

用户

用户一般来说是指使用计算机的人，计算机针对使用其的每一个人给了一个特定的名称，用户就可以使用这些名称来登录使用计算机，除了人之外，一些系统服务也需要含有部分特权的用户账户运行；因此出于安全考虑，用户管理应运而生，它加以明确限制各个用户账户的权限，root在计算机中拥有至高特权，所以一般只作管理用，非特权用户可以通过 SU 或 SUDO 程序来临时获得特权

GNU/Linux 通过用户和用户组实现访问控制----包括对文件访问、设备使用的控制

个人可以拥有很多账户，只是彼此名称不同，比如root名称已经占用就不能再用了，此外，任意用户可能从属某个用户组，此用户可以加入某些已经存在的组来获得该组的特权

GNU/Linux 系统中的每一个文件都有属一个用户（属主）和一个用户组（属组）。另外，还有三种类型的访问权限：读（read）、写（write）、运行（execute）。

用户账号、用户的分类

Linux识别用户：通过UID（用户的id 号码）

id 语法：

id 用户名

```
1 [root@exrcise1 opt]# id #如不接用户名，则显
   示当前用户名
2 uid=0(root) gid=0(root) 组=0(root)
3 [root@exrcise1 opt]# id sshd #接用户名，则显示指
   定用户名
4 uid=74(sshd) gid=74(sshd) 组=74(sshd)
5 [root@home ~]# id abdc #可以识别系统有没有
   指定
6 id: abdc: no such user
7
```

Linux用户三种角色

超级用户：root 拥有对系统的最高的管理权限 ID=0

普通用户：系统用户 UID:1-999(centos7 版本) 1-499(centos6 版本)\
本地用户 UID:1000+ 500+

UID:即每个用户的身份标示,类似于每个人的身份证号码。

虚拟用户：伪用户 一般不会用来登录系统的，它主要是用于维持某个服务的正常运行。如：ftp, apache

下图是用户和组的关系：

一对一：一个用户可以存在一个组中；

多对一：多个用户可以存在一个组中；

一对多：一个用户可以存在多个组中；

多对多：多个用户可以存在多个组中；



名称	账号信息	说明
用户配置文件	/etc/passwd	记录了每个用户的一些基本属性，并且对所有用户可读，每一行记录 对应一个用户，每行记录通过冒号进行分隔
用户组文件	/etc/group	用户组的所有信息存放地，并且组名不能重复
用户对应的密码信息	/etc/shadow	因为 passwd 文件对所有用户是可读的，为安全起见把密码从 passwd中分离出来放入这个单独的文件，该文件只有 root 用户拥有 读权限，从而保证密码安全性

用户管理

添加用户命令

命令：useradd

useradd -d -u "UID" -g "主组" -G "附加组" -s "登陆的 shell" -c 指定说明信息 用户

参数：

- 1 -d : 用户主目录路径， 可以指定用户家目录
- 2 -m : 如果没有目录，会自动创建新目录并且移到内容到新目录里面
- 3 -u : 指定用户ID号
- 4 -g : 设置用户初始组的名称或数字ID；该组必须是存在的；如果没有设置该选项，useradd会根据/etc/login.defs文件中的USERGROUPS_ENAB环境变量进行设置。
- 5 默认 USERGROUPS_ENAB yes 会用 和用户名相同的名字创建群组，GID 等于 UID.
- 6 -G : 用户要加入的附加组列表；使用逗号分隔多个组，不要添加空格；如果不设置，用户仅仅加入初始组。（一个用户只允许有一个主组，可以有多个附属组）
- 7 -s : 用户默认登录 shell 的路径；启动过程结束后，默认启动的登录 shell 在此处设定；请确保使用的 shell
- 8 已经安装，默认是 Bash。有时候需要禁止某些用户执行登录动作，例如用来执行系统服务的用户。将 shell 设置成 /sbin/nologin就可以禁止用户登录。
- 9 -c : 附加描述信息
- 10 -M : 给创建的用户不创建家目录
- 11 -r : 创建系统账户，默认无家目录

添加登录用户

- 1 例子：添加一个名为 lin01 的用户，并使用 bash 作为登录的shell
- 2 [root@exercise1 ~]# useradd lin01
- 3 [root@exercise1 ~]# tail -1 /etc/passwd
- 4 lin01:x:1000:1000::/home/lin01:/bin/bash
- 5 [root@exercise1 ~]#

