# 一、Linux用户、用户组概念

Linux系统是一个多用户多任务的分时操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统。

每个用户账号都拥有一个唯一的用户名和各自的口令。用户在登录时键入正确的用户名和口令后，就能够进入系统和自己的主目录。

计算机生成了可选文字:
Group 

|  |  |
| --- | --- |
| 用户 | 一个户口本中的每一个用户 |
| 用户组 | 户口本中的所有成员组成的组 |
| 其他人 | 除用户本人(户主)和用户组(户口本中的成员)外的成员都是其他人 |

每一个用户都可以拥有多个用户组。

每一个用户组都可以容纳多个用户。

## 1.用户及UID：

在Linux系统中，每一个用户默认都对应了一个UID，而这个UID可以理解是用户的身份证号。

CentOS系统中：

|  |  |
| --- | --- |
| UID 0 | root |
| UID 1-499 | 系统预留，作为系统用户来使用 |
| UID 500 - 65535 | 自定义账户 (注意，此处仅指的是CentOS6.5系统，其他系统的自定义账户则不一定是从500开始，例如Ubuntu系统是从1000开始。) -u指定id 超过65535也可以 |

## 2.用户信息存储的位置

|  |  |
| --- | --- |
| 用户信息 | /etc/passwd |
| 密码信息 | /etc/shadow |
| 用户组信息 | /etc/group |

passwd、shadows、group这三个文件不要随意的进行修改，如果想要进行文件内容的查看，建议大家copy出一份，对copy的文件进行修改。或把文件copy出来之后，进行保存，再去修改原文件。这样做即便是写错了文件也可以通过单用户的方式将文件恢复。

# 二、Linux用户账号管理

就是对passwd、shadow、group三个文件的增删改。增加一个用户时会赋予，UID，账户名、家目录以及shell程序。  
 新建的用户，默认的情况下是无法登录的。

创建一个用户的时候，默认情况下，会为其创建一个同名的用户组

## 1.创建用户(useradd)

语法：

useradd [选项] {UserName}

选项：

|  |  |
| --- | --- |
| -d 目录 | 指定用户家目录 |
| -g 用户组 | 指定用户所属的主用户组。 |
| -G 用户组,用户组 | 指定用户所属的附加组。 |
| -c comment | 指定一段注释性描述 |
| -s Shell程序 | 指定用户的登录Shell。有一个特殊的shell程序叫nologin，如果某帐号使用此shell，就表示用户不允许登录，该用户通常成为伪用户。最常用的/bin/bash。 其它/sbin/nologin，/bin/sh |
| -u 用户号 | 指定用户的用户号 |

