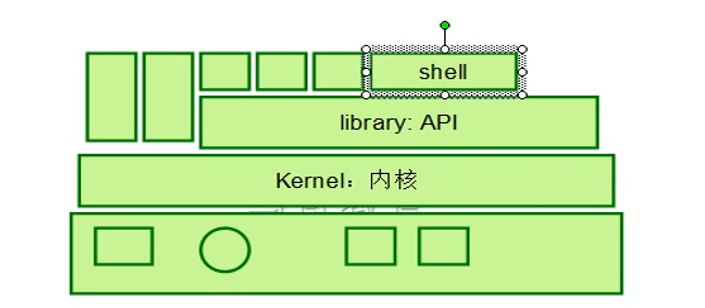
# Linux 基础

## 库



库： lib 可以理解为公用模块。为了免去重复开发，避免与内核直接进行系统调用而在内核的基础上封装的公用模块。

lib （API） 上层应用程序就可以调用这些库，程序员可以调用这些库编程，也可以不用。也可以直接在内核上开发。

shell 就是典型的lib , Linux系统中的库是 \*.so 称之为共享对象，有静态和动态的。lib 是不能独立运行的。用来被上层程序调用的。

## 进程

当程序被内核加载到内存，运行起来，就成为了进程，进程是程序的副本，是程序的运行实例。如果程序在编译的时候使用了某个共享库，这个进程启动前必须由内核将其所依赖的某个共享库载入内存。如果其他程序的进程也依赖这个共享库，就不必重复载入。

## 资源和服务

用户： 访问资源和服务的主体。对计算机而言就是使用资源的凭证或者标识符。

资源： 计算机的服务和资源。

权限： 定义某个资源和服务的访问的能力。

组： 用户组，（角色）定义了多个 资源和服务的访问能力。就是权限的集合，权限的容器。 可以方便的为用户指派权限，在系统中也是一个标识符。

Linux系统中对文件资源定义了3组用户的使用能力

1 文件的属主，就是文件的所有者有什么样的权限

2 文件的属组，文件属于哪一个组，具有什么样的权限

3 文件的其他用户，（除属主和属组外的其他用户）

对于计算机而言什么是用户，怎么识别用户的呢？

登陆输入用户凭证(用户名和密码)，用户就有一个系统唯一的标识符来标识该用户。

**用户发起命令，进程代为执行，进程也是有属主和属组的。**

**谁发起的进程那么进程就以谁的身份标识去执行。当一个文件资源被一个进程访问使用时，进程就会用标识去对比文件资源的权限标识。整个过程都是系统内核的资源的安全管理机制来完成的，称进程的安全上下文。**

## 权限类别

**文件：**

**r : read 可读，可以使用类似cat ，more ,less ，tail等这样的命令查看文件内容。**

**w : write 可写，可以编辑或者删除此文件。**

**x : excutable 可执行，可以在命令提示符下当作命令提交给内核运行。**

**-： 无权限**

**目录：**

**r: 可以对此目录执行ls，以列出内部的所有文件**

**w: 可以在此目录中创建文件**

**x： 可以使用cd切换进此目录，也可以使用ls -l 查看内部文件的详细信息。**

**注意：如果没有x权限，只能使用ls 是不能使用ls -l**

**-： 无权限**

**一般默认不能让文件带有x权限，防止文件中有恶意的命令，存在不安全隐患。**

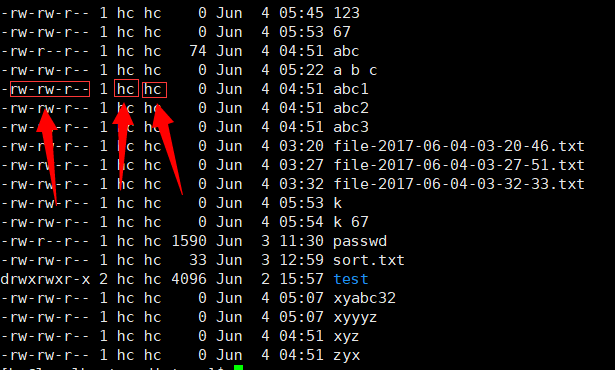
**而目录一般有x权限，以方便查看目录中的文件。**

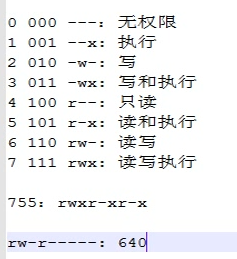
**rw-rw-r--**

**rw-** **属主的权限**

**rw- 属组的权限**

**r-- 其他用户的权限**





**八进制： 3位二进制**

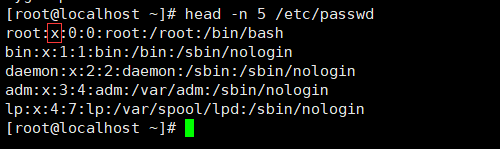
用户： UID /etc/passwd

组： GID /etc/group

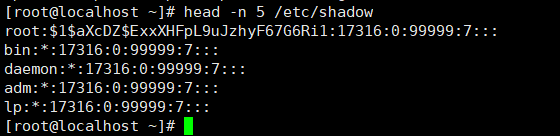
影子口令 /etc/shadow

组口令 /etc/gshadow

密码位都是x ，真正的密码保存在 /etc/shadow下



root的密码在这里



## 用户类别

管理员 ： 拥有全部权限 uid 为 0

普通用户： uid 1-65535

系统用户：1-499

一般用户：500-60000

系统用户

运行特定任务，如web服务，又不需要登陆，如果用管理员运行，太危险了，如进程 被劫持，管理员的权限太大了，所有资源都可以访问。这时系统用户有没有太大权限， 可以安全运行。

## 用户组类别

站在用户角度看

管理员组：

普通组：

系统组：

用户组：

用户组角度看

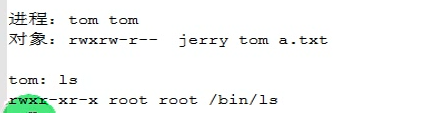
一个用户是属于一个或者多个组，他创建的文件属组是哪个组呢？

用户组类别：

基本组：也可以称为私有组，每一个用户都有一个默认组，称之为基本组，他创建的文 件属于这个基本组，属主就是他自己。用户的基本组未必是私有组。

私有组：如果一个用户在创建的时候没有为其指定所所属的组，那么创建用户的同时系 统将为其默认创建一个和用户同名的组，这个组就他一个人，称之为私有组。

额外组：基本组之外的称之为额外组或者附加组。



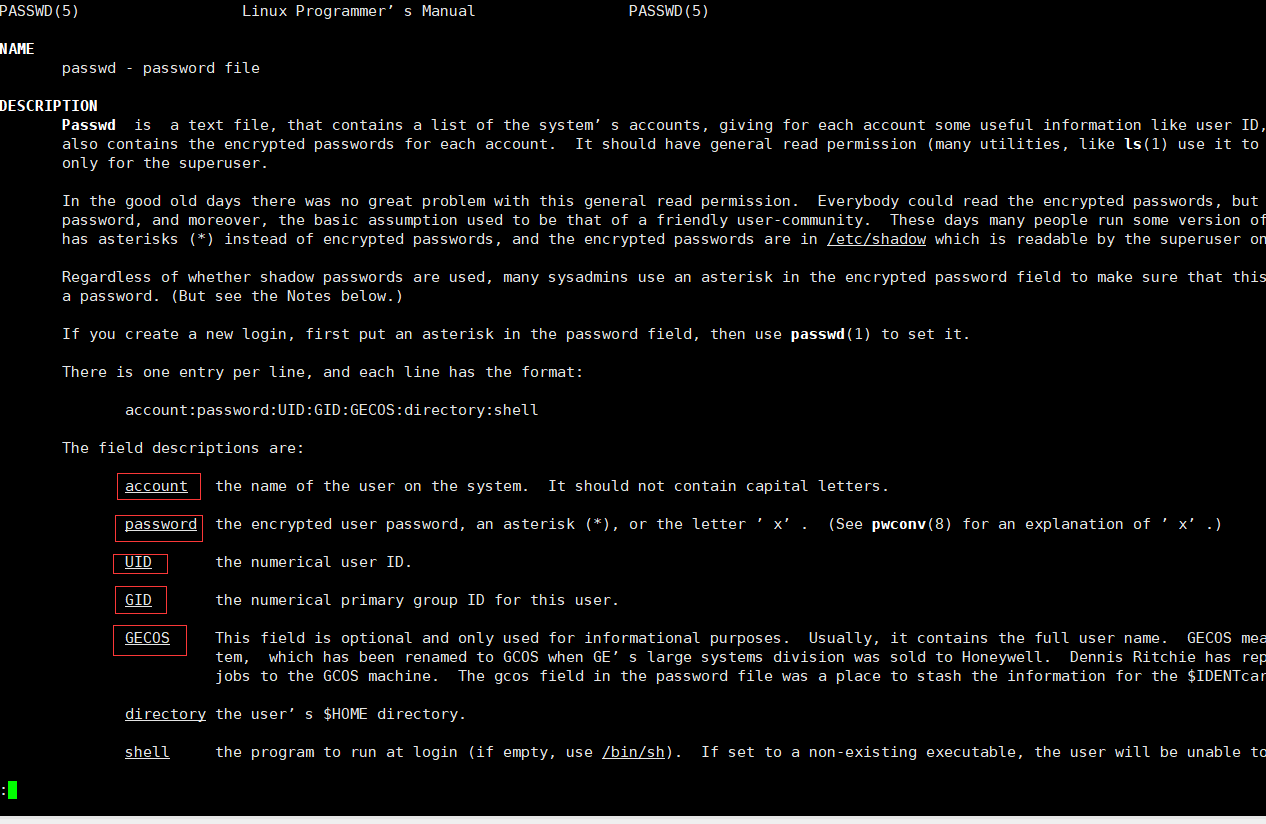
文件是否能被访问，取决与发起者（进程）的身份，tom，和a.txt的属主,属组进行比较 ，a.txt的属主是jerry , 第一组权限没有，属组是tom组，就看tom是不是tom组的属主，

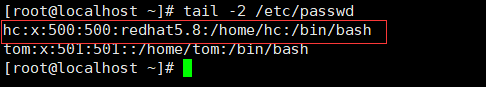
如果不是，第二组权限没有了，就看第3组权限是不是有读的权限，r--表明ls可以读取a.txt的内容。