说明：此命令会自动创建 lin01 组，并成为 lin01 用户的默认主组，同时默认的登录 shell 是 bash

用户帐户的全部信息被保存在/etc/passwd文件。这个文件以如下格式保存了每一个系统帐户的所有信息(字段以“:”分割)

lin01:x:1000:1000:./home/lin01:/bin/bash

/etc/passwd: 每个字段的作用:

例如: root:x:0:0:root:/root:/bin/bash

用户名 : 密码占位符 : UID : GID : 用户描述 : 用户主目录 (bash 中 "~" 代表哪个) : 登录后使用的 shell

lin01 : 用户名

x : 密码占位符

1001 : 用户的 UID , 它都是用数字来表示的

1001 : 用户所属组的GID , 它都是用数字来表示的

用户描述信息: 对用户的功能或其它来进行一个简要的描述

/home/lin01 : 用户主目录 (shell 提示符中 "~" 代表的那个)

/bin/bash : 用户登录系统后使用的shell

解释二:

/etc/passwd中一行记录对应着一个用户，每行记录又被冒号(:)分隔为7个字段，其格式和具体含义如下：

```
1 用户名:口令:用户标识号:组标识号:注释性描述:主目录:登录Shell
2
3  #查看系统中，支持哪些shell
4  [root@exercise1 ~]# cat /etc/shells
5  /bin/sh
6  /bin/bash
7  /sbin/nologin
8  /usr/bin/sh
9  /usr/bin/bash
10 /usr/sbin/nologin
```

指定用户 UID

```
1 [root@exercise1 ~]# useradd -u 1100 oracle
2 [root@exercise1 ~]# id oracle
3 uid=1100(oracle) gid=1100(oracle) 组=1100(oracle)
4 [root@exercise1 ~]# tail -1 /etc/passwd
5 oracle:x:1100:1100::/home/oracle:/bin/bash
6 [root@exercise1 ~]# ls /home/oracle/ -a
7 . .. .bash_logout .bash_profile .bashrc
8 [root@exercise1 ~]#
```

指定用户主目录

```
1 [root@exercise1 ~]# useradd -d /opt/lin01 lin01
2 [root@exercise1 ~]# tail -1 /etc/passwd
3 lin01:x:1101:1101::/opt/lin01:/bin/bash
4 [root@exercise1 ~]#
```

指定用户的主组

```
1 [root@exercise1 ~]# useradd lin02
2 [root@exercise1 ~]# id lin02
3 uid=1102(lin02) gid=1102(lin02) 组=1102(lin02)
4 [root@exercise1 ~]# useradd -g lin02 abc
5 [root@exercise1 ~]# id abc
6 uid=1103(abc) gid=1102(lin02) 组=1102(lin02)
7 [root@exercise1 ~]#
```

指定用户的附属组

我们也可以把这个附属组称为补充组，用户可以有 0 个或多个附加组的成员

```
1 [root@exercise1 ~]# useradd -G root,lin01,lin02 lin03
2 [root@exercise1 ~]# id lin03
3 uid=1104(lin03) gid=1104(lin03) 组
  =1104(lin03),0(root),1101(lin01),1102(lin02)
4 [root@exercise1 ~]#
```

创建用户的另外一个命令(了解)

```
1 [root@exercise1 ~]# adduser lin04
2 [root@exercise1 ~]# id lin04
3 uid=1105(lin04) gid=1105(lin04) 组=1105(lin04)
4 [root@exercise1 ~]# which adduser
5 /usr/sbin/adduser
6 [root@exercise1 ~]# ll /usr/sbin/adduser
7 lrwxrwxrwx. 1 root root 7 1月  9 09:29
  /usr/sbin/adduser -> useradd  注: adduser 是useradd的软
  链接
8 [root@exercise1 ~]#
```

指定用户默认登录 shell 的路径

```
1 [root@exercise1 ~]# useradd -s "/bin/nologin" lin05
2 [root@exercise1 ~]# su lin05
3 su: failed to execute /bin/nologin: 没有那个文件或目录
4 [root@exercise1 ~]#
```

修改用户信息

语法: usermod 【参数】 用户名

常用参数:

1	-u	指定要修改用户的UID	
2	-g	指定要修改用户的主组	#只有一个(只能修改系统已存在的组)
3	-G	指定要修改用户的附属组	#可以有多个
4	-d	指定要修改用户的家目录	
5	-s	指定要修改用户的登录 shell	
6	-c	指定要修改用户的指定说明信息	
7	-a	添加指定成员到附属组中	

