

Linux系统用户与用户组管理

教师: 李军辉

用 戶 (User)

組 (Group)

用户 (User)

- 使用Linux系统时, 需要以一个用户的身份登录.
- 运行一个程序时, 需要以一个用户的身份运行.
-

用户 (User)

- 使用Linux系统时, 需要以一个用户的身份登录.
- 运行一个程序时, 需要以一个用户的身份运行.
-

用户限制使用者或进程可以使用、不可以使用哪些资源.

组 (Group)

组用来方便组织管理用户

- 每个用户拥用一个UserID, 操作系统实际使用的是用户UserID, 而非用户名.
- 每个用户属于一个主组, 属于一个或多个附属组
- 每个组拥有一个GroupID
- 每个进程以一个用户身份运行, 并受该用户可访问的资源限制
- 每个可登录的用户拥有一个指定的shell

用户 (User)

- 用户 UserID 为 32 位, 从 0 开始. 但为了和老式系统兼容, 用户 ID 限制在 60,000 以下.
- 用户分为以下三种 (用户编号随不同发行版本不同):
 - root 用户 (ID 为 0 的用户为 root 用户, 根用户, 超级用户)
 - 系统用户 (1 - 499)
 - 普通用户 (500 以上)
- 系统中的文件都有一个所属用户及所属组.
- 使用 id 命令可以显示当前用户的信息.
- 使用 passwd 命令可以修改当前用户密码.

跟用户相关文件

- /etc/passwd 保存用户信息
- /etc/shadow 保存用户密码 (加密后)
- /etc/group 保存组信息

/etc/passwd 文件结构

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

- 1 账号名称
- 2 口令 (早期 Unix 系统的口令就是放在这字段上)
- 3 UID (user id)
- 4 GID (group id)
- 5 用户信息说明栏
- 6 家目录
- 7 Shell

- 0: 当 UID 是 0 时, 代表这个账号是『系统管理员』。
- 1~499: 保留给系统使用的 ID, 由于系统上面启动的服务希望使用较小的权限去运行, 因此不希望使用 `root` 的身份去运行这些服务, 所以我们就得要提供这些运行中程序的拥有者账号才行。这些系统账号通常是不可登陆的。根据系统账号的由来, 通常系统账号又约略被区分为两种:
 - 1~99: 由 `distributions` 自行创建的系统账号;
 - 100~499: 若用户有系统账号需求时, 可以使用的账号 UID。
- 500~65535: 给一般使用者用的。

/etc/shadow 文件结构

/etc/passwd 的权限需配置为 -rw-r-r-

```
root:$1$/30QpE5e$y9N/D0bh6rAACBEz.hqo00:14126:0:99999:7:::
```

- 1 账号名称: 必须要与 /etc/passwd 相同才行
- 2 口令: 经过编码的口令 (加密)
- 3 最近更动口令的日期: 以 1970 年 1 月 1 日作为 1 而累加的日期
- 4 口令不可被更动的天数: 这个账号的口令在最近一次被更改后需要经过几天才可以再被变更! 如果是 0 的话, 表示口令随时可以更动的意思。
- 5 口令需要重新变更的天数: 为了强制要求用户变更口令, 这个字段可以指定在最近一次更改口令后, 在多少天数内需要再次的变更口令才行。

/etc/shadow 文件结构

/etc/passwd 的权限需配置为 -rw-r-r-

```
root:$1$/30QpE5e$y9N/D0bh6rAACBEz.hqo00:14126:0:99999:7:::
```

- 6 口令需要变更期限前的警告天数: 当账号的口令有效期限快要到的时候(第5字段), 系统会依据这个字段的配置, 发出『警告』言论给这个账号。
- 7 口令过期后的账号宽限时间(口令失效日):
- 8 账号失效日期:
- 9 保留:最后一个字段是保留的, 看以后有没有新功能加入。

一般用户的口令忘记了,怎么办?

一般用户的口令忘记了,怎么办?