## 用户管理命令

### passwd

修改用户密码





passwd 中的字段含义

account: 账号

password: 密码占位符，实际密码在shadow中

UID ：用户id 管理员UID 为 0

GID： 组id 管理员基本组 GID 为 0

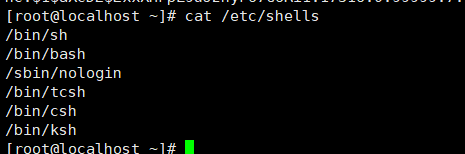
GECOS: 可选的，用户的注释信息，可以为空

home dir ：用户的家目录

shell ： 用户默认的shell

当前系统合法的shell , 安全的shell

cat /etc/shells

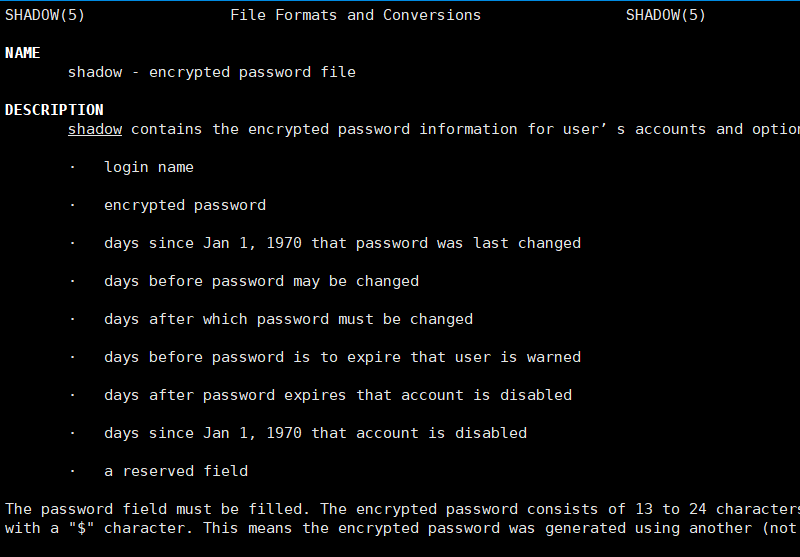


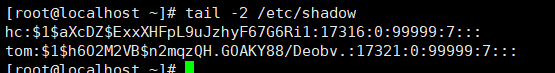
### shadow

密码存放文件

which shadow

man 5 shadow





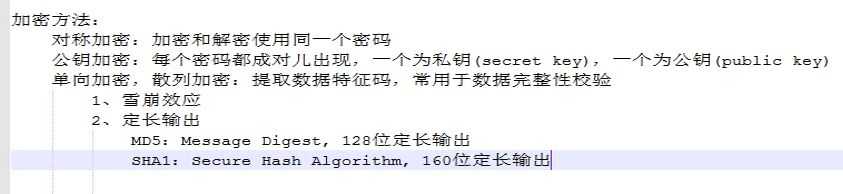
shadow中字段的意思

1. login name ： 登陆名
2. encrypted password : 加密了的密码

hc:$1$aXcDZ$ExxXHFpL9uJzhyF67G6Ri1:17316:0:99999:7::: MD5加密

**盐 随机生成**

**密码为\* 和 ！！的标识被锁定的，不允许登陆的，也可以把密码前面加两个！，他就不 能登陆了。**



1. days since Jan 1,1970 that password was last changed

上一次密码修改距离1970.1.1开始的天数。

1. days before password may be changed

密码被修改之前经过的天数，密码修改的最短使用期限。

0 表示没有限定，可以随时该密码。5 表示密码使用5天过后才能修改。

1. days after which password must be changed

密码的最长使用期限

1. days before password is to expire that user is warned

密码在过期之前多少天提示用户，密码要修改了，还可以正常工作

1. days after password expires that account is disabled

密码过期后的宽限期多少天，这段时间不能工作，要修改密码，如果过了宽限期，对不起，锁定 ！！ 找管理员解锁。

1. days since Jan 1, 1970 that account is disabled

明确指定多少天后密码过期，99999 相当于永久了

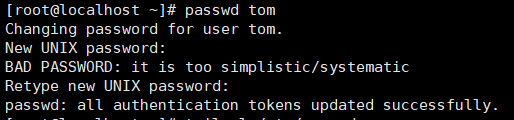
1. a reserved field

保留字段

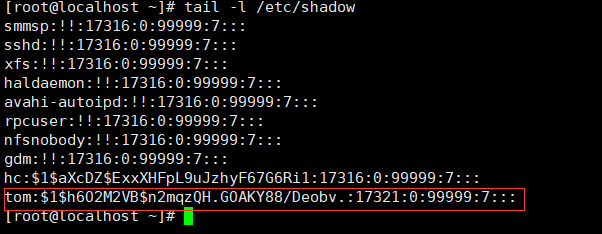
1普通用户可以使用passwd修改自己的密码

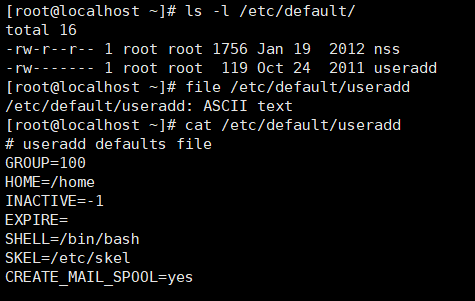
2管理员可以使用passwd加上用户名为普通用户添加密码。

passwd tom



密码就添加好了



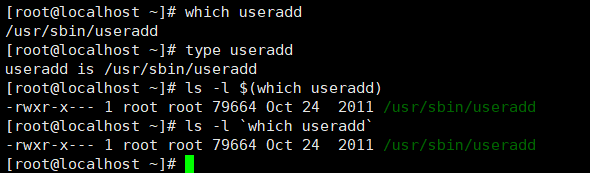


/etc/default/useradd 当管理员添加用户的时候，如果不指定，默认使用该文件下的配置。

### useradd

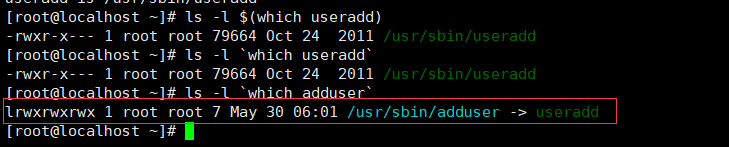
**用户添加**

useradd 在/usr/sbin/useradd



如果不知道useradd在哪个目录下，要查看该文件的，使用命令替换吧，

ls -l `which useradd` == ls -l $(which useradd)

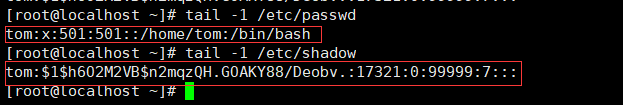


adduser->useradd adduser是一个快捷键指向useradd，实际上是同一个命令。

1. 默认添加，什么也不指定

useradd USERNAME

useradd tom

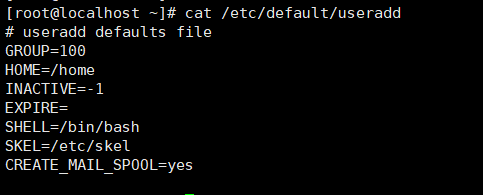


passwd中可以看出 tom是一个普通用户，UID 501 GID 501 默认使用bash

tom的基本组也添加好了。



如果不指定，按照默认配置文件

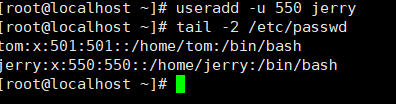


1. 指定添加

**-u**, --uid UID

The numerical value of the user’s ID. This value must be unique, unless the -o option is used. The value must be non-negative. The default is to use ID value equal to or greater than UID\_MIN and smaller then UID\_MAX. Values between 0 and UID\_MIN are typically reserved for system accounts.

-u 数字 这个数字就用户的id，这个数字必须唯一。

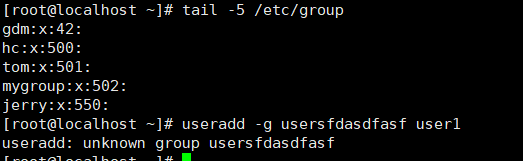


**-g**, --gid GROUP

The group name or number of the user’s initial login group. The group name must exist. A group number must refer to an already existing group. /etc/default/useradd.

指定用户基本组的id或者组名，这个组名必须存在。

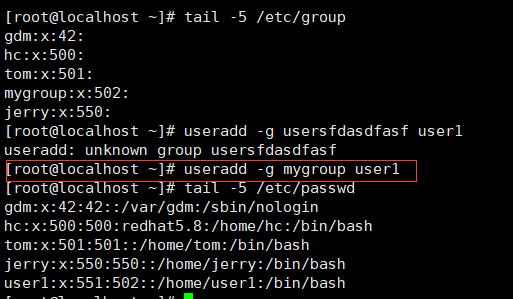
指定了要一个不存在的组名，unknown group



useradd -g mygroup user1

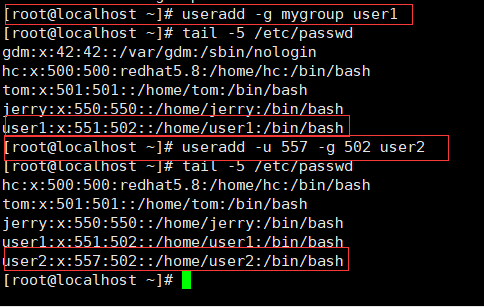
指定user1 的组名是mygroup(group中已存在)

如果没有指定uid 那么uid会在最后一个也是最大uid的基础上自增1作为自己的uid



useradd -u 557 -g 502 user2 == useradd –u 557 –g mygroup user2

指定uid 557 ，用户的基本组502 （group中的已经存在的组，mygroup:x:502）



**-G**, --groups GROUP1[,GROUP2,...[,GROUPN]]]

A list of supplementary groups which the user is also a member of. Each group is separated from the next by a comma, with no intervening whitespace. The groups are subject to the same restrictions as the group given with the -g option. The default is for the user to belong only to the initial group.

**为用户指定额外组也就是附加组。可以有多个，用逗号隔开，前提是这些组也必须存在**。

**-c,** --comment COMMENT

Any text string. It is generally **a short description of the login**, and is currently used as the field for the **user’s full name**.

**为用户指定注释信息。指定用户的全名**

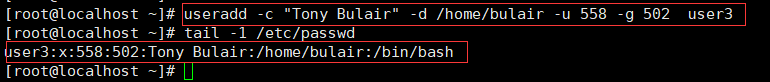
**-d,** --home **HOME\_DIR**

The new user will be created using HOME\_DIR as the value for the user’s login directory. The default is to append the LOGIN name to BASE\_DIR and use that as the login directory name. The directory HOME\_DIR does not have to exist but will not be created if it is missing.

为用户指定家目录

如：/xxx/xxx/xx

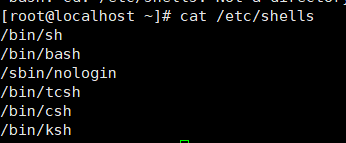
useradd –c “Tony bulair” –d /home/bulair –u 558 –g502 user3



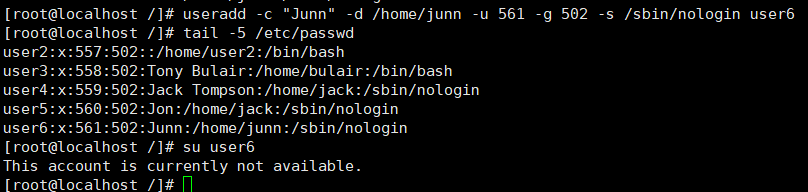
**-s**, --shell SHELL

The name of the new user’s login shell. The named program will be used for all future new user accounts.

