openEuler内核编程技术

实训指导书

第十章 第2讲

《DAC与ACL的配置与使用》

软件所制

**术语表**

|  |  |  |  |
| --- | --- | --- | --- |
| 英文 | 中文 | 含义 | 别名 |
|  |  |  |  |
|  |  |  |  |

第十章 第2讲 DAC与ACL的配置与使用

# 任务1：DAC的配置与使用(45min)

## 相关知识

DAC是传统的Linux访问控制方式，一般通过r、w、x 对文档对象进行权限设置；当某个进程想对文件进行访问时，系统是通过该进程的用户或者用户组，与文件的权限进行比较，再判定是否允许访问。

DAC主要的内容包括以下几个概念：主体、客体、权限（rwx）、所有权（ugo）。

（1）主体是用户的身份，客体是资源或者说是文件（linux中一切皆文件）。

由客体的属主对自己的客体进行管理，由主体自己决定是否将自己的客体访问权限或部分访问权限授予其他主体，这种控制方式是自主的。也就是说，在自主访问控制下，用户可以按自己的意愿，有选择地与其他用户共享他的文件。

（2）文件的所有权

系统中的每个文件（一些特殊文件可能没有，如块设备文件等）都有所有者。在DAC系统中，文件的所有者是创建这个文件的计算机的使用者（或事件，或另一个文件）。那么此文件的自主访问控制权限由它的创建者来决定如何设置和分配。

在Linux所支持的默认的文件系统（如ext系列）中，所有权有三类：文件的拥有者，文件的所属组，其他人；

* 文件的拥有者：也称为属主，标记为u；默认情况下，创建文件的用户就是其属主。
* 文件的所属组：也称为属组，标记为g；除了属主之外，还可以被哪些用户共同拥有。由于这样的用户可能不止一个，所以不方便用某个用户名来标识，因此以组的方式来标识。
* 其他人：标记为o；除了文件的属主和属组之外的其他所有使用者的统称。

（3）访问权限

文件的所有者拥有访问权限，并且可以将访问权限分配给自己及其他用户。

在Linux所支持的默认的文件系统（如ext系列）中，权限有三种：读权限，写权限，执行权限。

* 读权限：

1）对于目录来说，拥有读权限的用户可以浏览该目录中的文件名；如果同时拥有执行权限，则该用户可以查看目录的元数据，目录中的文件名及这些文件的元数据信息。

2）对于非目录文件来说，拥有读权限的用户在对存放该文件的目录有读权限和执行权限的前提下可以查看该文件的内容。

* 写权限：

1）对于目录来说，拥有写权限的用户必须要同时拥有执行权限，才可以在该目录中新建、修改和删除文件名；

2）对于非目录文件来说，拥有写权限的用户在对存放该文件的目录有读权限和执行权限的前提下可以在该文件的末尾追加内容；如果想要修改和删除该文件内容，必须同时拥有读权限。

* 执行权限：

1）对于目录来说，执行权限是最最基本的权限了。拥有执行权限的用户可以在路径表示中引用此目录，并且如果该用户同时拥有读权限，便可以进入到此目录中，可以查看目录的元数据。

2）对于非目录的文件来说，拥有执行权限的用户可以将此文件当作命令来执行，也就是说可以把这个文件直接载入内存，让其运行起来成为进程或者通过其他的解释工具将他里面保存的命令、语句解释出来并执行。

### DAC权限管理

（1）chgrp

语法：chgrp [选项] [所属群组] [文件或目录...]

功能：改变文件或目录所属的用户组

选项说明：

* -c或--changes 效果类似"-v"参数，但仅回报更改的部分。
* -f或--quiet或--silent 　不显示错误信息。
* -h或--no-dereference 　只对符号连接的文件作修改，而不更动其他任何相关文件。
* -R或--recursive 　递归处理，将指定目录下的所有文件及子目录一并处理。
* -v或--verbose 　显示指令执行过程。
* --help 　在线帮助。
* --reference=<参考文件或目录> 　把指定文件或目录的所属群组全部设成和参考文件或目录的所属群组相同。
* --version 　显示版本信息。

使用示例：

|  |
| --- |
| # chgrp -v testgroup test.txt //将test.txt文件的裙子群组改为bin  # chgrp --reference=file2 file1 //将file1的群组属性设置为与file2相同 |

（2）chown

语法：chown [-cfhvR] [--help] [--version] user[:group] [文件或目录...]

功能：将指定文件的拥有者改为指定的用户或组，用户可以是用户名或者用户ID；组可以是组名或者组ID；文件是以空格分开的要改变权限的文件列表，支持通配符。

选项说明：

* user : 新的文件属主用户名或 ID
* group : 新的文件属组(group)用户名或 ID
* -c : 显示更改的部分的信息
* -f : 忽略错误信息
* -h :修复符号链接
* -v : 显示详细的处理信息
* -R : 处理指定目录以及其子目录下的所有文件
* --help : 显示辅助说明
* --version : 显示版本

使用示例：

|  |
| --- |
| # chown user:usergroup file //将文件file的拥有者/属主设为 user，属组设为 usergroup  # chown -R user:usergroup \* //当前目录下的所有文件与子目录的属主都设为 user，属组设为usergroup |

（3）chmod

语法：chmod [-cfvR] [--help] [--version] 文件或目录...

功能：改变文件/目录的权限（r,w,x)

选项说明：

* mode : 权限设定字串，格式为 ：[ugoa...][[+-=][rwxX]...][,...]

