

用户账户与组账户、文件权限及归属管理

一、用户账号管理

1、用户账号概述

2.添加用户账号useradd命令 注意:在Ubuntu系列系统中 adduser

3.用户账号的初始配置文件

4.设置/更改用户口令passwd命令

5.修改用户属性usermod命令

6.删除用户账号 userdel命令

二、组账号管理

1.组账号概述

2.添加组账号 groupadd命令

3.设置组账号密码（极少用），添加、删除组成员 gpasswd命令

三、查询命令——id、groups、finger、w、whoami、who

1.id命令

2.groups命令

3.finger命令（需安装）

4.w命令

5.whoami

6.who

7.纯手工创建用户

四、权限及归属原理

1.基本访问权限

2.归属（所有权）

3.查看文件的权限和归属

4.各权限的字母及8进制表示

5.设置文件或目录的权限 chmod命令

7.权限掩码 umask

8.脚本---循环

Linux是一个多用户，多任务多进程的服务器操作系统

用户角色：超级用户（管理员），普通用户，程序用户

一、用户账号管理

1、用户账号概述

用户：使用者在计算机内部的身份标识

用户账号的常见分类

超级用户：root uid=0 gid=0 权限最大（使用需要严谨）

普通用户：1001<=uid<=60000 受到权限限制，一般在宿主目录下有完整权限

程序用户：1=<uid<=999 应用程序运行时需要通过用户身份获取相应的系统资源，通常不能用于登录系统或管理系统

用户账号文件：

/etc/passwd

作用：保存用户名称、宿主目录、登录Shell等基本信息，每一行对应一个用户的账号记录

```
[root@localhost ~]# useradd mysql
[root@localhost ~]# echo "123456" | passwd --stdin mysql
```

更改用户 mysql 的密码。

passwd：所有的身份验证令牌已经成功更新。

```
[root@localhost ~]# tail -1 /etc/passwd
```

```
mysql:x:1000:1000::/home/mysql:/bin/bash
```

共7个字段，各字段代表含义：（面试）

第一字段：用户账号名

第二字段：密码占位符（密码保存到了影子文件）

第三字段：uid编号

第四字段：gid编号

第五字段：用户备注信息（用户全名）

第六字段：用户宿主目录（家目录）

第七字段：登录时分配到的Shell解释器（若shell类型为/sbin/nologin则不能登录）

/etc/shadow 影子文件

作用：保存用户的密码、账号有效期等信息，每一行对应一个用户的密码记录

```
[root@localhost ~]# tail -1 /etc/shadow
```

```
mysql:$6$Dk21okj.$Qw93VXjsJLAgn3FYPFomdBI7ThzjV1U7Ydreg/NeACmsqVGEYWZFegXBmoE
trlcikGsvbKEpDpkR07dEgXi.:17881:0:99999:7:::
```

扩展：

- 密码复杂度：3/4原则：大写字母 小写字母 数字 特殊符号
- 长度：8位
- 密码破解：字典 穷举（暴力破解）

共9字段，目前只启用8字段，各字段代表的含义：

第一字段：用户账号名

第二字段：加密的密码，！！表示密码被锁定

第三字段：上次修改密码的时间，距1970-01-01过去多少天

第四字段：密码最短有效期（距上次密码修改起多少天内不能再次修改）单位“天”。“0”表示随时可修改密码。

第五字段：密码最长有效期（在修改密码后的多少天必须重新修改密码。99999表示永久可以使用）

第六字段：提前多少天警告用户口令将过期（7表示在密码过期前7天开始警告）

第七字段：在密码过期之后多少天禁用此用户

第八字段：密码过期日期，若设置则显示为过期日期距1970年1月1日多少天

第九字段：保留字段（未使用）

```
备份文件    /etc/passwd    /etc/shadow
[root@localhost ~]# ls -l /etc/passwd*
-rw-r--r--. 1 root root 2532 2月  25 01:18 /etc/passwd
-rw-r--r--. 1 root root 2487 2月  22 10:34 /etc/passwd-
[root@localhost ~]# ls -l /etc/shadow*
-----. 1 root root 1379 2月  25 07:13 /etc/shadow
-----. 1 root root 1349 2月  22 10:34 /etc/shadow-
[root@localhost ~]# rm -rf /etc/passwd    删除
[root@localhost ~]# cp /etc/passwd- /etc/passwd    备份
```

