

目录的相关操作：

- . 此层目录
- .. 上一层目录
- 前一个工作目录
- ~ 目前用户身份所在的家目录，比如~Ray 表示 Ray 这个用户的家目录

请问在 Linux 底下，根目录下有没有上层目录(..)存在？

```
[root@localhost ~]# cd /  
[root@localhost /]# ls -al  
总用量 74  
dr-xr-xr-x. 18 root root 4096 3月 7 20:17 .  
dr-xr-xr-x. 18 root root 4096 3月 7 20:17 ..  
lrwxrwxrwx. 1 root root    7 3月 7 20:01 bin
```

确实有，但是二者的权限、属性完全一致，就是说根目录的上一层目录就是它本身

几个常见的处理目录的指令：

- cd: 变换目录
- pwd: 显示当前目录（pwd 是 Print Working Directory 的缩写）
- mkdir: 建立一个新的目录
- rmdir: 删除一个空的目录

cd 是 Change Directory 的缩写

cd ~Ray

代表去到 Ray 这个用户乱家目录，亦即 /home/Ray

cd ~

回到自己的家目录，即/root

cd

同上回到/root

cd -

回到上一个目录

mkdir -p test1/test2/test3/test4 可以建立多层目录

mkdir -m 711 test2 建立权限为 rwx--x--x 的目录

如果没有加上 -m 来强制设定属性，系统会使用默认属性。

rmdir+目录 可以删除空目录，被删除的目录里面必定不能存在其他的目录或文件。

rmdir -p test1/test2/test3/test4 利用 -p 这个选项，立刻就可以将 test1/test2/test3/test4 一次删除

如果要将目录连同目录下所有东西都删除，则需要用指令：

rm -r +目录

环境变量：

当我们执行一个指令时，比如 ls，系统会先按照 PATH 的设定去每个 PATH 定义的目录下搜寻文件名为 ls 的可执行文件，先搜寻到的先执行

echo \$PATH 可以看到 PATH 定义了哪些目录

PATH 变量是由很多目录组成的，目录之间用 (:) 来隔开。/bin 这个目录 PATH 中，因此在

任何地方都能够找到 (/bin/ls) 以执行 ls

尝试把 bin 目录下的 ls 移动到 root 目录下，然后 ls 指令就执行不了了

```
mv: 无法移动 '/bin/ls' 到 '/root': 文件已存在
[root@localhost bin]# mv /bin/ls /root
[root@localhost bin]# cd /root
[root@localhost ~]# ls
bash: ls: 未找到命令...
相似命令是: 'lz'
```

这样是可以的，指定目录

```
[root@localhost home]# /root/ls
linuxidc lost+found Ray
```

再改一下 PATH，让系统能找到 root 目录下的 ls

```
[root@localhost home]# PATH="$PATH":/root
[root@localhost home]# ls
linuxidc lost+found Ray
```

如果我有两个 ls 指令在两个不同目录，例如 /usr/local/bin/ls 和 /bin/ls 那么当我下达 ls 指令的时候，哪个 ls 会被执行？

当然是 PATH 中写在前面的那个

目录、文件管理：

基本操作：显示属性、复制、删除、移动

ls -a 显示全部文件（包含隐藏文件）

ls -F 给予附加数据结构：*表示可执行文件、/表示目录、=表示 socket 文件、|表示 FIFO 档案

```
[root@localhost Ray]# ls -F
Desktop/   Downloads/ Music/     Public/   test0.txt
Documents/ haha_copy.txt Pictures/  Templates/ Videos/
```