指定用户使用那种shell, 指定路径 /etc/shells 里的shell



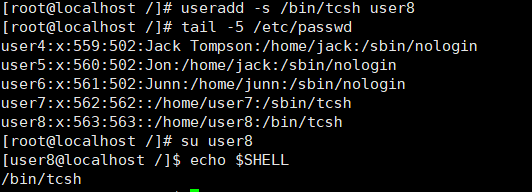
useradd -c "Junn" -d /home/junn -u 561 -g 502 **-s /sbin/nologin** user6



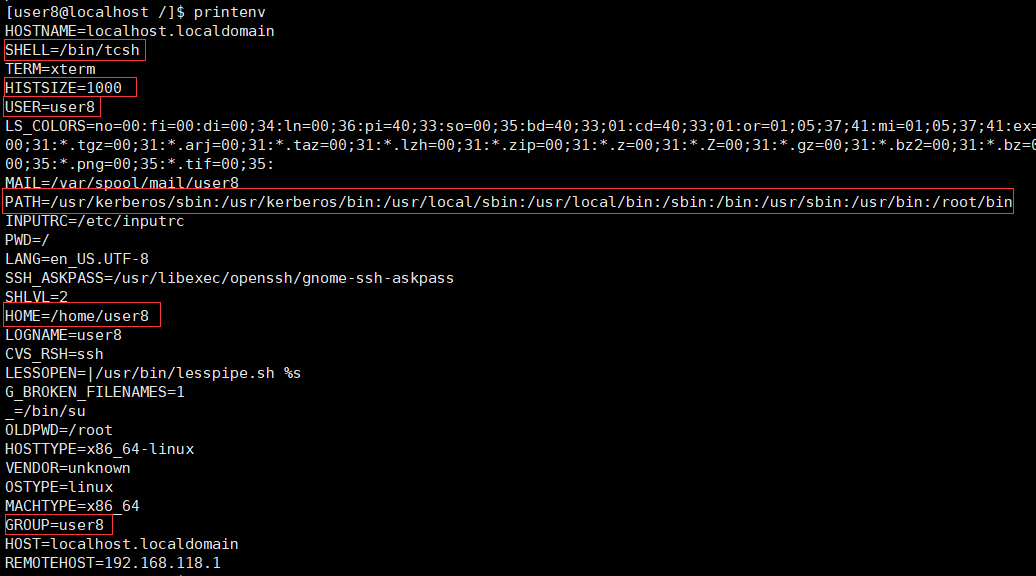
This account is currently not available

当前账户不可用 因为没有一个可以用的shell

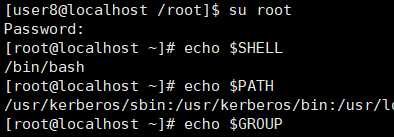
指定shell为/bin/tcsh 就ok



当用用户环境变量



echo $SHELL SHELL 为当前用户变量



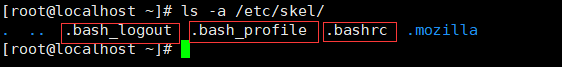
**-m**, --create-home

**The user’s home directory will be created if it does not exist.** The files contained in SKEL\_DIR will be copied to the home directory if the -k option is used, otherwise the files contained in /etc/skel will be used instead. Any directories contained in **SKEL\_DIR** （bash的配置文件）or /etc/skel will be created in the user’s home directory as well. The **-k** option is only valid in conjunction with the -m option. The default is to not create the directory and to not copy any files.

为用户创建家目录 –m 和 –k 一起使用

-m -k 强制复制 /etc/skel/ 目录下的3个文件到用户的家目录中，如果添加用户不指定，也会复制这3个文件到用户的家目录中去。

用户 bash 的配置文件

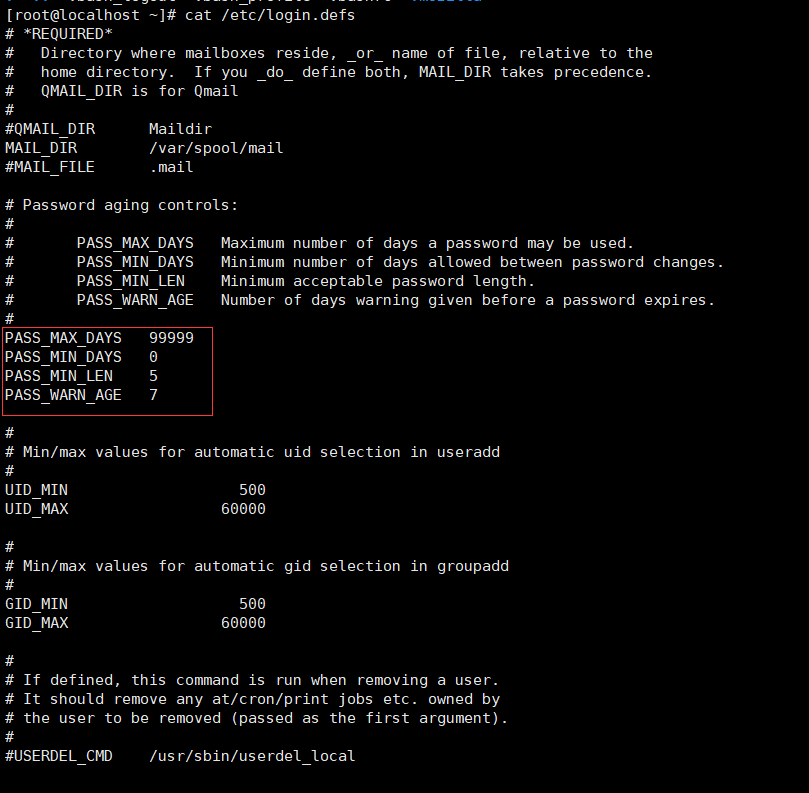


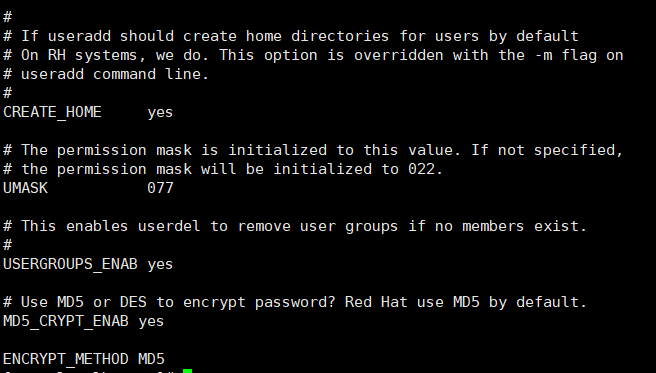
**-M : The user’s home directory will not be created**, even if the system wide settings from /etc/login.defs is to create home dirs.

不给用户创建家目录，就算是指定了家目录也不让创建

cat /etc/**login.defs**

创建用户时登入配置文件默认的配置，如果不指定，就是用下面的配置。





要不要创建家目录 默认yes

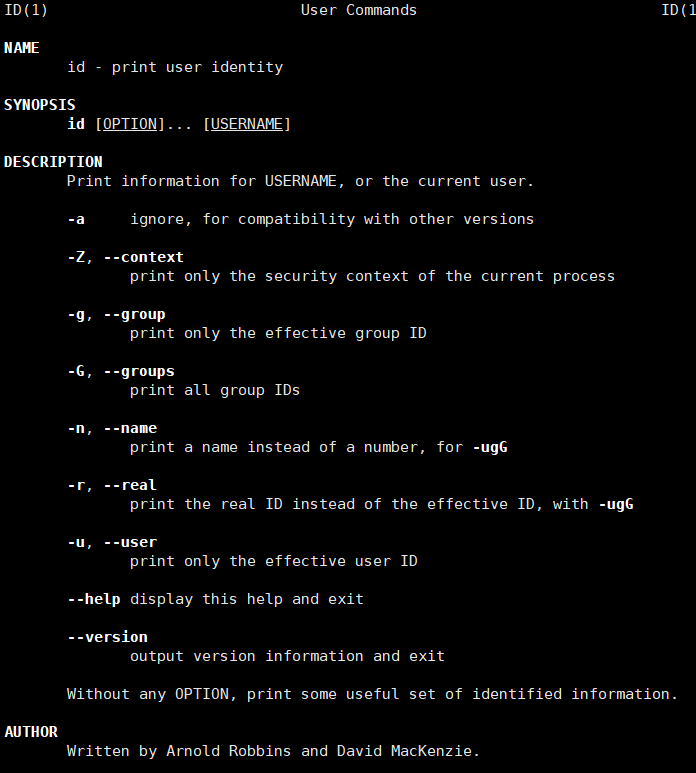
umask （反向掩码）用户的权限等。

**-r，**  **This flag is used to create a system account.** That is, a user with a UID lower than the value of UID\_MIN defined in /etc/login.defs and whose password does not expire. Note that useradd will not create a home directory for such an user, regardless of the default setting in /etc/login.defs. You have to specify -m option if you want a home directory for as ystem account to be created. This is an option added by Red Hat

