用户权限

一、基本权限UGO

1. UGO 的含义

- **U (User)**：文件或目录的**所有者**（创建者或通过 `chown` 指定的用户）-**属主**
- **G (Group)**：文件或目录的**所属组**（通过 `chgrp` 或 `chown` 指定的用户组）-**属组**
- **O (Other)**：既不是所有者，也不在所属组中的**其他所有用户**

---

2. 权限类型

每个角色（U/G/O）可分配以下三种权限：

- **r (Read)**：读取文件内容，或列出目录中的文件。
- **w (Write)**：修改文件内容，或在目录中创建/删除文件。
- **x (eXecute)**：执行文件（如脚本），或进入目录。

---

3. 查看权限

使用 `ls -l` 命令查看文件或目录的权限：

bash

```
ls -l /root
```

输出示例：

```
-rw-r--r-- 1 alice developers 1024 Jan 1 12:34 example.txt
```

- 权限字段：

  ```
  -rw-r--r--
  ```

  - 第1位：`-` 表示文件，`d` 表示目录。
  - 第2-4位：**User** 的权限（`rw-`：可读可写，不可执行）。
  - 第5-7位：**Group** 的权限（`r--`：仅可读）。
  - 第8-10位：**Other** 的权限（`r--`：仅可读）。

---

4. 权限的两种表示方式

符号表示法（rwx）

- 直接使用 `r`、`w`、`x` 表示权限。
- 示例：

  ```
  rwxr-x---
  750
  ```

  - User：`rwx`（读、写、执行）
  - Group：`r-x`（读、执行）
  - Other：`---`（无权限）

**数字表示法

- 将 `rwx` 转换为数字相加：
  - `r` = 4（2^2）
  - `w` = 2（2^1）
  - `x` = 1（2^0）
- 示例：

  ```
  rwxr-x---
  ```

  →

  ```
  750
  ```

  - User：`4+2+1=7`
  - Group：`4+0+1=5`
  - Other：`0+0+0=0`

---

5. 修改权限（`chmod` 命令）

语法

```
使用符号：u用户 g组  o其他  r读   w写  x执行
语法：  chmod   对象(u/g/o/a)赋值符(+/-/=)权限类型(r/w/x)     文件/目录
```

符号模式

bash

```
# 给 User 添加执行权限，Group 移除写权限，Other 设置只读
chmod u+x,g-w,o=r example.txt
-rw-r--r--
-rwxr--r--
```

数字模式

bash

```
# 设置权限为 rw-r-----（User可读可写，Group可读，Other无权限）
chmod 640 example.txt
```

实验1：查看当前目录的权限

```
[root@xnha ~]# ll /tmp    #无法看到/tmp 目录的权限
总用量 0
drwxr-xr-x. 2 root   root   26 3月   4 01:32 202411111
drwxrwxr-x. 3 abc    abc    35 3月   4 08:09 practice
drwx------. 3 root   root   17 3月   6 06:45 systemd-private-
f5a863dbe0dc478eb338abdd33ddce14-bluetooth.service-JX3LUK
drwx------. 3 root   root   17 3月   6 06:45 systemd-private-
f5a863dbe0dc478eb338abdd33ddce14-chronyd.service-AFacoY
drwx------. 3 root   root   17 3月   6 06:45 systemd-private-
f5a863dbe0dc478eb338abdd33ddce14-colord.service-hWBiQw
drwx------. 3 root   root   17 3月   6 06:45 systemd-private-
f5a863dbe0dc478eb338abdd33ddce14-fwupd.service-VGH3Z7
drwx------. 3 root   root   17 3月   6 06:45 systemd-private-
f5a863dbe0dc478eb338abdd33ddce14-geoclue.service-zO4W1b

[root@xnha ~]# ll -d /tmp          #-d 命令查看当前目录
drwxrwxrwt. 26 root root 4096 3月   6 06:46 /tmp
```

实验2：可执行文件

1.在/tmp目录中创建file1.txt，编辑文件写入以下内容

file1.txt

```
echo "hello zhouwu"
read -p "please your name?" name
echo "$name good"
```

2.为用户添加file1.txt的执行权限

```
[root@xnha ~]# chmod u+x /tmp/file1.txt
```

3.观察文件类型

```
[root@xnha ~]# ll /tmp/file1.txt
-rwxr--r--. 1 root root 71 3月   6 06:52 /tmp/file1.txt
```