### 1.1.创建普通用户

创建一个普通用户，使用默认值即可：

# bash

useradd tom

说明：该账户被创建完之后，会具备的信息

UID：由于是当前系统的第一个账户，那么它的UID是500。

家目录(主目录)：因为是默认创建，所以它的家目录所在位置是/home/tom

用户组：创建用户的同时，会为其创建一个同名的用户组，tom组

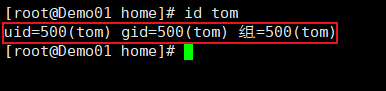
shell程序：默认会是bash

### 1.2.查看用户信息(id)

查看用户的信息包括UID以及所属的组及附加组

#bash

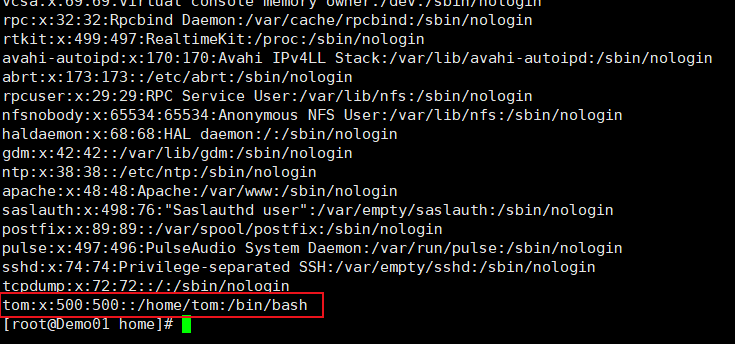
id tom



除了通过以上指令看到用户信息，还可以通过查看/etc/passwd文件下的信息获取

#bash

cat /etc/passwd



### 1.2.创建用户指定目录和附加组(-d -g -G)

如果需要指定用户在哪个文件夹生成home文件则需要先把父文件夹先建出来。例如/tedu/jerry就要先把tedu文件夹建立出来。

创建一个普通用户，其家目录是在/tedu/jerry，主组root,附加组tom,adm：

# bash

mkdir /tedu

useradd -d /tedu/jerry -g root -G tom,adm jerry

### 1.3.创建用户指定UID及shell程序(-s -u)

指定用户的shell程序为/bin/sh，UID为80000：

# bash

useradd -s /bin/sh -u 80000 test1

说明：虽然CentOS系统中默认的UID是65536个(0-65535)，但是可以手动指定它的UID超出此范围。

测试再创建账户是UID会不会延续上案例的UID。

# bash

useradd test2

说明：正常来应该按照上次创建用户的UID往后+1。但是80000已超过系统默认的范围，所以不会根据超出范围后UID进行+1延续。

### 1.4.创建伪用户

创一个nologin的用户，此用户是无法进行登录的。

# bash

useradd -s /sbin/nologin test3

## 2.修改用户(usermod)

语法：

usermod [选项] {UserName}

选项：

|  |  |
| --- | --- |
| -c comment | 指定一段注释性描述 |
| -d | 目录 指定用户主目录，如果目录不存在，则同时使用-m选项，可以创建主目录。 |
| -g 用户组 | 指定用户所属的用户组。 |
| -G 用户组,用户组 | 指定用户所属的附加组。如果想要在原有的基础上追加附加组，使用-aG的方式。 |
| -s Shell文件 | 指定用户的登录Shell。有一个特殊的shell程序叫nologin，如果某帐号使用此shell，就表示用户不允许登录，该用户通常成为伪用户。 |
| -u 用户号 | 指定用户的UID |

### 2.1.修改test账户的UID(-u)

# bash

usermod -u 9000 test1

### 2.2.修改账户的附加组为test、test1(-G)

# bash

usermod -G test,test1 jerry

### 2.3.为账户追加附加组件root、adm(-aG)

# bash

usermod -aG root,adm jerry

### 2.4.将伪用户状态修改为正常普通用户(-s)

# bash

usermod -s /bin/bash test3

说明：test5之前的shell程序是/sbin/nologin

## 3.删除用户(userdel)

语法：

userdel [选项] {UserName}

选项：

|  |  |
| --- | --- |
| -r | 删除账户的同时删除该账户的家目录 |

### 3.1.删除用户

# bash

userdel test2

说明：该命令执行完成之后，在home目录下依然保留了zhangsan的家目录。

### 3.2.删除用户及家目录(-r)

userdel -r test3

说明：该命令之后之后，home目录中的lisi目录会被一并删除，且不保留它的工作文件。一般来说，公司中删除账户的操作比较少，就算删除账户时，一般也不用-r的选项，因为人虽然离职，但是有可能此人还会回来继续任职。

如果员工离职，完全没有必要删除账户，可以使账户无法登录即可，比如nologin，锁定账户，删除密码等手段都是可以防止资料泄漏。

注：Windows和Linux系统(其他系统没测试)，没有密码的账户都是不允许通过远程的方式进行访问的。

## 4.用户密码管理(passwd)

语法：

passwd [选项] {UserName}

选项：

|  |  |
| --- | --- |
| -l | 锁定账户 |
| -u | 解锁账户 |
| -d | 删除密码 |

说明：

passwd 可以不跟选项、用户名，默认是修改自己的帐号密码。

修改他人密码，必须具备管理员权限(并不一定非要是root账户)

普通账户，只能修改自己的密码。

管理员修改他人的密码，不需要满足密码策略。

普通账户修改密码时，必须满足密码安全策略。-->as12AS!@ zx12ZX!@

管理员修改他人密码：

passwd {UserName}

修改自己的密码

Passwd

### 4.1.为test2用户添加密码

#bash

passwd test2

### 4.2.锁定test2账户(-l)

# bash

passwd -l test2

### 4.3.解锁test2账户(-u)

# bash

passwd -u test2

### 4.4.删除test2账户密码(-d)

# bash

passwd -d test2

案例中，锁定账户和删除账户密码都是可以达到不允许账户远程登录的效果。

## 5.用户身份切换(su)

|  |  |
| --- | --- |
| su {UserName} | 表示切换用户之后，依然停留在当前目录 |
| su - {UserName} | 表示切换用户之后，去到该用户的家目录 |