请系统管理员帮忙,他会重新配置好你的口令而不需要知道你的旧口令!利用root的身份使用passwd命令来处理即可。

root用户的口令忘记了，怎么办？

root用户的口令忘记了，怎么办？

root 的口令在 /etc/shadow 当中，因此你可以使用各种可行的方法启动进入 Linux 再去修改。例如重新启动进入单人维护模式(第二十章)后，系统会主动的给予 root 权限的 bash 接口，此时再以 passwd 修改口令即可；或以 Live CD 启动后挂载根目录去修改 /etc/shadow，将里面的 root 的口令字段清空，再重新启动后 root 将不用口令即可登陆！登陆后再赶快以 passwd 命令去配置 root 口令即可。

/etc/group 文件结构

这个文件就是在记录 GID 与组名的对应.

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
```

这个文件每一行代表一个群组，也是以冒号『:』作为字段的分隔符，共分为四栏，每一字段的意义是：

- 1 组名:
- 2 群组口令: 通常不需要配置，这个配置通常是给『群组管理员』使用的，目前很少有这个机会配置群组管理员！同样的，口令已经移动到 /etc/gshadow 去，因此这个字段只会存在一个『x』；
- 3 GID: 群组的 ID
- 4 一个账号可以加入多个群组，那某个账号想要加入此群组时，将该账号填入这个字段即可。

根据第4列可知，每个用户都可以拥有多个支持的群组。

假如某个用户同时加入多个群组，那么创建一个文件或文件夹时，属于哪个群组？

有效群组(effective group)与初始群组(initial group)

初始群组: /etc/passwd 里面的第四栏的 GID

有效群组: groups 命令第一个输出的群组即为有效群组.

有效群组的作用是在新建文件/文件夹时, 对应的群组.

切换有效群组

newgrp: 有效群组的切换

```
newgrp users #有效群组修改为users
```

查看登录的用户

- 命令whoami显示当前用户
- 命令who显示哪些用户已经登录系统
- 命令w显示有哪些用户已经登录并且在干什么

创建用户

useradd [-u **UID**] [-g **初始群组**] [-G **次要群组**] [-mM] [-c **说明栏**] [-d **家目录绝对路径**] [-s **shell**] **使用者账号名**

- **-u** : 后面接的是 **UID** , 是一组数字。直接指定一个特定的 **UID** 给这个账号 ;
- **-g** : 后面接的那个组名就是我们上面提到的 **initial group** , 该群组的 **GID** 会被放置到 **/etc/passwd** 的第四个字段内。
- **-G** : 后面接的组名则是这个账号还可以加入的群组。这个选项与参数会修改 **/etc/group** 内的相关数据 !
- **-M** : 强制 ! 不要创建用户家目录 ! (系统账号默认值)
- **-m** : 强制 ! 要创建用户家目录 ! (一般账号默认值)
- **-c** : 这个就是 **/etc/passwd** 的第五栏的说明内容 , 可以随便我们配置的

useradd [-u UID] [-g 初始群组] [-G 次要群组] [-mM] [-c 说明栏]
[-d 家目录绝对路径] [-s shell] 使用者账号名

- -d : 指定某个目录成为家目录, 而不要使用默认值。务必使用绝对路径!
- -r : 创建一个系统的账号, 这个账号的 UID 会有限制(参考 /etc/login.defs)
- -s : 后面接一个 shell, 若没有指定则默认是 /bin/bash
- -e : 后面接一个日期, 格式为『YYYY-MM-DD』此项目可写入 shadow 第八字段, 亦即账号失效日的配置项目;
- -f : 后面接 shadow 的第七字段项目, 指定口令是否会失效。0为立刻失效,
- -1 为永远不失效(口令只会过期而强制于登陆时重新配置而已。)

例1：完全参考默认值创建一个用户，名称为 vbird1

```
useradd -m vbird1  
|| -d /home/vbird1  
grep vbird1 /etc/passwd /etc/shadow /etc/group
```

需要使用-m参数，否则不会创建用户目录。

创建用户

命令useradd创建一个新用户：

```
useradd vbird1
```

- 在/etc/passwd中添加用户信息
- 如果使用passwd命令创建密码, 则将密码保存于/etc/shadow文件中
- 为用户建立一个新的主目录 /home/vbird1 (注意主目录权限)
- 将/etc/skel中的文件复制到用户主目录下
- 建立一个与用户名相同的组, 新建用户默认属于这个同名组

例2: 假设我已知道我的系统当中有个组名为 `users`，且 UID 700 并不存在，请用 `users` 为初始群组，以及 uid 为 700 来创建一个名为 `vbird2` 的账号.