修改 UID

```

1 [root@exercise1 ~]# id oracle
2 uid=1100(oracle) gid=1100(oracle) 组=1100(oracle)
3 [root@exercise1 ~]# usermod -u 1111 oracle
4 [root@exercise1 ~]# id oracle
5 uid=1111(oracle) gid=1100(oracle) 组=1100(oracle)
6 [root@exercise1 ~]#

```

修改附加组

```

[root@exercise1 ~]# usermod -G 1102 abc #修改的组要存在
[root@exercise1 ~]# id abc
uid=1103(abc) gid=1102(lin02) 组=1102(lin02)
[root@exercise1 ~]#

```

修改 shell

```

[root@exercise1 ~]# usermod -s "/bin/nologin" oracle
[root@exercise1 ~]# tail -2 /etc/passwd
oracle❌1111:1100::/home/oracle:/bin/nologin
abc❌1103:1102::/home/abc:/bin/bash
[root@exercise1 ~]#

```

修改用户主目录

```

[root@exercise1 ~]# usermod -m -d /opt/abc abc
-m 选项会自动创建新目录并且移到内容到新目录里面

```


添加说明信息

```
[root@exercise1 ~]# usermod -c "hello word" abc
[root@exercise1 ~]# tail -2 /etc/passwd
oracle✗1111:1100::/home/oracle:/bin/nologin
abc✗1103:1102:hello word:/opt/abc:/bin/bash
[root@exercise1 ~]#
```

删除用户

语法: `userdel 参数 用户名`

选项: `-r` 删除的时候, 会同时删除用户的家目录和`/var/mail`下的目录

```
1 [root@exercise1 ~]# userdel lin03
2
3 [root@exercise1 ~]# userdel -r lin03
```

提问: 如果当有一个用户正在使用

能否删除

```
1 [root@home ~]# userdel -r abc
2 userdel: user abc is currently used by process 19055
3 [root@home ~]# userdel -r abc
```

查看用户相关命令

```
1 [root@exercise1 ~]# id    #用户和组的信息
2 uid=0(root) gid=0(root) 组=0(root) 环境
   =unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
   #直接id命令默认查找当前用户啊，id 用户名 才是指定查看用户和组的信息
3
4 [root@exercise1 ~]# whoami    #查看当前有效用户名
5 root
6
7 [root@exercise1 ~]# who    #显示目前登入系统的用户信息
8 root      pts/0          2022-01-17 17:10 (192.168.119.1)
9 root      pts/1          2022-01-17 18:32 (192.168.119.1)
```

[root@exercise1 ~]# w #w命令用于显示已经登陆系统的用户列表

19:09:59 up 1:53, 2 users, load average: 0.08, 0.03, 0.05

USER	TTY	FROM	LOGIN@	IDLE	JCPU	PCPU	WHAT
------	-----	------	--------	------	------	------	------

root	pts/0	192.168.119.1	17:10	1:39m	0.02s	0.02s	-bash
------	-------	---------------	-------	-------	-------	-------	-------

root	pts/1	192.168.119.1	18:32	7.00s	0.07s	0.00s	w
------	-------	---------------	-------	-------	-------	-------	---

[root@exercise1 ~]# users #用于显示当前登录系统的所有用户的用户列表

root root

用户的创建流程

提问：为什么创建用户时，普通UID为什么是1000开始？系统用户UID为什么是从201-999？

系统创建用户参考的两个配置文件

/etc/login.defs

/etc/default/useradd

```
1 [root@exercise1 opt]# grep -vE "^#|^$" /etc/login.defs
2 MAIL_DIR      /var/spool/mail
3 PASS_MAX_DAYS  99999
4 PASS_MIN_DAYS  0
5 PASS_MIN_LEN   5
6 PASS_WARN_AGE  7
7
8 #定义普通用户的UID范围
9 UID_MIN        1000
10 UID_MAX        60000
11
12 #定义系统用户的UID范围
13 SYS_UID_MIN    201
14 SYS_UID_MAX    999
15
16 #定义组的GID范围
17 GID_MIN        1000
18 GID_MAX        60000
19
20 #定义系统组GID范围
21 SYS_GID_MIN    201
22 SYS_GID_MAX    999
23 CREATE_HOME yes                                #默认创建家目录
24 UMASK          077
25 USERGROUPS_ENAB yes                            #创建用户时都会创建主组
26 ENCRYPT_METHOD SHA512                          #密码加密模式
27
28 [root@exercise1 opt]# cat /etc/default/useradd
29 # useradd defaults file
30 GROUP=100
31 HOME=/home
32 INACTIVE=-1                                     #是否启用账号过期停权，-1表示不启用
33 EXPIRE=                                           #账号终止日期，不设置表示不启用
34 SHELL=/bin/bash
35 SKEL=/etc/skel                                #配置新用户家目录的默认文件存放路径
36 CREATE_MAIL_SPOOL=yes
```