普通用户切换root时，书写格式不需要写成su - root，直接su即可。

su与账户名之间加不加"-"区别在于路径。

### 5.1.切换到用户test2当前目录下

当前位置，在/root下：

# bash

su test2

该命令执行过后，切换到zhangsan账户，但是路径依然停留在/root目录下

计算机生成了可选文字:
[root尽localhost一］#pwd
/root
[root尽localhost一］#suzhangsan
[zhangsan@localhostroot」＄pwd
/root
[zhangsan尽localhostroot」＄

### 5.2.切换到用户test2家目录下(-)

# bash

su - zhangsan

该命令执行过后，切换到zhangsan账户，同时会去到zhangsan账户的家目录下。

计算机生成了可选文字:
[rootOlocalhost一』＃
[zhangsan尽localhost
/home/zhangsan
[zhangsan尽localhost
一zhangsan
pwd
晚甲迁甲
Ul..Jl..J
S}}

## 6.用户shell程序的简单介绍

/bin/sh和/bin/bash都是Linux的shell程序，但是表现的形式不同，具体执行功能是一样的，可以理解为bash是sh的升级版，兼容版本。

|  |  |
| --- | --- |
| /bin/sh | -sh-4.1$ |
| /bin/bash | [root@localhost tedu]# |

从上面的内容可以看得出来，bash确实是比sh提供的内容更多，更为详细。

# 三、Linux用户组管理

## 1.创建用户组(groupadd)

语法：

groupadd [选项] {groupName}

选项：

|  |  |
| --- | --- |
| -g GID | 指定新用户组的GID |
| -o | 通常与-g同时使用，使新用户组可以与系统已有的组ID相同。 |

系统底层会将两个GID相同用户组识别为同一个用户组，这样做的目的是让两个用户组的权限相同。识别规则，**后者遵循前者**。

### 1.1.新建group1组

#bash

groupadd group1

说明：此命令向系统中增加了一个新组group1，新组的组标识号，在当前已有的最大组标识号的基础上加1

### 1.2.新建group2组指定GID(-g)

# bash

groupadd -g 101 group2

说明：此命令向系统中增加了一个新组group2，同时指定新组的组标识号是101。

### 1.3.为Tom用户添加group1为附加组

创建一个普通的用户组，并将该组添加为tom账户的附加组

# bash

usermod -aG group1 tom

### 1.4.创建group3与group2的GID相同的用户组(-g -o)

由于group3的GID与group2的GID相同，所用两个组的权限是相同的，**后者遵循前者**。

# bash

groupadd -g 101 -o group3

## 2.用户组修改(groupmod)

语法：

groupmod [选项] {groupName}

选项：

|  |  |
| --- | --- |
| -g GID | 指定新用户组的GID |
| -o | 通常与-g同时使用，使新用户组可以与系统已有的组ID系统。 |
| -n | 用来修改组名 group -n newGroupName oldGroupName |

### 2.1.将组group2的组标识号修改为102(-g)

# groupmod -g 102 group2

### 2.3.新建一个group4，并将其组名修改成bigdata(-n)

# bash

groupadd group4

groupmod -n bigdata group4

## 3.用户组删除(groupdel)

语法：

groupdel {groupName}

### 3.1.删除bigdata用户组

# bash

groupdel bigdata

### 3.2.针对用户删除用户的附加组

gpasswd -d 用户名 组名

注意：

如果删除的用户组，已经被用户追加为附加组，对应的所有用户的该附加组会被撤销掉。

如果被删除的用户组，已经被用户指定为主组，则该用户组无法被删除。(可以理解为像Windows中文件被占用时不能被删除。)