（1）u 表示该文件的拥有者，g 表示与该文件的拥有者属于同一个群体(group)者，o 表示其他以外的人，a 表示这三者。

（2）+ 表示增加权限、- 表示取消权限、= 表示唯一设定权限。

（3）r 表示可读取，w 表示可写入，x 表示可执行，X 表示只有当该文件是个子目录或者该文件已经被设定过为可执行。

* -c : 若该文件权限确实已经更改，才显示其更改动作
* -f : 若该文件权限无法被更改也不要显示错误讯息
* -v : 显示权限变更的详细资料
* -R : 对目前目录下的所有文件与子目录进行相同的权限变更(即以递回的方式逐个变更)
* --help : 显示辅助说明
* --version : 显示版本

也可以用三位八进制数字的形式分别表示权限，第一位指定属主的权限，第二位指定组权限，第三位指定其他用户的权限。每位通过4(读)、2(写)、1(执行)三种数值的和来确定权限。

|  |  |  |  |
| --- | --- | --- | --- |
|  | r | w | x |
| 二进制代表 | 100 | 010 | 001 |
| 十进制代表 | 4 | 2 | 1 |

例如：

（1）文件属主的权限为：rw-，则使用八进制6（4+2）表示权限；

（2）文件的DAL权限为：rwxr-xr-x，则使用八进制755表示权限；7（4+2+1）表示属主权限，5（4+1）表示组用户的权限；5（4+1）表示其他用户权限。

使用示例：

|  |
| --- |
| # chmod 755 file //给file的属主分配读、写、执行(7)的权限，给file的属主所在组分配读、执行(5)的权限，给其他用户分配读、执行(5)的权限  # chmod u=rwx,g=rx,o=x file //与‘chmod 755 file’等效  # chmod u+x file 　//给file的属主增加执行权限  # chmod =r file 　//为所有用户分配读权限  # chmod 444 file 　　//为所有用户分配读权限，与上例等效  # chmod a-wx,a+r file 　　 　 //为所有用户分配读权限，与上例等效  # chmod -R u+r directory 　 //递归地给directory目录下所有文件和子目录的属主分配读的权限 |

（4）chattr

语法：chattr [-RV][-v<版本编号>][+/-/=<属性>] [文件或目录...]

功能：设置文件/目录的隐藏权限。**只有超级权限的用户才具有使用该命令的权限**，这项指令可改变存放在ext2、ext3、ext4、xfs、ubifs、reiserfs、jfs等文件系统上的文件或目录属性。

选项说明：

* -R 递归处理，将指定目录下的所有文件及子目录一并处理。
* -v<版本编号> 设置文件或目录版本。
* -V 显示指令执行过程。
* +<属性> 开启文件或目录的该项属性。
* -<属性> 关闭文件或目录的该项属性。
* =<属性> 指定文件或目录的该项属性

属性说明：

a：让文件或目录仅供附加用途。

b：不更新文件或目录的最后存取时间。

c：将文件或目录压缩后存放。

d：将文件或目录排除在倾倒操作之外。

i：不得任意更动文件或目录。

s：保密性删除文件或目录。

S：即时更新文件或目录。

u：预防意外删除。

使用示例：

|  |
| --- |
| # chattr +i /etc/resolv.conf //添加 "i" 隐藏属性，无法更动此文件了；防止该文件被修改  # chattr -i /etc/shadow //解除 "i" 这个隐藏属性  # chattr +a /var/log/messages //只能往指定文件里追加数据，但不能删除，适用于各种日志文件 |

（5）lsattr

语法：lsattr [-adlRvV] [文件或目录...]

功能：显示文件/目录的隐藏权限。

参数说明：

* -a 　显示所有文件和目录，包括以"."为名称开头字符的额外内建，现行目录"."与上层目录".."。
* -d 　显示，目录名称，而非其内容。
* -l 　此参数目前没有任何作用。
* -R 　递归处理，将指定目录下的所有文件及子目录一并处理。
* -v 　显示文件或目录版本。
* -V 　显示版本信息。

使用示例：

|  |
| --- |
| # lsattr /etc/resolv.conf //显示指定文件的隐藏权限  # lsattr -a //将当前目录的文件或目录下的文件所有属性(包括隐藏属性)列出 |

（6）文件特殊权限：

1. SUID（Set User ID）：

功能：让非文件属主在执行文件时暂时具有属主的权限，执行结束后权限恢复。

设置：在u的x位用s/S设定，原位置有执行权限，就置为 s，否则为 S。

1. SGID（Set Group ID）：

功能：让非文件属组的组员在执行文件是暂时具有属组组员的权限，执行结束后权限恢复。

设置：在g的x位用s/S设定，原位置有执行权限，就置为 s，否则为 S 。

1. SBIT（sticky bit，粘滞位）

功能：只对目录有效，使目录下的文件，只有文件拥有者才能删除。

（如果用户不属于owner，仅属于group或者other，就算他有w权限，也不能删除文件）。

设置：粘滞位，在o的x位用t/T设定，原位置有执行权限，就置为 t ，否则为T 。

设置SUID、SGID的方法：

1）方法一：

设置SUID：chmod u+u filename

设置SGID：chmod g+u filename

设置SBIT：chmod o+t filename

2）方法二：chmod 4位遵循权限设置的数字 filename

* 若设置SUID，则将权限位最前面（第一个短横线\_所占据的）的那一位设置为4；
* 若设置SGID，则将权限位最前面的那一位设置为2；
* 若设置SBIT，则将权限位最前面的那一位设置为1；
* 如果希望三者都置位，那么将权限位最前面的那一位设置为7（4+2+1）。

使用示例：

|  |
| --- |
| # chmod u+s filename //文件被设置了SUID  # chmod g+s filename //文件被设置了SGID  # chmod o+t dirname //目录被设置了SBIT  # chmod 4755 filename //rws r-x r- x文件被设置了SUID，文件属主具有读、写和执行的权限，所有其他用户具有读和执行的权限  # chmod 6711 filename //rws --s --x 文件被设置了SUID和SGID，文件属主具有读、写和执行的权限，所有其他用户具有执行的权限  # chmod 1744 dirname //rwx -w- -wt 文件被设置了SBIT，只有文件属主具有删除目录的权限 |

