Linux权限的概念

- ps axj | grep bash 查看进程
- ps axj就可以查看所有进程，who可以查看有谁在用，每一次登录系统都会自动创建一个bash

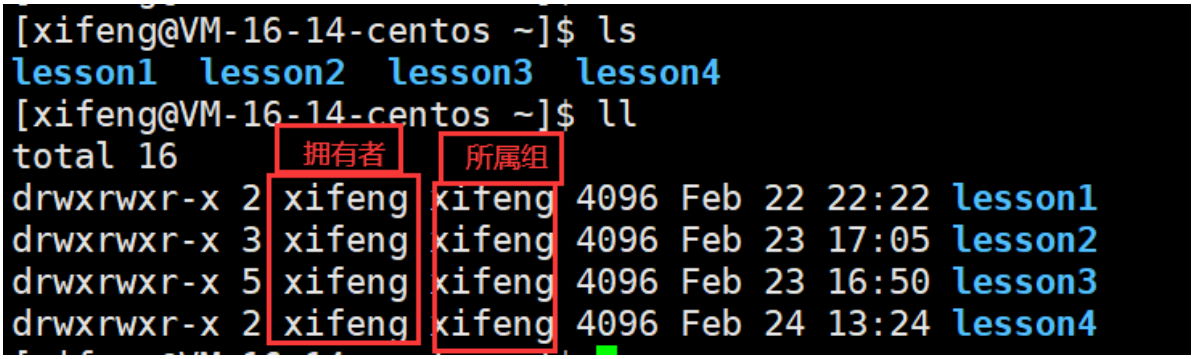
1.Linux中，默认有两类用户

- root：超级管理员，具有非常高的权限
- 普通用户：具有一般权限，需要受权限约束的(root创建的用户)
- su - 普通转换成超级管理员并且返回根目录（root密码）
- su 普通用户转换为超级管理员，在当前位置不发生变化，也就是在哪里su就在哪里
- 用户回退：建议exit或者ctrl+d，不建议su - 用户
- sudo 临时权限提升，执行后续命令，以root身份执行（要的是普通用户的密码）

2.理解什么是权限

- 权限约束的是人，文件本身就有的天然的权限属性 r + w +x
- 简单来说：一件事情是否允许被特定的人做（权限=人+事物的属性）

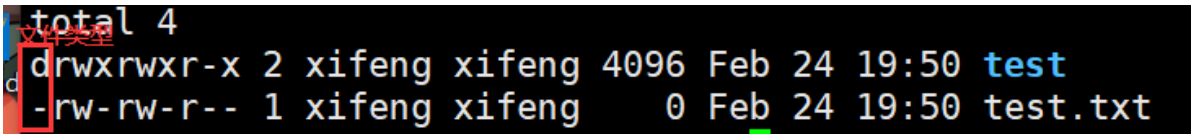
3.Linux 中的用户类别：



- 拥有者 owner
- 所属组 group
  - 主要就是方便能让自己和同组的人看到，而不然其他人看到
- 其他 other
- 拥有者，所属组，other：指的是一种角色身份，而root和普通用户是一个确定的人

4.标识文件类型

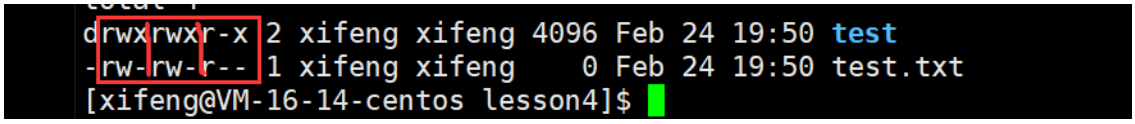
1. 第一列是文件类型



- -: 普通文件 [文本，各种动静态库，可执行文件，源程序]
- d：目录文件
- c：字符设备文件（主要键盘与显示器）:简单查看/dev/tty
- b：块设备文件 （主要磁盘df -h可以查看磁盘大小）:/dev/vda1
- p：管道文件（主要通信）
- l：链接文件（软链接）