## 4.用户组的切换(newgrp)

某用户属于多个用户组时，想要访问其他用户组中的内容时，必须切换用户组才行。前提是该账户确实是拥有多个用户组。

 newgrp {GroupName}

### 4.1.使用jerry以test1的组新建文件

切换为用户jerry在自己的主目录下使用以test2组新建一个自定义名的文件，新建之后这个文件将讲会看到权限用户为jerry，权限组为test2。

#bash

su jerry

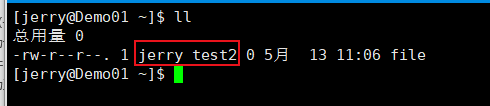
cd ~

newgrp test2

touch file

ll

可以看到文件的使用者是jerry，属组是test2。



## 5.查看用户组下有什么用户(grep '{组名}')

grep '{组名}' /etc/group

### 5.1.查看group1下有多少个用户

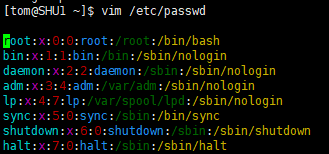
由于是基于查看etc/group文件进行工作的，所以此命令对权限有要求。

grep group1/etc/group



# 四、passwd、shadow、group文件详解

## 1.passwd



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 账户名 | 密码 | 用户id | 主组id | 账户的描述 | 家目录 | Shell程序 |
| root: | x: | 0: | 0: | root: | /root: | /bin/bash: |

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

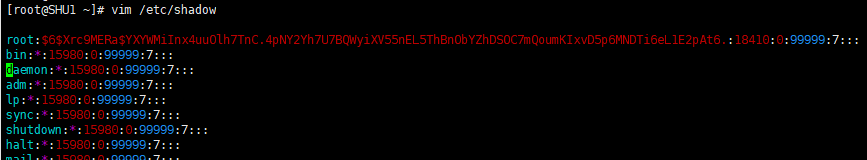
passwd这个文件的每行内容由冒号隔开，分为7段

|  |  |  |
| --- | --- | --- |
| 第一段 | 账户名 | 不要使用：. - + / |
| 第二段 | 密码 | passwd这个文件是所有人都可查看，所以密码是加密的，显示的x。 |
| 第三段 | UID | 系统用来标识内部的账户，通常UID和账户是对应的关系。如果出现了两个不同账户名但是却使用的相同的UID，那么系统就将它们两个识别为同一个账户，只不过它们拥有不同的账户名、shell程序、家目录。后者遵循前者。 |
| 第四段 | GID | 此处记录的是该账户的主组信息。 |
| 第五段 | 注释描述 | 对账户的描述信息，通常自定义账户都不写这个。 |
| 第六段 | 家目录 | 用户登录系统之后的默认工作空间，该空间除root和本人意外，默认是不允许其他人访问。 |
| 第七段 | Shell程序 | Linux、类Unix系统中的特有程序。 |

用户名 密码 UID GID 家目录 shell程序

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

## 2.shadow



root:$6$Xrc9MERa$....:18410:0:99999:7:::

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 账户名 | 密码 | 从密码创建至今的天数 | 修改密码之间的间隔 | 最大的有效期（天） | 密码到期前的N天 | 缺省值为空 |
| root: | $6$Xrc9MERa$: | 18410: | 0(无间隔) | 99999: | 7: |  |

|  |  |
| --- | --- |
| 第一段 | 用户名 |
| 第二段 | 加密后的口令(注：不允许手动修改密码内容,如果含有不属于集合 { $./0-9A-Za-z }中的字符，则对应的用户不能登录) |
| 第三段 | 从密码创建至今的天数。从1970年1月1日开始计算值。 |
| 第四段 | 表示上次和下次修改密码之间的间隔，如果是0表示无间隔限制 |
| 第五段 | 自密码创建时刻起，最大的有效期（天） |
| 第六段 | 密码到期前的N天，提醒用户修改密码。 |
| 第七段 | 缺省值为空，该字段允许密码到期之后N天之内还依然可以登录。 |
| 第八段 | 缺省值为空，该字段表示一个绝对的天数，意为到期之后不允许登录，也可以理解为密码的有效存活期。 |

## 3.group

root:x:0:jerry

|  |  |
| --- | --- |
| 第一段 | 用户组的名称 |
| 第二段 | 组的密码，通常用x或者\*来表示。部分系统中没有组密码的设定 |
| 第三段 | GID |
| 第四段 | 组内的成员列表。文件中显示的账户名都是将该组作为附加组。如果是主组，在此不给与显示。 |

用户组名 组密码 GID 组内成员

root :x :0 :jerry

说明：

root账户和root用户组是两码事，某个账户就算是加入了root组，那么也不代表它具备root权限。因为在Linux系统，默认情况下，用户的权限高于用户组。