## 任务描述

1. 创建测试使用的用户、用户组、文件、目录等，完成以下DAC权限设置：

分别使用chgrp、chown、chmod、chattr、lsattr以及特殊权限（SUID、SGID、SBIT）设置DAC权限；

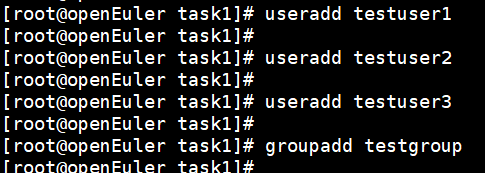
2. 测试验证已设置权限是否成功设置与删除。

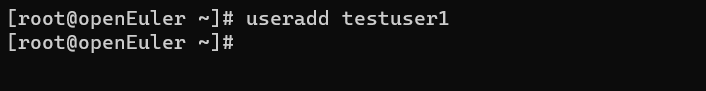
## 审核要求

1. 正确设置文件/目录的DAC权限。
2. 提交关键过程的截图。

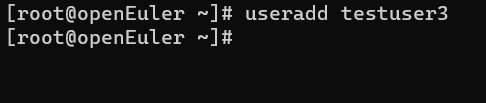
## 参考答案

# 添加需要试验的用户和用户组，省略设定密码的过程

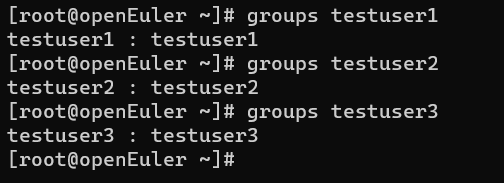








# 查看添加的用户所属的组，以及用户组包含的用户





**getent group <group\_name>**

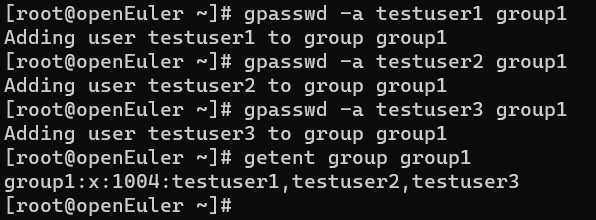
查询/etc/group文件，并从文件中获取提到的组的条目。

输出格式为：

**group:password:GID:user(s)**

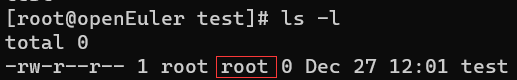
* group 表示组名
* password是加密的组密码，空白字段表示无密码，x位表示密码在文件/etc/gshadow中
* GID 是组ID
* user(s) 是该组成员的用户列表，为空表示该组没有成员。

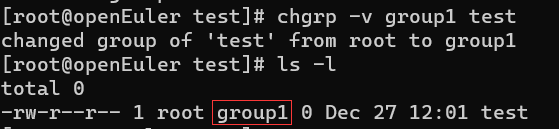
把测试用的用户添加到group1中

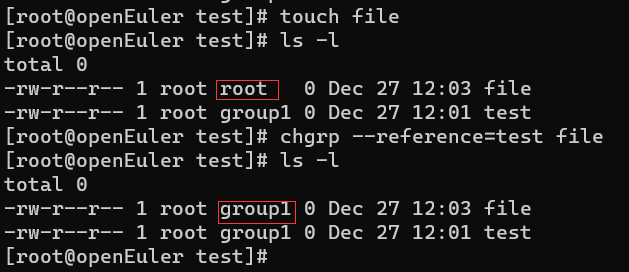


**一、设置DAC权限**

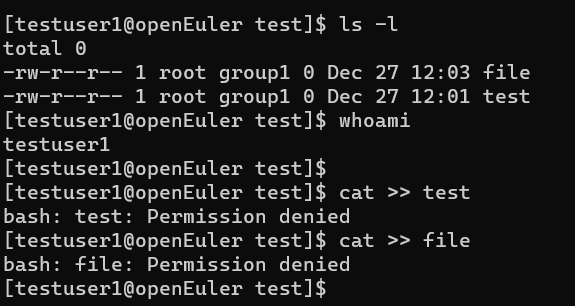
1、chgrp





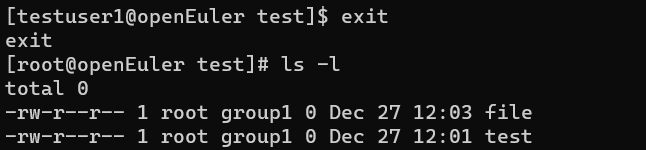


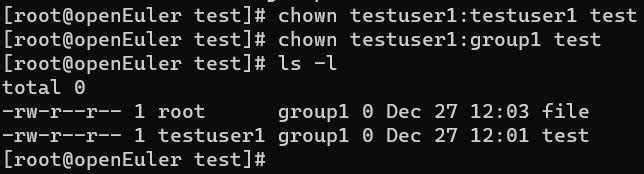
使用su testuser1 切换用户后，该用户则没有这两个文件的写权限，因为testuser1用户不是文件的属主（exit返回root用户）：



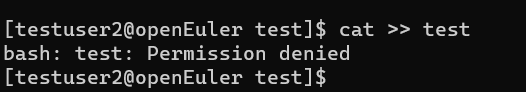
2、chown

root用户下，将test的拥有者与群组都设置为testuser1：

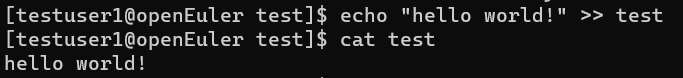


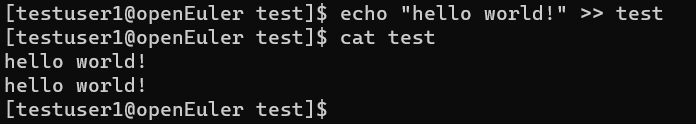


root用户下，使用su testuser2 切换用户后，该用户则没有test文件的写权限，因为testuser2用户不是文件的属主：



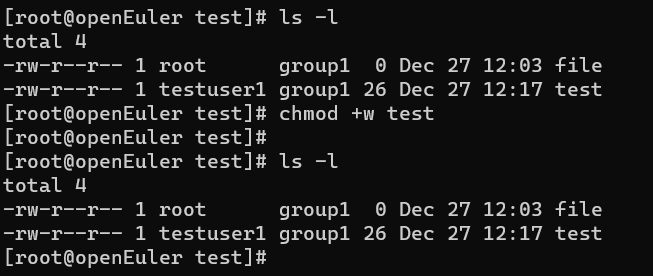
exit返回root用户，在root用户下，使用su testuser1 切换用户后，可正常写入test文件：