2.添加用户账号useradd命令

注意:在Ubuntu系列系统中：adduser

格式：useradd[选项]...用户名

常用选项：

- **-u: 指定uid标记号**
- **-d: 指定宿主目录，默认为/home/用户**
- **-e: 指定账号失效时间**
- **-M: 不为用户建立初始化宿主目录（创建程序目录的时候使用）**
- **-s: 指定用户的登录shell，与-M配合使用 /sbin/nologin 无法登陆**
- **-g:指定用户的基本组名（或gid号），系统中创建用户时，默认会创建一个同名的基本组**
- **-G: 指定用户的附加组名（或gid号）**
- **-c:添加备注，显示在/etc/passwd第五字段**

```
[root@localhost ~]# useradd tomcat
[root@localhost ~]# tail -2 /etc/passwd
mysql:x:1000:1000::/home/mysql:/bin/bash
tomcat:x:1001:1001::/home/tomcat:/bin/bash
```

```
[root@localhost ~]# useradd -u 1010 nginx
[root@localhost ~]# tail -3 /etc/passwd
mysql:x:1000:1000::/home/mysql:/bin/bash
tomcat:x:1001:1001::/home/tomcat:/bin/bash
nginx:x:1010:1010::/home/nginx:/bin/bash
```

```
[root@localhost ~]# mkdir /bigdata
[root@localhost ~]# useradd -d /bigdata/hadoop hadoop
[root@localhost ~]# tail -1 /etc/passwd
hadoop:x:1011:1011::/bigdata/hadoop:/bin/bash
```

`useradd -d` 这条命令只能帮助创建最后一层目录，例如上面的/bigdata/hadoop，我们要事先准备好/bigdata这个目录

```
[root@localhost ~]# useradd -M -s /sbin/nologin spark
```

```
[root@localhost ~]# tail -1 /etc/passwd
```

```
spark:x:1012:1012::/home/spark:/sbin/nologin
```

```
[root@localhost ~]# ls /home/
```

```
mysql nginx tomcat
```

```
[root@localhost ~]# useradd -e "2020-12-30" zhanglong
```

```
[root@localhost ~]# tail -3 /etc/shadow
```

```
hadoop:!!:17881:0:99999:7::: I
```

```
spark:!!:17881:0:99999:7:::
```

```
zhanglong:!!:17881:0:99999:7::18626:
```

3.用户账号的初始配置文件

文件来源：新建用户账号时，用户宿主目录内容从/etc/skel/目录中复制而来

主要的用户初始配置文件-->脚本（只对当前用户生效，不同的时机执行）：

`~/.bash_profile`:每次登陆时执行

`~/.bashrc`:每次进入新bash环境时执行

`~/.bash_logout`:每次退出登陆时执行

`~/.bash_history`:用户登陆时从该文件加载历史命令记录

`history` 查看历史命令信息

`history -c` : 清除历史命令信息

历史命令引用：

! 序号

! 命令开头

Ctrl+r

全局初始配置文件，对所有用户生效

/etc/bashrc

/etc/profile

注意：/etc/profile和~/.bash_profile冲突了，~/.bash_profile的配置覆盖全局文件配置

4.设置/更改用户口令

passwd命令

格式：passwd[选项]...[用户名]

不指定用户名，默认修改的是当前登录用户的密码，普通用户可以修改自己的密码，root可以修改任意用户的密码

常用选项：

-stdin: 通过前一个管道的数据作为密码输入，对shell script有用。

-l: lock 锁定的意思，将 /etc/shadow的第二列前面加!，使密码失效。

-u: unlock 解锁的意思，与l相反。

S: 列出密码相关的信息。 查看用户账号的状态（是否被锁定）

nxwi: 接天数, 分别shadow的第4567段。

-d: 清空用户密码 (和不设置密码不一样)
-x, --maximum=DAYS: 密码的最长有效时限
-n, --minimum=DAYS: 密码的最短有效时限
-w, --warning=DAYS: 在密码过期多少天开始提醒用户
-i, --inactive=DAYS: 当密码过期后经过多少天该账号会被禁用

```
[root@localhost ~]# grep "tomcat" /etc/shadow
```

```
tomcat:!!:17881:0:99999:7:::
```

```
[root@localhost ~]# passwd tomcat
```