4.执行文件内容，观察程序运行

```
[root@xnha ~]# cd /tmp
[root@xnha tmp]# ./file1.txt
hello zhouwu
please your name?test
test good
[root@xnha tmp]#
```

5.去除权限，运行失败

```
[root@xnha tmp]# chmod u-x file1.txt
[root@xnha tmp]# ./file1.txt
bash: ./file1.txt: 权限不够
```

实验3：为所有用户添加执行权限

```
chmod +x file1.txt  # 所有角色添加执行权限（不推荐）
```

实验4：目录场景

允许其他人查看目录内容，但禁止修改：

```
mkdir /tmp/mydir
chmod 755 mydir/  # rwxr-xr-x
```

- User：可读、写、进入目录。
- Group/Other：可读、进入目录，但不能创建/删除文件。

实验5：-R选项

1.在tmp中创建递归目录dir1，观察新建文件夹的默认权限是多少（755）

```
[root@xnha tmp]# ll -d dir1/
drwxr-xr-x. 2 root root 6 3月   6 07:47 dir1/
```

2.在dir1目录中创建3个文件111、222、333，观察文件默认权限是多少（644）

```
[root@xnha dir1]# touch 111 222 333
[root@xnha dir1]# ls
111  222  333
[root@xnha dir1]# cd ..
[root@xnha tmp]# ll dir1
总用量 0
-rw-r--r--. 1 root root 0 3月   6 07:49 111
-rw-r--r--. 1 root root 0 3月   6 07:49 222
-rw-r--r--. 1 root root 0 3月   6 07:49 333
```

3.修改dir1权限为777，观察文件是否变化

```
[root@xnha tmp]# chmod 777 dir1/
[root@xnha tmp]# ll -d dir1/
drwxrwxrwx. 2 root root 39 3月   6 07:49 dir1/
[root@xnha tmp]# ll dir1/
总用量 0
-rw-r--r--. 1 root root 0 3月   6 07:49 111
-rw-r--r--. 1 root root 0 3月   6 07:49 222
-rw-r--r--. 1 root root 0 3月   6 07:49 333
```

4.通过-R递归修改dir1中所有文件权限为700

```
[root@xnha tmp]# chmod -R 700 dir1/
[root@xnha tmp]# ll -d dir1/
drwx------. 2 root root 39 3月   6 07:49 dir1/
[root@xnha tmp]# ll dir1/
总用量 0
-rwx------. 1 root root 0 3月   6 07:49 111
-rwx------. 1 root root 0 3月   6 07:49 222
-rwx------. 1 root root 0 3月   6 07:49 333
```

6.更改属主/组改变权限（chown命令）

语法： change owner

```
chown：  设置一个文件属于谁，属主
语法：  chown    用户名.组名    文件
```

实验1：改属主、属组

1.更改file1权限为600

```
[root@xnha tmp]# chmod 600 file1.txt
[root@xnha tmp]# ll file1.txt
-rw-------. 1 root root 71 3月   6 06:52 file1.txt
```

2. 切换其他用户尝试访问

```
[root@xnha tmp]# su - abc
[abc@xnha ~]$ cat /tmp/file1.txt
cat: /tmp/file1.txt: 权限不够
```

3. 返回root用户，更改文件属主/组

```
[abc@xnha ~]$ exit
注销
[root@xnha tmp]# chown abc.hr file1.txt
[root@xnha tmp]# ll file1.txt
-rw-------. 1 abc hr 71 3月   6 06:52 file1.txt
```

4. 切换abc用户再次尝试访问

```
[root@xnha tmp]# su - abc
[abc@xnha ~]$ cat /tmp/file1.txt
echo "hello zhouwu"
read -p "please your name?" name
echo "$name good"
```

实验2：只改属主

```
[root@xnha tmp]# ll file1.txt
-rw-------. 1 abc hr 71 3月   6 06:52 file1.txt
[root@xnha tmp]# chown user999 file1.txt
[root@xnha tmp]# ll file1.txt
-rw-------. 1 user999 hr 71 3月   6 06:52 file1.txt
```