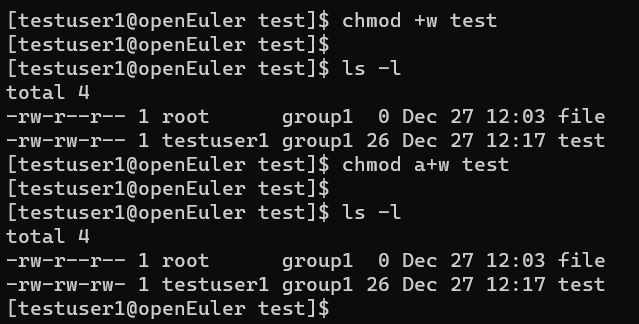


3、chmod

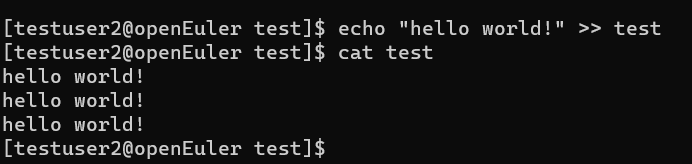
由于test的属主不是root，因此root修改权限无效：



su testuser1切换testuser1用户后，可正常修改权限：

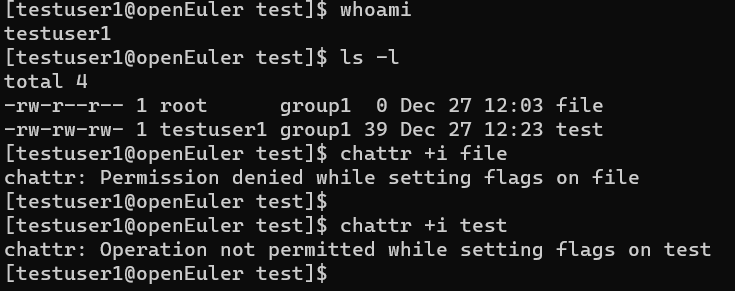


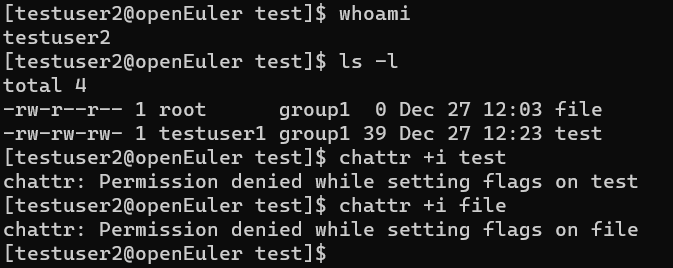
exit返回root用户，在root用户下，使用su testuser2 切换用户后，可正常写入test文件：



4、chattr与lsattr

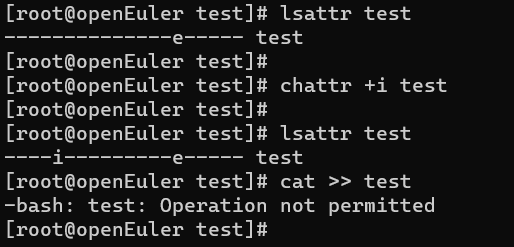
由于chattr命令只能在root用户下才能生效，因此使用su testuser1或su testuser2切换用户后，无法修改权限：



