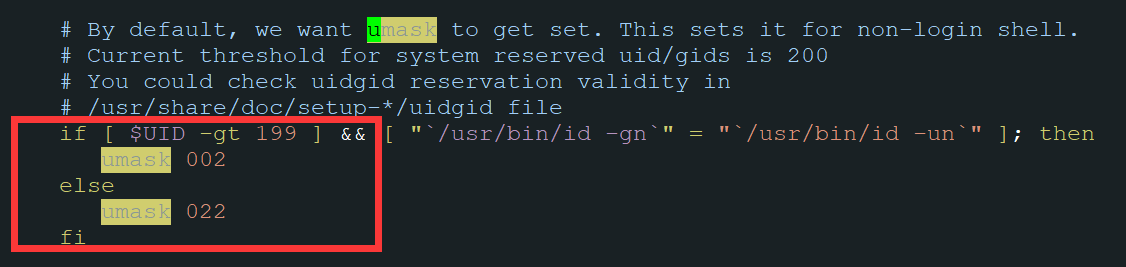
root的umask 默认是 022

查看umask



持久保存umask

全局设置： /etc/bashrc



7 ACL权限设置

为不同的用户设置不同的访问权限

chmod只能够设置一个用户（所有者），组，其它

比如：要想设置tom rose jack等用户不同的访问权限，这个时候就要通过acl访问控制列表来进行设置，windows下NTFS具有这个功能

Linux系统下，ext4文件系统开始默认支持acl功能，ext4以前，需要在挂载磁盘时增加acl的选项

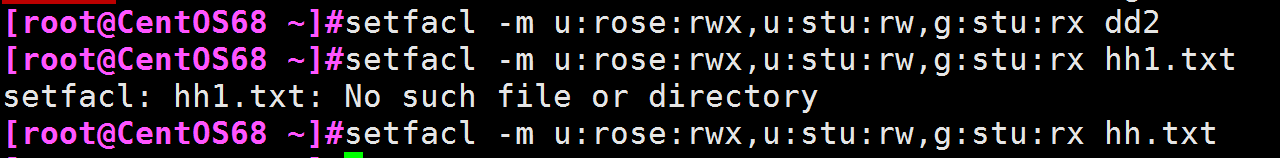
1 设置acl权限setfacl

范例：

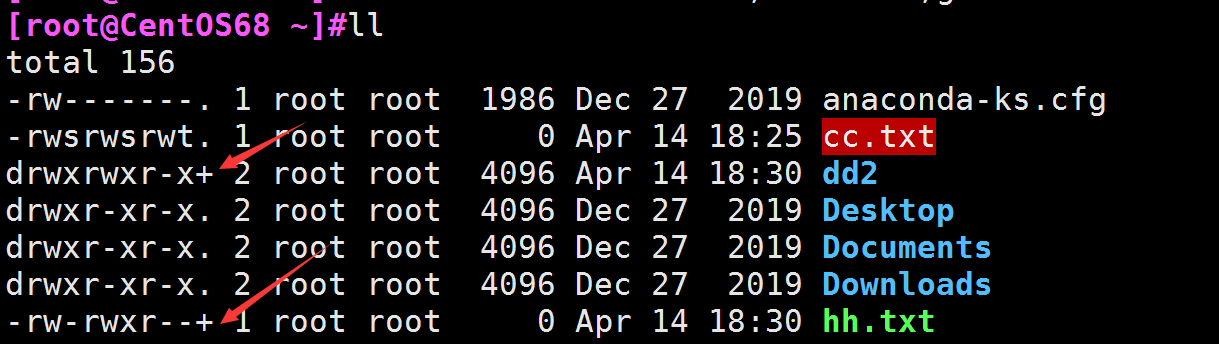
让tom用户对sh1.sh这个脚本也有rwx权限

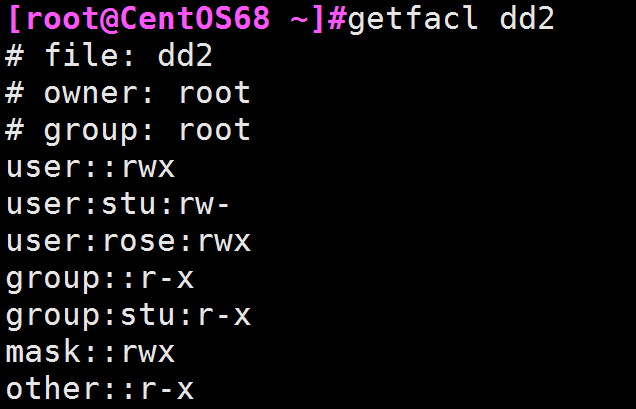
#setfacl –m **u:zhangsan:rwx, u:lisi:r,u:wangwu:rw, g:bakgroup:rx**

**#setfacl -m u:用户:权限，u:用户:权限，….g:组名称:权限 目录或者文件**



2查看acl权限：getfacl





8 实验安排

1 按下面列出命令查看权限

1. 利用ls -ld 查看如下3个目录权限 /root /home/stu /
2. 利用ls -l查看如下两个文件权限/usr/bin/passwd /usr/bin/locate
3. 利用ls -ld 查看/tmp目录权限
4. 利用stat 查看/tmp目录权限，主要观察数字权限

2 权限设置与执行脚本

利用vim创建一个脚本，脚本名称 hello.sh

步骤

[root@localhost ~]# vim hello.sh

脚本内容如下：

#!/bin/bash

echo "hello world!!!";

[root@localhost ~]# chmod u+x hello.sh

#执行脚本 /root/hello.sh ./hello.sh

**利用数字设定所有人都有可执行权限，试试看怎么设置？**

# chmod 755 hello.sh

执行hello.sh

[root@localhost ~]# ./hello.sh

./ 表示的是pwd所代表的路径

3创建teacher用户，设置密码123

4 将hello.sh复制到teacher的主目录下

#cp hello.sh /home/teacher

将/home/teacher/hello.sh所有者给teacher

#chown teacher /home/teacher/hello.sh

5 再利用 su - teacher切换到teacher用户

对脚本hello.sh执行看是否有结果，执行完成后退出teacher用户(exit)

6 acl权限设置

1. 将hello.sh文件复制到/tmp目录下
2. 利用acl权限设置，对stu用户给予rwx权限，并查看权限结果
3. 利用su - stu
4. 进入/tmp目录，执行hello.sh，看是否可以执行，执行完成后退出exit

7 创建目录 /public 将其设置为具有 t 特殊权限,设置好后观察设置结果

o+t

将/public的权限设置为1777，即和/tmp目录相同的权限

8 将/public的所有者给teacher用户，并观察结果

操作如下：

创建用户jack和tom

切换到tom身份下

创建文件 /public/tomtest.txt

切换到jack身份下

创建文件 /public/jacktest.txt

以jack身份删除tomtest.txt看看是否能删除成功

9 SGID操作，创建一个组sales

1） 将上面的/public文件夹的所属组修改为sales

2）将/public权限设置SGID

3）切换到用户tom身份，进入/public目录，在此目录里面创建一个文件tom.txt，4）利用ls -l 查看tom.txt的所属组是哪个