实验3：只改属组

```
[root@xnha tmp]# ll file1.txt
-rw-------. 1 user999 hr 71 3月   6 06:52 file1.txt
[root@xnha tmp]# chown .dev file1.txt
[root@xnha tmp]# ll file1.txt
-rw-------. 1 user999 dev 71 3月   6 06:52 file1.txt
```

---

7. 关键注意事项

1. 目录的执行权限（x）：
   - 无 `x` 权限时，即使有 `r` 权限，也无法 `cd` 进入目录或读取其中文件列表。
2. 权限优先级：
   - 系统按 **User → Group → Other** 的顺序匹配权限，**首次匹配成功则生效**。
3. 安全风险：
   - 避免给文件设置 `777`（所有用户可读可写可执行），尤其是敏感文件（如 `/etc/passwd`）。

---

8. 修改所属组（chgrp）

- `chgrp` **命令**：

```
# 修改所属组
[root@xnha tmp]# chgrp hr file1.txt
[root@xnha tmp]# ll file1.txt
-rw-------. 1 user999 hr 71 3月   6 06:52 file1.txt
```

9.综合练习

需求：

文件file10.txt

属主是user100，读写执行-7（可以看内容，可以改内容，可以执行）

属组是jishuzu（user200），读取　-4（只能看，不能改，不能执行）

其他人　　没有权限-0（既不能看，又不能改和执行）

测试：

1. 使用user100，访问文件。写入文件，执行文件
2. 使用jishuzu成员，访问文件，不可写和执行
3. 使用其他用户user300，访问文件失败。写入失败，执行失败。

操作：

1.创建文件

```
[root@xnha ~]# cd /tmp
[root@xnha tmp]# touch file10.txt
[root@xnha tmp]# ll file10.txt
-rw-r--r--. 1 root root 0 3月   7 01:03 file10.txt
```

2.创建用户

```
[root@xnha tmp]# useradd user100
[root@xnha tmp]# useradd user200
[root@xnha tmp]# useradd user300
[root@xnha tmp]# passwd user100
更改用户 user100 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：所有的身份验证令牌已经成功更新。
[root@xnha tmp]# passwd user200
更改用户 user200 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：所有的身份验证令牌已经成功更新。
[root@xnha tmp]# passwd user300
更改用户 user300 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：所有的身份验证令牌已经成功更新。
```

3.创建组

```
[root@xnha tmp]# groupadd jishuzu
[root@xnha tmp]# usermod -aG jishuzu user200
```

4.授予文件属主/组，以及其他人的权限

```
[root@xnha tmp]# chmod 740 file10.txt
[root@xnha tmp]# ll file10.txt
-rwxr-----. 1 root root 0 3月   7 01:03 file10.txt
[root@xnha tmp]# chown user100.jishuzu file10.txt
[root@xnha tmp]# ll file10.txt
-rwxr-----. 1 user100 jishuzu 0 3月   7 01:03 file10.txt
[root@xnha tmp]# id user200
uid=1108(user200) gid=1108(user200) 组=1108(user200),2002(jishuzu)
[root@xnha tmp]#
```

5.测试user100,是否拥有rwx权限

```
[root@xnha tmp]# su - user100
[user100@xnha ~]$ cd /tmp
[user100@xnha tmp]$ ls
202411111   practice
         tracker-extract-files.0     vmware-root_847-4013198920
dir1        systemd-private-e1d3ed26c96a4be6a11540bcdfc49862-bluetooth.service-
J9Ppv3     tracker-extract-files.1006  vmware-root_853-4022308820
file100     systemd-private-e1d3ed26c96a4be6a11540bcdfc49862-chronyd.service-
edcf0N       tracker-extract-files.1104  vmware-root_856-2731086721
file10.txt  systemd-private-e1d3ed26c96a4be6a11540bcdfc49862-colord.service-
20GFnf        user03                      vmware-root_862-2731217798
file1.txt   systemd-private-e1d3ed26c96a4be6a11540bcdfc49862-geoclue.service-
822iGd       vmware-root_826-2990547547  vmware-root_866-2722763301
file2.txt   systemd-private-e1d3ed26c96a4be6a11540bcdfc49862-
ModemManager.service-zUqqcH  vmware-root_835-3988097475  vmware-root_873-
4013854327
file3.txt   systemd-private-e1d3ed26c96a4be6a11540bcdfc49862-rtkit-
daemon.service-BPausn  vmware-root_841-4013329999
[user100@xnha tmp]$ cat file10.txt
echo 123456
[user100@xnha tmp]$ vim file10.txt
[user100@xnha tmp]$ cat file10.txt
echo 123456
echo 456789
[user100@xnha tmp]$ ./file10.txt
123456
456789
```