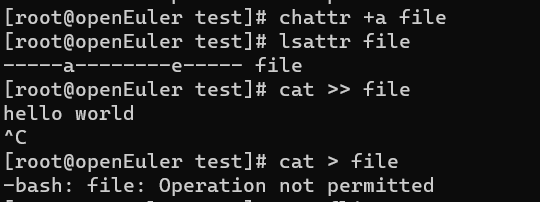


exit返回root用户，可正常修改权限：

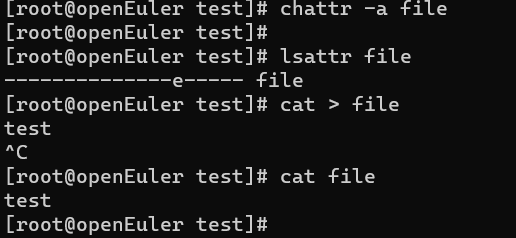
添加i权限设置，即便root用户也无法修改文件：



添加 a 属性权限后，只能往file文件里追加数据，但不能覆盖/删除原有内容（cat命令使用>> 是追加，>是覆盖）；删除a 属性权限后正常编辑：

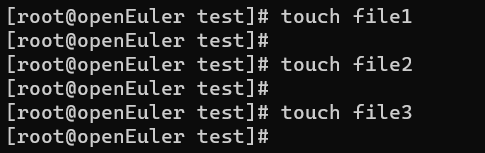






5、特殊权限 SUID与SGID

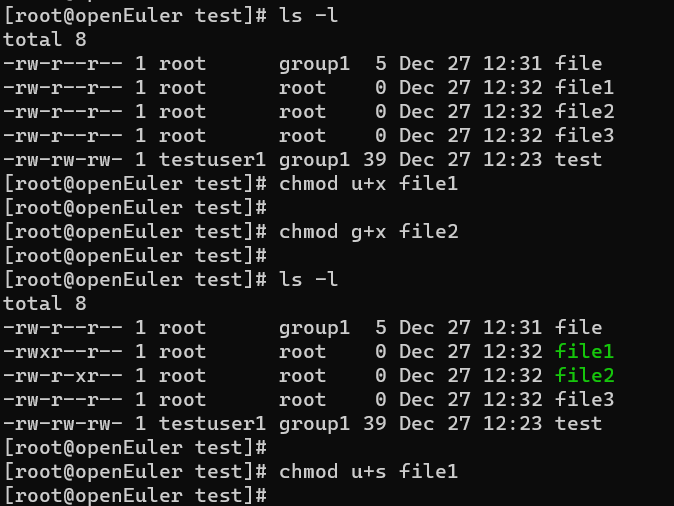
新建测试文件：

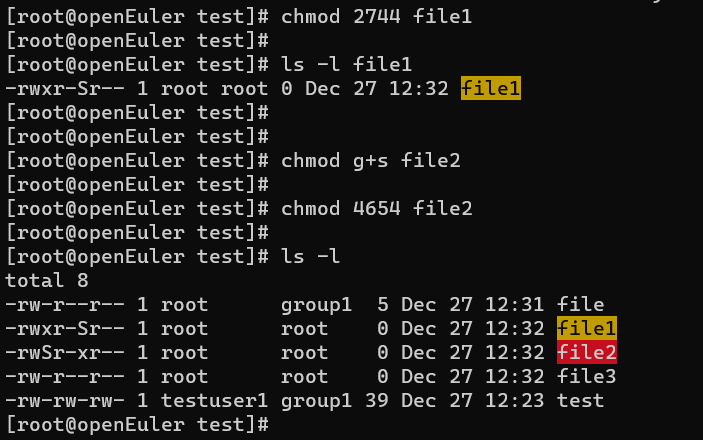


设置特殊权限：

当执行权限位已设置x，则设置SUID或SGID时，该位设置为s；

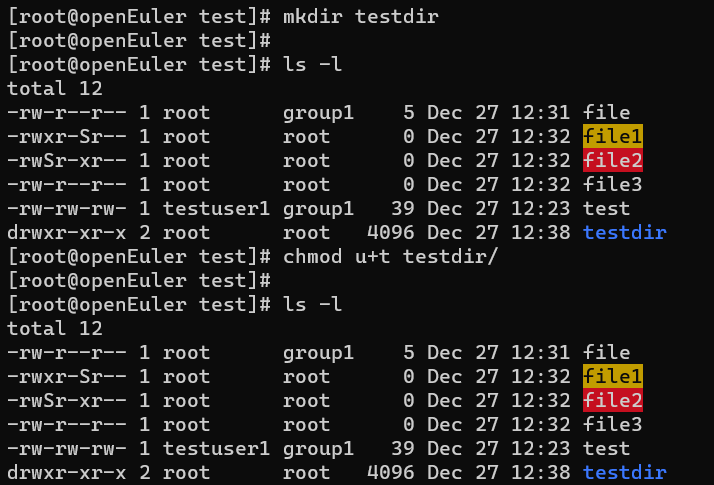
当执行权限位未设置x，则设置SUID或SGID时，该位设置为S。

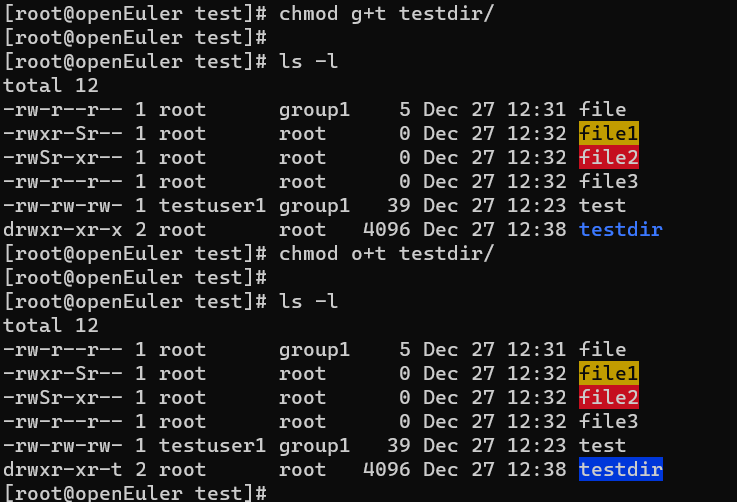




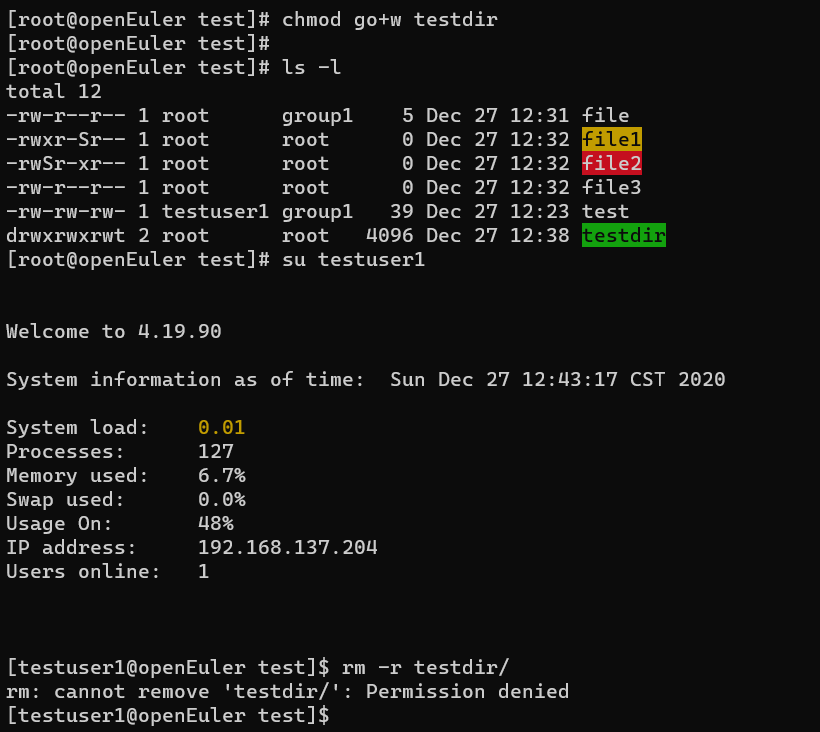
6、特殊权限SBIT

（1）只能给o（other）用户组用户设置 SBIT权限。





（2）当切换用户testuser1后，因testuser1不是testdir的owner，虽然给他添加w权限，也不能删除文件：



# 任务2：ACL的配置与使用(45min)

## 相关知识

访问控制列表（ACL，Access Control List）

ACL用于更精准地针对单一用户、单一文件或者目录进行r、w、x的权限设置。它在UGO权限管理的基础上为文件系统提供一个额外的、更灵活的权限管理机制。是UNIX文件权限管理的一个补充。