**添加一个系统用户 id 在1-499 中找一个，系统用户通常不登入系统，没有家目录**

### id

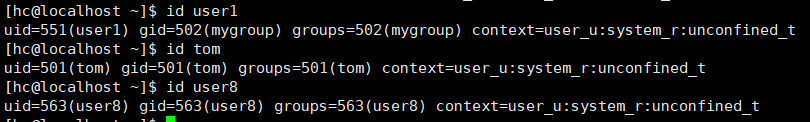
**id查看用户账户的属性信息**



id 显示当前用户（hc）账户的属性

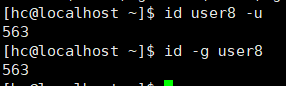


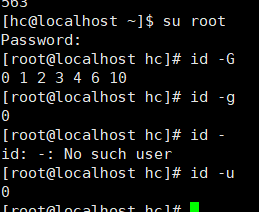
id user1 id跟上其他用户显示其他用户的账户属性



id –g user8 只查看user8的组信息的id号

id –u user8 只查看user8的用户id号



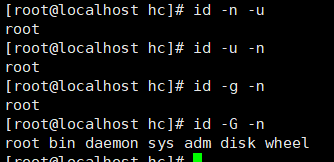


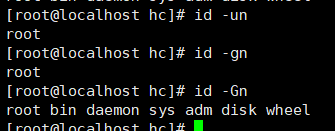
-n (name) 显示名称，不显示id号

id –u –n 显示用户的名称

id –g –n 显示用的所属基本组的名称

id –G –n 所属每一个的组的组名

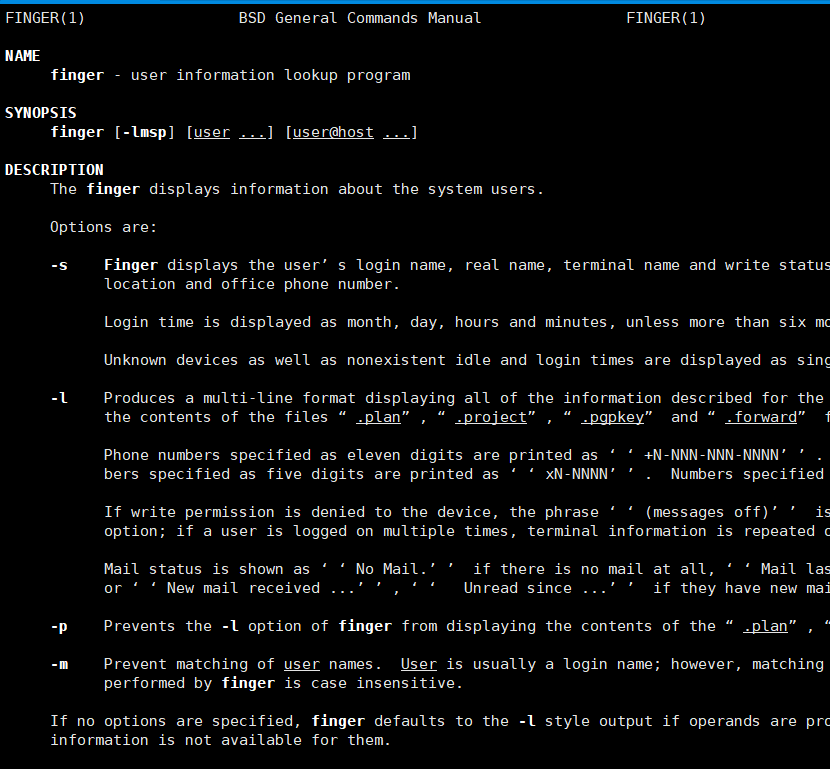




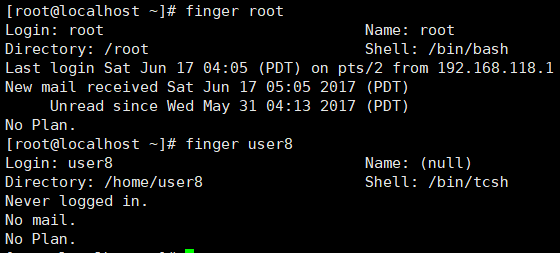
### finger

**检索用户信息的**

finger – user information lookup program



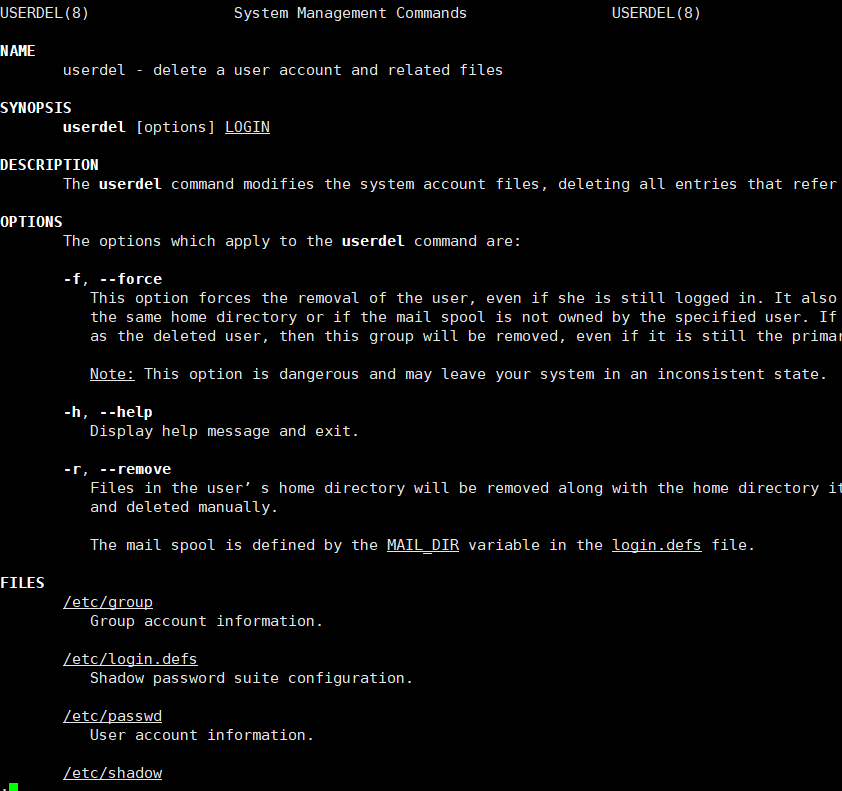
finger root 查看root账号的信息



### userdel

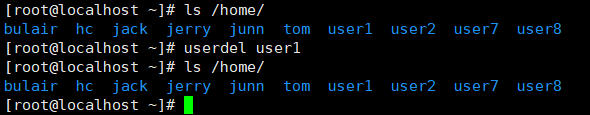
userdel delete user 删除用户

man userdel



如果不指定任何选项，用户的家目录是不会被删除的

userdel user1 删除user1 但是user1的家目录还在

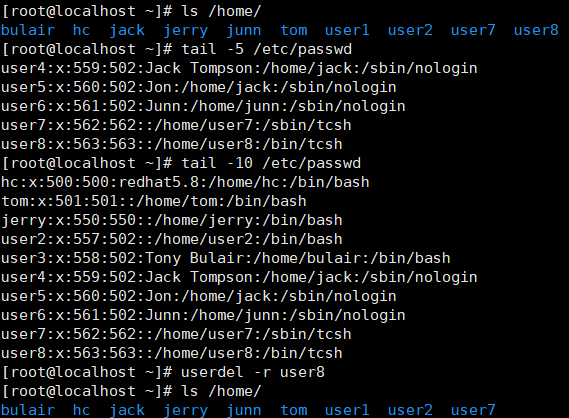


**-r** ,--remove

Fiels in the user’s home directory weill be remvoed along with the home directory…

