

## 1.文件管理

2021年8月11日 14:30

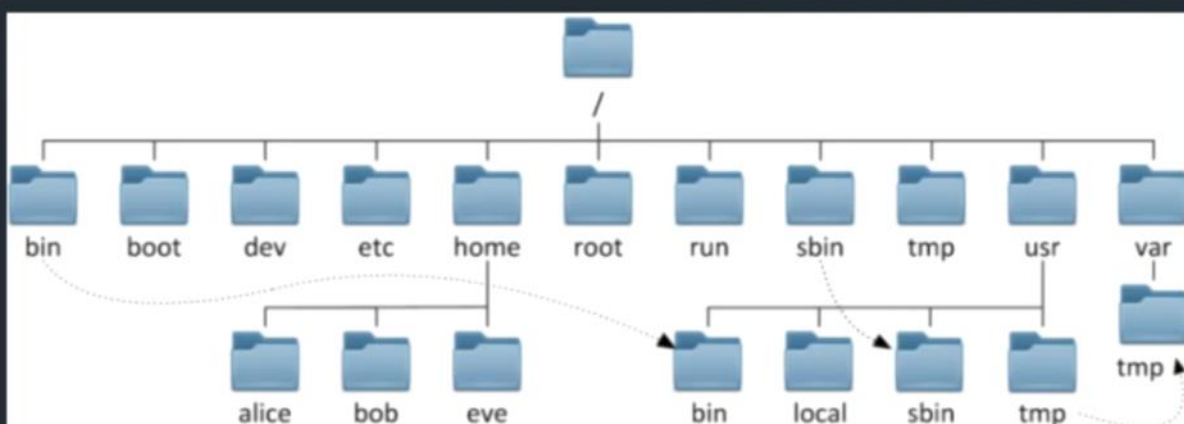
### 1.linux目标结构简洁

Windows:以多根的方式组织文件c:\D\E\

**Linux: 以单根的方式组织文件/**

1. Bin:执行命令文件
2. Boot:启动文件
3. Dev:驱动配置文件
4. Etc:配置文件
5. Home:存用户自己的文件, 普通用户的家目录
6. Root:root用户的目录
7. Sbin:超级命令目录, 只有管理员可以访问
8. Tmp:临时目录
9. Usr:程序安装的默认目录
10. Var:通常放的日志文件

## 2.Linux目录结构视图



常见文件类型: 查询文件列表命令: ls -l

- 普通文件 (文本文件, 二进制文件, 压缩文件, 电影, 图片。。。)

d 目录文件 (蓝色)

切换目录: cd /

cd 绝对路径或相对路径

cd 绝对路径 cd /home/alice

cd 相对路径 cd . ..

## 1.1Linux 命令语法

2021年8月21日 9:48

1. 清屏命令: `ctrl + l`
2. Mkdir 创建目录
  - a. 级联创建目录: `mkdir -p(父系)`
3. 复制: `cp` 原文件路径 目标文件夹 级联复制 `cp -r` 表示递归
4. 移动: `mv` 源文件路径 目标文件路径
5. 删除: `rm -rf` 文件或目录的路径 注: 删除 `rm -rf *` 通配符删除不掉隐藏文件



6. 查看文件内容:
  - a. Cat 全部内容
  - b. More 翻页
  - c. head 头部 查看文件的头两行, `head -2 /路径`
  - d. Tail 尾部
  - e. Grep 过滤关键字
7. 修改文件
  - a. 重定向: `ls -a > list.txt`
  - b. 文件编辑器1:
    - i. vim insert 模式 按yy复制 按p;
    - ii. 删除5行是5dd;按u撤销;
    - iii. dd 删除;
    - iv. 一起复制多行3yy就是复制3行;
  - c. 文件编辑器2: 可视化
    - i. 选中复制, 没有鼠标怎么办?
    - ii. 先按v,再动光标上下左右, 再按y, 再按p
  - d. 进入其他模式:

```
进入其它模式
a 进入插入模式
i 进入插入模式
o 进入插入模式
A 进入插入模式

: 进入末行模式 (扩展命令模式)
v 进入可视模式
ESC 返回命令模式
```