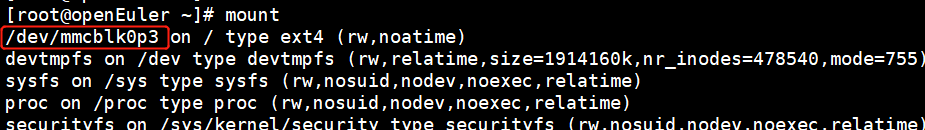
### ACL权限管理

（1）开启ACL权限

在 openEuler-20.03 系统中，ACL 权限默认是开启的，不需要手工开启。

如何查看 ACL 权限是否开启：

1. 使用 mount 命令可以査看到系统中已经挂载的分区

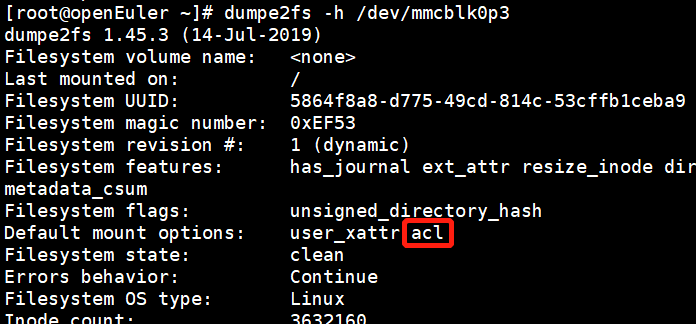


1. 使用 dumpe2fs 命令査看分区文件系统的详细信息

dumpe2fs 命令可选的选项及其含义如下：

-h：仅显示超级块中的信息，而不显示磁盘块组的详细信息；

`dumpe2fs -h /dev/mmcblk0p3`



由截图可见，ACL 权限是 /dev/mmcblk0p3分区的默认挂载选项（Default mount options），不需要手工挂载。

若操作系统的默认挂载选项不包含ACL，则需手动挂载：

1. 使用 mount 命令重新挂载，加入 ACL 权限：

|  |
| --- |
| # mount -o remount, acl / |

不过使用此命令是临时生效的。要想永久生效，需要修改 /etc/fstab 文件。

1. 手动修改 /etc/fstab 文件

|  |
| --- |
| # vim /etc/fstab  UUID=a2cb2f57-c76e-43ea-bca0-f2325r3d8bd2 /ext4 defaults, acl 1 1  # 在需要开启 ACL 权限的分区行上（ACL 权限针对的是分区），手动在 defaults 后面加入",acl"即可永久在此分区中开启 ACL 权限。  # mount -o remount /  # 重新挂载文件系统或重启系统，使修改生效 |

（2）ACL权限管理

|  |
| --- |
| # getfacl 目录/文件名 #查看ACL权限  # setfacl 选项 目录/文件名 #设定ACL权限 |

选项说明：

* -m：给指定文件设定 ACL 权限。  
  设置用户的ACL 权限，格式为： "u:用户名:权限"；  
  设置用户组的ACL 权限，格式为： "g:组名:权限"。
* -x：删除指定的 ACL 权限。
* -b：删除所有的 ACL 权限。
* -d：设定默认 ACL 权限。只对目录生效，目录中新建立的文件会继承此默认权限。
* -k：删除默认 ACL 权限；
* -R：递归设定 ACL 权限。指设定的 ACL 权限会对目录下的所有子文件生效。
* -h：显示setfacl的用法。

使用示例：

|  |
| --- |
| # setfacl -m u:user:rw testfile //为 user用户设置testfile 文件的读写权限  # getfacl testfile // 查看testfile文件的acl权限 |