ls -l 显示文件属性与权限等（常用）

复制、删除与移动： cp, rm, mv

cp -i 如果目标文件已存在，会先询问是否覆盖

cp -p 同时复制文件的属性，而不是使用默认属性

cp -s 复制成为符号链接文件，即快捷方式

复制前：

```
-rw-rw-r--. 1 root utmp 77184 3月 15 21:14 wtmp
```

直接复制后：

```
-rw-r--r--. 1 root root 77184 3月 15 21:47 wtmp
```

用 Ray 账号复制：

```
-rw-rw-r--. 1 Ray Ray 77184 3月 15 21:54 wtmp3
```

如果使用 cp -a 复制：

```
-rw-rw-r--. 1 root utmp 77184 3月 15 21:14 wtmp2
```

可见属性也完全复制了过去

建立快捷方式：

```
[Ray@localhost tmp]$ cp -s bashrc bashrc_slink
[Ray@localhost tmp]$ ls -l bashrc*
-rw-r--r--. 1 root root 176 3月 15 21:46 bashrc
lrwxrwxrwx. 1 Ray Ray 6 3月 15 21:59 bashrc_slink -> bashrc
```

对这个链接文件进行复制操作：

```
[Ray@localhost tmp]$ cp bashrc_slink bashrc_slink1
[Ray@localhost tmp]$ cp -d bashrc_slink bashrc_slink2
```

```
lrwxrwxrwx. 1 Ray Ray 6 3月 15 21:59 bashrc_slink -> bashrc
-rw-r--r--. 1 Ray Ray 176 3月 15 22:05 bashrc_slink1
lrwxrwxrwx. 1 Ray Ray 6 3月 15 22:05 bashrc_slink2 -> bashrc
```

可见要加上-d 才能复制到链接文件，不然直接把原文件复制过去了

可以复制多个文件到同一个目录：

cp 源文件 1 源文件 2 目标文件夹

rm -i 删除前会询问是否确定要删除

rm -r 递归删除，将目录下全部东西都删除掉

可以通过通配符*，将前缀相同的文件、目录全部删除

mv -f 如果目标文件已存在，则强制覆盖

mv -i 如果目标文件已存在，则询问是否覆盖

mv -u 如果目标文件已存在，若 source 更新则更新

mv name1 name2 实现重命名

mv file1 file2 document 将两个文件移动到一个文件夹里

文件内容查阅：cat、more、less

cat 由第一行开始显示文件内容

more 一页页的显示

less 一页页显示、同时可以向前翻页

cat -n 可以打印行号

more 打开文件后：

空格键：向下翻页

Enter：下翻一行

/字符串：搜寻此字符串

q：立即离开 more

b：往回翻页

less：

- 空格键 : 向下翻动一页 ;
- [pagedown] : 向下翻动一页 ;
- [pageup] : 向上翻动一页 ;
- /字符串 : 向下搜寻『字符串』的功能 ;
- ?字符串 : 向上搜寻『字符串』的功能 ;
- n : 重复前一个搜寻 (与 / 或 ? 有关 !)
- N : 反向的重复前一个搜寻 (与 / 或 ? 有关 !)
- q : 离开 less 这个程序 ;

head 文件名: 默认显示前 10 行

head -n 20 文件名: 显示前 20 行

同理, tail 是从后面显示

查看纯文本文件用以上指令, 而如果用以上指令查看非纯文本文档 (比如执行文件 /usr/bin/passwd, 是 binary file), 会显示乱码。

这个时候用 od 即可

touch 指令的作用:

建立空文件、将某个文件

日期修订为当前时间 (mtime (内容数据改变时更新)、atime (文件内容被取用时更新))

ls -l 默认显示 mtime

ctime (状态改变时更新)