### f. 文本编辑:

```
文本编辑
y 复制
d 删除
p 粘贴
u undo撤销
```

### g. 光标定位:

```
光标定位
h j k l //上下左右
0 $ //行首行尾
gg G //页首页尾
3G 进入第三行
/string (n N 可以循环的) //查找字符, 按n键选下一个 (重要)
```

### h. 显示行号: `set nu`;取消行号: `set nonu`

设置环境

```
:set nu 设置行号
:set list 显示控制字符
:set nonu 取消设置行号
```

### i. 保存退出:

```
保存退出
:w 保存
:q 退出
:wq 保存并退出
```

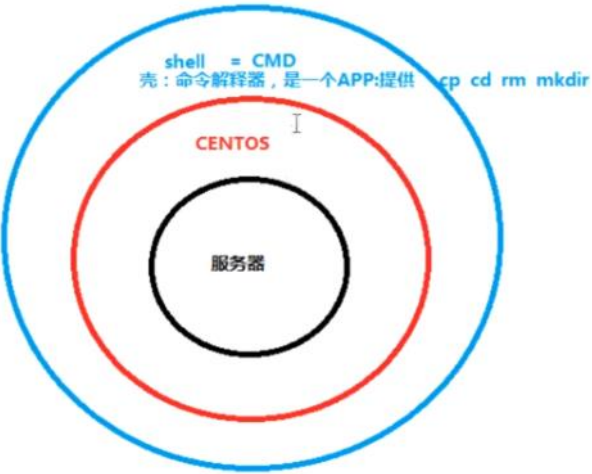
### j. 查找替换、另存:

vim 是一个应用程序  
CD 是另一个程序提供的。  
shell, 壳, 命令解释。

**bash: 是ashell和bshell的合并叫abshell**

```
查找替换
:范围 s/原内容/新内容/全局
:1,5 s/root/qianfeng/g 从1 - 5行的root 替
读入文件/写文件(另存为)
:w /tmp/aaa.txt 另存为/tmp/aaa.txt
```

命令（空格）	选项（空格）	参数
Ls /		
Ls -l		



## 1.2临时文件

2021年8月21日 14:25

场景：比如正在vim /1.txt,突然卡住重启了，没有保存，此时会生成临时文件/.1.txt.swap

需要删除

1. 查看隐藏文件 ls / ls -a /
2. 使用rm删除即可

```
[ root@localhost ~]# ls -a /  
.  
..  
1. txt  
2. txt  
. 1. txt. swp  
2. txt. swp  
4. txt  
5. txt  
6. txt  
88. txt  
aaa  
bin  
boot  
dev  
etc  
home  
lib  
lib64  
media  
mnt  
opt  
proc  
root  
run  
sbin  
srv  
sys  
tmp  
usr  
var  
var1  
[ root@localhost ~]# rm -rf /. 1. txt. swp
```

## 1.3相对路径

2021年8月21日 14:32

什么是绝对路径？有啥用？好吗？

从根开始描述。

举例：/home/abc/1.txt

定位文件。

能找到所有的文件。

绝对路径的缺点：路径太长

相对路径：前提是要cd到当前目录

什么是相对路径？有啥用？好吗？

两种表达式：

当前目录 .

当前目录 不打

上一级目录..

省去输入那么多的路径。

## 1.4 文件管理的作业

2021年8月21日 15:23

- 1.熟悉10个根目录下的一级目录的名称和功能。
- 2.熟悉至少3中文件类型
- 3.掌握改变目录的命令。
- 4.理解绝对路径和相对路径
- 5.掌握创建文件文件和目录的命令。
- 6.掌握两种查看文件的命令。
- 7.掌握vim文本编辑器的使用。

## 2.1 用户管理-用户信息存储的文件

2021年8月21日 15:24

用户信息存储的文件: /etc/passwd

文件内容由7列组成:

/etc/passwd (冒号分割为7列字段)

root:x:0:0:root:/root:/bin/bash

用户名:x:uid:gid:描述:HOME:shell

注: /bin/nologin不能登录, 没有shell解释器服务

root:用户名:登录系统的名字

X:密码占位符:, 具体内容不在此处

0:UID: 用户的身份证号

0:GID:GROUP 组号

root:描述: 比如经理 manager

/root:家目录: 登录系统时, 所在目录

/bin/bash:登录shell: 命令解释器

系统约定: RHEL7

uid: 0 特权用户

uid: 1~499 系统用户

uid: 1000+ 普通用户

2.2用户密码信息存储文件

2021年8月21日 16:14

文件: /etc/shadow

```
aaa:$6$DBd:18302:0:99999:7: :  
-----  
1         2         3         4         5         6         7         8         9
```