更改用户 tomcat 的密码

新的密码:

无效的密码: 密码少于8个字符

重新输入新的密码:

passwd: 所有身份验证令牌已经成功更新

```
[root@localhost ~]# grep "tomcat" /etc/shadow
```

```
tomcat:$6$xaEBADad$u6yhhyhHQ/g5Cn6xWOv.W.myG32FkHbBPegsl0cyl2bCDfYJO8mZVo8lnQ8
```

```
QAzF/k7/zYv7xkLnn06gmSjdT31:17881:0:99999:7:::
```

```
[root@localhost ~]# passwd -S tomcat
```

```
tomcat PS 2018-12-16 0 99999 7 -1 (密码已设置, 使用 SHA512 算法。)
```

```
[root@localhost ~]# passwd -l tomcat
```

锁定用户 tomcat 的密码。

passwd: 操作成功

```
[root@localhost ~]# grep "tomcat" /etc/shadow
```

```
tomcat:!!$6$xaEBADad$u6yhhyhHQ/g5Cn6xWOv.W.myG32FkHbBPegsl0cyl2bCDfYJO8mZVo8lnQ8
```

```
QAzF/k7/zYv7xkLnn06gmSjdT31:17881:0:99999:7:::
```

```
[root@localhost ~]# passwd -S tomcat
```

```
tomcat LK 2018-12-16 0 99999 7 -1 (密码已被锁定。)
```

```
[root@localhost ~]#
```

```
[root@localhost ~]# passwd -u tomcat
```

解锁用户 tomcat 的密码。

passwd: 操作成功

5.修改用户属性usermod命令

格式: usermod[选项]... 用户

常用选项:

-l: 更改用户账号的登录名字

-c: 修改用户备注

-L: 锁定用户账号

-S: 锁定用户

-U: 解锁用户账户 (passwd -l 锁定的用户, 通过usermod -U解锁两次)

以下选项与useradd命令中的含义相同: -u、-d、-e、-s、-c、-g、-G

```
[root@localhost ~]# grep "mysql" /etc/passwd
```

```
mysql:x:1000:1000::/home/mysql:/bin/bash
```

```
[root@localhost ~]# usermod -l mariadb mysql
```

```
[root@localhost ~]# grep "mysql" /etc/passwd
```

```
mariadb:x:1000:1000::/home/mysql:/bin/bash
```

锁定用户: `passwd -S 用户`

```
[root@localhost ~]# passwd -S mariadb
mariadb PS 2018-12-16 0 99999 7 -1 (密码已设置, 使用 SHA512 算法。)
[root@localhost ~]# usermod -L mariadb
[root@localhost ~]# passwd -S mariadb
mariadb LK 2018-12-16 0 99999 7 -1 (密码已被锁定。)
[root@localhost ~]# usermod -U mariadb
[root@localhost ~]# passwd -S mariadb
mariadb PS 2018-12-16 0 99999 7 -1 (密码已设置, 使用 SHA512 算法。)
```

6.删除用户账号 `userdel`命令

格式: `userdel 用户名`

常用选项: `-r` :删除用户的同时删除用户的宿主目录

```
[root@localhost ~]# ls /home/
mysql nginx tomcat zhanglong
[root@localhost ~]# userdel zhanglong
[root@localhost ~]# ls /home/
mysql nginx tomcat zhanglong
[root@localhost ~]# userdel -r tomcat
[root@localhost ~]# ls /home/
mysql nginx zhanglong
```

7. 关于用户及密码相关控制文件:

```
[root@localhost ~]# grep -v -E "^$|^#" /etc/login.defs
```

MAIL_DIR	/var/spool/mail	#用户邮件存放目录
PASS_MAX_DAYS	99999	#密码默认最长有效期
PASS_MIN_DAYS	0	#密码默认最短有效期
PASS_MIN_LEN	5	#密码默认长度
PASS_WARN_AGE	7	#密码过期警告时间
UID_MIN	1000	#普通用户起始UID范围
UID_MAX	60000	#普通用户结束UID范围
SYS_UID_MIN	201	#系统用户起始UID范围
SYS_UID_MAX	999	#系统用户结束UID范围
GID_MIN	1000	#普通组起始GID范围
GID_MAX	60000	#普通组结束GID范围
SYS_GID_MIN	201	#系统组起始GID范围
SYS_GID_MAX	999	#系统组结束GID范围
CREATE_HOME	yes	#是否创建用户宿主目录
UMASK	007	#用户宿主目录默认权限
USERGROUPS_ENAB	yes	#表示userdel删除用户时, 如果该用户用户组如果没有成员存在, 则会删除该用户组
ENCRYPT_METHOD	SHA512	#表示用户密码加密方式, 此处表示用MD5加密密码

```
[root@localhost ~]# grep -v -E "^$|^#" /etc/default/useradd
```

