10. 用户管理

笔记本:	优秀笔记		
创建时间:	2018/12/24 15:46	**更新时间:**	2018/12/24 16:05
作者:	306798658@qq.com		

1：用户管理

用户配置文件：/etc/passwd

```
[root@www ~]# cat /etc/passwd  #有很多内容，这里只复制了几行做参考。
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

一行代表一个用户，每一行用冒号分隔为7段：
第一段：用户名
第二段：早期为用户的密码,后来因为安全问题将密码保存在/etc/shadow文件里
第三段：用户的uid
第四段：用户的gid
第五段：用户的注释描述信息，没有实际作用
第六段：为用户的家目录路径
第七段：用户登录的shell

密码配置文件：/etc/shadow

```
root:$6$QisNn8DZ$NWC.ZyydKOT7gghozYSx2BmAO/NZzwZp4UhdD1tYQJEInNe3Jx4dChnDRKEaB5PIqNqpkAkDtdXCU07j825TX.:17660:0:99999:7:::
bin:*:17110:0:99999:7:::
daemon:*:17110:0:99999:7:::
adm:*:17110:0:99999:7:::
```

跟/etc/passwd一一对应，每一行一个用户，用冒号分割为九段
第一段：用户名
第二段：用户密码，
    1.如果密码为星号，或者密码的第一位是星号，那么这个用户就是禁用状态，不可登录。
    2.如果为空，则此用户登录不需要密码。
    3.双叹号表示这个密码已经过期了
    4.6开头的，表明是用SHA-512加密；
    5.1 表明是用MD5加密；
    6.2 是用Blowfish加密；
    7.5 是用 SHA-256加密；
第三段：最近一次更改密码距离1970年1月1日总共多少天
第四段：表示多少天以后才可以更改密码, 默认为0 不限制
第五段：表示多少天内必须更改密码，否则密码过期，无法登录。
第六段：警告提醒，提示距离密码过期还有几天
第七段：不活动时间，表示的是用户没有登录活动但账号仍能保持有效的最大天数。
第八段：用户距离1970年多少天后过期
第九段：保留字段.

查看用户gid和uid：id命令

id命令可以查看用户的UID，GID，和附加组的GID，怎么添加附加组下面会写到。

```
[root@long01 ~]# id                     #不带参数直接使用id命令是查看当前用户
uid=0(root) gid=0(root) 组=0(root)      #括号里面的为用户名和组名
[root@long01 ~]# id user1               #带用户名参数可以查看指定用户。
uid=1001(user1) gid=1001(user1) 组=1001(user1)
```

-u选项：只显示用户的uid

```
[root@long01 ~]# id -u user1
1000
```

-g选项：只显示用户的gid

```
[root@long01 ~]# id -g user1
1000
```

-G选项：只显示用户的附加组的gid

```
[root@long01 ~]# id -G user1
1000
```

-n选项：配合-u、-g、-G选项使用，只显示显示用户名，所属组名，和附加组名。 #-n只能配合任意一个选项

```
[root@long01 ~]# id -un user1
user1
[root@long01 ~]# id -gn user1
user1
[root@long01 ~]# id -Gn user1
user1
```

添加用户：useradd命令

格式：useradd [选项] 用户名 #[中括号表示这一段内容可有可无。]

```
[root@long01 ~]# useradd  testuser1
[root@long01 ~]# id testuser1
uid=1002(testuser1) gid=1002(testuser1) 组=1002(testuser1)
```

-u选项：指定uid。 root用户的uid为0， CentOS7的普通用户UID是从1000开始使用

```
[root@long01 ~]# useradd  -u 1010  testuser2   #-u指定的UID必须大于1000，并且没有被使用。
[root@long01 ~]# id testuser2
uid=1010(testuser2) gid=1010(testuser2) 组=1010(testuser2)
```

-g选项：指定gid gid必须是已经存在的。

```
[root@long01 ~]# useradd -g 1002 testuser3
[root@long01 ~]# id testuser3
uid=1011(testuser3) gid=1002(testuser1) 组=1002(testuser1)   这里的GID就是1002了#
```

-b选项：指定用户的家目录

```
[root@long01 ~]# useradd -d /home/test2  testuser4
[root@long01 ~]# su - testuser4
[testuser4@long01 ~]$ pwd
/home/test2
```

-s选项：指定用户登陆的shell 一般不需要指定, 指定不能登陆的用户 为/sbin/nologin