6.测试user200

```
[user100@xnha tmp]$ su - user200
密码:
[user200@xnha ~]$ cat /tmp/file10.txt
echo 123456
echo 456789
[user200@xnha ~]$ vim /tmp/file10.txt
[user200@xnha ~]$ ./file10.txt
-bash: ./file10.txt: 没有那个文件或目录
[user200@xnha ~]$ cd /tmp
[user200@xnha tmp]$ ./file10.txt
-bash: ./file10.txt: 权限不够
[user200@xnha tmp]$
```

7.测试user300

```
[user200@xnha tmp]$ su - user300
密码:
[user300@xnha ~]$ cat /tmp/file10.txt
cat: /tmp/file10.txt: 权限不够
[user300@xnha ~]$ vim  /tmp/file10.txt
[user300@xnha ~]$ cd /tmp
[user300@xnha tmp]$ ./file10.txt
-bash: ./file10.txt: 权限不够
```

二、基本权限ACL

一、ACL 与 UGO 的区别

1. UGO 权限的局限性

- **仅支持三个对象**：用户（User）、组（Group）、其他人（Other）。
- **无法精细化控制**：无法为同一文件设置多个独立用户的差异化权限。

示例需求：

bash

```
user01  rwx  file1
user02  rw   file1
user03  r    file1
user04  rwx  file1
user05  rw   file1
```

UGO 无法实现以上需求，需依赖 ACL 补充权限。

2. ACL 的优势

- **多对象支持**：可为同一文件设置多个用户/组的独立权限。
- **精细化控制**：支持 `r`（读）、`w`（写）、`x`（执行）的灵活组合。

二、ACL 核心命令语法

1. 设置权限：`setfacl`

bash

```
setfacl -m <用户/组>:<名称>:<权限> <文件/目录>
```

参数解析：

- `-m`：修改权限（`-x` 删除，`-b` 清空所有 ACL）。
- `<用户/组>`：`u`（用户）、`g`（组）、`o`（其他人）。
- `<名称>`：用户名或组名（留空表示默认属主/组）。
- `<权限>`：`r`/`w`/`x`（可组合，如 `rw`）。

示例：

bash

```
# 为用户 alice 设置读写权限
setfacl -m u:alice:rw /home/test.txt

# 为组 hr 设置只读权限
setfacl -m g:hr:r /home/test.txt

# 禁止用户 jack 访问
setfacl -m u:jack:- /home/test.txt
```

三、ACL 权限设置与验证

1. 创建测试文件

bash

```
[root@xnha ~]# touch /home/test.txt
[root@xnha ~]# ll /home/test.txt
-rw-r--r--. 1 root root 0 3月   6 09:34 /home/test.txt
```

2. 设置 ACL 权限

设置用户user01拥有读写权限，设置hr组只读权限

```
[root@xnha ~]# setfacl -m u:user01:rw /home/test.txt
[root@xnha ~]# setfacl -m g:hr:r /home/test.txt
```

3. 查看 ACL 权限：`getfacl`

bash

```
[root@xnha ~]# getfacl /home/test.txt
getfacl: Removing leading '/' from absolute path names
# file: home/test.txt
# owner: root
# group: r.oot
user::rw-
user:user01:rw-
group::r--
group:hr:r--
mask::rw-
other::r--
```

字段解析：

- `user::rw-`：属主权限。
- `user:user01:rw-`：用户 user01的 ACL 权限。
- `mask::rw-`：实际生效权限（与用户权限取交集）

四、权限删除与恢复

1. 删除单条 ACL

bash

```
# 删除组 hr 的 ACL 权限
[root@xnha tmp]# setfacl -x g:hr /home/test.txt
[root@xnha tmp]# getfacl /home/test.txt
getfacl: Removing leading '/' from absolute path names
# file: home/test.txt
# owner: root
# group: root
user::rw-
user:user01:rw-
group::r--
mask::rw-
other::r--
```