## 任务描述

1. 创建测试使用的用户、用户组、文件、目录等，添加ACL权限设置：

包括设置、删除指定文件和目录的用户/用户组的ACL权限；并设置目录子文件的ACL权限实现递归。

2. 测试验证已设置权限是否成功设置与删除。

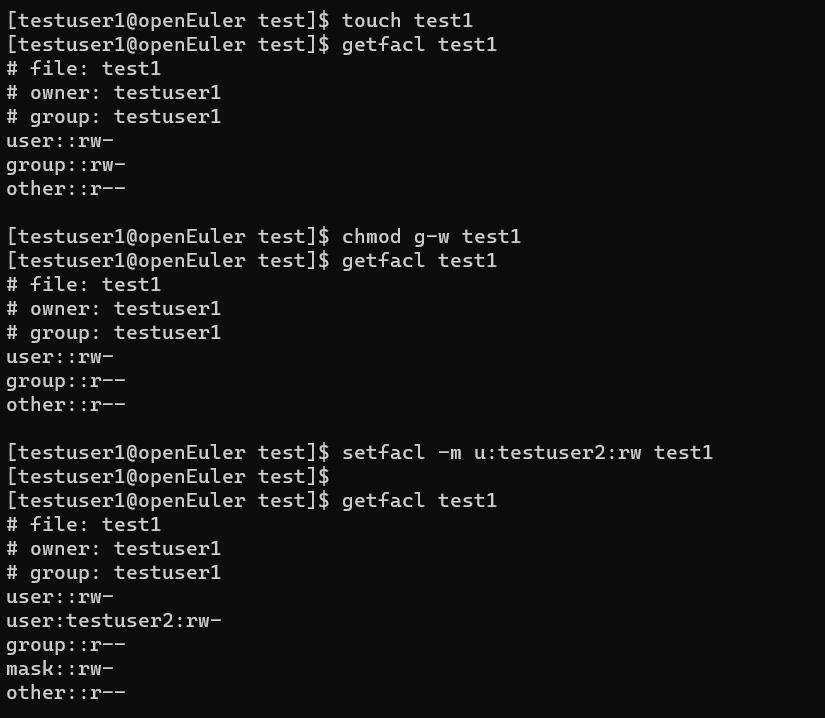
## 审核要求

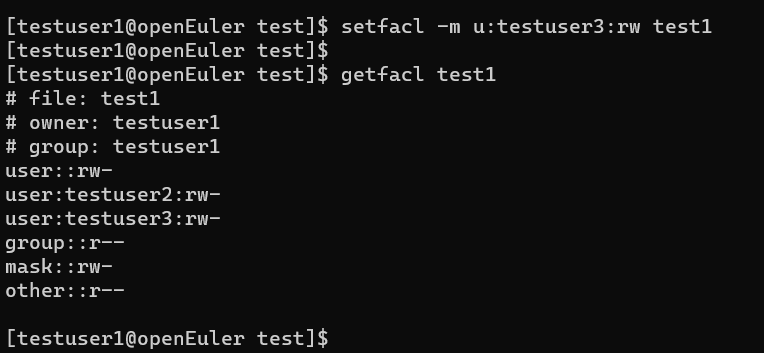
1. 正确设置与删除ACL权限。
2. 提交关键过程的截图。

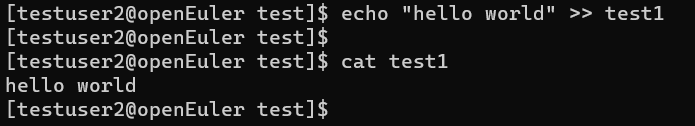
## 参考答案

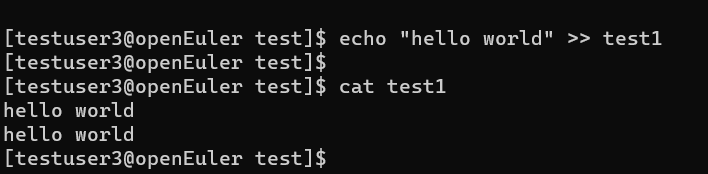
**二、设置ACL权限**

1、给指定文件设置用户/用户组的ACL权限

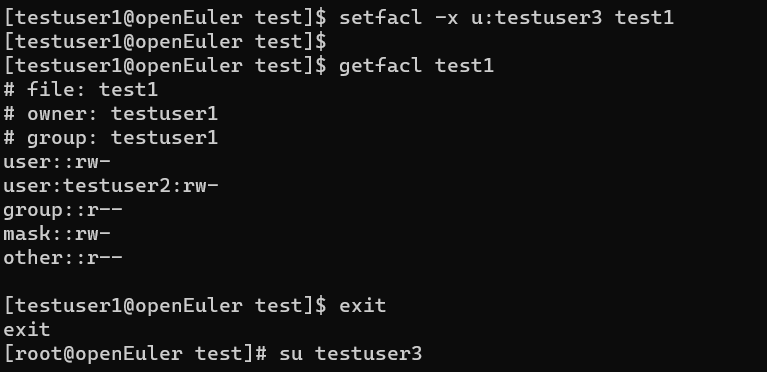


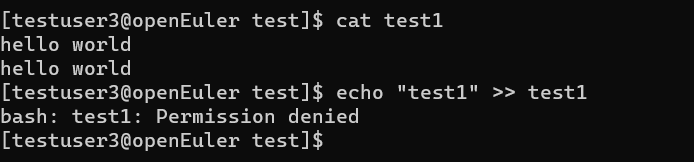




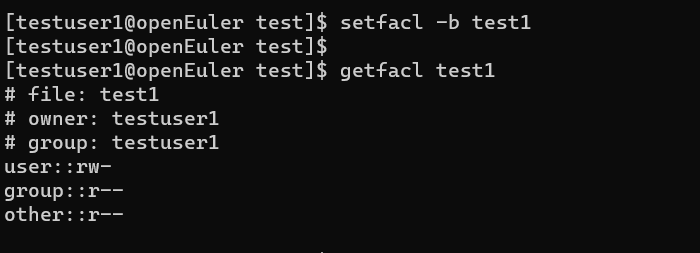


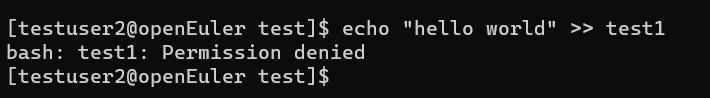
删除指定用户/用户组的ACL权限（-x）：