共8列:

- 1. 用户名
- 2. 密码加密值
- 3. 最后一次修改时间 (从1970/1/1距离现在多少天)
- 4. 最小时间间隔 (0代表当天可以改密码, 如果是5那么必须过5天才  
能改)
- 5. 密码最大时间间隔: 密码有效期, 99999代表永久. 14
- 6. 警告时间: (7), 是软规则, 提醒密码快过期
- 7. 不活动时间 (28, 用户不登陆系统. 超过28天, 禁用你)
- 8. 失效时间 (30, 到了30天账户就不能用了)
- 9. 保留位置

```
root:$1$MYG2NDG6$a1wtYr5GDM2esAPjug0YP0:15636:0:99999:7:::
```

1) “登录名” 是与/etc/passwd文件中的登录名相一致的用户账号

2) “口令” 字段存放的是加密后的用户口令字, 如果为空, 则对应用户没有口令, 登录时不需要口令;

星号代表帐号被锁定;

双叹号表示这个密码已经过期了。

\$6\$开头的, 表明是用SHA-512加密的,

\$1\$ 表明是用MD5加密的

\$2\$ 是用Blowfish加密的

\$5\$ 是用 SHA-256加密的。



## 2.3用户组信息存储文件

2021年8月21日 18:31

```
/etc/group
root:x:0:
组名:组密码:组ID:组成员
```

注：组员默认为空

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[ root@localhost ~]# head - 1 /etc/passwd
root: x: 0: 0: root: /root: /bin/bash
[ root@localhost ~]# head - 1 /etc/shadow
root: $6$Ns6m4pPdJom0FM52$T1lGkBLHpxBBz0LPYnLGct3naNP2e90CBvHnpzs
cqk25j2y. /MpAx0mgnn3LDPbvq9U2FLUIDbNl. : : 0: 99999: 7: : :
[ root@localhost ~]#
[ root@localhost ~]#
[ root@localhost ~]# head - 1 /etc/group
root: x: 0:
[ root@localhost ~]#
```

用户信息

组信息

## 2.4创建、删除用户

2021年8月21日 19:28

### 1. 创建用户命令: `useradd user01`

### 2. 查询用户是否创建成功:

`id user01`

`grep user01 /etc/passwd`

### 3. 修改密码: `passwd user01`

创建用户指定选项:

`-u` //创建用户, 指定用户的uid

```
[ root@localhost ~]# tail -1 /etc/passwd
user01: x: 1001: 1001: : /home/user01: /bin/bash
[ root@localhost ~]# useradd user02 -u 1503
[ root@localhost ~]# tail -2 /etc/passwd
user01: x: 1001: 1001: : /home/user01: /bin/bash
user02: x: 1503: 1503: : /home/user02: /bin/bash
[ root@localhost ~]#
```

`-d` //创建用户指定用户的家目录。注: 可以随便放, 但是不要放在  
boot, root下

`Useradd --help`寻求帮助

### 1. 删除用户命令: `userdel user02;`

`-r`一起删除用户的目录。

修改密码只能是root账户

### 2. 修改用户信息命令: `usermod -s`

SHELL	命令语法结构
<code>usermod -s niuniu2</code>	???
命令	选项 什么shell 用户参数

```
root@localhost:~# useradd niuniu
root@localhost:~# tail -1 /etc/passwd
niuniu:x:1001:1001:~/home/niuniu:/bin/bash
root@localhost:~# usermod -s /sbin/nologin niuniu
bash: usermod: command not found
root@localhost:~# usermod -s /sbin/nologin niuniu
root@localhost:~#
```

## 2.5 用户组

2021年8月22日 12:51

**创建命令:** `groupadd hr`

选项:

-g 修改组id (注: 该-g是基本组, 如果是-G是附加组)

**删除组命令:** `groupdelete hr`

用户AAA, 基本组1508改为1510, 附加组空

用户BBB, 基本组1509, 附加组1510

组CCC, 组号 1510, 开始和谁都没关系, 最有有两个成员, 一个AAA, 一个BBB