```
[root@long01 ~]# useradd -s /sbin/nologin testuser5
[root@long01 ~]# su - testuser5
This account is currently not available.
```

-M选项：创建用户的同时不创建家目录。一个用户没有家目录会出现很多问题，所以一般只在创建不可登陆的用户时使用。

```
[root@long01 ~]# useradd -M testuser6
[root@long01 ~]# su - testuser6
su: 警告：无法更改到 /home/testuser6 目录：没有那个文件或目录
-bash-4.2$
```

删除用户：userdel

userdel [选项] user_name 删除用户
基本用法：直接删除一个用户，但是不会删除该用户的家目录

```
[root@long01 ~]# userdel testuser2
[root@long01 ~]# ls -dl /home/testuser2
drwx------ 2 1010 1010 62 5月  17 09:53 /home/testuser2
```

-r选项：删除用户的同时删除该用户的家目录

```
[root@long01 ~]# userdel -r testuser3
[root@long01 ~]# ls -dl /home/testuser3
ls: 无法访问/home/testuser3: 没有那个文件或目录
```

修改用户属性：usermod

命令格式：usermod 选项 user_name #注：使用usermod命令至少需要一个选项。
-u选项：更改用户的uid

```
[root@long01 ~]# usermod -u 1100 testuser4
[root@long01 ~]# id testuser4
uid=1100(testuser4) gid=1012(testuser4) 组=1012(testuser4)
```

-g选项：更改用户gid 可以用组名也可以用gid

```
[root@long01 ~]# usermod -g user1  testuser4
[root@long01 ~]# id testuser4
uid=1100(testuser4) gid=1001(user1) 组=1001(user1)
```

-d选项：更改用户的家目录，如果更改用户的家目录不存在，需要将用户原有的家目录重命名为更改的家目录。或者复制系统的家目录模板到指定的路径下。

/etc/skel/ 为系统的家目录配置模板

```
[root@long01 etc]# usermod -d /home/user4 testuser4
[root@long01 etc]# cp -r  /etc/skel/ /home/user4                #将目录复制到/home下改名为user4
[root@long01 etc]# chown -R testuser4:testuser4 /home/user4/    #修改属主和属组。
[root@long01 etc]# ls -al /home/user4
总用量 12
drwxr-xr-x   2 testuser4 testuser4  62 5月  18 13:56 .
drwxr-xr-x. 10 root      root      118 5月  18 13:56 ..
-rw-r--r--   1 testuser4 testuser4  18 5月  18 13:56 .bash_logout
-rw-r--r--   1 testuser4 testuser4 193 5月  18 13:56 .bash_profile
-rw-r--r--   1 testuser4 testuser4 231 5月  18 13:56 .bashrc
[root@long01 etc]# su - testuser4                               #通过以上操作才能正常切换到这个用户。
上一次登录：五 5月 18 13:52:21 CST 2018pts/0 上
[testuser4@long01 ~]$
```

-s选项：更改用户的shell，将testuser4更改为不能登录的shell

```
[root@long01 etc]# usermod -s /sbin/nologin testuser4
[root@long01 etc]# su - testuser4
This account is currently not available.
```

-G选项：扩展组,给用户指定多个组 格式:usermod -G grp1,grp2,grp3... user+name

```
给testuser4同时添加testuser5组和testuser6组
[root@long01 etc]# usermod -G testuser5,testuser6  testuser4
[root@long01 etc]# id testuser4
uid=1100(testuser4) gid=1001(user1) 组=1001(user1),1013(testuser5),1014(testuser6)
```

-L选项：锁定用户。
#注：锁定用户后，ROOT用户可以使用数su命令切换到该用户。
#但是，该用户无法直接登录，在普通用户下也不能使用su命令登录被锁定的用户。

```
[root@long01 ~]# usermod  -L testuser4
[root@long01 ~]#  cat /etc/shadow |grep testuser4
testuser4:!$6$RmQEV3Wz$kn6rFqEl276ihUdfMt9j9mvedBCp.8AQROmFPRtCqnKqdd8NhiOe2L1ntxdkHL.v.ugXXEiPL3nCOk23LA0g/0:17669:0:99999:7:::
```

#锁定后的用户在/etc/shadow文件中的密码最前面加上了一个感叹号。此时这个用户就无法登录了。

```
[root@long01 ~]# su - user1
上一次登录：五 5月 18 14:40:28 CST 2018pts/0 上
[user1@long01 ~]$ su - testuser4
密码：
su: 鉴定故障
```

-U选项：解锁用户