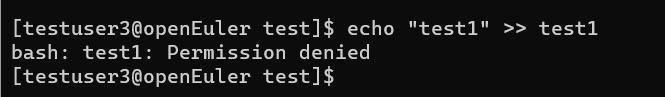




删除文件的所有ACL权限（-b）：

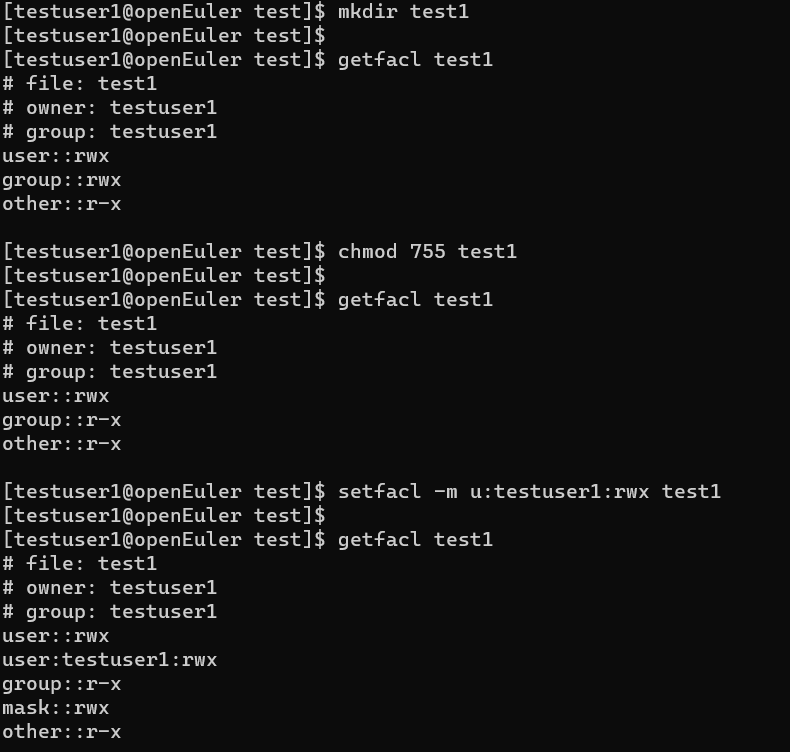


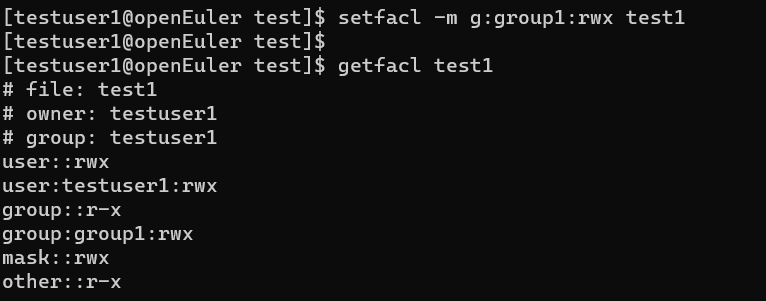




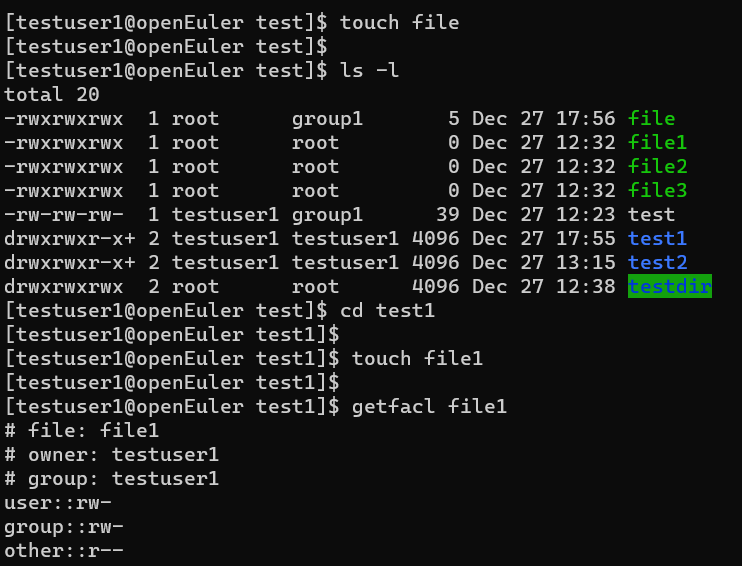
2、给指定目录及子目录/文件设置用户/用户组的ACL权限

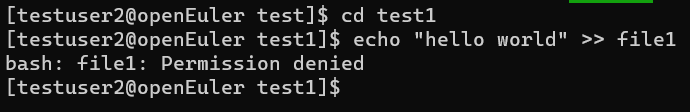
同上。





进入目录创建文件，新建文件，查看ACL权限是否继承：

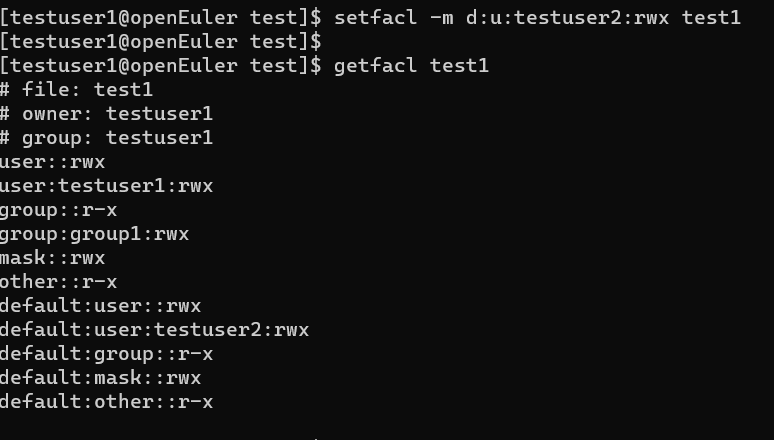




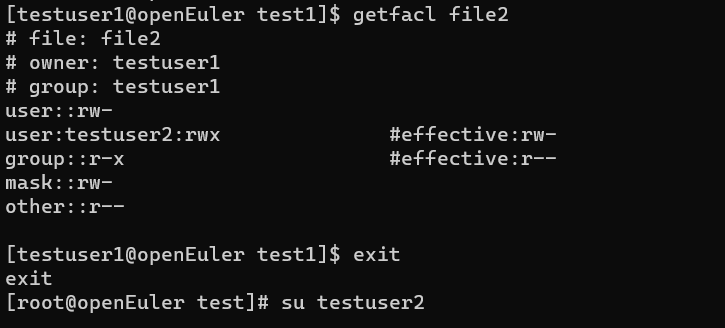
说明子文件file没有继承父目录testdir的ACL权限；

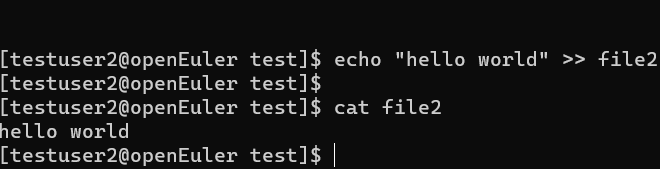
如何使子目录/子文件继承父目录的ACL权限——默认ACL权限

默认 ACL 权限的作用是：如果给父目录设定了默认 ACL 权限，那么父目录中所有新建的子文件都会继承父目录的 ACL 权限。









给目录设置默认ACL后，目录下的原有文件权限不变，新建文件会继承父目录的ACL权限。