```
root@localhost home# useradd AAA
root@localhost home# grep AAA /etc/passwd
AAA:x:1002:1003:/home/AAA:/bin/bash
root@localhost home# groupadd CCC
root@localhost home# usermod AAA -g CCC
root@localhost home# grep AAA /etc/passwd
AAA:x:1002:1004:/home/AAA:/bin/bash
root@localhost home# grep CCC /etc/group
CCC:x:1004:
root@localhost home# usermod BBB
usermod: user 'BBB' does not exist
root@localhost home# useradd BBB
root@localhost home# usermod BBB -G CCC
root@localhost home# grep BBB /etc/passwd
BBB:x:1003:1005:/home/BBB:/bin/bash
root@localhost home# grep CCC /etc/passwd
root@localhost home# grep CCC /etc/group
CCC:x:1004:BBB
root@localhost home# AAA
```

**修改组命令:** `groupmod -g 修改组id`

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

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

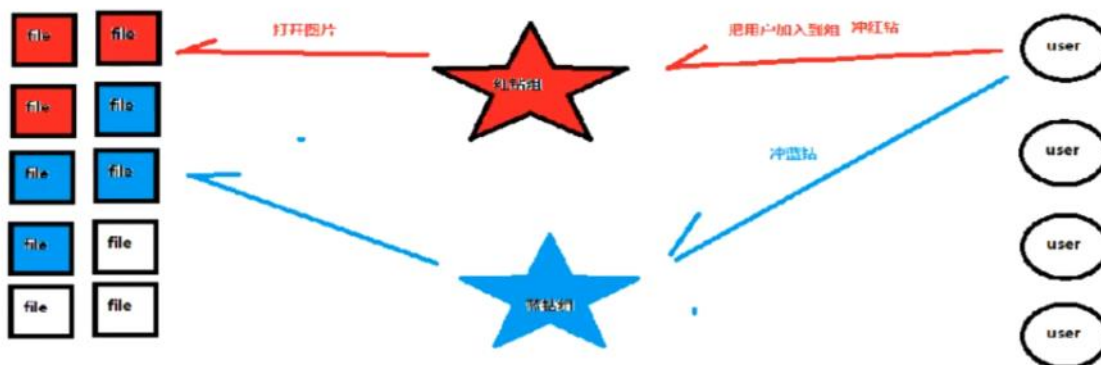
```
DDD: x: 1511:
```

```
[root@localhost ~]# groupmod -g 1512 DDD
```

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