删除用户的同时，同时删除用户的家目录

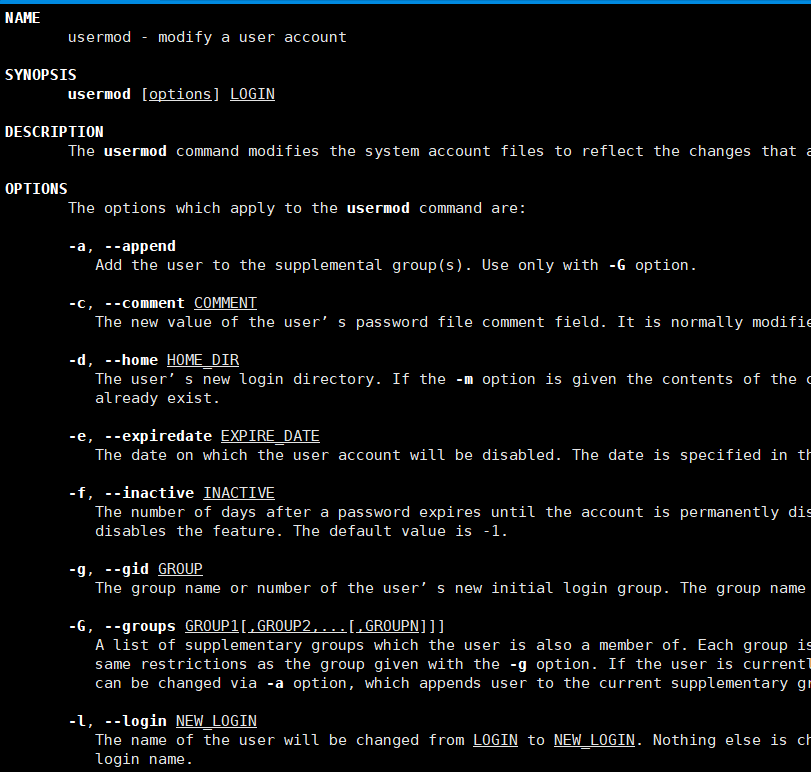
userdel –r user8



### usermod

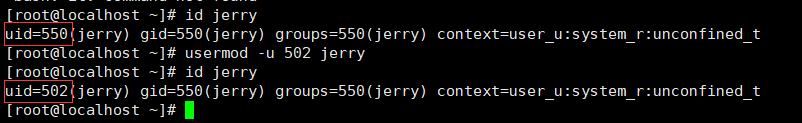
修改用户账户属性

与useradd用法相同。



**-u** 修改用户的id

usermod –u 502 jerry

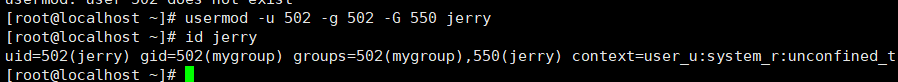


**-g** 修改用户的基本组id

**-G 修改用户额外组（附加组）此前的附加组就没有了，覆盖了所用就需要用 –a**

通常-G 和 –a 就一块使用，在原来附加组的基础上添加，就不会覆盖原来的附加组。

usermod –u 502 –g 502 –a –G 550 jerry



**-c** 修改注释信息

**-d** 为用户重新指定家目录 此前用户登录过的家目录就不能用了

所用要 –d –m 一块使用，移动用户此前家目录中的文件至新指定的家目录。

**-s** 修改用户使用shell

**-l** 修改用户的登录名

**-e** 指定用户的过期时间（用的少）

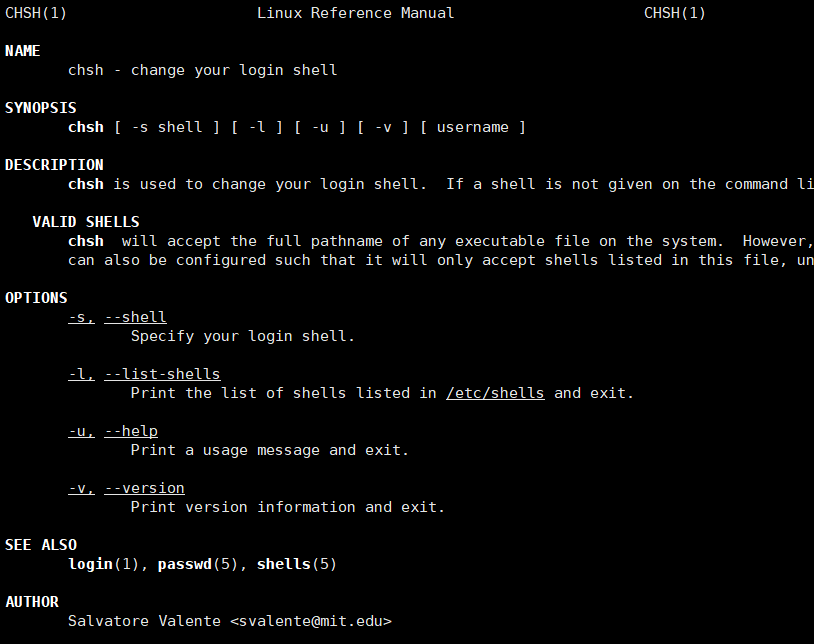
**-f** 指定用非活动时间（用的少）

**-L** 锁定账号，禁用

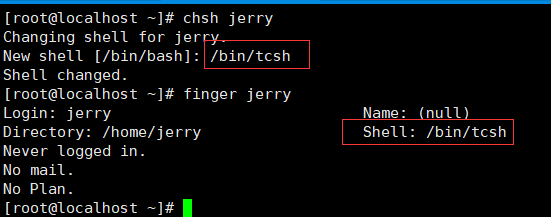
**-U** 解锁账号

### chsh

change shell 修改用户使用的shell



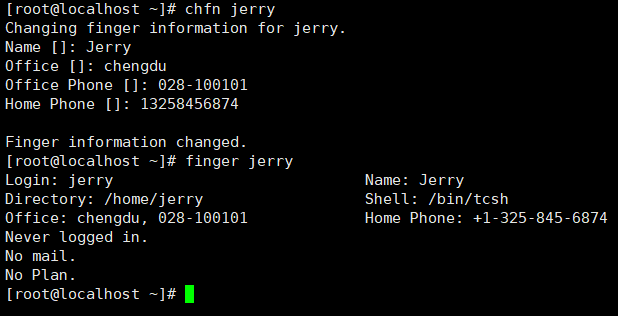
chsh jerry 修改用户的shell



### chfn

改变finger change finger

修改用户的注释信息

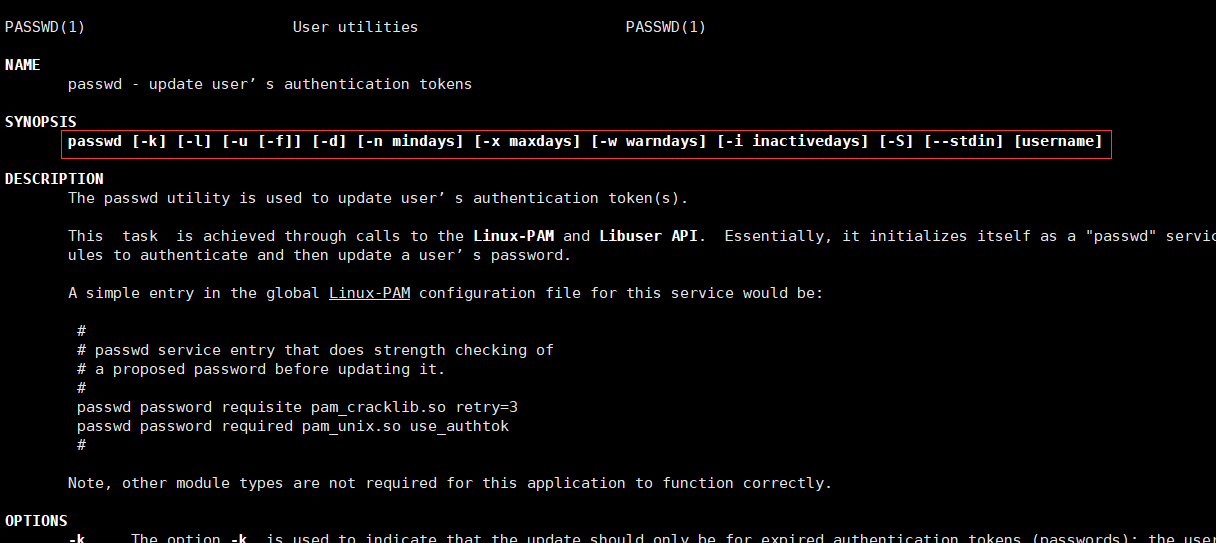


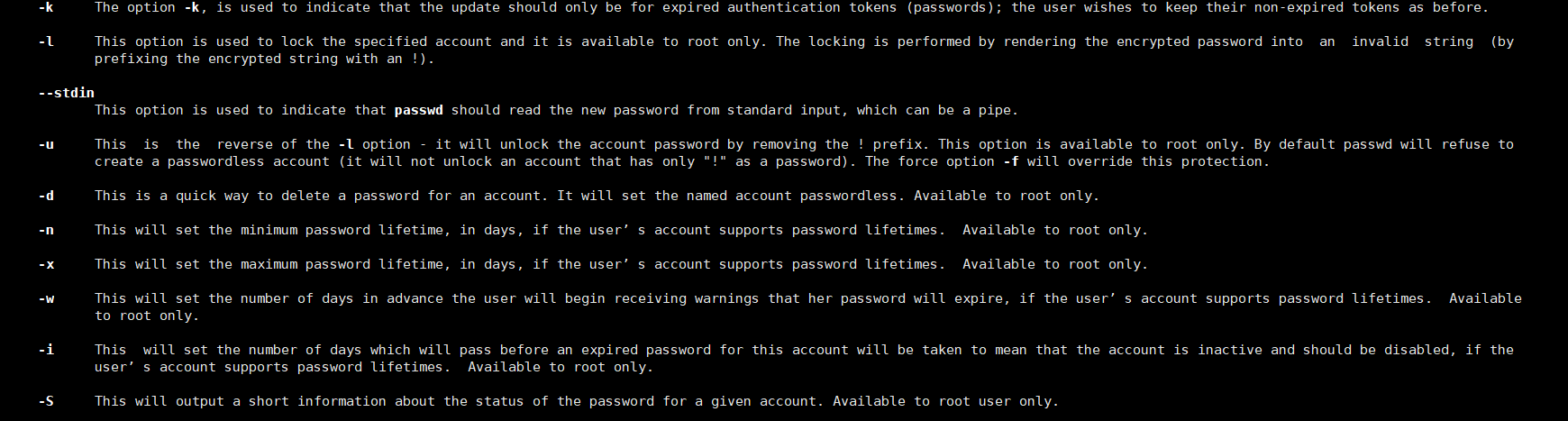
## 密码管理命令

### passwd

1普通用户只能使用passwd来该自己的密码

2管理员使用passwd 用户名 修改其他用户的密码



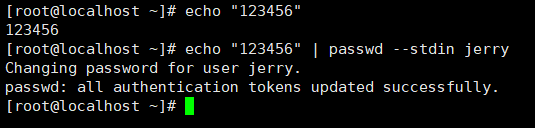


**--stdin** 标准输入

从标准输入读取密码

**echo “123456” | passwd --stdin jerry**

将123456字符串通过管道传入stdin来接收字符串 passwd 用字符串来修改jerry的密码



**-l 锁定账号**

**-u 解锁账号**

**-n 密码最短使用期限**

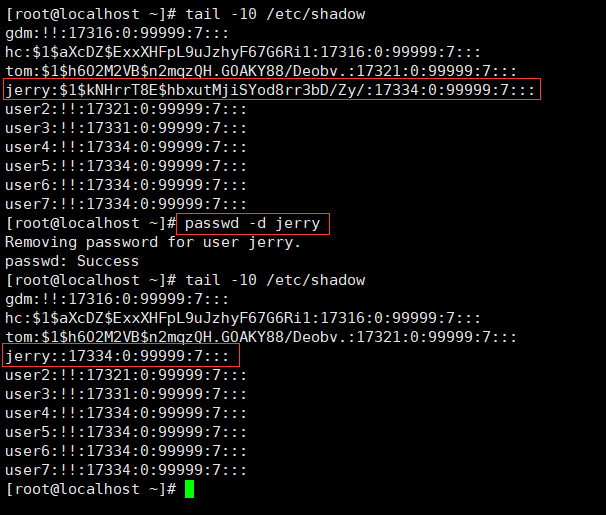
**-x 密码最大使用期限**

**-w 密码过期警告时间**

**-i 密码非活动期限的时间**

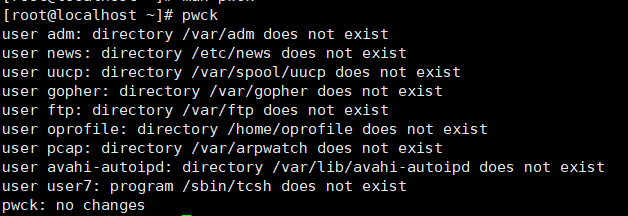
**-d 删除密码**

passwd –d jerry



### pwck

pwck 检查用户账号的完整性



## 组管理的命令

### groupadd

which group

man 5 group



用户组字段意思

group\_name 字段名

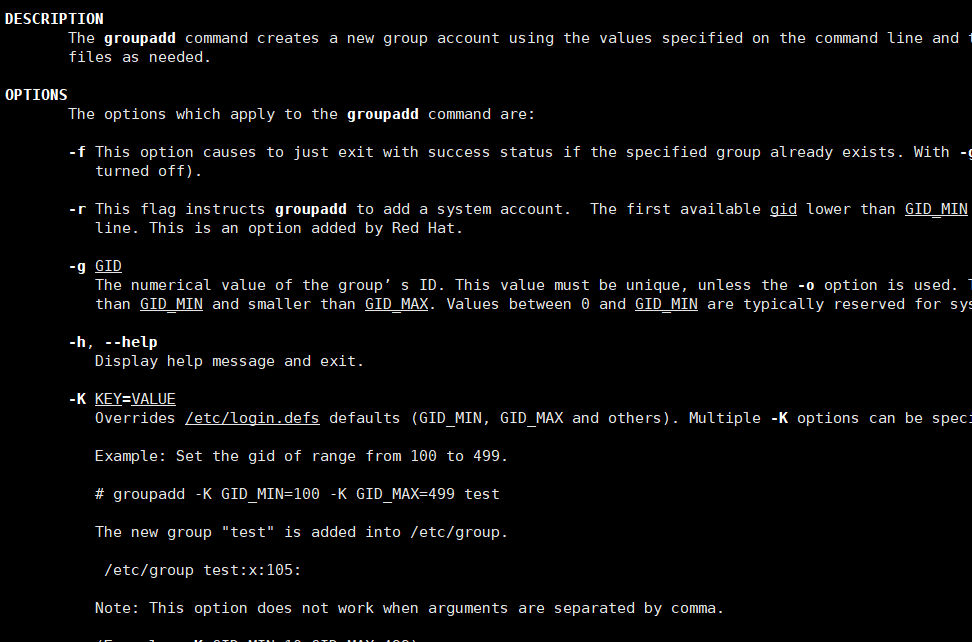
password 密码占位符

GID 组ID

user\_list 以这个组为附加组（额外组），多个用逗号隔开，所用是个列表，没有即为空。



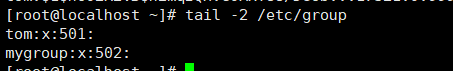
man groupadd



添加用户组

1.默认添加

groupadd GROUPNAME



2.指定添加

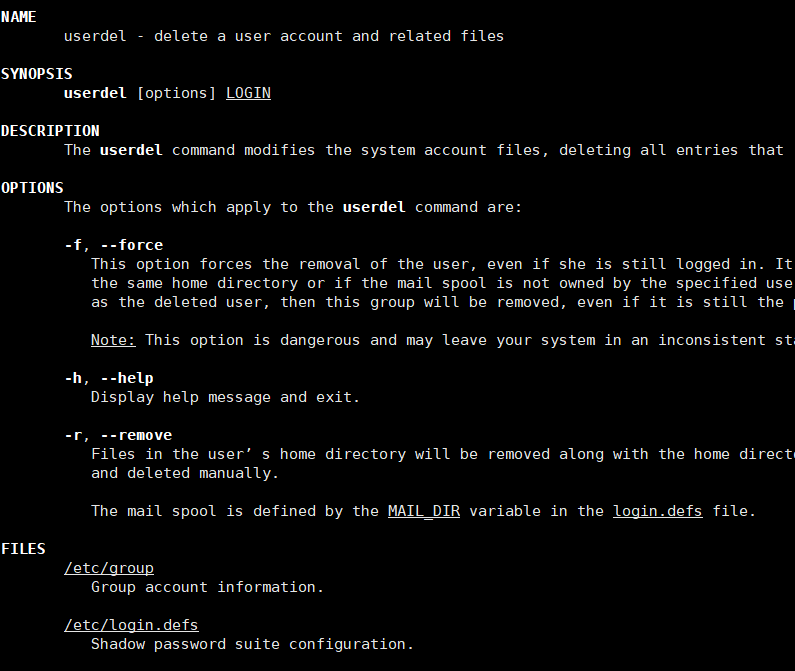
**-g 指定GID 普通用户通常500以后，如果不指定，默认GID最大的一个后面自增1**