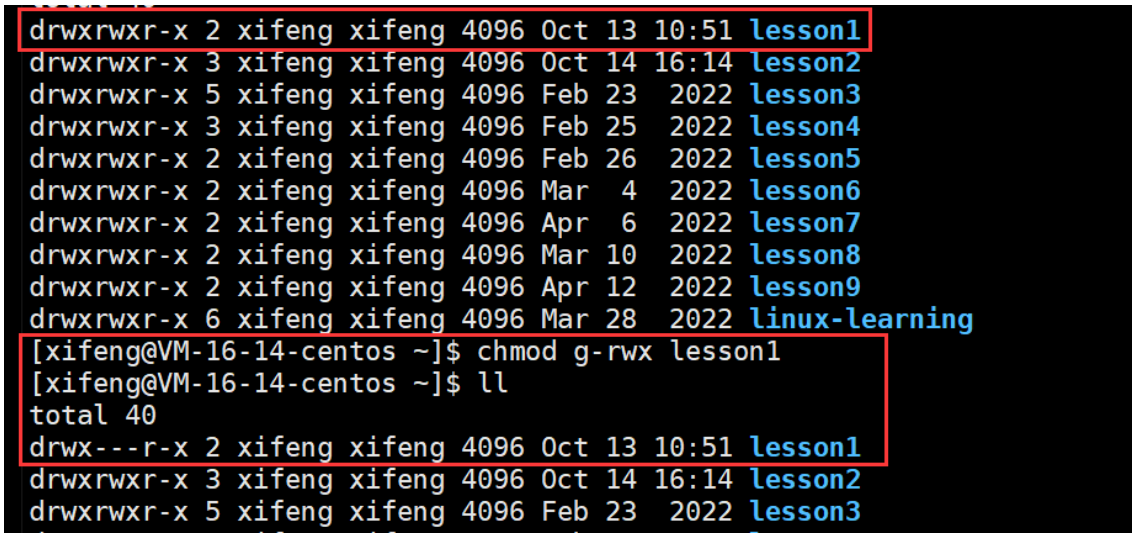
Linux下一切皆文件，Linux系统中，不以文件后缀区分文件类型，gcc，g++....都是系统上的命令

2. 后面九列字符

- 
```
drwxrwxr-x 2 xifeng xifeng 4096 Feb 24 19:50 test
-rw-rw-r-- 1 xifeng xifeng    0 Feb 24 19:50 test.txt
[xifeng@VM-16-14-centos lesson4]$
```
    - 后面九列三三为一组，分别对应拥有者，所属组，其他人的权限。
    - 举例：r(是否具有读)w(是否具有写)-(是否具有可执行)： 这就代表可以具有读写权限，但是没有执行权限

5.修改权限

- **chmod 修改文件权限（读写执行，且是永久生效）**


```
drwxrwxr-x 2 xifeng xifeng 4096 Oct 13 10:51 lesson1
drwxrwxr-x 3 xifeng xifeng 4096 Oct 14 16:14 lesson2
drwxrwxr-x 5 xifeng xifeng 4096 Feb 23  2022 lesson3
drwxrwxr-x 3 xifeng xifeng 4096 Feb 25  2022 lesson4
drwxrwxr-x 2 xifeng xifeng 4096 Feb 26  2022 lesson5
drwxrwxr-x 2 xifeng xifeng 4096 Mar  4  2022 lesson6
drwxrwxr-x 2 xifeng xifeng 4096 Apr  6  2022 lesson7
drwxrwxr-x 2 xifeng xifeng 4096 Mar 10  2022 lesson8
drwxrwxr-x 2 xifeng xifeng 4096 Apr 12  2022 lesson9
drwxrwxr-x 6 xifeng xifeng 4096 Mar 28  2022 linux-learning
[xifeng@VM-16-14-centos ~]$ chmod g-rwx lesson1
[xifeng@VM-16-14-centos ~]$ ll
total 40
drwx---r-x 2 xifeng xifeng 4096 Oct 13 10:51 lesson1
drwxrwxr-x 3 xifeng xifeng 4096 Oct 14 16:14 lesson2
drwxrwxr-x 5 xifeng xifeng 4096 Feb 23  2022 lesson3
```
    - chmod u(改的是拥有者)+(增加权限) / -(删除权限) file_name(文件名)
    - chmod g(改的是所属组)+/- file_name
    - chmod o(改的是其他人)+/- file_name
    - 可以连着修改的例如：chmod u+rwx,g+rwx,o+rwx file_name
    - chmod a(代表all就是所有的都一起加) +/- file_name