```
#将 testuser4 用户解锁：
[root@long01 ~]# usermod -U testuser4
[root@long01 ~]#  cat /etc/shadow |grep testuser4
testuser4:$6$RmQEV3Wz$kn6rFqEl276ihUdfMt9j9mvedBCp.8AQROmFPRtCqnKqdd8NhiOe2L1ntxdkHL.v.ugXXEiPL3nCOk23LA0g/0:17669:0:99999:7:::
#/etc/shadow文件中该用户的密码最前面的感叹号已经不在了， 这个用户就可以正常登录了。

[root@long01 ~]# su - user1                          #切换到普通用户user1.
上一次登录：五 5月 18 14:46:18 CST 2018pts/0 上
[user1@long01 ~]$ su - testuser4                     #切换到testuser4
密码：                                               #输入testuser4的密码
上一次登录：五 5月 18 14:40:33 CST 2018pts/0 上
最后一次失败的登录：五 5月 18 14:46:35 CST 2018pts/0 上
最有一次成功登录后有 1 次失败的登录尝试。
[testuser4@long01 ~]$                                #成功登录。
```

用户切换：su命令

切换用户命令格式：su [选项] [参数] #参数为指定的用户名

```
[root@long01 ~]# su user1    #直接切换到user1
[user1@long01 root]$ pwd     #切换后当前目录还是root。并且环境变量和用户的各种配置都不会加载。
/root
```

#使用su命令的时候 加上一个 “ - ” 符号。 可以加载要切换到用户的环境变量和配置等。

```
[user1@long01 ~]$ 登出
[root@long01 ~]# su - user1
上一次登录：五 5月 18 15:15:34 CST 2018pts/0 上
[user1@long01 ~]$ pwd
/home/user1
```

-c选项：使用指定用户执行一条命令，需要知道指定用户的密码。。
#命令格式： su -c 命令 指定用户　#如果命令中带有空格，需要用单引号或双引号括起来。

```
[user1@long01 ~]$ mkdir /etc/test/
mkdir: 无法创建目录"/etc/test/": 权限不够
[user1@long01 ~]$ su -c 'mkdir /etc/test/' root
密码：
[user1@long01 ~]$ ls -ld /etc/test/
drwxrwxr-x 2 root root 6 5月  18 15:30 /etc/test/
```

修改用户密码：passwd

passwd [username] 修改用户密码，不带username则修改当前用户密码，带username则修改username用户的密码（只有管理员可以这样操作）。

```
[user1@long01 ~]$ passwd
更改用户 user1 的密码 。
为 user1 更改 STRESS 密码。
（当前）UNIX 密码：                    #普通用户修改自己的密码需要输入当前的密码。
新的 密码：
无效的密码： 密码少于 8 个字符         #密码不能低于八位，也不能过于简单，而且只能重试三次。
新的 密码：
无效的密码： 密码少于 8 个字符
新的 密码：
无效的密码： 密码未通过字典检查 - 过于简单化/系统化 #
passwd：已经超出服务重试的最多次数
[user1@long01 ~]$ passwd
更改用户 user1 的密码 。
为 user1 更改 STRESS 密码。
（当前）UNIX 密码：                  #输入当前的密码
新的 密码：                          #输入新密码
重新输入新的 密码：                  #确认新密码
passwd：所有的身份验证令牌已经成功更新。
```

-l选项：锁定用户。效果跟usermod -L 一样

```
[root@long01 ~]# passwd -l user1
锁定用户 user1 的密码 。
passwd：操作成功
[root@long01 ~]# su - testuser4
上一次登录：五 5月 18 14:54:26 CST 2018pts/0 上
[testuser4@long01 ~]$ su - user1
密码：
su：鉴定故障
```

-u选项：解锁用户，效果跟usermod -U 一样

```
[root@long01 ~]# passwd -u user1
解锁用户 user1 的密码。
passwd：操作成功
[root@long01 ~]# su - testuser4
上一次登录：五 5月 18 16:02:43 CST 2018pts/0 上
[testuser4@long01 ~]$ su - user1
密码：
上一次登录：五 5月 18 16:01:17 CST 2018pts/0 上
最后一次失败的登录：五 5月 18 16:02:58 CST 2018pts/0 上
最有一次成功登录后有 1 次失败的登录尝试。
```

–stdin选项： 更改用户密码的另一种方式, 可以echo password| passwd –stdin user_name

