

用户和组

一、用户:

是用来操作系统中不同功能的身份.

Linux系统中用户的分类

- 1) 超级用户: root uid=0
- 2) 系统用户: 不需要登录系统, 服务于应用程序, 维护系统的运行
bin nobody uid=1-499
- 3) 普通用户: 可以登录的用户 uid>=500

UID: 用户标识

添加用户: useradd username 创建的时候可以指定u、g、c等选项

查看当前用户的ID: id username

查看当前用户的名字: whoami

设置用户密码: passwd username (普通用户只能通过passwd修改自己本身密码)

删除一个用户: userdel username (-r 选项会连同创建的家目录一同删除)

创建组: groupadd groupname

删除组: groupdel groupname (用户的主要组不可以删除)

修改用户信息: usermod -[u,g,G等] username

修改组的信息: groupmod groupname

为组添加一个管理员: gpasswd -A username groupname

用组管理员为组添加一个用户成员: gpasswd -a username groupname

删除一个组成员: gpasswd -d username group

为组设定一个密码: gpasswd groupname

系统文件:

/etc/passwd: 用户信息

/etc/shadow: 用户密码相关的信息

useradd user1

- 1、会在/etc/passwd添加一行
- 2、会在/etc/group添加一行
- 3、会在/etc/shadow添加一行
- 4、会在/etc/gshadow添加一行
- 5、会在/home目录下建立一个与用户同名的目录作为家目录, 同时会从/etc/skel目录中复制出所有的隐藏文件到该目录
- 6、修改权限

-u 指定用户的uid

-g 指定用户的gid

-G 指定附属组, 可以有多个, 但是这些附属组必须是系统内已经存在的

-c 指定描述

-d 指定家目录

-M 不建立家目录

-s 指定shell

-e 指定用户过期时间, 日期

-f 指定用户过期时间, 天数

```
#useradd u1 -u 2000 -g 2000 -G 516 -d /u1 -c user1 -s /sbin/nologin
```

usermod

与useradd参数差不多, 一个是在创建用户时设定, 一个是用户创建好

-u -g -G -d -s
-L 锁定用户
-U 解除锁定
-l 改名
-a 添加用户到组，结合-G使用

passwd -l 锁定用户

-u 解锁用户
-n 密码最小使用期限 天数
-x 密码最大使用期限 天数
-w 密码过期前几天提醒 天数

whoami
who
who am i
w
users
id

finger username 查看用户的相关信息
chfn username 修改用户相关信息
userinfo username X window下修改用户相关信息

对用户的操作：新增、删除、修改、查看、设置密码

1、查看用户 id

```
# id //默认查看的是当前用户的信息
uid=0(root) gid=0(root) groups=0(root)
# whoami //查看当前用户是谁
root
# id lp //查看指定用户的信息
uid=4(lp) gid=7(lp) groups=7(lp)
uid: 用户标识
gid: 组标识, 用户的主组
groups: 用户所属组信息
```

2、添加用户 useradd

注意：添加用户的操作只有root能操作

```
# useradd niulang
```

反例：

```
# su - niulang // switch user切换用户
```

```
$ useradd zhinv
```

-bash: /usr/sbin/useradd: Permission denied 权限被拒绝，普通用户是无法添加用户的

```
$ exit
```

```
logout
```

```
# id niulang
```

```
uid=500(niulang) gid=500(niulang) groups=500(niulang)
```

添加用户时，如果不指定用户的组，那么默认会创建一个与用户名同名的组作为用户的主要组。

1) 添加用户时指定选项 man useradd

-c: 注释性的描述信息

-d: 指定用户的家目录

默认情况下：

root的家目录是/root

普通用户的家目录是/home

-g: 指定用户的主要组, 前提: 组是必须存在的

-G: 指定用户的附加组(附属组), 前提: 组是必须存在的

-s: 指定用户的登录shell

-u: 指定用户的uid

head -1 /etc/passwd

root:x:0:0:root:/root:/bin/bash

用户名:密码:uid:gid:注释性描述:用户的家目录:用户的登录shell

-u -g -c -d -s

echo \$SHELL //查看当前shell, echo可以用来打印变量

/bin/bash

要求区分的两种shell

/bin/bash: 可登录shell, 如果用户的shell是这个, 那么该用户可以登录到系统中

/sbin/nologin: 非登录shell, 如果用户的shell是这个, 那么该用户是不允许登录到系统的

格式: useradd 选项 选项的参数 ... 用户名

例子:

useradd -u 600 -s /sbin/nologin zhinv

id zhinv

uid=600(zhinv) gid=600(zhinv) groups=600(zhinv)

tail -1 /etc/passwd

zhinv:x:600:600::/home/zhinv:/sbin/nologin

useradd -g 502 wangmu 错的

useradd: group '502' does not exist

useradd -g zhinv wangmu

id wangmu

uid=601(wangmu) gid=600(zhinv) groups=600(zhinv)

3、删除用户 userdel

userdel wangmu //删除的时候不会删除用户的家目录和邮箱

useradd wangmu //能添加, 但是会有提示消息

useradd: warning: the home directory already exists.