```
DDD: x: 1512:
```

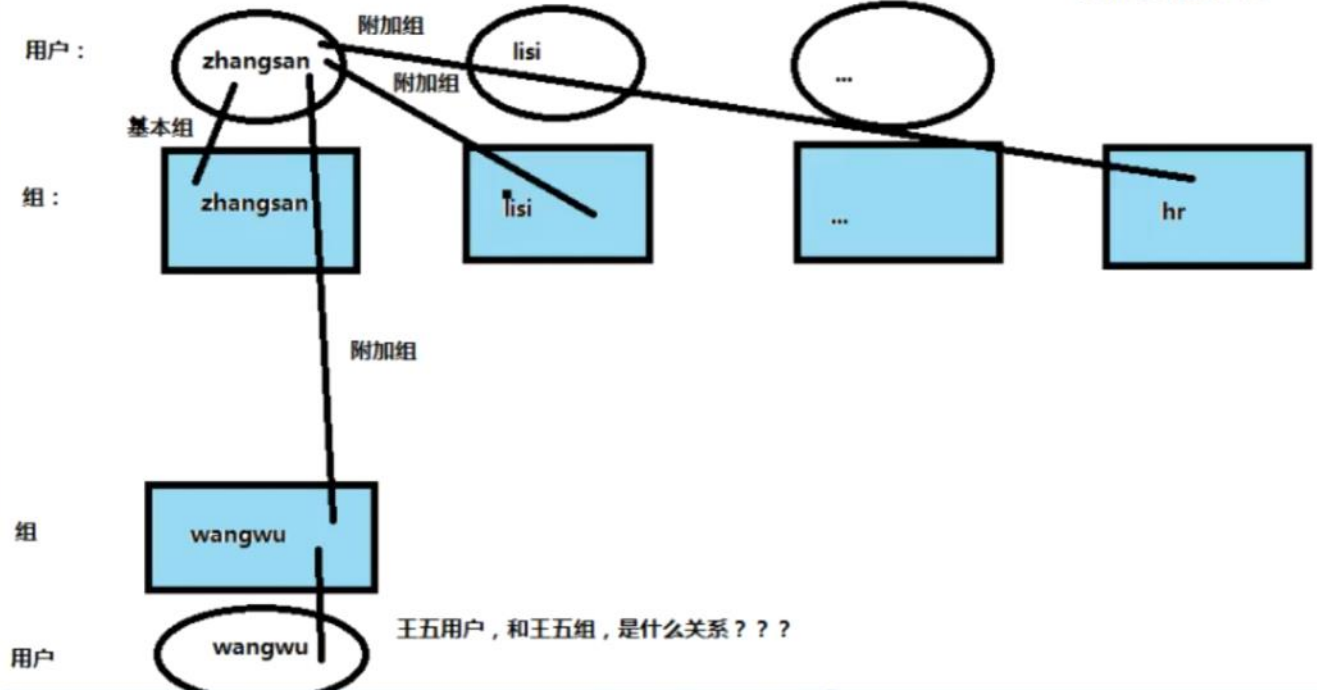
用户和组 和文件和权限的关系



知识点 组的类型

基本组 随用户而创建，组名同用户名  
附加组

基本组只有一个  
附加组可以有多个。



## 2.6提权

2021年8月22日 14:34

### 3.提权 (了解)

永久提权Switching users with su

临时提权Running commands as root with sudo

sudo的配置文件: /etc/sudoers

以root身份授权普通用户xulei

```
[root@qianfeng ~]# vim /etc/sudoers
```

```
%wheel    ALL=(ALL)    ALL
```

第107行: 允许wheel用户组中的用户在不输入该用户的密码的情况下使用所有命令

```
[root@qianfeng ~]# useradd xulei -G wheel
```

```
[root@qianfeng ~]# id xulei
```

```
uid=1001(xulei) gid=1001(xulei) 组=1001(xulei),10(wheel)
```

```
# passwd xulei
```

切换用户xulei登陆

```
[xulei@qianfeng ~]$ useradd gougou10
```

```
-bash: /usr/sbin/useradd: 权限不够
```

```
[xulei@qianfeng ~]$ sudo useradd gougou10
```

```
[xulei@qianfeng ~]$ id gougou10
```

```
uid=1002(gougou10) gid=1002(gougou10) 组=1002(gougou10)
```

## 2.7作业

2021年8月22日 15:15

8.理解账户和组的配置文件中字段含义

9.掌握创建和删除用户的命令。

10.理解su和sudo命令的区别。

### 3.1用户权限UGO

2021年8月22日 15:17

基本权限三个对象UGO：U代表自己，G代表组，O代表其他人

所有对象A（rwx） 修改给所有对象都赋予相同的权限

权限的三种类型：读写执行：4， 2， 1（r,w,x）

使用符号：u用户 g组 o其他 r读 w写 x执行  
语法 ① 语法：chmod 对象(u/g/o/a)赋值符(+/-/=)权限类型(r/w/x) 文件/目录

查看文件的属性。讲解有哪些类型。还没有验证。

```
[root@localhost tmp]# ll file1
-rw-r--r--. 1 root root 0 4月 13 20:49 file1
类型和权限 链接 属主 属组 大小 创建时间 文件名
```

1. 给file授予，属主读的权限，没有写和执行。