```
useradd -m -u 700 -g users vbird2
```

```
ll -d /home/vbird2
```

```
grep vbird2 /etc/passwd /etc/shadow /etc/group
```

例3: 创建一个系统账号, 名称为 vbird3

```
useradd -r vbird3
```

```
ll -d /home/vbird3
```

```
grep vbird3 /etc/passwd /etc/shadow /etc/group
```

般账号是500号以后, 那用户自己创建的系统账号则一般是由 100号以后起算的。

修改密码

可以通过直接修改/etc/passwd的方式实现 (但不建议).

修改密码

- 口令不能与账号相同；
- 口令尽量不要选用字典里面会出现的字符串；
- 口令需要超过 8 个字符；
- 口令不要使用个人信息，如身份证、手机号码、其他电话号码等；
- 口令不要使用简单的关系式，如 $1+1=2$, `Iamvbird` 等；
- 口令尽量使用大小写字符、数字、特殊字符(\$,#,等)的组合。

修改密码

请 root 给予 vbird2 口令

```
passwd vbird2
```

Changing password for user vbird2.

New UNIX password:

Retype new UNIX password:

passwd: all authentication tokens updated successfully.

修改密码

用 vbird2 登陆后，修改 vbird2 自己的口令

```
passwd
```

```
Changing password for user vbird2.
```

```
(current) UNIX password:
```

```
New UNIX password:
```

```
Retype new UNIX password:
```

```
passwd: all authentication tokens updated successfully.
```

修改密码

使用 standard input 创建用户的口令

```
echo "abc543CC" | passwd --stdin vbird2
```

Changing password for user vbird2.

passwd: all authentication tokens updated successfully.

- 好处: 不需要两次输入, 方便处理.
- 坏处: 可以通过文件.bash_history找到这个命令.
- 这个动作通常仅用在 shell script 的大量创建使用者账号当中.

for 应用举例: 批量增加用户账号

- 给定一个用户文件user.txt, 每行一个用户名
- 将初始密码设置为123456, 首次登录后必须更改

for 应用举例: 批量增加用户账号

```
#!/bin/bash

for i in $(cat ./users.txt)
do
    useradd $i
    echo -e "123456\n123456" | passwd $i
    chage -d 0 $i
done
```

如果你想要让 vbird2 的口令具有相当的守则，举例来说你要让 vbird2 每 60 天需要变更口令， 口令过期后 10 天未使用就宣告口令失效.

```
passwd -S vbird2 #查看口令样信息
```

```
passwd -x 60 -i 10 vbird2 #修改
```

```
passwd -S vbird2 #查看口令样信息
```

让 vbird2 的账号失效，和恢复

p

```
asswd -S vbird2
```

```
passwd -l vbird2
```

```
grep vbird2 /etc/shadow
```

```
passwd -u vbird2
```

```
grep vbird2 /etc/shadow
```

chage命令 (change age)

chage [-l|-d|-m|-M|-W] 账号名

- **-l**：列出该账号的详细口令参数；
- **-d**：后面接日期，修改 shadow 第三字段(最近一次更改口令的日期)，格式 YYYY-MM-DD
- **-E**：后面接日期，修改 shadow 第八字段(账号失效日)，格式 YYYY-MM-DD
- **-l**：后面接天数，修改 shadow 第七字段(口令失效日期)
- **-m**：后面接天数，修改 shadow 第四字段(口令最短保留天数)
- **-M**：后面接天数，修改 shadow 第五字段(口令多久需要进行变更)
- **-W**：后面接天数，修改 shadow 第六字段(口令过期前警告日期)

列出 vbird2 的详细口令参数