小结：如果在创建时自行指定了参数，那么优先使用定义的；如果没有定义则使用默认配置文件（/etc/login.defs；/etc/default/useradd）

解决模板文件被删之后显示不正常的问题

```
1 [root@exercise1 ~]# rm -rf /home/abc/.bash*
2 [root@exercise1 ~]# su - abc
3 -bash-4.2$ exit #出现这个不完整的 shell 提示符，如何处理？
4 [root@exercise1 ~]# cp /etc/skel/.bash* /home/abc/ #存
   放模板的地方，也可以直接cp其他家目录 cp -r /home/oracle
   /home/abc
5 [root@exercise1 ~]# chown abc:abc /home/abc/.bash*
6 [root@exercise1 ~]# su - abc
7 [root@abc ~]$
```

用户组管理

/etc/group 用户组信息配置文件

```
[root@base home]# vi /etc/group
```

```
root:x:0:
```

以“:”为分隔符，分为4列：\

第1列为用户组\

第2列为密码位\

第3列为GID\

第4列为附属组列表

添加用户组

语法: groupadd 【参数】 用户组名

常用参数:

-g 指定新组的GID

```
1 [root@exercise1 ~]# groupadd -g 1400 game
2 [root@exercise1 ~]# tail -1 /etc/group
3 game:x:1400:
4 [root@exercise1 ~]#
```

修改用户组信息

语法: groupmod 【参数】 用户组名

常用参数:

```
1 -g 修改既有的GID
2 -n 修改既有的组名
3 -d 从组中删除某一个成员
```

用户组切换

```
1 语法: newgrp 用户组名
2
3 注: 想要切换的用户组必须有支持的用户组
4
5 [root@exercise1 ~]# newgrp oracle
6 [root@exercise1 ~]# id
7 uid=0(root) gid=1100(oracle) 组=1100(oracle),0(root) 环境
  =unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
8 [root@exercise1 ~]#
```

删除用户组

```
1 语法: groupdel 用户组名
2
3 注: 必须确认/etc/passwd内的账号没有任何人使用该用户组作为主用户组
4
5 [root@exercise1 ~]# groupdel oracle
6 groupdel: 不能移除用户“oracle”的主组
7 [root@exercise1 ~]# groupdel game
8 [root@exercise1 ~]#
```

用户组管理员功能

注: 只有root能将用户组的主控权交由后面的用户管理

语法: gpasswd 【参数】用户组名

对于root而言, 常用参数:

```
1 -A 用户名 将该用户组的主控权交由后面的用户管理
2 -M 用户名 将该用户加入这个用户组里
3 -r 将用户组的密码删除
```

例子:

```
1 [root@exercise1 ~]# groupdel game
2 [root@exercise1 ~]# useradd lin05
3 [root@exercise1 ~]# groupadd game
4 [root@exercise1 ~]# gpasswd -A lin05 game
5 [root@exercise1 ~]# gpasswd -M abc game
6 [root@exercise1 ~]# tail -2 /etc/group
7 lin05:x:1112:
8 game:x:1113:abc
9 [root@exercise1 ~]#
```

对于用户组管理员而言, 常用参数:

```
1 -a 用户名 将该用户加入到用户组当中
2 -d 用户名 将该用户删除出用户组
```

例子:

```

1 [root@exercise1 ~]# gpasswd -a lin05 game
2 正在将用户“lin05”加入到“game”组中
3 [root@exercise1 ~]# gpasswd -d lin05 game
4 正在将用户“lin05”从“game”组中删除
5 [root@exercise1 ~]# su - lin05
6 [lin05@exercise1 ~]$ gpasswd -a lin05 game
7 正在将用户“lin05”加入到“game”组中
8 [lin05@exercise1 ~]$ gpasswd -d lin05 game
9 正在将用户“lin05”从“game”组中删除
10 [lin05@exercise1 ~]$

