

用户数据：即文件数据块，记录文件的真实内容
元数据：文件的附加属性，如文件大小，创建时间等，
inode是文件的唯一标识

第5章文件和目录

5.1

Linux 文件

5.2

Linux 目录结构

5.3

文件和目录操作

5.4

链接文件

Linux系统与Windows系统有很大的不同，它以目录的形式挂载文件系统，其目录结构是一个分层的树形结构。链接是一种在共享文件和访问它的用户的若干目录项之间建立联系的方法，Linux系统中包括硬链接和软链接两种方式。

5.1 Linux文件

在Linux系统下除了一般文件之外，所有的目录和设备（光驱、硬盘等）都是以文件的形式存在的，本节主要讲述Linux文件扩展名和Linux文件类型。

5.1.1 Linux文件的扩展名

文件扩展名是文件名的最后一个点（.）之后的部分，下面简单列出了一些在Linux系统中的文件扩展名和它们的含义。

1. 压缩和归档的文件

压缩和归档的文件扩展名及其含义如下。

- .bz2: 使用bzip2压缩的文件。
- .gz: 使用gzip压缩的文件。
- .tar: 使用tar压缩的文件，又称tar文件。
- .tbz: 使用tar和bzip压缩的文件。
- .tgz: 使用tar和gzip压缩的文件。
- .zip: 使用zip压缩的文件，Linux下使用gzip命令压缩的文件。

2. 文件格式

文件格式扩展名及其含义如下。

- . au: 音频文件。
- . gif: GIF图像文件。
- . html/. htm: HTML文件。
- . jpg: JPEG图像文件。
- . pdf: PDF文档。

- .png: PNG图像文件。
- .ps: PostScript文件，为打印而格式化过的文件。
- .txt: 纯ASCII文本文件。
- .wav: 音频文件。
- .bmp: 图像文件。

3. 系统文件

系统文件扩展名及其含义如下。

.conf: 一种配置文件，配置文件有时也使用.cfg。

.lock: 锁文件，用来判定程序或设备是否正在被使用。

.rpm: Linux系统中用来安装软件的软件包管理器文件。

4. 编程和脚本文件

编程和脚本文件扩展名及其含义如下。

- .c: C语言的源码文件。
- .cpp: C++语言的源码文件。
- .h: C或C++语言的头文件。
- .o: 程序的对象文件。

- . pl: Perl脚本。
- . py: Python脚本。
- . so: 库文件。
- . sh: Shell脚本。
- . tcl: TCL脚本。

文件扩展名不总是被使用或被一致地使用。如果一个文件没有扩展名，或者与它的扩展名不相符时，可以使用file命令查看该文件的类型。

```
[root@PC-LINUX ~]# file it  
it: ASCII text
```

5.1.2 Linux文件类型

Linux文件类型和Linux文件的文件名所代表的意义是两个不同的概念。通过一般应用程序创建的文件，比如file.txt, file.tar.gz, 这些文件虽然要用不同的程序来打开，但放在Linux文件类型中衡量的话，大多称之为普通文件。

Linux文件类型常见的有：普通文件、目录文件、字符设备文件、块设备文件和符号链接文件等。

1. 普通文件

用“ls -lh”命令查看某个文件的属性，可以看到有类似“-rw-r--r--”的属性符号，其属性第一个符号是“-”，这样的文件在Linux系统中就是普通文件。这些文件一般是用一些相关的应用程序创建，比如图像工具、文档工具或归档工具等。

2. 目录文件

当在某个目录下执行“ls -lh”命令，看到有类似“drwxr-xr-x”的属性符号，其属性第一个符号是“d”，这样的文件在Linux系统中就是目录文件。

3. 设备文件

Linux系统下的/dev目录中有大量的设备文件。

(1) 块设备文件

块设备的主要特点是可以随机读写，而最常见的块设备就是磁盘，如/dev/hda1、/dev/sda2等。用“ls -l”命令查看某个文件的属性，可以看到有类似“brw-rw----”的属性符号，其属性第一个符号是“b”，这样的文件在Linux系统中就是块设备文件。

(2) 字符设备文件

最常见的字符设备文件是打印机和终端，可以接收字符流。`/dev/null`是一个非常用的字符设备文件，送入这个设备的所有内容都被忽略。用“`ls -l`”命令查看某个文件的属性，可以看到有类似“`crw-r-----`”的属性符号，其属性第一个符号是“`c`”，这样的文件在Linux系统中就是字符设备文件。

4. 管道文件

管道文件有时候也被叫做FIFO文件（FIFO是先进先出的意思），管道文件就是从一头流入，从另一头流出。用“ls -l”命令查看某个文件的属性，可以看到有类似“prw-----”的属性符号，其属性第一个符号是“p”，这样的文件在Linux系统中就是管道文件。