```
chage -l vbird2
```

使用者在第一次登陆时， 强制他们一定要更改口令后才能够使用系统资源

```
useradd agetest
```

```
echo "agetest" | passwd --stdin agetest
```

```
chage -d 0 agetest
```

修改用户信息

命令usermod修改用户信息:

usermod 参数 username

支持以下参数:

- -c : 后面接账号的说明, 即 /etc/passwd 第五栏的说明栏, 可以加入一些账号的说明。
- -d : 后面接账号的家目录, 即修改 /etc/passwd 的第六栏;
- -e : 后面接日期, 格式是 YYYY-MM-DD 也就是在 /etc/shadow 内的第八个字段数据
- -f : 后面接天数, 为 shadow 的第七字段
- -g : 后面接初始群组, 修改 /etc/passwd 的第四个字段, 亦即是 GID 的字段
- -G : 后面接次要群组, 修改这个使用者能够支持的群组, 修改的是 /etc/group
- -a : 与 -G 合用, 可『添加次要群组的支持』而非『配置』

命令usermod修改用户信息:

usermod 参数 username

支持以下参数:

- -l : 后面接账号名称。亦即是修改账号名称, /etc/passwd 的第一栏!
- -s : 后面接 Shell 的实际文件, 例如 /bin/bash 或 /bin/csh 等等。
- -u : 后面接 UID 数字! 即 /etc/passwd 第三栏的数据;
- -L : 暂时将用户的口令冻结, 让他无法登陆。其实仅改 /etc/shadow 的口令栏。
- -U : 将 /etc/shadow 口令栏的!

例3：修改使用者 vbird2 的说明栏，加上『VBird's test』的说明。

```
usermod -c "VBird's test" vbird2
```

```
grep vbird2 /etc/passwd
```

例4：用户 vbird2 口令在 2009/12/31 失效。

```
usermod -e “2009-12-31” vbird2
```

```
grep vbird2 /etc/shadow
```

例5：我们创建 vbird3 这个系统账号时并没有给予家目录，请创建他的家目录

```
ll -d ~vbird3
cp -a /etc/skel /home/vbird3
chown -R vbird3:vbird3 /home/vbird3
chmod 700 /home/vbird3
usermod -d /home/vbird3 vbird3
ll -a vbird3
```

删除用户

命令useradd删除指定用户：

`userdel username` (不删除用户主目录)

`userdel -r username` (同时删除用户主目录)

例6：删除 vbird2，连同家目录一起删除

```
userdel -r vbird2
```

新增与移除群组

基本上，群组的内容都与这两个文件有关：`/etc/group`,
`/etc/gshadow`。

- 几乎所有操作系统都有组的概念.
- 通过组, 可以更加方便的归类, 管理用户.
- 一般来讲, 我们使用部门、职能或地理区域的分类方式来创建使用组.

- 几乎所有操作系统都有组的概念.
- 通过组, 可以更加方便的归类, 管理用户.
- 一般来讲, 我们使用部门、职能或地理区域的分类方式来创建使用组.

- 每个级有一个组ID
- 组信息保存在/etc/group中
- 每个用户有一个主组, 同时还可以拥有最多31个附属组

- 创建组
 - `groupadd groupname`
- 修改组信息
 - `groupmod 参数 groupname`
- 删除组
 - `groupdel groupname`

groupadd

groupadd [-g gid] [-r] 组名

- -g : 后面接某个特定的 GID , 用来直接给予某个 GID
- -r : 创建系统群组

例7：新建一个群组，名称为 group1

```
groupadd group1  
grep group1 /etc/group /etc/gshadow
```

groupmod

groupmod [-g gid] [-n group_name] 群组名

- -g : 修改既有的 GID 数字；
- -n : 修改既有的组名

例8：将刚刚上个命令创建的 group1 名称改为 mygroup， GID 为 201

```
groupmod -g 201 -n mygroup group1  
grep mygroup /etc/group /etc/gshadow
```

不要随意的更动 GID， 容易造成系统资源的错乱!

groupdel

groupdel [groupname]

例9：将刚刚的 mygroup 删除！

groupdel mygroup

例10：若要删除 vbird1 这个群组的话？

groupdel vbird1

groupdel

groupdel [groupname]

例9：将刚刚的 mygroup 删除！

```
groupdel mygroup
```

例10：若要删除 vbird1 这个群组的话？

```
groupdel vbird1
```

必须要确认 /etc/passwd 内的账号没有任何人使用该群组作为 initial group

任务一：单纯的完成上头交代的任务，假设我们需要的账号数据如下，你该如何实现？

账号名称	账号全名	支持次要群组	是否可登陆主机	口令
myuser1	1st user	mygroup1	可以	password
myuser2	2nd user	mygroup1	可以	password
myuser3	3rd user	无额外支持	不可以	password