**-r 添加一个系统组 1-499**

**groupadd和useradd 用法相同**

### groupdel

delete group删除一个组

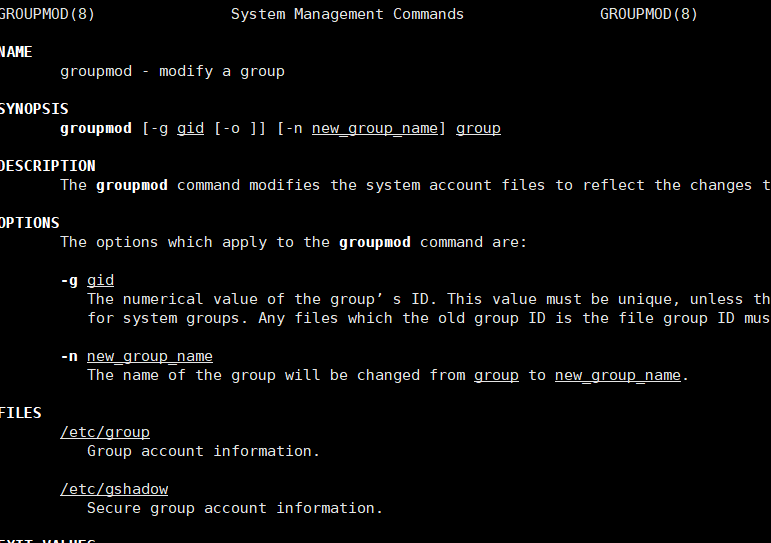


**-r --remove 删除**

**-f --force 强制删除**

### groupmod

modify group修改组



**-g** 修改GID

**-n** 修改组名

### gpasswd

设置组密码

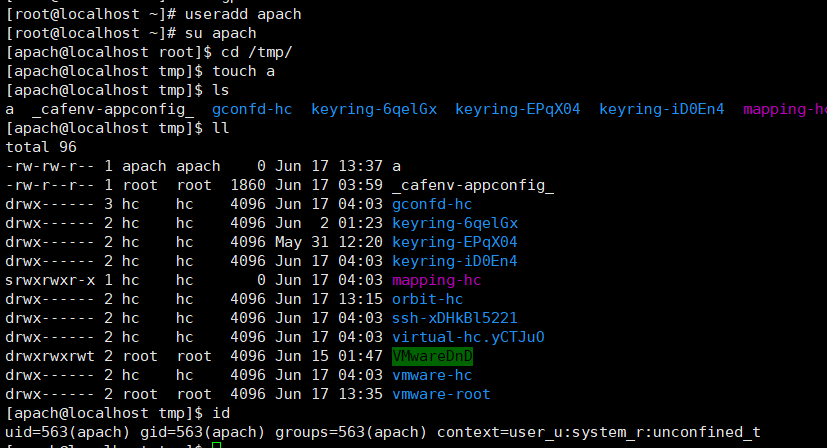


gpasswd 组名 设置组密码

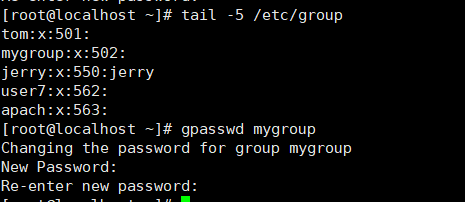
### newgrp

**登陆到一个新组中去**

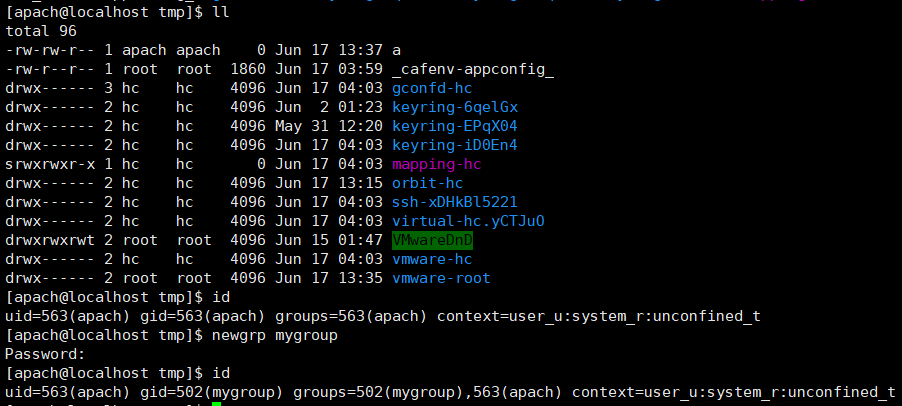
创建一个apach的账号



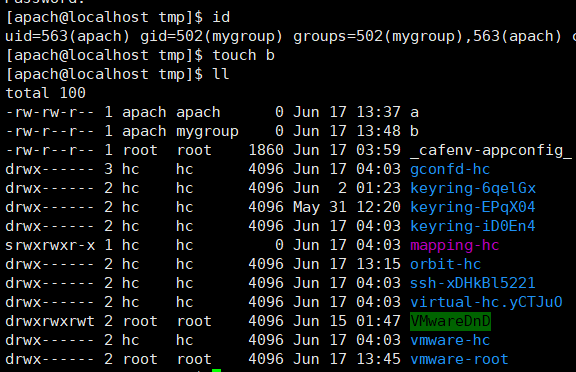
gpasswd mygroup 修改组密码



newgrp mygroup **切换用户的基本组和附加组为mygroup** 前提是mygroup这个组存在

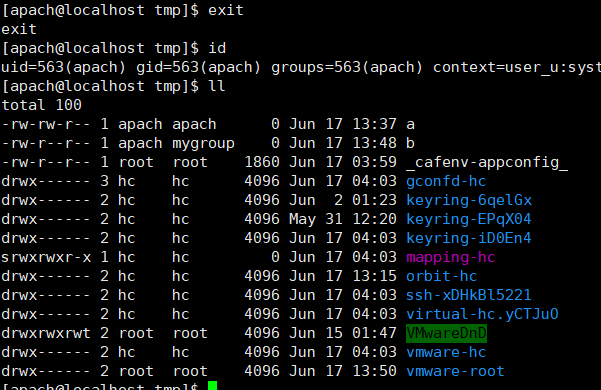


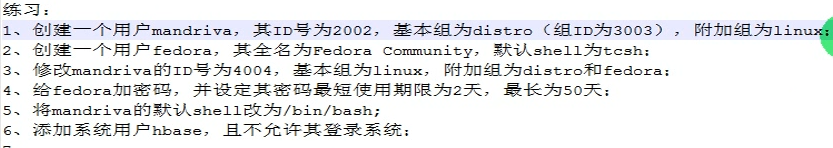
当切换组后，创建的文件的附加组就是切换后的组



### exit

如果退出当前的基本组和附加组使用exit





1. groupadd -g 3003 distro

groupadd –g 3004 linux

useradd –u 2002 –g distro –G linux mandriva

1. useradd –c ”Fedora Community” –s /bin/tcsh fedora
2. usermod –u 4004 –g linux –G distro,fedora mandriva
3. passwd –n 2 –x 50 fedora
4. chsh mandriva