```
chmod u=r file
```

2. 给file授予，属组读，写和执行的权限。

```
chmod g=rwx file
```

3. 给file授予，其他用户，没有读写和执行的权限。

```
chmod o= file 对
```

```
chmod o-rwx file
```

```
chmod o=--- file
```

```
ls: cannot access file1.txt: No such file or directory
[liai@localhost ~]$ ls -l file1.txt
-rw-rw-r--. 1 liai liai 0 Aug 22 06:37 file1.txt
[liai@localhost ~]$ chmod a=r-x file1.txt
[liai@localhost ~]$ ls -l file1.txt
-r--r--r--. 1 liai liai 0 Aug 22 06:37 file1.txt
[liai@localhost ~]$ chmod a=rwx file1.txt
[liai@localhost ~]$ ls -l file1.txt
-rwxrwxrwx. 1 liai liai 0 Aug 22 06:37 file1.txt
[liai@localhost ~]$ chmod u=r file1.txt
[liai@localhost ~]$ ls -l file1.txt
-r--rwxrwx. 1 liai liai 0 Aug 22 06:37 file1.txt
[liai@localhost ~]$ chmod g=rwx file1.txt
[liai@localhost ~]$ ls -l file1.txt
-r--rwxrwx. 1 liai liai 0 Aug 22 06:37 file1.txt
[liai@localhost ~]$ chmod o=--- file1.txt
[liai@localhost ~]$ ls -l file1.txt
-r--rwx---. 1 liai liai 0 Aug 22 06:37 file1.txt
[liai@localhost ~]$ _
```

1. 通过chmod 数字授权，用户读写执行，组读写，其他无  
chmod 760 file1.txt

2. 通过chmod 数字授权，用户读写执行，组读写执行，其他读写执行  
chmod ?? file1.txt

3. 通过chmod 数字授权，用户无，组无，其他读写执行  
chmod ?? file1.txt



## 3.2更改属主、组

2021年8月22日 19:50

Chown :修改主人和所属组

Chgrp:修改所属组

-r 递归修改权限，对于文件夹中所有文件及文件夹都生效  
(chmod,chown,chgrp都适用)

## 用户权限ACL

2021年8月22日 20:50

ACL access control list 限制用户对文件的访问  
ACL是UGO 的补充，或者说是加强版。

ACL 基础命令格式示例：

```
setfacl -m u:alice:rw /home/test.txt
```

命令 设置 用户或组:用户名:权限 文件对象

命令

Setfacl -m g:hr:rw /home/files 设置文件访问 -设置 对象: 对象名: 权限

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

拒绝权限:

jack的权限是拒绝权限，就算其他用户的权限是读取，但是拒绝权限也是优先于其他用户权限

组设置: 组的统一权限是R，设置ACL后查看时就是叠加权限rwx了

```
[root@localhost tmp]# setfacl -m g:hr:r-x /home/test.txt
[root@localhost tmp]# getfacl /home/test.txt
getfacl: Removing leading '/' from absolute path names
# file: home/test.txt
# owner: root
# group: root
user::rw-
user:alice:rw-
user:jack:---
group::r--
group:hr:r-x
mask::rwx
other::r--
```

```
[root@localhost tmp]# ls -l /home/test.txt
-rw-rwxr-- + 1 root root 0 2月 17 11:37 /home/test.txt
[root@localhost tmp]#
```

有ACL权限查看时有+

```
[root@localhost tmp]# ls -l /home/test.txt
-rw-rwxr-- + 1 root root 0 2月 17 11:37 /home/test.txt
[root@localhost tmp]#
```

请思考命令中的o是什么作用 ☹ [root@qianfeng ~]# setfacl -m o::rw /home/test.txt

注：两个：就是用户名省略了

```
setfacl -m u:alice:rw file1.txt
setfacl -m o::rw file1.txt
```

移除ACL权限命令：

Setfacl -x 对象：对象名字 路径

Setacl -b可以把所有ACL扩展出的权限全都移除

```
[root@localhost tmp]# setfacl -x u:alice /home/test.txt
[root@localhost tmp]# getfacl /home/test.txt
getfacl: Removing leading '/' from absolute path names
# file: home/test.txt
# owner: root
# group: root
user::rw-
user:root:rwX
user:jack:---
group::r--
group:hr:r-x
mask::rwX
other::rw-
[root@qianfeng ~]# setfacl -m g:hr:r /home/test.txt
[root@qianfeng ~]# setfacl -x g:hr /home/test.txt //删除组hr的acl权限
[root@qianfeng ~]# setfacl -b /home/test.txt //删除所有acl权限
```