```
groupadd mygroup1
useradd -G mygroup1 -c “1st user” myuser1
useradd -G mygroup1 -c “2nd user” myuser2
useradd -c “3rd user” -s /sbin/nologin myuser3 echo "password" |
passwd –stdin myuser1
echo "password" | passwd –stdin myuser2
echo "password" | passwd –stdin myuser3
```

任务二：我的使用者 pro1, pro2, pro3 是同一个项目计划的开发人员，我想要让这三个用户在同一个目录底下工作，但这三个用户还是拥有自己的家目录与基本的私有群组。假设我要让这个项目计划在 /srv/projecta 目录下开发，可以如何进行？

第1步. 假设这三个账号都尚未创建, 可先创建一个名为 projecta 的群组,

再让这三个用户加入其次要群组的支持即可:

```
groupadd projecta
useradd -G projecta -c "projecta user" pro1
useradd -G projecta -c "projecta user" pro2
useradd -G projecta -c "projecta user" pro3
echo "password" | passwd --stdin pro1
echo "password" | passwd --stdin pro2
echo "password" | passwd --stdin pro3
```

第2步. 开始创建此项目的开发目录：

```
mkdir /srv/projecta
```

```
chgrp projecta /srv/projecta
```

```
chmod 2770 /srv/projecta
```

```
ll -d /srv/projecta
```

ACL 的使用

ACL 是 Access Control List 的缩写，主要的目的是在提供传统的 owner,group,others 的 read,write,execute 权限之外的细部权限配置。

ACL 可以针对单一使用者，单一文件或目录来进行 r,w,x 的权限规范，对于需要特殊权限的使用状况非常有帮助。

ACL可以：

- 使用者 (user)：可以针对使用者来配置权限；
- 群组 (group)：针对群组为对象来配置其权限；
- 默认属性 (mask)：还可以针对在该目录下在创建新文件/目录时，规范新数据的默认权限；

- getfacl: 取得某个文件/目录的 ACL 配置项目；
- setfacl : 配置某个目录/文件的 ACL 规范。

针对特定用户的方式

创建一个文件，并单独设置vbird1用户可以访问和修改这个文件。

```
touch acl_test1
ll acl_test1
setfacl -m u:vbird1:rx acl_test1
ll acl_test1
getfacl acl_test1
```

针对特定群组的方式

配置规范：『g:[群组列表]:[rwx]』，例如针对 mygroup1 的权限规范 rx：

```
setfacl -m g:mygroup1:rx acl_test1  
getfacl acl_test1
```

任务三: 将前一小节任务二中 /srv/projecta 这个目录, 让 myuser1 可以进入查阅, 但 myuser1 不具有修改的权力。

任务三: 将前一小节任务二中 /srv/projecta 这个目录, 让 myuser1 可以进入查阅, 但 myuser1 不具有修改的权力。

#1. 先测试myuser1用户对文件夹的/srv/projecta的权限

```
cd /srv/projecta
```

#2. 用管理员权限修改/srv/projecta的权限 setfacl -m u:myuser1:rx

```
/srv/projecta
```

```
getfacl /srv/projecta
```

#3. 再要使用 myuser1 去测试看看结

```
cd /srv/projecta
```

```
ll -a
```

```
touch testing
```

权限的继承, 上面的做法myuser1可以进入/srv/projecta这个目录, 但在/srv/projecta目录中再建立新文件或新文件夹, 新文件或新文件夹的acl权限并没有被继承。

```
setfacl -m d:u:myuser1:rx /srv/projecta
```

使用者身份切换

在 Linux 系统当中还要作身份的变换？

- 使用一般账号：系统平日操作的好习惯

- 尽量以一般身份使用者来操作 Linux 的日常作业！等到需要配置系统环境时，才变换身份成为 root 来进行系统管理，相对比较安全啦！避免作错一些严重的命令，如 rm -rf

在 Linux 系统当中还要作身份的变换？

- 使用一般账号：系统平日操作的好习惯

- 尽量以一般身份使用者来操作 Linux 的日常作业！等到需要配置系统环境时，才变换身份成为 root 来进行系统管理，相对比较安全啦！避免作错一些严重的命令，如 rm -rf

- 用较低权限启动系统服务

- 相对于系统安全，有的时候，我们必须要有某些系统账号来进行程序的运行。

在 Linux 系统当中还要作身份的变换？

- 使用一般账号：系统平日操作的好习惯
 - 尽量以一般身份使用者来操作 Linux 的日常作业！等到需要配置系统环境时，才变换身份成为 root 来进行系统管理，相对比较安全啦！避免作错一些严重的命令，如 rm -rf
- 用较低权限启动系统服务
 - 相对于系统安全，有的时候，我们必须要有某些系统账号来进行程序的运行。
- 软件本身的限制

su命令

su 是最简单的身份切换命令了，它可以进行任何身份的切换.

Ubuntu下如何切换到root用户??