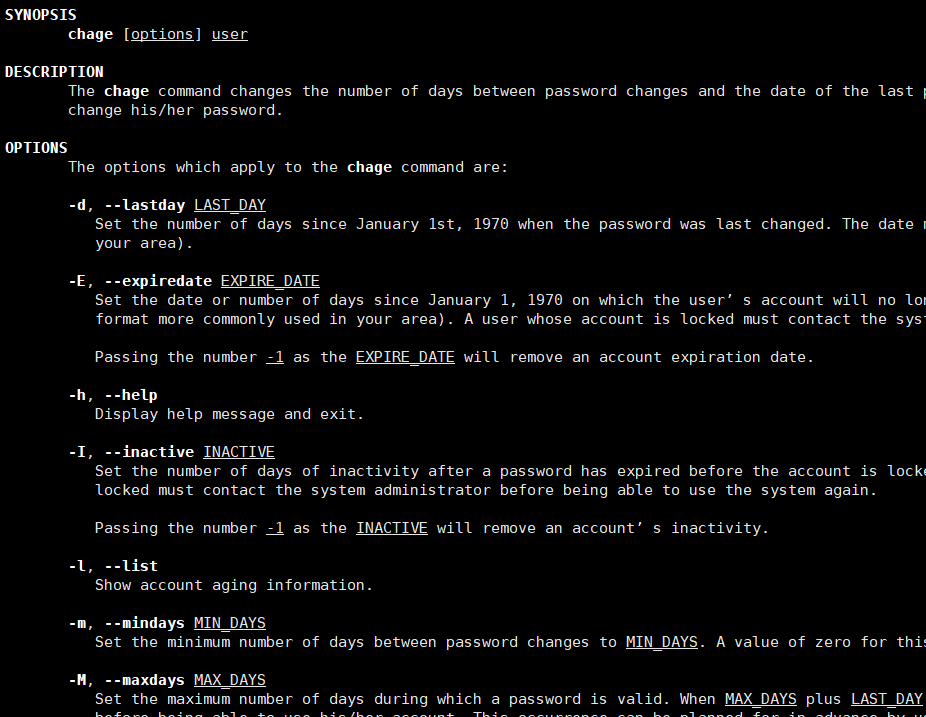
usermod –s /bin/bash mandriva

1. useradd –r –s /bin/nologin hbase



### chage

改变用户密码过期信息



-d 最近一次修改时间

-E 过期时间

-T 非活动时间

-m 最短使用期限

-M 最长使用期限

-w 警告时间

## 权限管理命令

三种权限：

**r** 对于文本文件可以读取内容，对于目录可以对此目录ls 但是不能 切换 ls -l

**w** 对于文本文件可以编辑和删除内容，对目录 可以在此目录中创建文件或目录

**x** 对文件可以执行,提交给内核，由内核为其创建一个进程副本，对于目录而言 可以对其使用cd 或者 ls –l 这样的命令来查看目录里面每个文件的详细的属性信息。

三类用户：

**u** 属主

**g** 属组

**o** 其他用户

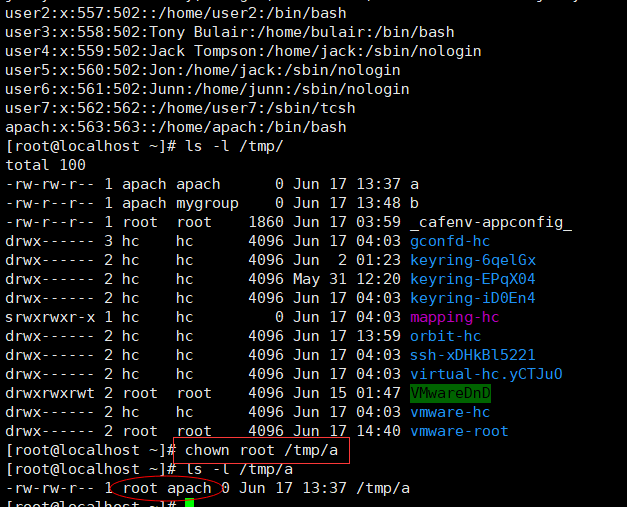
### chown

**只有管理员有这个权限**

**1.改变文件属主**



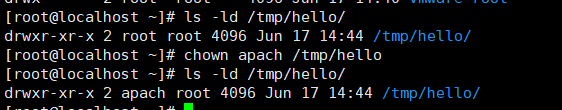
chown root /tmp/a



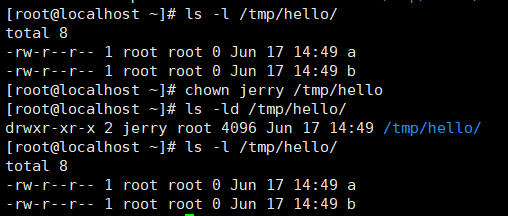
**2.改变目录的属组**

chown apach /tmp/hello

ls –l –d /tmp/hello 查看目录要使用-d 不然是该目录下的文件详情



修改目录，但是目录中的文件是权限没有变

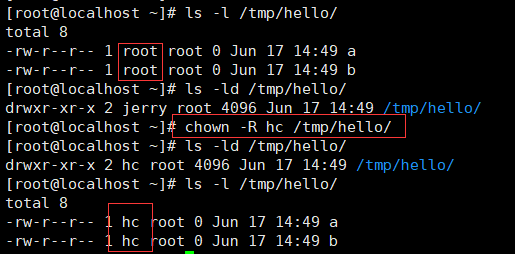


那么就使用

**-R**, --recursive (递归)

operate on files and directories recursivelyc hgrp

包括目录中的文件一起修改

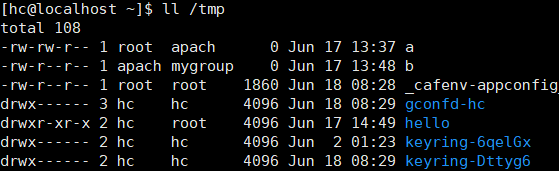


**--reference** = RFILE

use RFILE’s owner and group rather than the specifying OWNER:GROUP values

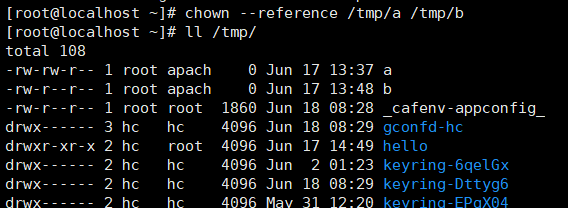
参考 tmp中a文件的权限来修改 test文件 reference 引用 参考

chown --reference =/tmp/a /tmp test



参考a文件的权限 修改b文件的权限

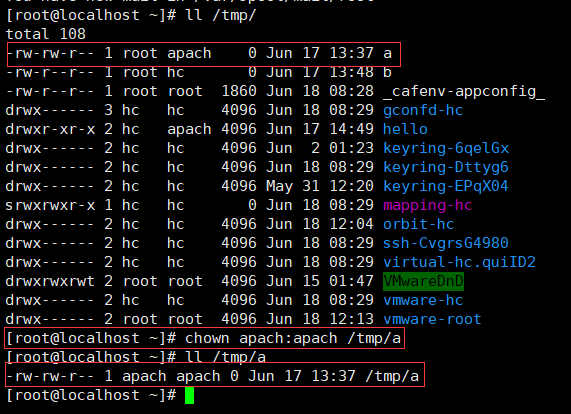
chown –reference /tmp/a /tmp/b



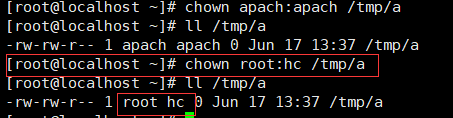
**chown 也可以同时修改属主和属组**

chown USERNAME:GRPNAME file = chown USERNAME.GRPNAME file

chown apach:apach /tmp/a

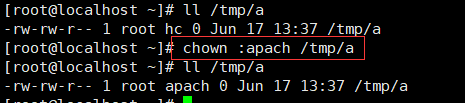


chown apach:apach /tmp/a



chown :GRPNAME file 只改变属组

chown :apach /tmp/a 只改变a的属组



**-R**

chown -R USERNAME:GRPNAME file 递归修改

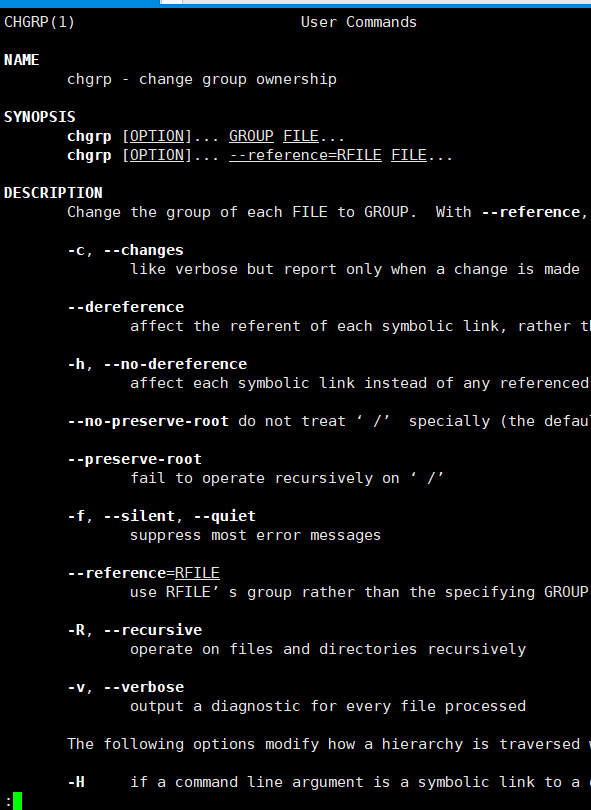
**--reference = /xxx/xxx/x**

chown –R –reference=/tmp/a /tmp/test

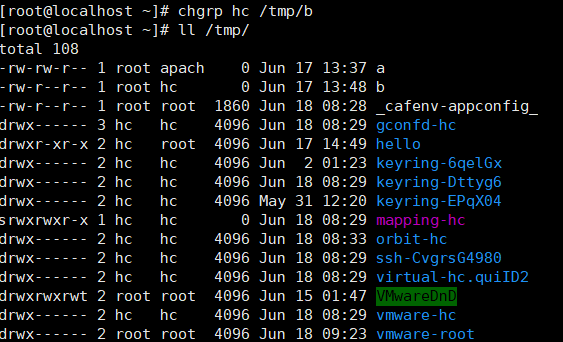
### chgrp

**同样只有管理员有这个权限**

修改文件的属组



**chgrp hc /tmp/b**

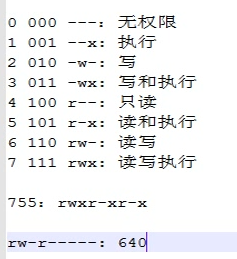


**--reference=RFILE**

### chmod

修改文件的权限



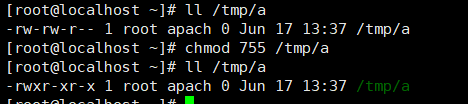


1. 修改三类用户的权限

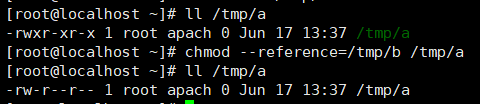
ugo 一起修改

chmod (644 775) file

chmod 75 file = chmod 075 file 缺少3位高位自动补零



**--reference 参考其他文件权限修改**

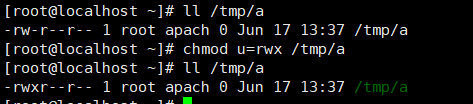


**-R 递归修改 目录和目录中的文件一起修改**

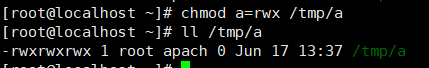
1. 修改某类用户或某些类用户的权限

u(属主) g (属组) o(其他) a(all 所有)

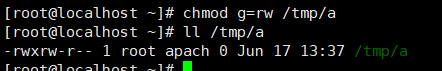
chmod u=rwx /tmp/a



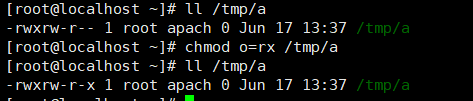
chmod a=rwx /tmp/a == chmod 777 /tmp/a 让三类用户有全部权限



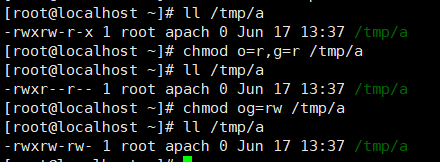
chmod g=rw /tmp/a == chmod g=rw- /tmp/a



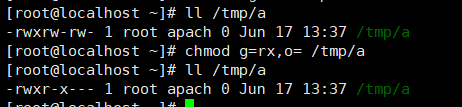
chmod o = rx /tmp/a == chmod o=r-x /tmp/a



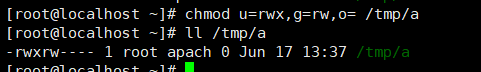
chmod o=r,g=r /tmp/a == chmod go=r /tmp/a



chmod g=rx,o= /tmp/a 不给值就是没有权限



chmod u=rwx,g=rw,o= /tmp/a == chmod u=rwx,g=rw /tmp/a

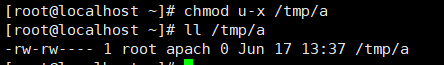


1. 修改某类用户的某位或者某些位权限

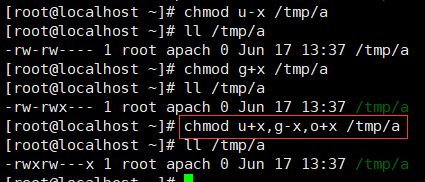
u g o a

chmod 用户类别 + | - file

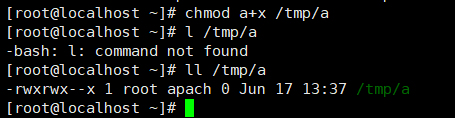
chmod u-x /tmp/a 去掉用户的执行权限



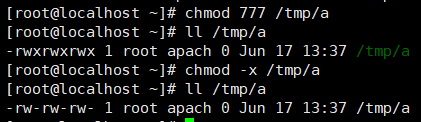
chmod u-x,g+x,o+x /tmp/a



chmod a+x /tmp/a 让三类用户都有执行权限

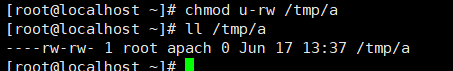


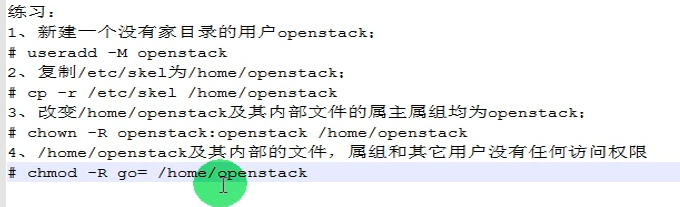
chmod a-x /tmp/a == chmod –x /tmp/a 去掉所有用户的执行权限，a 可以不写



可以操作多位权限

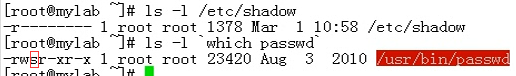
chmod u-rw /tmp/a





r------- 那么密码是怎么保存从呢？ 权限对管理员来说无效的，权限本来就是管理员来设定，那么对于普通用户修改自己的密码也可以的保存，这是为什么？

特殊权限位 s



当然不使用命令行添加用户也可以，手动添加用户，在

/etc/passwd

/etc/shadow

/etc/group

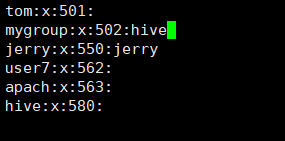
这3个文件中添加信息

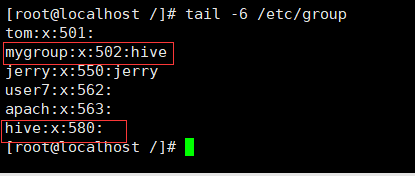
**需求**：手动添加用户hive(580),基本组位hive（580），附加组是mygroup

1. 先创建一个组 **/etc/group**

hive:x:580:

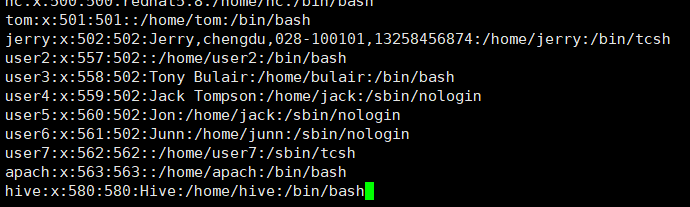
并在mygroup中添加附加组hive



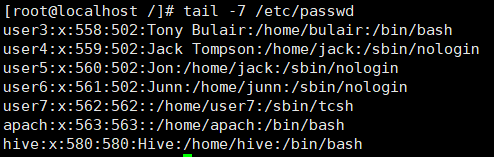


1. 创建一个用户 **/etc/passwd**

hive:x:580:580:Hive:/home/hive:/bin/bash



tail -7 /etc/passwd



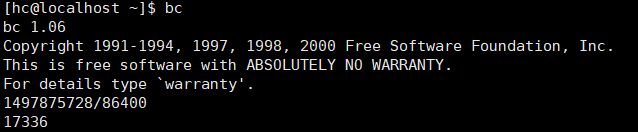
1. 添加密码 /etc/shadow

1970.1.1 到当前经过多少秒

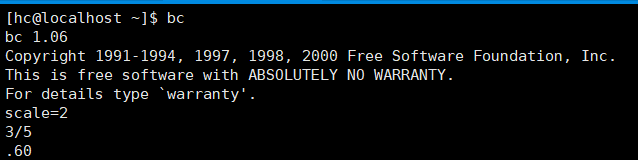
date +%s



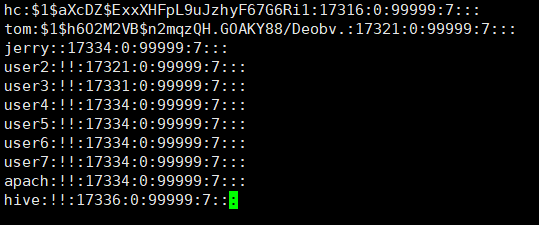
**bc** **Linux中的计算器** 可以算天数 1497875728/86400 （每天86400秒）=17336



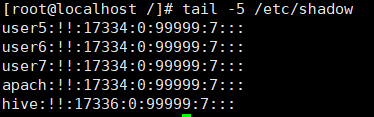
bc 默认是元正，需要指定小数位 scale =2



hive:!!:17336:0:99999:7:::

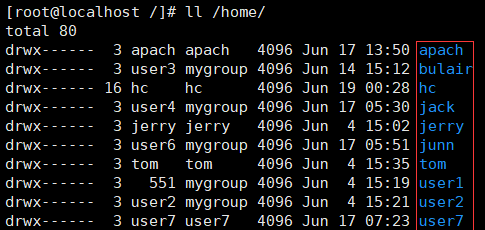


tail -5 /etc/shadow

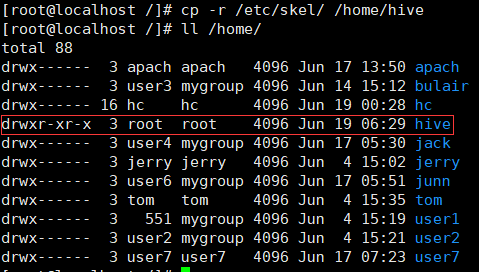


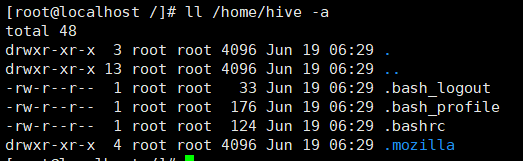
1. cp家目录并设置权限

现在并没有hive的家目录



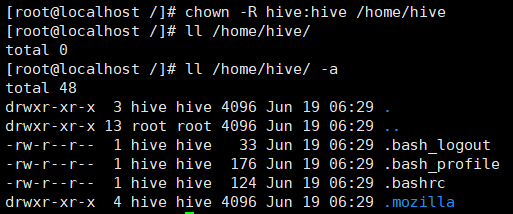
cp –r /etc/skel/ /home/hive 递归复制skel下的文件到/home/hive(如果没有自动创建)