Not copying any file from skel directory into it.

Creating mailbox file: File exists

-r: 可以干净的删除一个用户

userdel -r wangmu

4、给用户设置密码

passwd:

超级用户: 既可以修改自己的密码, 又可以修改别人的密码, 并且不需要提供当前密码

普通用户: 只能修改自己的密码, 要求严格, 并且需要提供当前密码

passwd // 默认修改当前用户的密码

Changing password for user root.

New password:

BAD PASSWORD: it is based on a dictionary word

BAD PASSWORD: is too simple

Retype new password:

passwd: all authentication tokens updated successfully.

passwd niulang //修改指定用户的密码

Changing password for user niulang.

New password:

BAD PASSWORD: it is too short

BAD PASSWORD: is too simple

Retype new password:

passwd: all authentication tokens updated successfully.

反例:

```
# su - niulang
```

```
$ passwd zhinv
```

passwd: Only root can specify a user name.

普通用户修改自己的密码，需要当前密码，而且长度最少6位，而且不能是字典单词

```
$ passwd
```

Changing password for user niulang.

Changing password for niulang.

(current) UNIX password:

New password:

BAD PASSWORD: it is based on a dictionary word

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

/etc/shadow: 存放用户密码信息的文件

```
# head -1 /etc/shadow
```

```
root:$6$9A0mQbxm$3X1qbP0srKNql7jCXpmwhyNxGTHmlbT65zF.:16276:0:99999:7:::
```

用户名:加密后的密码:....

\$6\$: 加密算法是sha512

\$1\$: 加密算法是md5

5、修改用户：常用

usermod: //modify: 修改

-G: 修改用户的附属组，不管用户原来的附属组如何，直接覆盖

-g: 修改用户的主要组

-a: 一般和-G联合使用，追加附属组

-u: 修改用户的uid

-d: 修改用户的家目录

-s: 修改用户的shell

```
# useradd -G niulang,zhinv sha
```

```
# id sha
```

```
uid=601(sha) gid=601(sha) groups=601(sha),500(niulang),600(zhinv)
```

```
# useradd sha2
```

```
# usermod -G sha2 sha
```

```
# id sha
```

```
uid=601(sha) gid=601(sha) groups=601(sha),602(sha2)
```

```
# usermod -a -G niulang sha
```

```
# id sha
```

```
uid=601(sha) gid=601(sha) groups=601(sha),500(niulang),602(sha2)
```

练习:

1) 添加三个用户usr1, usr2, usr3

```
# useradd usr1
```

```
# useradd usr2
```

```
# useradd usr3
```

2) 添加用户usr4，同时指定其主组为usr1，uid为2014，并验证用户创建是否正确

```
# useradd -g usr1 -u 2014 usr4
```

```
# id usr4
```

3) 添加用户usr5，家目录为/tmp/usr5，附加组为usr2, usr3

```

# useradd -d /tmp/usr5 -G usr2,usr3 usr5
4) 修改usr4的附加组为usr2, usr3, 注意验证
# usermod -G usr2,usr3 usr4
# id usr4
5) 修改usr1的shell为非登录shell
# usermod -s /sbin/nologin usr1
# grep usr1 /etc/passwd //仅过滤出文件中含有usr1的行
6) 彻底删除usr5用户
# userdel -r usr5
7) 为用户usr1设置密码redhat
# passwd usr1

```

二、组 group

用户和组的关系：组中可以没有用户、也可以有一个或者多个用户。用户只有一个主要组，可以同时属于多个组，但是除了主组以外的其他组称为附属组。

组的分类：从用户的角度分的，分为：

主组：primary group、initial group、first group，用户的默认组，也就是gid所标识的组

附属组：secondary group、supplementary group

1、添加组 groupadd

```

# groupadd g1
-g: 添加组时，指定gid是多少
# groupadd -g 800 g2
# tail -2 /etc/group
g1:x:603:
g2:x:800:
组名:密码:gid:组里成员

```

2、修改组：groupmod

```

# groupmod -g 888 g2
# grep g2 /etc/group
g2:x:888:
修改组的名字： -n 新名字
# groupmod -g 588 -n group1 g1
# grep gr /etc/group
group1:x:588:

```

3、删除组：groupdel

注意：不能够删除用户的主要组，能够删除用户的附属组

```

# groupdel group1
# grep group1 /etc/group //没有任何输出，表示文件中没有含有group1关键字的行
#

```

小实验：

```

# useradd t1
# useradd t2
# id t1
uid=603(t1) gid=603(t1) groups=603(t1)
# id t2
uid=604(t2) gid=604(t2) groups=604(t2)
# tail -2 /etc/group
t1:x:603:
t2:x:604:
# groupdel t1 //t1组是t1用户的主要组，不能删除

```

```

groupdel: cannot remove the primary group of user 't1'
# usermod -g t2 t1
# usermod -G t1 t2
# id t1
uid=603(t1) gid=604(t2) groups=604(t2)
# id t2
uid=604(t2) gid=604(t2) groups=604(t2),603(t1)
# groupdel t1 //t1组仅仅是t2用户的附属组，可以被删除
# grep t1 /etc/group