ll 指令就是 ls -l 指令别名

可以连续下达几条指令, 指令之间用分号 (;) 分隔即可, 指令按顺序执行:

```
[root@localhost ~]# ls
anaconda-ks.cfg  Desktop  Documents  haha_copy.txt  Music  Public  Videos
Downloads  haha.txt  Pictures  Templates
[root@localhost ~]# rm haha.txt;ll
rm: 是否删除普通文件 "haha.txt"? y
总用量 40
-rw-----. 1 root root 1614 3月  7 20:20 anaconda-ks.cfg
drwxr-xr-x. 2 root root 4096 3月  8 19:57 Desktop
drwxr-xr-x. 2 root root 4096 3月  8 19:57 Documents
drwxr-xr-x. 2 root root 4096 3月  8 19:57 Downloads
-rwxr-----. 1 root root  9 3月 11 22:56 haha_copy.txt
drwxr-xr-x. 2 root root 4096 3月  8 19:57 Music
drwxr-xr-x. 2 root root 4096 3月  8 19:57 Pictures
drwxr-xr-x. 2 root root 4096 3月  8 19:57 Public
drwxr-xr-x. 2 root root 4096 3月  8 19:57 Templates
drwxr-xr-x. 2 root root 4096 3月  8 19:57 Videos
```

ctime 是改变不了的, 但 atime 和 mtime 可以改变, 用 touch 指令

```
[root@localhost tmp]# touch -d "3 days ago" bashrc
[root@localhost tmp]# ll bashrc;ll --time=atime bashrc;ll --time=ctime bashrc
-rw-r--r--. 1 root root 176 3月 13 17:39 bashrc
-rw-r--r--. 1 root root 176 3月 13 17:39 bashrc
-rw-r--r--. 1 root root 176 3月 16 17:39 bashrc
```

或者改为具体时间:

```
[root@localhost tmp]# touch -t 1803151742 bashrc
[root@localhost tmp]# ll bashrc;ll --time=atime bashrc;ll --time=ctime bashrc
-rw-r--r--. 1 root root 176 3月 15 17:42 bashrc
-rw-r--r--. 1 root root 176 3月 15 17:42 bashrc
-rw-r--r--. 1 root root 176 3月 16 17:42 bashrc
```

mtime、atime 都改变为具体时间，而 ctime 是目前时间

平时看的文件属性里，这三个时间最重要的还是 mtime（内容改变时更新）

例题：在/tmp 下建立一个目录，名称为 chapter7_1，其拥有者为 Ray，群组为 users，任何人都可以进入该目录，但除了 dmtsai 外，其他人不能修改该目录下的档案。

权限属性应该是 d rwx r-x r-x Ray users ...

```
[root@localhost tmp]# mkdir chaptr7_1
[root@localhost tmp]# chown -R Ray:users /tmp/chaptr7_1/
[root@localhost tmp]# chmod -R 755 /tmp/chaptr7_1/
[root@localhost tmp]# ll
总用量 732
-rw-r--r--. 1 root root 2784 3月 8 20:12 anaconda.log
-rw-r--r--. 1 root root 176 3月 15 17:42 bashrc
lrwxrwxrwx. 1 Ray Ray 6 3月 15 21:59 bashrc_slink -> bashrc
-rw-r--r--. 1 Ray Ray 176 3月 15 22:05 bashrc_slink1
lrwxrwxrwx. 1 Ray Ray 6 3月 15 22:05 bashrc_slink2 -> bashrc
drwxr-xr-x. 2 Ray users 4096 3月 16 17:56 chaptr7_1
```

新建文件或目录时的默认权限是什么？

```
[root@localhost tmp]# umask -S
u=rwx,g=rX,o=rX
```

按道理来说，对于文件，它一般是用来记录数据的，并不需要 x 执行的权限，预设权限应为：

-rw-rw-rw 即 666

对于目录，有 x 权限才能进入此目录，因此预设全部权限：drwxrwxrwx

umask 的分数指的是『该默认值需要减掉的权限！』

umask=022 代表 group 和 others 被拿掉了 w 权限

这样子新建的文件，就只有自己能编辑，同一群组的人不能编辑，为了改变这种状况，默认权限应该设置为 002，因此可以：

原本：

```
[root@localhost tmp]# touch a1
[root@localhost tmp]# ll
总用量 732
-rw-r--r--. 1 root root 0 3月 16 18:59 a1
```