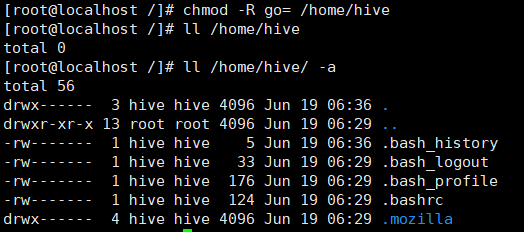


**chown –R hive.hive /home/hive**

递归设置hive家目录中的文件属主是hive，属组是hive组



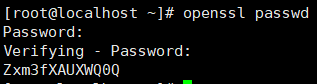
让g o 没有任何权限



一个用户就手动添加好了 **这时可以su 但是不能自己登陆，需要为其设置密码**

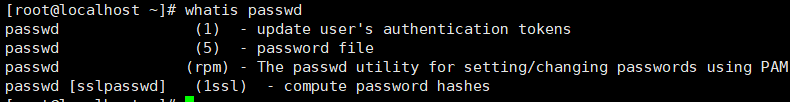
使用openssl加密

**openssl passwd**

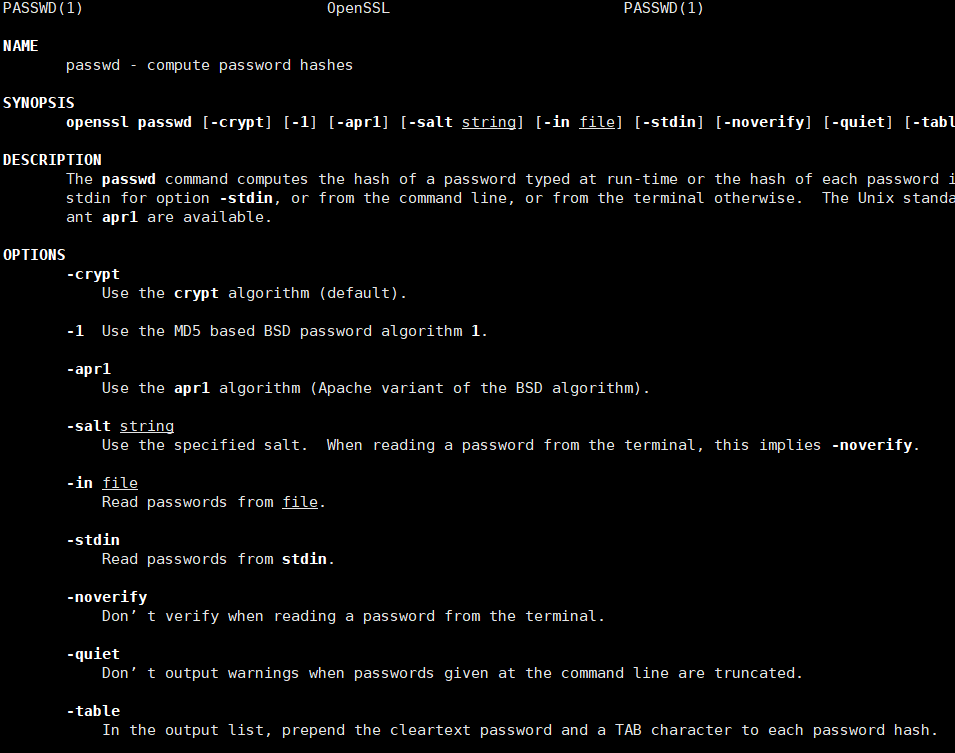


生成密码串，没有加盐

**man passwd**



**man sslpasswd**



**-1** **Use the MD5** based BSD password algorithm 1

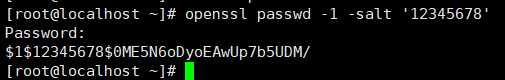
-1 不是字母l 表示使用md5加密

**-salt** Use the specified salt. When reading a password form the terminal,this implies

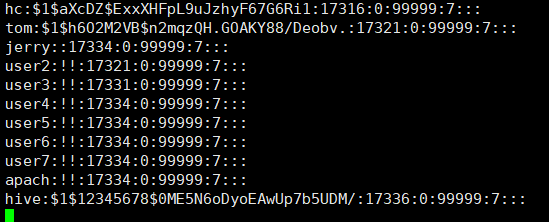
加点盐

openssl passwd -1 –salt ‘12345678’

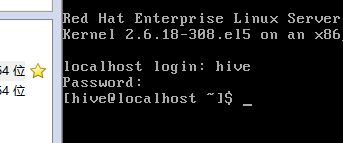
使用MD5加密，并加盐 ‘12345678’ 盐可以随便给几个，但是最好给8个



去掉两个！！ 将密码写入



用户就可以登录了



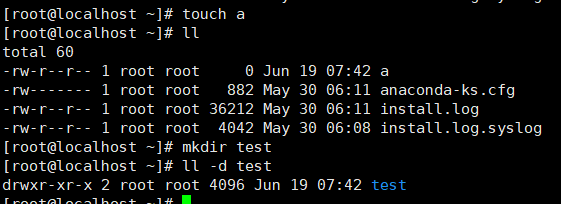
### umask

**umask ： 遮罩码或者反向掩码**

root 创建一个文件和目录

文件：rw-r--r-- a 644

目录：rwxr-xr-x test 755



普通用户创建一个文件和目录

文件：rw-rw-r-- a 664

目录：rwxrwxr-x test 775



两种用户创建的文件和目录不一样？

root 用户umask 022



hive用户umask 002



**创建文件规则：666-umask**

**创建目录规则：777-umask**

root:系统用户

666-022=644

777-022=755

hive:普通用户

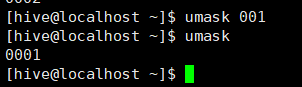
666-002=664

777-002=775

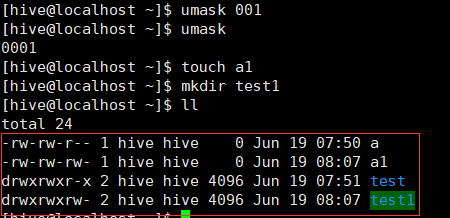
查看用umask

给用户设定umask 普通用户默认002

umask 001



创建文件和目录 权限又变了



umask 023

遵守一个原则： 文件默认不能有执行权限，如果有计算的结果有执行权限，自动加1

**umask决定了文件和目录的默认权限。**

如果我们创建的文件不想让其他用户没有任何权限，不能访问： **umask 027**

如果退出了当前登录，下次登录进入系统用户的umask还是**002**

站在用户的角度来说，shell类型

**登录式shell**

正常通过某终端登录

su - USERNAME

su –l USERNAME

**非登录式shell**

su USERNAME

图形终端下打开的命令窗口

自动执行的shell脚本

他们的区别：

bash的配置文件：

全局配置：默认的对所有用户都生效

**/etc/profile**

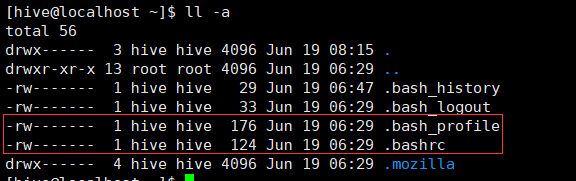
**/etc/profile.d/\*.sh**

**/etc/bashrc**

个人配置：用户自己修改，只对当前用户生效，如果全局和个人配置不一致，以个人的为准

**~/.bash\_profile**

**~/.bashrc**



**profile**类的文件：

**设定环境变量**

**运行命令或脚本**

**bashrc**类的文件：

**设定本地变量**

**定义命令别名**

**登录式shell如何读取配置文件的？**

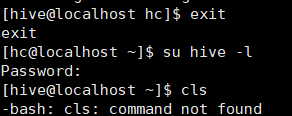
1. /etc/profile
2. /etc/profile.d/\*.sh
3. ~/.bash\_profile
4. ~/.bashrc
5. /etc/bashrc

**非登录式shell如何读取配置文件的？**

1. ~/.bashrc
2. /etc/bashrc
3. /etc/profile.d/\*.sh

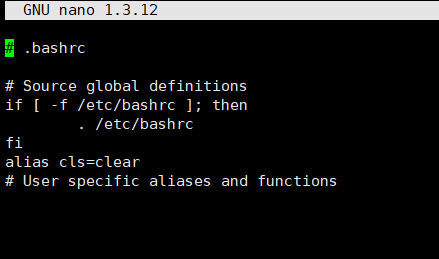
所以su 和 (su – su –l)是不一样的 半切换是不读取用户配置文件的

例如：我们使用alias定义 alias cls=clear，使用cls是有效的，但是我们logout登出系统再次进入系统的时候，使用cls命令就失效了



让这个别名的命令不失效，可以把这个 alias cls=clear 这个命令配置在用户的.bashrc的文件中，只对当前用户生效。如果要让这个命令对多有用户都有效，将这个命令配置在全局的bashrc中。

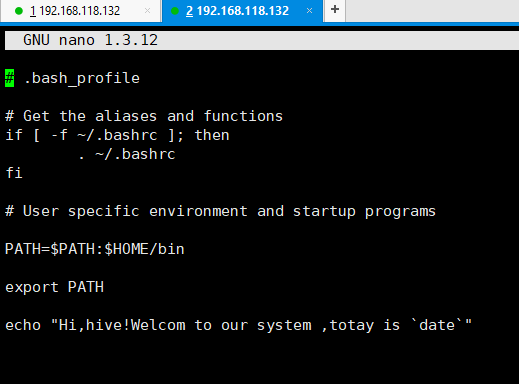
[hive@localhost ~]$ nano .bashrc



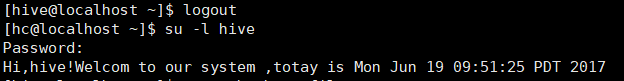
立马使用是没有效果的，只有logout后登录，cls 命令可以用来，不会失效。

登录后提示

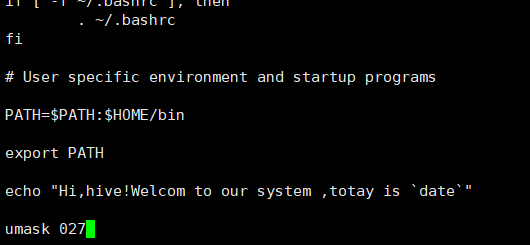
[hive@localhost ~]$ nano .bash\_profile



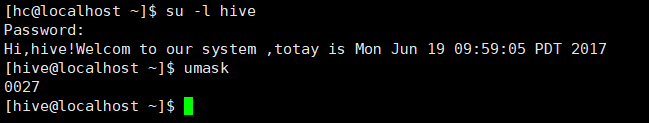
登入系统后



定义umask 027

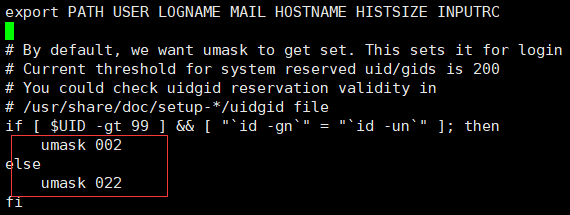


umask 027



为什么普通用户的umask是002

[root@localhost ~]# nano /etc/profile 查看全局的profile



bash. 脚本解释器