对于非法请求，外壳程序是有权利直接拒绝的

shell:

1. 传递请求指令，让OS执行命令
2. 保护内核(创建子进程来执行具有风险的事情)

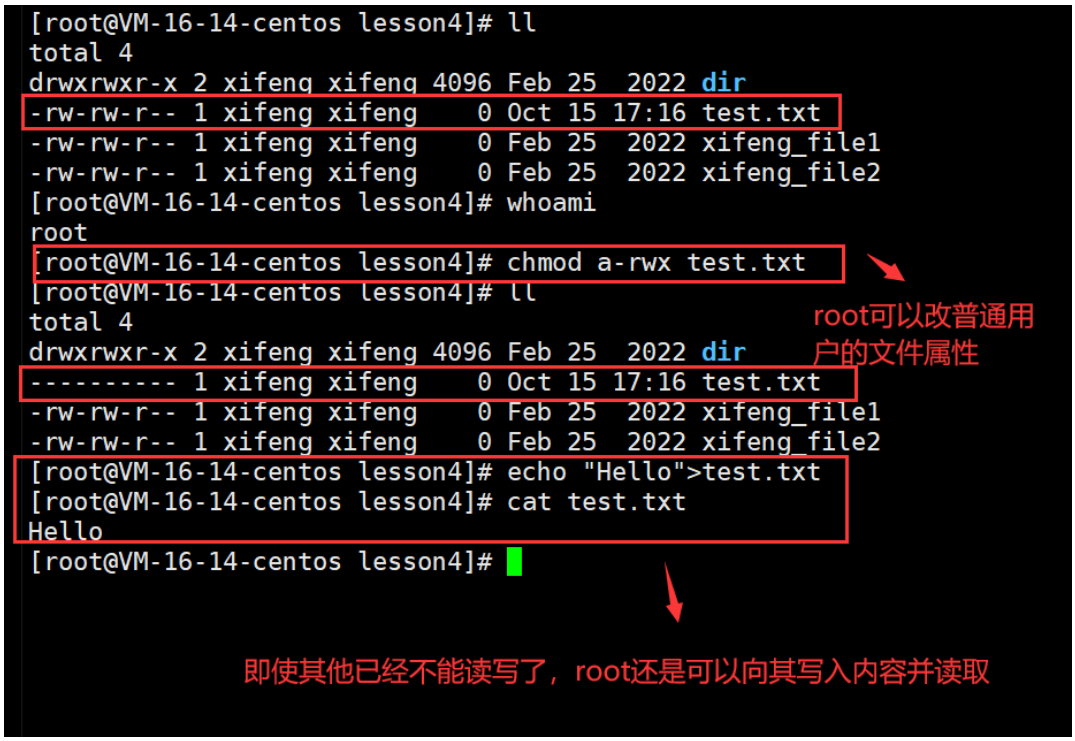
---

su如果不加-也可以切换root用户，跟su -的区别就是su -切换后返回根目录，而su还是在当前切换的目录

5.修改文件权限

- **chmod 修改文件权限（读写执行，且是永久生效）**
    - chmod u(改的是拥有者)+(增加权限) / -(删除权限) file_name(文件名)
    - chmod g(改的是所属组)+/- file_name
    - chmod o(改的是其他人)+/- file_name
    - 可以连着修改的例如：chmod u+rwx,g+rwx,o+rwx file_name
    - chmod a(代表all就是所有的都一起加) +/- file_name