2. 清空所有 ACL

bash

```
[root@xnha tmp]# setfacl -b /home/test.txt
[root@xnha tmp]# getfacl /home/test.txt
getfacl: Removing leading '/' from absolute path names
# file: home/test.txt
# owner: root
# group: root
user::rw-
group::r--
other::r--

[root@xnha tmp]# ll /home/test.txt
-rw-r--r--. 1 root root 0 3月   6 09:34 /home/test.txt
```

3. 恢复默认权限

bash

```
# 重置为 UGO 权限（需提前备份）
[root@xnha tmp]# chmod 644 /home/test.txt
[root@xnha tmp]# ll /home/test.txt
-rw-r--r--. 1 root root 0 3月   6 09:34 /home/test.txt
```

五、高级特性与注意事项

1. 递归设置目录权限

bash

```
# 递归设置目录及子文件
setfacl -R -m u:alice:rwx /data/
```

2. 注意事项

- **权限优先级**：ACL > UGO（若冲突，以 ACL 为准）。
- 备份与恢复：

  bash

  ```
  # 备份
  getfacl -R /path > acl_backup.txt
  # 恢复
  setfacl --restore=acl_backup.txt
  ```

总结

- **ACL 核心价值**：解决 UGO 无法多用户精细化授权的问题，适用于团队协作场景。
- **最佳实践**：优先使用 UGO 满足基础需求，复杂场景再通过 ACL 补充。
- 命令速查：
  - 设置权限：`setfacl -m`

- 查看权限：`getfacl`
- 删除权限：`setfacl -x` 或 `-b`

三、特殊权限

一、特殊权限位

1. SUID（safe uid）

作用：suid针对文件/程序时，具备临时获得属主的权限。

通过实验理解特殊权限位

实验1：通过SUID临时提权

实验目的：让普通用户通过SUID临时获得root权限，查看受保护文件
前置条件：

- 使用root账号创建测试文件 /root/file1.txt
- 创建普通用户 alice
- 编辑 file1.txt

```
[root@xnha ~]# touch /root/file1.txt
[root@xnha ~]# useradd alice
[root@xnha ~]# vim /root/file1.txt
[root@xnha ~]# cat /root/file1.txt
echo 1111111111
[root@xnha ~]# ll /root/file1.txt
-rw-r--r--. 1 root root 16 3月   7 00:03 /root/file1.txt
```

- 切换用户 alice ,尝试访问/root/file1.txt

```
[root@xnha ~]# su - alice
[alice@xnha ~]$ cat /root/file1.txt
cat: /root/file1.txt: 权限不够
```

- 改变文件属主，再次尝试访问，依旧不行

```
[root@xnha ~]# chown alice /root/file1.txt
[root@xnha ~]# su - alice
[alice@xnha ~]$ cat /root/file1.txt
cat: /root/file1.txt: 权限不够
```

实验步骤：

步骤1：查看默认权限

bash

```
[root@xnha tmp]# ll /usr/bin/cat
-rwxr-xr-x. 1 root root 38480 1月  18 2023 /usr/bin/cat
```

步骤2：为`cat`命令添加SUID权限

bash

```
[root@xnha tmp]# ll /usr/bin/cat
-rwxr-xr-x. 1 root root 38480 1月  18 2023 /usr/bin/cat
[root@xnha tmp]# chmod u+s /usr/bin/cat
[root@xnha tmp]# ll /usr/bin/cat
-rwsr-xr-x. 1 root root 38480 1月  18 2023 /usr/bin/cat
 # 权限位出现's'表示SUID生效
```

步骤3：普通用户尝试访问受保护文件

bash

```
[root@xnha tmp]# su - user01
[user01@xnha ~]$ cat /root/file1.txt
dddd
```

步骤4：移除SUID权限（实验后必做！）

bash

```
[user01@xnha ~]$ exit
注销
[root@xnha tmp]# chmod u-s /usr/bin/cat
[root@xnha tmp]# ll /usr/bin/cat
-rwxr-xr-x. 1 root root 38480 1月  18 2023 /usr/bin/ca
```

思考：：

- 为什么设置SUID后普通用户能读取root文件?
  答案：SUID使程序运行时继承属主（root）权限，而非用户自身权限。
- 观察命令passwd的权限，为什么不一样