```

如果你记不住命令，那么直接用root权限修改 vi /etc/passwd 也是一样的

密码管理

```
root:$6$.BTyNB8Q397zR.KY$412.....M8ZHWiidd/:16274: 0 : 99999 : 7 : : :
```

- 1 用户名：密码(REHL5 使用 MD5/RHEL6 使用 SHA512 加密)
- 2 被加密后的密码。
- 3 最近更改密码的日期，从 19700101 开始计算。
- 4 密码不能更改的天数：最近被改过之后几天后才可以再次更改。0 表示随时可以修改。
- 5 密码过期时间，即多少天后必须再次修改。
- 6 密码需要更改期限到来前 7 发出警告。
- 7 宽限天数：密码过了几天后还能改密码。
- 8 帐号过期时间。
- 9 保留。

```

1 [root@exercise1 ~]# head -3 /etc/shadow
2 root:$6$DL36xqQ9Z4lIC96A$TySYbPQmrysstnePos8..QeKSZsAQ.x
   ctrglJvVJDVml40mbhtjezPppJWW6c0PijQweQT5dn9rVaz2rFB9tg.:
   :0:99999:7:::
3 bin:*:17110:0:99999:7:::
4 daemon:*:17110:0:99999:7:::
5 [root@exercise1 ~]#

```

格式如下

- 1 名称 | 说明
- 2 **name** | 登录名称，这个必须是系统中的有效账户名
- 3 **password** | 我们发现所谓的加密算法，其实就是用明文密码和一个叫 **salt** 的东西通过函数 **crypt()** 完成加密。而所谓的密码域密文也是由三部分组成的，即: **\$id\$salt\$encrypted**。[注]:**tid**为1时，采用md5进行加密；**id**为5时，采用SHA256进行加密；**id**为6时，采用SHA512进行加密。
- 4 **lastchange** | 最近一次更改密码的日期，以距离 1970/1/1 的天数表示
- 5 **min-age** | 不能更改密码的最少天数，最近更改过后几天才可以更改；如果为 0 表示“最短期限要求”
- 6 **maxage** | 密码过期时间，必须更改密码前的最多天数
- 7 **warning** | 密码即将到期的警告期，以天数表示，0 表示“不提供警告”
- 8 **inactive** | 宽限天数，密码到期后
- 9 **expire** | 账号过期时间，以距离 1970/1/1 的天数计算（千年虫）
- 10 **blank** | 预留字段

给用户添加密码

注：在默认的情况下，新建账号登陆是暂时封锁的，可以看下/etc/shadow文件的第二个字段

- 1 #交互式
- 2 [root@exercise1 ~]# passwd oracle #交互(这种方式只能root使用)
- 3 更改用户 oracle 的密码。
- 4 新的 密码:
- 5 无效的密码: 密码少于 8 个字符
- 6 重新输入新的 密码:
- 7 passwd: 所有的身份验证令牌已经成功更新。 #root设置密码可以忽略字典
- 8
- 9 1. 超级管理员
- 10 1.1 随意修改任何人的密码
- 11 1.2 密码强度可以随意
- 12

- 13 2.普通用户执行
- 14 2.1 仅修改自己的密码，不可以修改其他人；
- 15 2.2 密码强度必须要高

#非交互式

[root@exercise1 ~]# echo 123456 | passwd --stdin abc #不交互
更改用户 abc 的密码。

passwd: 所有的身份验证令牌已经成功更新。

[root@exercise1 ~]#

#(--stdin 可以通过来自前一个管道的数据，作为密码输入)

创建随机密码

```
1 [root@exrcise1 opt]# mkpasswd
2 -bash: mkpasswd: 未找到命令
3 [root@exrcise1 opt]# yum provides mkpasswd
4 已加载插件: fastestmirror
5 Determining fastest mirrors
6 * base: mirrors.aliyun.com
7 * extras: mirrors.aliyun.com
8 * updates: mirrors.aliyun.com
9 expect-5.45-14.el7_1.x86_64 : A program-script
  interaction and testing utility
10 源      : local
11 匹配来源:
12 文件名   : /usr/bin/mkpasswd
13
14 [root@exrcise1 opt]# yum install -y expect
15 [root@exrcise1 opt]# man mkpasswd
16 [root@exrcise1 opt]# mkpasswd -l 10 -d 3 -c 3 -C 3 -s 1
17 3tjVwt3_v7
```