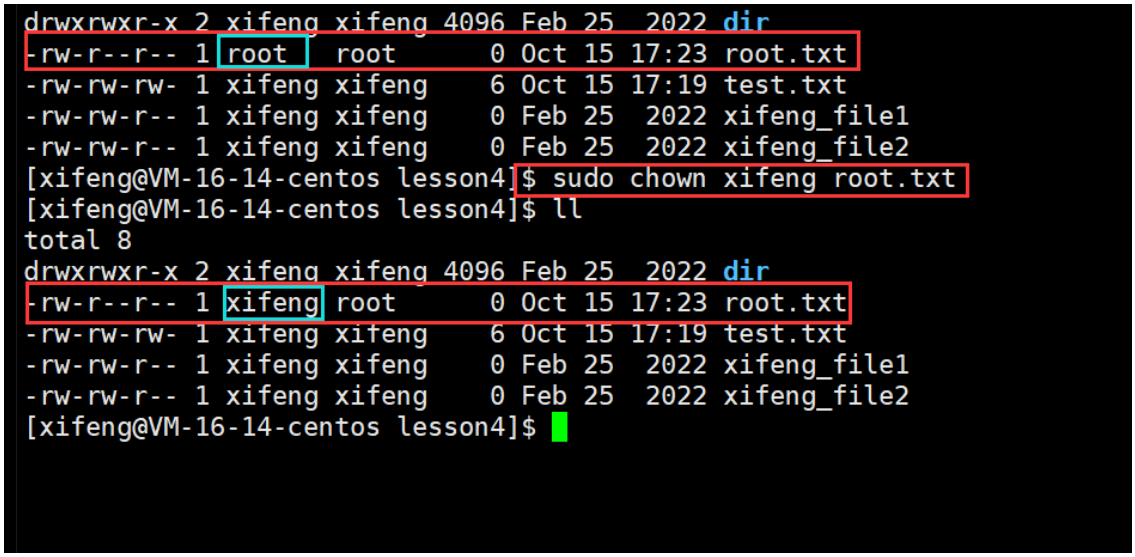
- **root用户是几乎不受权限约束的**，权限是用来约束普通用户的



- **权限的位置是确定的而且是两态的**
  - 拥有者权限的修改还可以用八进制来修改
  - 原因：首先因为权限位置是确定的，是否就可以用0/1序列来表示
    - 例如：-rw-r--r-- 转换的话可以转化为1 1 0 1 0 0 1 0 0，然后能换算成八进制为6 4 4
    - 改法：chmod 664 file_name

6.修改文件拥有者

- **chown 修改文件拥有者**



  - chown user(要给的用户) file_name
  - 但是如果是普通用户，这种更改拥有者的操作是不被允许的，需要sudo来提升权限
  - 或者直接用root用户来修改