GROUP=100	
HOME=/home	#吧用户的主目录建在/home中
INACTIVE=-1	#是否启用账号过期停权, -1表示不启用

EXPIRE= #账号终止日期。不设置表示不启用
SHELL=/bin/bash #所用SHELL的类型
SKEL=/etc/skel #默认添加用户开的目录默认文件存放位置；也就是说，当我们用adduser添加用户时，用户家目录下的文件，都是从这个目录中复制过去的

/lib64/security/pam_cracklib.so

控制密码复杂的关键文件，Redhat公司专门开发了cracklib这个安装包来判断密码的复杂度

[root@localhost ~]# man pam_cracklib

retry=N #改变输入密码的次数，默认值是1。就是说，如果说，如果用户输入的密码强度不够就退出。可以使用这个选项设置输入的次数，以免一切都从头再来

minlen=N #新密码最低可接受的长度

difok=N #默认值为10。这个参数设置允许的新、旧密码相同字符的个数。不过，如果新密码中1/2的字符和旧密码不同，则新密码被接受

dcredit=N #限制新密码中至少有多少个数字

ucredit=N #限制新密码中至少有多少个大写字符

lcredit=N #限制新密码中至少有多少个小写字符

二、组账号管理

1.组账号概述

组：用户集合，组的存在便于管理多个用户权限

组账号分类：

基本组（私有组）

附加组（公共组）

组账号文件

/etc/group:保存组账号基本信息

/etc/gshadow:保存组账号的密码信息（较少使用）

[root@localhost ~]# head -1 /etc/group

root:x:0:zhangsan

第1个字段：组名

第2个字段：密码占位符

第3个字段：GID

第4个字段：组内的成员信息

[root@localhost ~]# head -1 /etc/gshadow

root:::

2.添加组账号 groupadd命令

格式：groupadd [-g GID] 组账号名

常用选项：-g GID

[root@localhost ~]# groupadd students

[root@localhost ~]# groupadd class

[root@localhost ~]# tail -2 /etc/group

students:x:1013:

class:x:1014:

3.设置组账号密码（极少用），添加、删除组成员 gpasswd命令

格式：gpasswd [选项]... 组账号名

常用选项：

-a: 向组内添加一个用户

-d: 从组内删除一个用户成员

-M: 定义组成员列表，以逗号分隔

增加或删除组成员，也可以用vi编辑器对/etc/group 文件直接编译修改

```
[root@localhost ~]# grep "class" /etc/group
class:x:1014:
```

```
[root@localhost ~]# gpasswd -a nginx class
```

正在将用户“nginx”加入到“class”组中

```
[root@localhost ~]# gpasswd -a hadoop class
```

正在将用户“hadoop”加入到“class”组中

```
[root@localhost ~]# grep "class" /etc/group
class:x:1014:nginx,hadoop
```

```
[root@localhost ~]# gpasswd -d hadoop class
```

正在将用户“hadoop”从“class”组中删除

```
[root@localhost ~]# grep "class" /etc/group
class:x:1014:nginx
```

I

```
[root@localhost ~]# gpasswd -M nginx,hadoop,spark,mariadb class
```

```
[root@localhost ~]# grep "class" /etc/group
```

```
class:x:1014:nginx,hadoop,spark,mariadb
```

4. 删除组账号 groupdel命令

格式：groupdel 组账号名

```
[root@localhost ~]# groupdel students
```

三、查询命令——id、groups、finger、w、whoami、who

1.id命令

作用：查询用户身份标识

格式：id [用户名]

```
[root@localhost ~]# id root
```

```
uid=0(root) gid=0(root) 组=0(root) I
```

```
[root@localhost ~]# id nginx
```

```
uid=1010(nginx) gid=1010(nginx) 组=1010(nginx),1014(class)
```