```
#例：将user1的密码改为123456
[root@long01 ~]# echo "123456"|passwd --stdin user1
更改用户 user1 的密码 。
passwd：所有的身份验证令牌已经成功更新。
```

#也可以passwd --stdin user_name，然后输入一个密码，此方法会以明文显示密码

```
[root@long01 ~]# passwd --stdin user1
更改用户 user1 的密码 。
123456
passwd：所有的身份验证令牌已经成功更新。
```

2：用户组管理

用户组配置文件：/etc/group

```
[root@www ~]# cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
```

用户组的配置文件分为四段：
第一段：用户组的名称
第二段：用户组的密码，实际保存在/etc/gshadow文件中
第三段：用户组的GID。
第四段：用户组的成员。

/etc/gshadow 用户组的密码配置文件

```
此文件与/etc/group文件中的行一一对应，也有四段：
第一段：用户组名
第二段：加密码后的密码
第三段：组管理员(多个用,分隔)
第四段：组成员(多个用,分隔)
```

创建用户组：
命令格式：groupadd 选项 group_name

```
[root@long01 ~]# groupadd group1
[root@long01 ~]# tail -1 /etc/group
group1:x:1015:                              #查看刚才创建的用户组，gid为1015
```

-g选项：指定创建的用户组的gid

```
[root@long01 ~]# groupadd -g 1111 group2   #创建一个gid为1111的用户组
[root@long01 ~]# tail -1 /etc/group        #查看/etc/group文件的最后一行
group2:x:1111:
```

修改用户组：
groupmod 选项 组名 #用法和usermod类似

-g选项：#改变组的ID号

```
[root@long01 ~]# groupmod -g 1122 group1   #将用户组group1的gid改为1122
[root@long01 ~]# tail -2 /etc/group
group1:x:1122:
group2:x:1111:
```

-n选项：改变用户组组名

```
[root@long01 ~]# groupmod -n group3 group2
[root@long01 ~]# tail -2 /etc/group        #将用户组group2更改为group3
group1:x:1122:
group3:x:1111:
```

删除用户组
groupdel group_name 前提用户组里面没有任何成员.

```
[root@long01 ~]# tail -3 /etc/group
testuser6:x:1014:testuser4              #首先testuser6用户组有一个成员testuser4。
group1:x:1122:                          #group1和group3用户组都没有成员用户。
group3:x:1111:

[root@long01 ~]# groupdel testuser6
groupdel：不能移除用户“testuser6”的主组   #删除testuser6用户组失败。

[root@long01 ~]# groupdel group3
[root@long01 ~]# groupdel group1        #可以成功删除group1，group2
[root@long01 ~]# tail -3 /etc/group
testuser4:x:1012:
testuser5:x:1013:testuser4
testuser6:x:1014:testuser4
```

mkpasswd命令
mkpasswd可以生成一个随机的字符串。使用前需要安装该命令程序。

```
# yum install -y expect    #安装mkpasswd工具
```

mkpasswd [选项] 生成随机密码

```
[root@long01 ~]# mkpasswd   #不带选项默认生成9位的随机字符串。
zJHg!ao86
```

-l选项：指定生成的字符串长度

```
[root@long01 ~]# mkpasswd -l 12     #生成12位随机字符串
hKFqwk2=deh0
```

-s 指定特殊符号数量

```
[root@long01 ~]# mkpasswd -s 0 -l 12    #生成12位没有特殊符号的字符串。
oVbnot1rhwX5

[root@long01 ~]# mkpasswd -s 2 -l 12    #生成带有2个特殊符号的12位字符串。
7g,!yJh3jjYq
```

例子：使用一条命令生成随机字符串，保存这个字符串，使用这个字符串修改密码。

```
[root@long01 ~]# touch save_pass.info       #创建保存密码的文件。
[root@long01 ~]# chmod 600 save_pass.info   #将文件权限修改为600，防止其他用户查看与修改。

###下面这条命令中的‘&&’符号的功能是前面的命令执行成功才会执行后面的命令。
###‘&&’符号前面的命令是生成一个10位的字符串，并保存到sava_pass.info文件中
###‘&&’符号后面的命令是查看保存字符串文件的最后一行，用这一行字符串来修改user1的密码。
[root@long01 ~]# mkpasswd -s 2 -l 10 >> save_pass.info && tail -1 save_pass.info | passwd --stdin user1
更改用户 user1 的密码 。
passwd：所有的身份验证令牌已经成功更新。
```