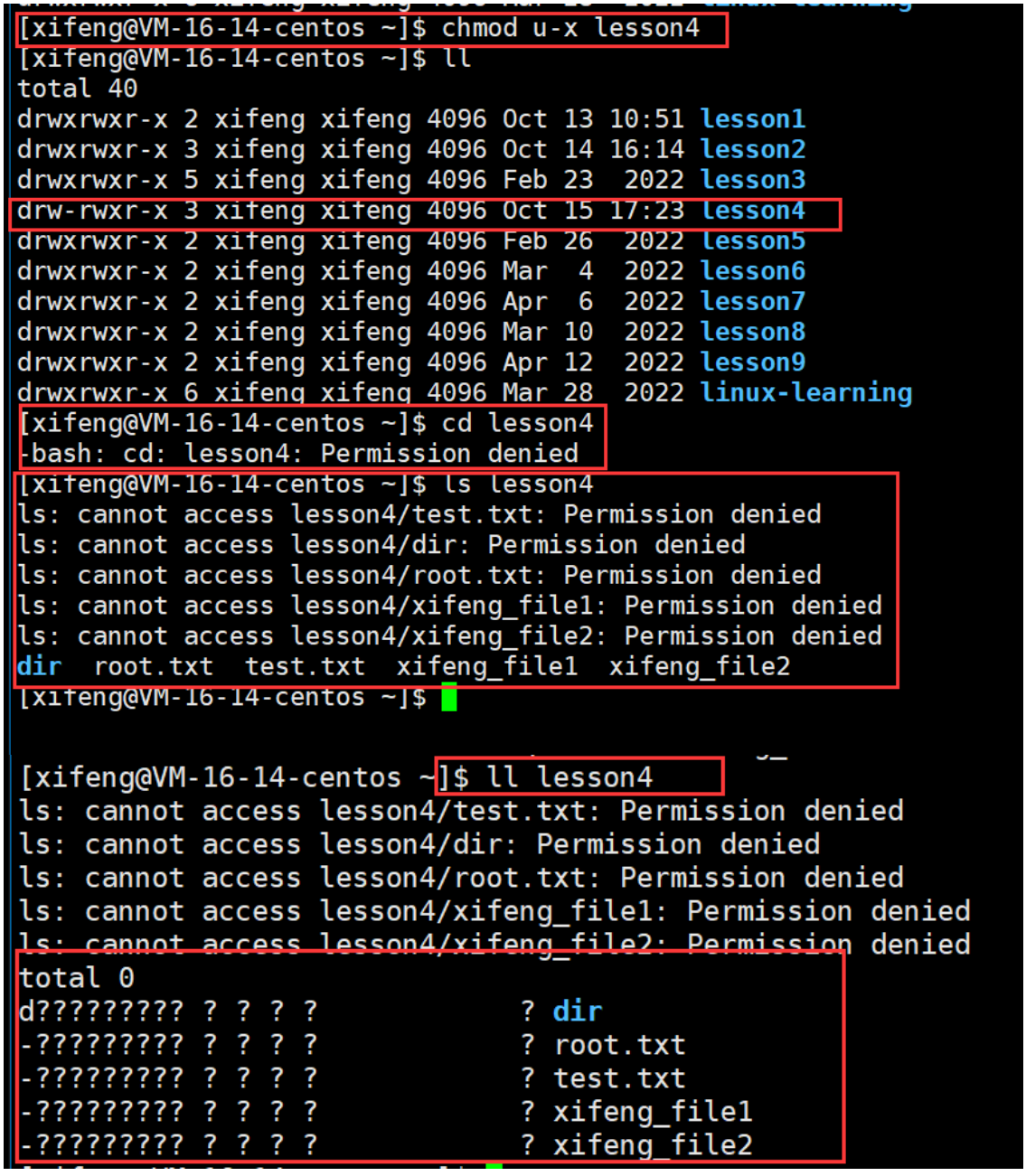
7.修改文件所属组

- **chgrp 修改文件所属组**
  - 同chown都是需要sudo或者转换成root用户
  - 如果文件本身属于我们，那么我们要把组更改回来不需要sudo或者换成root用户，可以直接回收回来
  - 修改文件拥有者和所属组可以同时修改具体操作：

```
sudo chown user:user file_name
```