修改 umask 后：

```
[root@localhost tmp]# umask 002
[root@localhost tmp]# touch a2
[root@localhost tmp]# ll
总用量 732
-rw-r--r--. 1 root root 0 3月 16 18:59 a1
-rw-rw-r--. 1 root root 0 3月 16 18:59 a2
```


chattr 配置文件的隐藏属性

+ 增加属性 -移除属性 =设定属性

a: 设定 a 后，这个文件将只能增加数据，不能删除或修改数据，且只有 root 能设定这个属性

i: 设定 i 后，文件不能被移除、重命名、设置连接文件（快捷方式）、写入或新增资料。

s: 删除文件后会彻底移出硬盘

u: 删除文件后仍留在硬盘，还可以恢复

试一下隐藏属性 i，i 属性可以让一个文件无法被变动：

```
[root@localhost tmp]# touch attrtest
[root@localhost tmp]# chattr +i attrtest
[root@localhost tmp]# rm attrtest
rm: 是否删除普通空文件 "attrtest"? y
rm: 无法删除"attrtest": 不允许的操作
[root@localhost tmp]# chattr -i attrtest
[root@localhost tmp]# rm attrtest
rm: 是否删除普通空文件 "attrtest"? y
[root@localhost tmp]#
```

可以用 lsattr 查看文件的隐藏属性

```
[root@localhost tmp]# touch test0316
[root@localhost tmp]# chattr +aij test0316
[root@localhost tmp]# lsattr test0316
----ia---j----- test0316
```

文件特殊权限：SUID、SGID、SBIT

当属性 s 出现在 x 的位置上时，就成为 Set UID，简称 SUID 特殊权限（主要功能：执行者将暂时具有该程序拥有者（owner）的权限）

Linux 系统中所有账号密码都记录在/etc/shadow 这个文件里，其权限为：

-r-- --- --- 1 root root，照理说应该只有 root 能读和写（这里很迷，明明不能写？）

为什么一般用户却能通过 passwd 指令修改自己的密码？

原因：Ray 用户对/usr/bin/passwd 有 x 权限，即能执行；

passwd 的拥有者是 root；

```
-rwsr-xr-x. 1 root root 27832 6月 10 2014 /usr/bin/passwd
```

passwd 有 SUID 特殊权限

Ray 执行 passwd 过程，能暂时获取 root 权限，于是 Ray 能修改/etc/shadow

同理 SGID 就是执行者暂时获取群组的权限

如果一个目录设定了 SGID 的权限，则

- 1、用户若对这个目录有 r、x 权限，则用户能进入该目录
- 2、用户在此目录下的有效群组变为该目录的群组
- 3、用户在此目录下新建的文件的群组与此目录的群组相同

SBIT 只能用于目录，它的作用是，某个文件只能由它所属的用户和 root 能删除、重命名、移动，而其他用户即使拥有这个目录的 w 权限也无法删除该文件。

设置以上三个特殊权限：

4: SUID

2: SGID

1: SBIT

比如文件原本权限为 755，如果要设置 SUID 和 SGID 权限，则可以 `chmod 6755 filename`

观察文件类型：file

查看文件是属于 ASCII、data、还是 binary 类型的文件等，可以用 file 指令

```
[root@localhost ~]# file ~/.bashrc
/root/.bashrc: ASCII text
```

比如 说明是 ASCII 格式的纯文本文件

指令与文件的搜寻

用 which 寻找指令的执行文件：

```
[root@localhost ~]# which ifconfig
/usr/sbin/ifconfig

[root@localhost ~]# which cd
/usr/bin/cd
```

which 预设是寻找 PATH 内规范的目录

搜寻文件一般先用 whereis 或 locate 来检查，找不到再用 find，因为 Linux 系统将所有文件都记录在一个数据库文件中，而前者利用数据库来搜寻数据，后者直接搜寻硬盘。

```
[Ray@localhost root]$ whereis ifconfig
ifconfig: /usr/sbin/ifconfig /usr/share/man/man8/ifconfig.8.gz
```