修改密码

命令: chage

参数:

-m : 密码可更改的最小天数。为 0 时代表任何时候都可以更改密码

-M : 密码保持有效的最大天数

-W : 用户密码到期前, 提前收到警告信息的天数

-E : 帐号到期的日期。过了这天, 此帐号将不可用

-d : 上一次更改的日期, 为 0 表示强制在下次登录时更新密码

-l : 账号到期后, 可以延期的天数

-l: 例出当前的设置。由非特权用户来确定他们的密码或帐号何时过期。

```
1 例子1: 修改用户 abc 密码信息: 让这个用户 abc 首次登录系统时必须
   更改其密码
2
3 [root@exercise1 ~]# chage -m 30 abc    #将两次改变密码之间
   相距的最小天数设为30天, 从最近修改密码的日期开始的5天内, 用户不能
   再次修改密码
4 [root@exercise1 ~]# chage -M 30 abc    #每隔30天    更新密
   码
5 [root@exercise1 ~]# chage -w 7 abc     #还有7天密码到期
6 [root@exercise1 ~]# chage -E '2023-01-01' abc #账号在
   2023年1月1号到期
7 [root@exercise1 ~]# chage -d 0 abc
8 [root@exercise1 ~]# ssh abc@192.168.119.142
9 The authenticity of host '192.168.119.142
   (192.168.119.142)' can't be established.
10 ECDSA key fingerprint is
   SHA256:8aWF+QxUSRyxSCUKFPSJLug+iSZuYpjE8/M4Nl6it38.
11 ECDSA key fingerprint is
   MD5:f2:a9:40:4d:28:ac:bb:38:46:dd:2f:98:a0:c3:6a:7e.
```

```

12 Are you sure you want to continue connecting (yes/no)?
13 yes
14 Warning: Permanently added '192.168.119.142' (ECDSA) to
15 the list of known hosts.
16 abc@192.168.119.142's password:
17 You are required to change your password immediately
18 (root enforced)
19 WARNING: Your password has expired.
20 You must change your password now and login again!  #
21 提示必须改密码
22 更改用户 abc 的密码 。
23 为 abc 更改 STRESS 密码。
24 （当前）UNIX 密码：
25 新的 密码：
26 无效的密码： 这个密码和原来的相同
27 新的 密码：
28 无效的密码： 密码与原来的太相似
29 新的 密码：
30 重新输入新的 密码：
31 抱歉，密码不匹配。
32 passwd: 已经超出服务重试的最多次数
33 Connection to 192.168.119.142 closed.
34
35 [root@exercise1 ~]# chage -I 3 abc #账号到期后，还可以延
36 期3天

```

例子2：帐户密码策略信息

```
[root@exercise1 ~]# chage -l abc
```

最近一次密码修改时间 : 密码必须更改

密码过期时间 : 密码必须更改

密码失效时间 : 密码必须更改

帐户过期时间 : 1月 01, 2023

两次改变密码之间相距的最小天数 : 30

两次改变密码之间相距的最大天数 : 30

在密码过期之前警告的天数 : 7

密码过期：设置的密码经过一段的时间后，系统会认为该密码不安全，于是将密码设置为过期状态，用户登录的时候，系统会提示用户进行密码修改

密码失效：经过一段时间，如果用户没有进行密码修改，则系统会将该密码设置为失效状态（此时用户不可通过该密码进行登录）

用户如何提权

往往公司的服务器对外都是禁止root用户直接登录，所以我们通常使用的都是普通用户，那么问题来了？当我们使用普通用户执行/sbin目录下的命令时，会发现没有权限运行，这种情况下我们无法正常的管理服务器，那如何才能不使用root用户直接登录系统，同时又保证普通用户能完成日常工作？

