

# Shell文件权限和脚本执行

## 1.查看文件属性

ls -l file

## 2.权限类型

rwX 读,写,执行

-rw-r--r-- 1 root root 41 Jul 19 10:49 file

#root用户对file具有rw权限

#root组其他用户对file具有r权限

#other其他所有人对file具有r权限

#第一个root是创建file文件的所有者

#第二个root是创建file文件的所属组

1. 用户管理
2. 用户组
3. 权限分配

## 1.新建用户

```
useradd user1
```

## 2.删除用户

```
userdel -r user1
```

## 3.修改密码

```
passwd user1
```

## 4.查看用户

```
id user1
```

## 1.新建组

```
groupadd grp1
```

## 2.删除组

```
groupdel grp1
```

## 1.把用户加入组

```
gpasswd -a user1 grp1
```

#把用户user1加入组grp1

## 2.把用户从组删除

```
gpasswd -d user1 grp1
```

#把用户user1从grp1组中删除

1.chmod

2.acl

3.sudo

1)chmod 755 /mnt

2)chmod a+x file



1)设置权限

```
setfacl -m u:user1:rwx /test
```

2)查看权限

```
getfacl /test
```

3)删除user1在/test上的权限

```
setfacl -x u:user1 /test
```

4)删除/test上的所有acl权限

```
setfacl -b /test
```

5)设置acl的默认权限

```
setfacl -m d:u:user1:rwx test
```

#当前目录的子目录会继承目录的acl权限

visudo

- 需要密码:

```
user1 localhost=/usr/sbin/useradd
```

- 不需要密码:

```
user1 ALL=NOPASSWD: /usr/bin/passwd
```

- Shell脚本

- 1.用途：完成特定的、较复杂的系统管理任务
- 2.格式：集中保存多条Linux命令，普通文本文件
- 3.执行方式：按照预设的顺序依次解释执行

- 建立包含执行语句的脚本文件
  - #脚本文件中包括的内容
    - 运行环境设置：#!/bin/bash
    - 注释信息：以#开始的说明性文字
    - 可执行的Linux命令行
- 为脚本文件添加可执行权限

- **[root@localhost ~]# vi repboot.sh**
- `#!/bin/bash`
- `# To show usage of /boot directory and mode of kernel file.`
- `echo "Useage of /boot: "`
- `du -sh /boot`
- `echo "The mode of kernel file:"`
- `ls -lh /boot/vmlinuz-*`
- **[root@localhost ~]# chmod a+x repboot.sh**

- 1.直接执行具有 “x” 权限的脚本文件
  - ✓ 例如：`./repboot.sh`
- 2.使用指定的解释器程序执行脚本内容
  - ✓ 例如：`bash repboot.sh`、`sh repboot.sh`

每周五17:30清理FTP服务器的公共共享目录

- 检查 `/var/ftp/pub/` 目录，将其中所有子目录及文件的详细列表、当时的时间信息追加保存到 `/var/log/pubdir.log` 日志文件中，然后清空该目录

```
[root@localhost ~]# vi /opt/ftpclean.sh
#!/bin/bash
date >> /var/log/pubdir.log
ls -lhR /var/ftp/pub >> /var/log/pubdir.log
rm -rf /var/ftp/pub/*
```

```
[root@localhost ~]# crontab -e
30 17 * * 5 /opt/ftpclean.sh
```



云知梦，只为有梦想的人