```

4、组管理 gpasswd

管理/etc/group和/etc/gshadow文件的，每一个组可以有管理员，成员和密码
给组加密码：可以让知道组密码的人，临时的切换到该组中，能够使用该组的资源。

1) 指定组的管理员列表

```

# groupadd groups
# useradd u1
# gpasswd -A u1 groups //u1用户就是groups组的管理员了

```

2) 向组中添加成员

```

# su - u1
$ gpasswd -a sha groups
Adding user sha to group groups
# tail /etc/group
groups:x:889:sha

```

3) 给组设置密码

```

# gpasswd groups
Changing the password for group groups
New Password:
Re-enter new password:

```

4) 切换组

```

# su - niulang
$ id
uid=500(niulang) gid=500(niulang) groups=500(niulang)
$ newgrp groups //临时切换组
Password:
$ id
uid=500(niulang) gid=889(groups) groups=500(niulang),889(groups)

```

5) 将用户从组中删除

```

# gpasswd -d sha groups
Removing user sha from group groups

```

/etc/passwd: 保存用户信息的文件

/etc/shadow: 保存用户密码相关信息的文件

/etc/group: 保存组信息的文件

/etc/gshadow: 保存组密码相关信息的文件

1、/etc/passwd: *****

```
# man 5 passwd
```

```
# head -1 /etc/passwd
root:x:0:0:root:/root:/bin/bash
用户名:密码占位符:UID:GID:描述信息:家目录:登录shell
```

2、/etc/shadow

```
# useradd zhangsan
用户有无密码两种情况:
# tail -1 /etc/shadow.
zhangsan:!!:16276:0:99999:7:::
# passwd zhangsan
Changing password for user zhangsan.
New password:
BAD PASSWORD: it is WAY too short
BAD PASSWORD: is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
# tail -1 /etc/shadow
```

```
zhangsan:$6$/23ax9CR$IDSfqNF3gZqDq4L0.Bdqd19SLbzBa6CuSpuOn.9YrFmSWYy09xPGa47SVEsQXfQJDybUD0Zsu4vG.FV
KMT91l/:16276:0:99999:7:::
```

```
# man 5 shadow
```

```
zhangsan:!!:16276:0:99999:7:::
```

- 1) 用户名
- 2) 加密后的密码
- 3) 上一次密码修改时间(单位是天, 是距离1970年1月1日多少天)
- 4) 密码最小生存周期 单位: 天 0表示随时可以修改密码; 2表示两天内不能修改
- 5) 密码最大生存周期 单位: 天, 密码使用的最长时间
- 6) 密码到期前几天开始警告
- 7) 密码过期之后到账户失效之前的时间限制 密码过期之后还可以用几天
- 8) 账户过期时间
- 9) 保留列

查看用户的密码相关的时间

```
# chage -l root
Last password change          : Jul 25, 2014
Password expires               : never
Password inactive              : never
Account expires                : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
# chage -d 0 用户名 用户下次登录必须更改密码
chage -m 2 u1 修改密码最少使用天数
chage -M 50 u1 修改密码最大使用天数 (密码过期时间)
chage -I 5 u1 密码过期后到账户失效的天数 (密码过期后还能使用的天数)
chage -E "2019-3-5" u1 修改的是账户的过期时间
```

3、/etc/group

```
# head -1 /etc/group
root:x:0:
组名:密码占位符:GID:用户列表
```

4、/etc/gshadow

```
# head -1 /etc/gshadow
root:::
组名: 加密后的密码:组管理员:组成员
```

/etc/login.defs 默认创建新用户的配置文件

和用户登录相关的文件或目录

/etc/profile

此文件为系统的每个用户设置环境信息,当用户第一次登录时,该文件被执行. 并从/etc/profile.d目录的配置文件中搜集shell的设置

/etc/bashrc

为每一个运行bash shell的用户执行此文件.当bash shell被打开时,该文件被读取

~username/.bashrc

每个用户都可使用该文件输入专用于自己使用的shell信息,当用户登录时,该文件仅仅执行一次!默认情况下,他设置一些环境变量,执行用户的.bashrc文件

~username/.bash_profile

该文件包含专用于你的bash shell的bash信息,当登录时以及每次打开新的shell时,该文件被读取。

/etc/profile.d/*

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

站在用户登录的角度来说, SHELL的类型:

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(使用su切换用户, 不带任何参数)

GUI下打开命令窗口

自动执行的shell脚本

登录式shell如何读取配置文件:

/etc/profile -->/etc/profile.d/*.sh --> ~/.bash_profile --> ~/.bashrc -->/etc/bashrc

非登录式shell如何读取配置文件:

~/.bashrc -->/etc/bashrc -->/etc/profile.d/*.sh

su - : 完全登录shell, 切换后用户拥有自己的一套环境变量

su: 不完全登录shell, 切换后用户使用的是其他用户的环境变量

和新创建用户相关的文件: /etc/login.defs /etc/default/useradd

/etc/rc.local 是系统开机后执行的最后一个脚本, 一般被管理员执行一些系统开机运行的命令