```
sudo passwd root
```

su切换到root用户时,需要输入root用户密码.sudo的运行则仅需要自己的口令即可!

su切换到root用户时,需要输入root用户密码.sudo的运行则仅需要自己的口令即可!到底哪些用户可以使用sudo?

su 切换到 root 用户时, 需要输入 root 用户密码. sudo 的运行则仅需要自己的口令即可! 到底哪些用户可以使用 sudo?

由于 sudo 可以让你以其他用户的身份运行命令 (通常是使用 root 的身份来运行命令), 因此并非所有人都能够运行 sudo, 而是仅有规范到 /etc/sudoers 内的用户才能够运行 sudo 这个命令.

如何设置某个普通用户(如vbird1)具有运行sudo的权限??

如何设置某个普通用户(如vbird1)具有运行sudo的权限??

```
sudo usermod -aG sudo vbird1
```

特殊的 shell, /sbin/nologin

在/etc/passwd文件中，系统账号对应的Shell通常为 /sbin/nologin。这是无法登陆的合法 shell。

- 这个使用者无法使用 bash 或其他 shell 来登陆系统
- 并不是说这个账号就无法使用其他的系统资源

/etc/nologin.txt 当Shell为/sbin/nologin的用户想要登陆系统时，屏幕上出现的就会是 /etc/nologin.txt 这个文件的内容

- 1 root 的 UID 与 GID 是多少？而基于这个理由，我要让 test 这个账号具有 root 的权限，应该怎么样作？
- 2 假设我是一个系统管理员，我有一个用户最近不乖，所以我想暂时将他的账号停掉，让他近期无法进行任何动作，等到未来他乖一点之后，我再将他的账号激活，请问：我可以怎么样作比较好？？
- 3 我在使用 useradd 的时候，新增的账号里面的 UID, GID 还有其他相关的口令控制，都是在哪几个文件里面配置的？
- 4 我希望我在配置每个账号的时候(使用 useradd)，默认情况下，他们的家目录就含有一个名称为 www 的子目录，我应该怎么样作比较好？

- 6 简单说明系统账号与一般用户账号的差别？
- 7 如何创建一个使用者名称 alex, 他所属群组为 alexgroup, 预计使用 sh, 他的全名为 "Alex Tsai", 且他还得要加入 users 群组当中！
- 8 由于种种因素，导致你的用户家目录以后都需要被放置到 /account 这个目录下。请问，我该如何作，可以让使用 useradd 时，默认的家目录就指向 /account ？
- 9 我想要让 dmtsai 这个使用者，加入 vbird1, vbird2, vbird3 这三个群组，且不影响 dmtsai 原本已经支持的次要群组时，该如何动作？

示例

假设某台服务器有以下类别员工使用：

- 计算机

- 教师

- 李老师
 - 张教师

- 学生

- 李同学
 - 张同学

- 电子

- 教师

- 陈老师
 - 刘教师

- 学生

- 陈同学
 - 刘同学

示例 (续)

设计如下组:

- 计算机: 李老师, 张老师, 李同学, 张同学
- 电子: 陈老师, 刘老师, 陈同学, 刘同学
- 计算机.教师: 李老师, 张老师
- 计算机.学生: 李同学, 张同学
- 电子.教师: 陈老师, 刘老师
- 电子.学生: 陈同学, 刘同学

示例 (续)

先创建组:

- `groupadd cs`
- `groupadd ee`
- `groupadd cs_faculty`
- `groupadd cs_student`
- `groupadd ee_faculty`
- `groupadd ee_student`

示例 (续)

先创建组:

- groupadd cs
- groupadd ee
- groupadd cs_faculty
- groupadd cs_student
- groupadd ee_faculty
- groupadd ee_student

再创建用户:

- useradd li -G cs,cs_faculty
- useradd zhang -G cs,cs_student
-