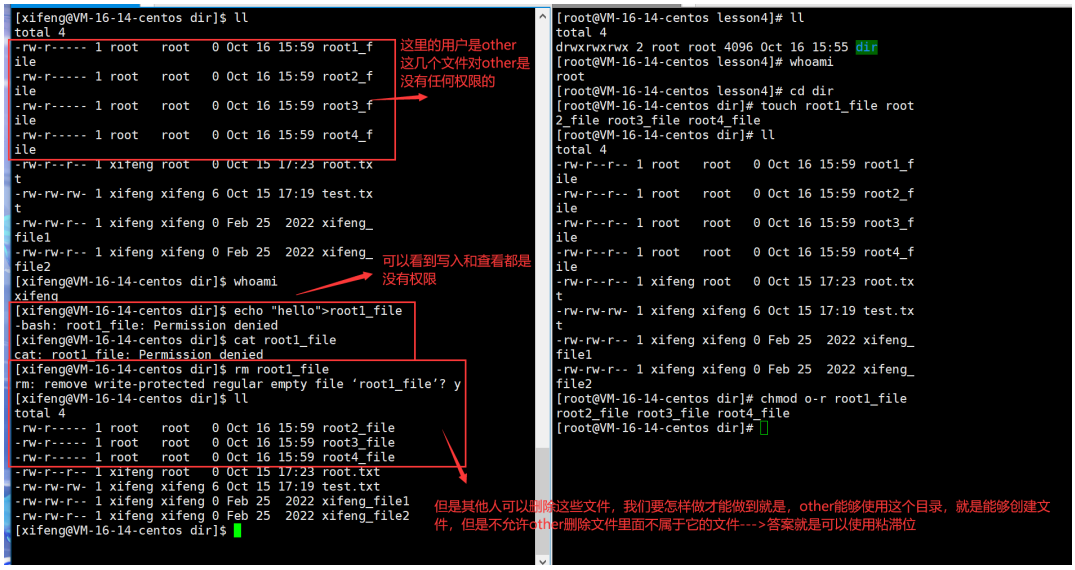
8.有关目录文件操作

- **如果要进入目录，需要的权限是x(可执行权限)**
- 目录的w(写)权限是对目录里面可以增加文件
- 目录的r(读)权限是查看目录里面的文件信息的权限
- **如果只有读写没有可执行，linux最多可以查看到目录里面的文件名，其它查看操作都不支持**

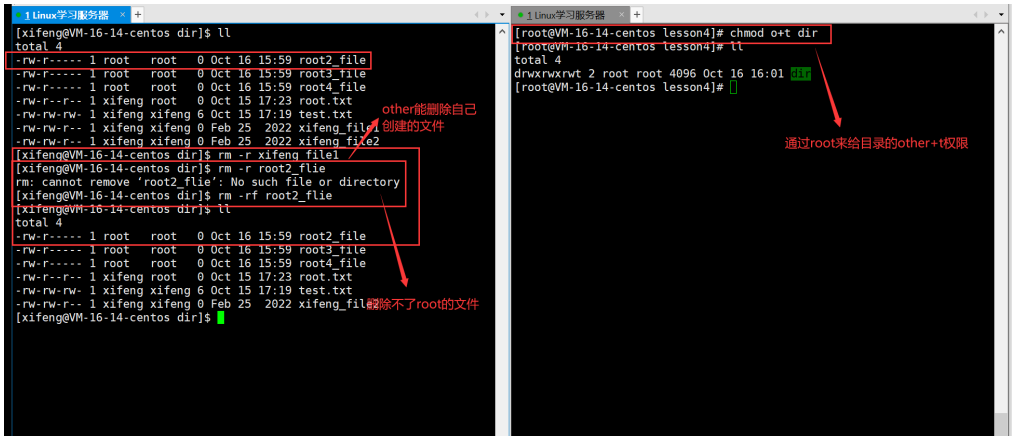


- 如果目录本身对other具有w权限，other可以删除任何的目录下的文件；如果目录本身对other没有w权限，other不可以删掉任何文件

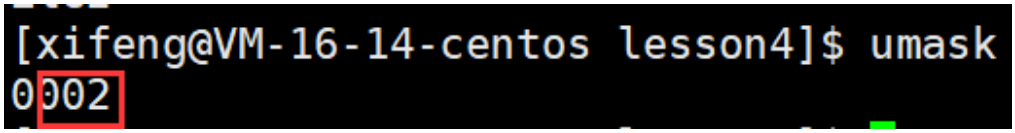
- 有一种情况是让other可以创建和写入文件，但是不能删除原来目录中的文件



  - 方法就是：粘滞位
  - 操作：chmod o+t dir(目录名)



  - 特点：**只能对目录进行设置**，一般是限制other权限的；对设置了粘滞位的目录，在改目录下，只有文件的拥有者(root)可以删除，其他人不能删除
  - 在Linux里面的tmp目录就是用的粘滞位，tmp的用途就是用来存储的临时文件

- **Linux默认权限**
  - 权限位置是固定的而且是两态的
  - 普通文件，起始权限(666)
  - 目录文件，起始权限(777)
  - **umask 权限掩码**
    - 

(看后三位)
    - 简单理解就是凡是在umask中出现的，都应在起始权限中去掉
    - 具体理解就是 002 转换成二进制就是**default = default & ~(mask)**

      等号右边default指(666),mask指的是(002)

```
000 000 010   （002）
110 110 110   （666）
--------------------
111 111 101   （取反）
110 110 110
--------------------
110 110 100   （按位与）
```

最后的664就是普通文件的默认权限，目录文件同理

- **自定义默认权限**
  - umask 0333(后面的数字可以根据自己需要写)
  - 定制的权限只有本次登录有效(可以通过更改系统文件来使其一直有效，但是不推荐这么操作)