whereis 可以加入选项来寻找相关的数据

-b 只搜寻 binary 可执行文件等等。

locate 指令的使用更简单，locate+部分文件名 就可以了，即使忘记完整文件名也能找出来

有时候 whereis 和 locate 会找不到一些新近的文件，因为数据库每天更新一次。

可以手动更新数据库：updatedb，但是要等几分钟时间

find 参数：

1、与时间有关：

-mtime -n: 指 n 天之前（不包含 n 天本身）被更改过内容的文件

-mtime n: 指 n 天之前的那一天内被更改过内容的文件

-mtime +n: n 天之内被改动过的文件

比如：过去 24 小时内更改过内容的文件：

```
[root@localhost ~]# find -mtime 0
./.cache/abrt
./.cache/abrt/lastnotification
```

寻找/etc 目录下那些文件日期比/etc/passwd 更新的文件：

```

[rechner:/usr/c/usbnotifier]
[root@localhost ~]# find /etc -newer /etc/passwd
/etc
/etc/dbus-1/system.d
/etc/systemd/system
/etc/systemd/system/sockets.target.wants
/etc/systemd/system/sockets.target.wants/avahi-daemon.socket
/etc/systemd/system/dev-virtio\x2dports-org.qemu.guest.agent.0.dev

```

2、与使用者有关

`find /home -user Ray`

找出系统中属于 Ray 用户的所有文件

`find / -nouser`

找出系统中不属于任何人的文件（不正常文件）

还有 `-group` 等等

3、与文件权限或名字有关的参数：

`-name filename` :名字为 filename 的文件

`-size [+|-]SIZE` :文件大小 大于小于 SIZE 的文件 单位 c:byte k:1024bytes

`-type TYPE` :一般正规文件（f），目录（d），连接文件（l）等

`-perm mode` :权限等于 mode 的文件

`-perm -mode` :权限包括 mode 的文件

寻找名为 passwd 的文件

```

[rechner:/usr/c/usbnotifier]
[root@localhost ~]# find / -name passwd
/etc/pam.d/passwd
/etc/passwd
/usr/bin/passwd
/usr/share/bash-completion/completions/passwd

```

寻找/var 目录下类型为 socket 的文件

```

[rechner:/usr/c/usbnotifier]
[root@localhost ~]# find /var -type s
/var/lib/gssproxy/default.sock
/var/spool/postfix/private/discard
/var/spool/postfix/private/proxywrite
/var/spool/postfix/private/defer

```

`find / -perm +7000`

就是找出含有 SGID 或 SUID 或 SBIT 属性的文件

`find /bin /sbin -perm +6000`

`find` 后也可以接多个目录

`find / -size +1000k`

找出系统中大于 1MB 的文件

第六章主要讲文件权限的意义，第七章讲文件系统的管理指令，下面结合回顾：

一、让用户进入某目录工作的权限：

1、使用指令：`cd`（变换工作目录）

2、目录所需权限：用户对此目录有 x 权限

3、附：如果用户想用 `ls` 查阅文件名，则需要 r 权限

二、用户在某个目录下读取一个文件

- 1、指令：cat、more、less
- 2、目录所需权限：用户对这个目录有 x 权限
- 3、文件所需权限：用户对这个文件有 r 权限

三、使用者可以修改一个文件的权限

- 1、指令：nano、vi
- 2、目录所需权限：x
- 3、文件所需权限：r、w

四、使用者可以创建新文件的权限

目录所需权限：w、x 权限

五、用户进入某目录并执行该目录下某个指令的权限

- 1、目录所需权限：x
- 2、文件所需权限：x

问一个使用者 能够进行『cp /dir1/file1 /dir2』的指令时，请说明 dir1, file1, dir2 的最小所需权限为何？

dir1:x file1:r, dir2:w,x

d rwx rws --- root project