2.groups命令

作用：查询客户所属的组

格式：groups [用户名]

```
[root@localhost ~]# groups root
```

```
root : root
```

```
[root@localhost ~]# groups nginx
```

```
nginx : nginx class
```

3.finger命令（需安装）

作用：查询账号的详细信息

格式: finger [用户名]

```
[root@localhost ~]# finger root
```

```
Login: root          Name: root
```

```
Directory: /root     Shell: /bin/bash
```

```
On since 日 12月 16 21:27 (CST) on tty1  28 minutes,57 seconds idle
```

```
On since 日 12月 16 21:27 (CST) on pts/0 from 192.168.200.2
```

```
1 second idle
```

```
On since 日 12月 16 21:33 (CST) on pts/1 from 192.168.200.2
```

```
23 minutes 23 seconds idle
```

```
No mail.
```

```
No Plan.
```

4.w命令

uptime

作用: 查询已登录到主机的用户信息

```
[root@localhost ~]# w
```

```
21:57:33 up 35 min, 3 users, load average: 0.00, 0.01, 0.05
```

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

root	tty1		21:27 29:57	0.17s	0.17s		-bash
------	------	--	-------------	-------	-------	--	-------

root	pts/0	192.168.200.2	21:27 5:00s	0.57s	0.01s		w
------	-------	---------------	-------------	-------	-------	--	---

root	pts/1	192.168.200.2	21:33 24:23	0.14s	0.14s		-bash
------	-------	---------------	-------------	-------	-------	--	-------

5.whoami

作用: 查询当前登录的账号名

示例:

```
[root@localhost ~]# whoami
```

```
root
```

6.who

作用: 与w命令类似, 查询已登录到主机的用户

示例:

```
[root@localhost ~]# who
```

```
root tty1 2018-12-16 21:27
```

```
root pts/0 2018-12-16 21:27 (192.168.200.2)
```

```
root pts/1 2018-12-16 21:33 (192.168.200.2)
```

7.纯手工创建用户

```
[root@localhost ~]# vim /etc/passwd
```

```
[root@localhost ~]# vim /etc/shadow
```

```
[root@localhost ~]# vim /etc/group
```

```
[root@localhost ~]# cp -r /etc/skel/ /home/u10
```

ps:

清退用户

```
fuser -k /dev/pts/1
```

```
fuser -k /dev/tty1
```

load average: 0.00, 0.01, 0.05, 系统负载平均值, 分别表示1分钟内, 5分钟内, 15分钟内, 值和CPU核心做对比

命令集锦：

useradd passwd usermod userdel groupadd groupdel groups id finger whoami who w uptime fuser

非交互设置密码：echo "123456" | passwd --stdin ul

四、权限及归属原理

1.基本访问权限

读权限r：针对目录可以查看目录的列表（ls），针对文件可以查看文件内容（cat more less head tail）

写权限w：针对目录可以改动目录的列表内容（rm touch mkdir cp mv），针对文件可以修改文件内容（vim sed）

可执行x：针对目录可以切换（cd），针对文件允许运行程序（二进制文件，脚本文件）

2.归属（所有权）

属主：拥有该文件的用户账号

属组：拥有该文件的组账号

3.查看文件的权限和归属

第一位表示文件类型

-：表示一般文件

d：表示目录

l：表示软链接（快捷方式）

p：表示PIPE管道文件

s：表示socket通信套接字文件

c：表示字符设备文件

b：表示块设备文件

4.各权限的字母及8进制表示

权限	字母表示	8进制	描述
读权限	r	4	
写权限	w	2	
可执行	x	1	
无权限	-	0	

drwxr-xr-x 的意思是一个权限为755的目录

-rw-r--r-- 的意思是一个权限为644的文件

目录满权限：777，默认是755

文件满权限：666.，默认是644

5.设置文件或目录的权限 chmod命令

格式：

chmod [-R] [ugoa] [+|=] [rwx] 参数

chmod [-R] [nnn] 参数

常用选项:

-R: 表示以递归的方式设置目录及目录下的所有子目录及文件的权限

u: 属主

g: 属组

o: 其他人

a: 所有人

+: 添加

-: 删除

=: 重置

nnn: 8进制的数字权限, 如: 777 666 644 755等

```
[root@localhost ~]# echo "hello,student" > stu.txt
```

```
[root@localhost ~]# ls -l stu.txt
```

```
-rw-r--r-- 1 root root 14 12月 16 22:55 stu.txt
```

```
[root@localhost ~]# chmod u+x,g+w stu.txt
```

```
[root@localhost ~]# ls -l stu.txt
```

```
-rwxrw-r-- 1 root root 14 12月 16 22:55 stu.txt
```

```
[root@localhost ~]# chmod a+x stu.txt
```

```
[root@localhost ~]# ls -l stu.txt
```

```
-rwxrwxr-x 1 root root 14 12月 16 22:55 stu.txt
```

```
[root@localhost ~]# chmod 644 stu.txt
```

```
[root@localhost ~]# ls -l stu.txt
```

```
-rw-r--r-- 1 root root 14 12月 16 22:55 stu.txt
```

```
[root@localhost ~]# mkdir /test
```

```
[root@localhost ~]# mkdir /test/dir1
```

```
[root@localhost ~]# touch /test/file1
```

```
[root@localhost ~]# ls -ld /test/
```

```
drwxr-xr-x 3 root root 31 12月 16 23:01 /test/
```

```
[root@localhost ~]# ls -l /test/
```

总用量 0

```
drwxr-xr-x 2 root root 6 12月 16 23:00 dir1
```

```
-rw-r--r-- 1 root root 0 12月 16 23:01 file1
```

```
[root@localhost ~]# chmod -R 700 /test/
```

```
[root@localhost ~]# ls -ld /test/
```

```
drwx----- 3 root root 31 12月 16 23:01 /test/
```

```
[root@localhost ~]# ls -l /test/
```

总用量 0

```
drwx----- 2 root root 6 12月 16 23:00 dir1
```

```
-rwx----- 1 root root 0 12月 16 23:01 file1
```

注意: 在建立目录时用-m选项可以直接设置新建目录的权限

6. 设置文件的归属 chown命令

格式:

chown 属主 文件

chown :属组 文件 = chown. 属组 文件 = chgrp 属组

chown 属主:属组 文件

chown -R 递归

```
[root@localhost ~]# chown nginx stu.txt
[root@localhost ~]# ls -l stu.txt
-rw-r--r-- 1 nginx root 14 12月 16 22:55 stu.txt
[root@localhost ~]# chown :class stu.txt
[root@localhost ~]# ls -l stu.txt
-rw-r--r-- 1 nginx class 14 12月 16 22:55 stu.txt
[root@localhost ~]# chown root:root stu.txt
[root@localhost ~]# ls -l stu.txt
-rw-r--r-- 1 root root 14 12月 16 22:55 stu.txt
```

```
[root@localhost ~]# ls -ld /test/
drwx----- 3 root root 31 12月 16 23:01 /test/
[root@localhost ~]# ls -l /test/
总用量 0
drwx----- 2 root root 6 12月 16 23:00 dir1
-rwx----- 1 root root 0 12月 16 23:01 file1
```

7.权限掩码 umask

- 1. 作用：控制新建的文件或目录的权限
- 2. umask值与新建文件、目录权限对照表

umask值	目录权限值	文件权限值
0	7	6
1	6	6
2	5	4
3	4	4
4	3	2
5	2	2
6	1	0
7	0	0

例如:

umask值为0000，则目录权限值777，文件权限值666

umask值为0022，则目录权限值755，文件权限值644

```
[root@localhost ~]# umask
0022
[root@localhost ~]# 777-022=755 ^C
[root@localhost ~]# 666-022=644 ^C
[root@localhost ~]# █
```

- 3. 查看umask值

格式: umask

```
[root@localhost ~]# umask  
0022
```

4. 设置umask值

格式: umask nnn

```
[root@localhost ~]# umask 0044  
[root@localhost ~]# umask  
0044
```

8.脚本---循环

建20个普通用户

```
[root@localhost ~]# cat user.txt  
for i in user{1..20}  
do  
useradd $i  
echo "123456" | passwd --stdin $i  
done
```

倒计时

```
for x in {10..1}  
do  
    echo $x  
    sleep 1  
done
```

输入文字

```
for x in {10..1}  
do  
    echo "开小灶啦"  
    sleep 1  
done
```