  答案：因为所有用户都有要修改自己密码的权力
- 观察文件 /etc/shadow的权限，为什么不一样

  答案：因为密码不允许任何人随意修改

二、文件属性chattr

作用：常用于锁定某个文件，拒绝修改。

分类：

有两个命令 **lsattr** 和 **chattr** 用来管理属性。下面是常用属性的列表。

属性	描述
a (append)	允许在文件中进行追加操作
A	这个属性不允许更新文件的访问时间
c (compressed)	启用这个属性时，文件在磁盘上会自动压缩
d (dump)	不能使用dump命令备份文件
D	设置了文件夹的D属性时，更改会在同步保存在磁盘上
e (extent format)	它表明，该文件使用磁盘上的块的映射扩展
i (immutable)	在文件上启用这个属性时，我们不能更改、重命名或者 删除这个文件
j (journaling)	设置了这个属性时，文件的数据首先保存在日志中，然后再写入文件
S (synchronous)	设置了这个属性时，变更或更改同步保存到磁盘上

chattr属性中可以使用的不同选项 ：

- **-R** 递归地修改文件夹和子文件夹的属性
- **-V** chattr命令会输出带有版本信息的冗余信息
- **-f** 忽略大部分错误信息

实验2：通过chattr锁定文件

实验目的：使用 `chattr` 防止文件被修改/删除
实验步骤：

步骤1：创建测试文件并查看默认属性

bash

```
[root@xnha tmp]# touch file100
[root@xnha tmp]# lsattr file100
-------------------- file100
```

步骤2：添加不可删除属性`i`

bash

```
[root@xnha tmp]# chattr +i file100
[root@xnha tmp]# lsattr file100
----i--------------- file100
```

步骤3：尝试删除文件（失败）

bash

```
[root@xnha tmp]# rm -rf file100
rm: 无法删除'file100': 不允许的操作
```

步骤4：恢复文件属性

bash

```
[root@xnha tmp]# chattr -i file100
[root@xnha tmp]# lsattr file100
-------------------- file100
 # 验证'i'属性移除
```

扩展属性：

- `a`**属性**：允许追加内容但禁止修改（适用于日志文件）

bash

```
[root@localhost ~]# chattr +a file200  # 追加内容可用`echo "text" >> file200`
```

三、进程掩码umask

实验3：修改umask控制默认权限

实验目的：理解umask如何影响新建文件/目录的默认权限
实验步骤：

步骤1：查看当前umask值

bash

```
[root@localhost ~]# umask
0022  # 默认掩码为0022
```

步骤2：创建文件/目录观察默认权限

bash

```
[root@localhost ~]# touch file800
[root@localhost ~]# mkdir dir800
[root@localhost ~]# ll file800 dir800 -d
-rw-r--r--. 1 root root 0 Mar 11 19:40 file800  # 文件权限644（666-022）
drwxr-xr-x. 2 root root 4096 Mar 11 19:40 dir800 # 目录权限755（777-022）
```

步骤3：临时修改umask为000（开放全部权限）

bash

```
[root@localhost ~]# umask 000
[root@localhost ~]# touch file900
[root@localhost ~]# mkdir dir900
[root@localhost ~]# ll file900 dir900
-rw-rw-rw-. 1 root root 0 Mar 11 19:44 file900  # 文件权限666
drwxrwxrwx. 2 root root 4096 Mar 11 19:44 dir900 # 目录权限777
```

步骤4：恢复umask为默认值

bash

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

计算公式：

- 文件默认权限 = `666 - umask`
- 目录默认权限 = `777 - umask`

四、安全注意事项

1. **SUID风险**：滥用SUID可能导致权限提升漏洞，生产环境慎用！
2. **chattr优先级**：`i`/`a`属性对root同样有效，需通过移除属性恢复操作。
3. **umask持久化**：临时修改仅对当前会话有效，永久修改需编辑`/etc/profile`或用户shell配置文件。

课后练习：

1. 为`/usr/bin/vim`设置SUID，观察普通用户能否编辑`/etc/shadow`。
2. 使用`chattr +a`保护日志文件，测试追加与覆盖写入的区别。
3. 计算umask为`0007`时，新建文件和目录的默认权限。