我们可以使用如下两种方式su、sudo

1.su切换用户，使用普通用户登录，然后使用su命令切换到root。优点简单；缺点：需要知道root密码

2.sudo提权，当需要使用root权限时进行提权，而无需切换至root用户，优点：安全、方便缺点：复杂

用户身份切换

命令：su

语法：su 用户名

```
1 [root@exercise1 ~]# su abc
2 [root@exercise1 ~]# su - abc
```

在使用su切换前，我们需要了解一些预备知识，比如shell分类、环境变量配置文件有哪些

1.Linux Shell主要分为如下几类

登陆shell，需要输入用户名和密码才能进入Shell，日常接触的最多的一种

非登陆shell，不需要输入用户和密码就能进入Shell，比如运行bash会开启一个新的会话窗口

2.bash shell配置文件介绍文件主要保存用户的工作环境

个人配置文件：~/.bash_profile；~/.bashrc。

全局配置文件：/etc/profile；/etc/profile.d/*.sh；/etc/bashrc

profile类文件，设定环境变量，登陆前运行的脚本和命令。

bashrc类文件，设定本地变量，定义命令别名

注意：如果全局配置和个人配置产生冲突，以个人配置为准。

3.登陆系统后，环境变量配置文件的应用顺序是？

登录式shell配置文件执行顺序/etc/profile->/etc/profile.d/*.sh->~/.bash_profile->~/.bashrc->/etc/bashrc

非登陆shell配置文件执行顺序~/.bashrc->/etc/bashrc->/etc/profile.d/*.sh

提问：如何验证

```
1 执行前先备份，不然测试完要删好麻烦
2 echo "echo /etc/profile" >>/etc/profile
3 echo "echo /etc/profile.d/1.sh" >>/etc/profile.d/1.sh
4 echo "echo ~/.bash_profile" >>~/.bash_profile
5 echo "echo ~/.bashrc" >>~/.bashrc
```

```
6 echo "echo /etc/bashrc" >>/etc/bashrc
7
8 [root@exercise1 ~]# su - root
9 /etc/profile.d/1.sh
10 /etc/profile
11 /etc/bashrc
12 /root/.bashrc
13 /root/.bash_profile
14
15 [root@exercise1 ~]# su root
16 /etc/profile.d/1.sh
17 /etc/bashrc
18 /root/.bashrc
```

4.说了这么多预备知识，那这些和su命令切换用户有什么关系？

su - username，属于登陆式shell，su username，属于非登陆式hell，区别在于加载的环境变量不一样。

普通用户su - 可以直接切换root用户，但需要输入root用户的密码。

超级管理员root用户使用su - username切换普通用户不需要输入任何密码。

小结：区别在于，加载的环境变量不一样（有可能会造成程序运行异常）

5.su 还有一种用法

```
1 [root@exercise1 ~]# useradd abc -r -M
2 [root@exercise1 ~]# su - abc -c "ifconfig"
   #指定让abc执行ifconfig命令
3 su: 警告: 无法更改到 /home/abc 目录: 没有那个文件或目录
4 ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu
   1500
5         inet 192.168.245.130  netmask 255.255.255.0
   broadcast 192.168.245.255
```

```
6      inet6 fe80::250:56ff:fe27:3876 prefixlen 64
scopeid 0x20<link>
7      ether 00:50:56:27:38:76 txqueuelen 1000
(Ethernet)
8      RX packets 18885 bytes 6877604 (6.5 MiB)
9      RX errors 0 dropped 0 overruns 0 frame 0
10     TX packets 2123 bytes 232746 (227.2 KiB)
11     TX errors 0 dropped 0 overruns 0 carrier 0
collisions 0
12
13 lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
14     inet 127.0.0.1 netmask 255.0.0.0
15     inet6 ::1 prefixlen 128 scopeid 0x10<host>
16     loop txqueuelen 1 (Local Loopback)
17     RX packets 0 bytes 0 (0.0 B)
18     RX errors 0 dropped 0 overruns 0 frame 0
19     TX packets 0 bytes 0 (0.0 B)
20     TX errors 0 dropped 0 overruns 0 carrier 0
collisions 0
```

小结：因为有些程序不支持root直接启动; 必须是普通用户;

- 1.普通用户不需要登录系统;
- 2.只是用来完成服务的启动;
- 3.创建该用户时,一般都创建为一个系统用户;
- 4.最后通过 `su - username -c "command"` 来完成服务的启动;

提权命令：sudo

sudo可以让你以root的身份来执行命令

```
1 [root@exercise1 ~]# su - abc    #切换身份
```

```

2 上一次登录: 一 1月 17 20:28:22 CST 2022pts/1 上
3 [abc@exercise1 ~]$ tail /etc/shadow #以abc用户读
   取/etc/shadow
4 tail: 无法打开"/etc/shadow" 读取数据: 权限不够
5 [abc@exercise1 ~]$ sudo tail /etc/shadow
6
7 我们信任您已经从系统管理员那里了解了日常注意事项。
8 总结起来无外乎这三点:
9
10     #1) 尊重别人的隐私。
11     #2) 输入前要先考虑(后果和风险)。
12     #3) 权力越大, 责任越大。
13
14 [sudo] abc 的密码:
15 对不起, 请重试。
16 [sudo] abc 的密码:
17 对不起, 请重试。
18 [sudo] abc 的密码:
19 sudo: 3 次错误密码尝试
20 [abc@exercise1 ~]$ sudo tail /etc/shadow
21 abc 不在 sudoers 文件中。此事将被报告。
22
23 [abc@exercise1 ~]$ 登出 #Ctrl+D登出回到root会收到邮件
24 您在 /var/spool/mail/root 中有新邮件
25 [root@exercise1 ~]# cat /var/spool/mail/root #查看邮件

```

```
Date: Mon, 17 Jan 2022 20:33:15 +0800 (CST)
```

```
exercise1 : Jan 17 20:33:15 : abc : user NOT in sudoers ; TTY=pts/1 ; PWD=/opt/abc ; USER=root ; COMMAND=/bin/tail /etc/shadow
```

小结: 通过sudo执行的命令, 都会记录在/var/log/secure文件里

解决方法

用root用户使用visudo去修改/etc/sudoers ==> vim /etc/sudoers

1	用户账号	登陆者的来源主机ip地址	可切换的身份	可执行的命令
2	92 root	ALL=	(ALL)	ALL
3				


```

4  99 %wheel  ALL=(ALL)      ALL      #这一行的设置会造成加入
   wheel组（可自定义用户组）的用户都可以
5
   使用sudo切换身份进行操作
6
7  1.快速配置sudo方式
8  [root@exercise1 ~]# usermod abc -G wheel
9
10 2.一般正常配置sudo方式
11 [root@exercise1 ~]# visudo
12 93 abc          ALL=(ALL)      ALL      #添加普通用户权
   限
13 abc          192.168.245.0=(ALL)  ALL      #设置某一网段能使
   用sudo
14 abc          ALL=(ALL)      /usr/sbin/useradd      #允
   许使用sudo执行指定命令
15 abc          ALL=(ALL)
   NOPASSWD:/usr/sbin/useradd      #NOPASSWD不需要使用密码
16 abc          ALL=(root)      ALL      #只可以切换成
   root用户
17
18 修改后语法检测：
19 [root@exercise1 ~]# visudo -c
20
21 测试：
22 [root@exercise1 ~]# mkdir /test
23
24 3.普通用户正常情况下是无法删除test目录的
25 [abc@exercise1 ~]$ rm -rf /test
26
27 4.使用sudo提权，需要输入普通用户的密码
28 sudo rm -rf /test

```

5.查看自己有哪些sudo权限

```
[abc@exercise1 ~]$ sudo -l
```

提问：提升的权限太大，能否有办法限制仅开启某个命令的使用权限？其他命令不允许？

方法1：使用sudo中自带的别名操作，将多个用户定义成一个组

```
1 # 1.使用sudo定义分组,这个系统group没什么关系
2 User_Alias OPS = abc,xiaohuo
3
4
5 # 2.定义可执行的命令组,便于后续调用
6 Cmnd_Alias NETWORKING = /sbin/ifconfig, /bin/ping
7 Cmnd_Alias SOFTWARE = /bin/rpm, /usr/bin/yum
8 Cmnd_Alias SERVICES = /sbin/service, /usr/bin/systemctl
   start
9 Cmnd_Alias STORAGE = /bin/mount, /bin/umount
10
11
12 # 3.使用sudo开始分配权限
13 OPS    ALL=(ALL) NETWORKING, SOFTWARE, SERVICES, STORAGE
14
```

方法2：使用groupadd添加组，然后给组分配sudo的权限，如果有新用户加入，直接将用户添加到该组

```
1 #1.添加真实的系统组
2 [root@exercise1 ~]# groupadd OPS
3
4 #2.使用sudo开始分配权限
5 [root@exercise1 ~]# visudo
6 %OPS    ALL=(ALL)          ALL          #添加组，分配sudo权限
7
8 #3.添加用户加入OPS组
9 [root@exercise1 ~]# usermod ming -G OPS
```