5. 链接文件

链接文件有点类似于Windows系统中的快捷方式，链接有两种方式：软链接和硬链接。

5. 链接文件

(1) 软链接文件

软链接（也叫符号链接），文件用户数据块中存放的内容是另一文件的路径名的指向。软链接就是一个普通文件，只是数据块内容有点特殊。软链接可对文件或目录创建。删除软链接并不影响被指向的文件，但若被指向的原文件被删除，则相关软连接就变成了死链接。

用“ls -l”命令查看某个文件的属性，可以看到有类似“lrwxrwxrwx”的属性符号，其属性第一个符号是“l”，这样的文件在Linux系统中就是软链接文件。

5. 链接文件

(2) 硬链接文件

硬链接就是同一个文件使用了多个别名（他们有共同的 **inode**）。由于硬链接是有着相同 **inode** 号仅文件名不同的文件，因此，删除一个硬链接文件并不影响其他有相同 **inode** 号的文件。硬链接不能对目录进行创建，只可对文件创建。

用“`ls -l`”命令查看某个文件的属性，可以看到第二列的文件硬链接数大于1，这样的文件在Linux系统中就是硬链接文件。

5.2 Linux目录结构

Linux系统都有根文件系统，它包含系统引导和使其他文件系统得以挂载所必要的文件，根文件系统需要有单用户状态所必须的足够的内容，还应该包括修复损坏系统、恢复备份等工具。

Linux系统的目录结构是分层的树形结构，都是挂载在根文件系统“/”下。

Linux系统中的常见目录

- | | | |
|----------|------------|-----------------|
| 1. /home | 7. /tmp | 13. /selinux |
| 2. /root | 8. /mnt | 14. /var |
| 3. /bin | 9. /boot | 15. /etc |
| 4. /sbin | 10. /proc | 16. /usr |
| 5. /dev | 11. /opt | 17. /lost+found |
| 6. /lib | 12. /media | 18. /srv |
| | | /run |

目录文件解释：

<https://www.cnblogs.com/diantong/p/10165360.html>

<https://www.cnblogs.com/dengyungao/p/8426878.html>

5.3 文件和目录操作

本节主要讲述在Linux系统下如何使用命令对文件和目录进行操作，如显示工作目录、切换目录、创建目录、删除目录、创建文件、删除文件、复制文件以及移动文件等。

5.3.1 pwd：显示工作目录路径

使用pwd命令可以显示当前用户所处的工作目录的绝对路径。

命令语法：

pwd

相对路径：相对于当前的工作目录
绝对路径：相对于根目录

【例5.1】 显示用户当前工作目录路径。

```
[root@PC-LINUX sysconfig]# pwd
```

```
/etc/sysconfig
```

//用户当前工作目录路径是/etc/sysconfig

5.3.2 cd：更改工作目录路径

使用cd命令可以更改工作目录路径。

命令语法：

cd [目录名]

说明：工作目录路径可以使用绝对或相对路径名，绝对路径从/（根）开始，然后循序到所需的目录下；相对路径从当前目录开始。

【例5.2】 更改用户工作目录路径为/etc。

```
[root@PC-LINUX sysconfig]# cd /etc
```

```
[root@PC-LINUX etc]# pwd
```

```
/etc
```

//查看当前目录，已经更改为/etc了

【例5.3】 更改目录位置至当前目录的父目录。

```
[root@PC-LINUX etc]# pwd
```

```
/etc
```

```
[root@PC-LINUX etc]# cd ..
```

```
[root@PC-LINUX /]# pwd
```

```
/
```

//可以看到目录已经更改为当前目录的父级目录“/”了

【例5.4】 更改目录位置为用户主目录。

```
[root@PC-LINUX /]# pwd
```

```
/
```

```
[root@PC-LINUX /]# cd ~
```

```
[root@PC-LINUX ~]# pwd
```

```
/root
```

//可以看到目录已经更改为当前用户的主目录

【例5.5】 更改目录位置至用户zhangsan的主目录。

```
[root@PC-LINUX ~]# cd ~zhangsan
```

```
[root@PC-LINUX zhangsan]# pwd
```

```
/home/zhangsan
```

//可以看到目录已经更改为用户zhangsan的主目录/home/zhangsan

5.3.3 ls：列出子目录和文件信息

使用ls命令，对于目录而言将列出其中的所有子目录与文件信息；对于文件而言将输出其文件名以及所要求的其他信息。

命令语法：

ls [选项] [目录或是文件]

```
ls -l
```



表5-1 ls命令显示的详细信息

列 数	描 述
第1列	第1个字符表示文件的类型
	第2~4个字符表示文件所有者对此文件的访问权限
	第5~7个字符表示用户组对此文件的访问权限
	第8~10个字符表示其他用户对此文件的访问权限
第2列	文件的链接数
第3列	文件的所有者
第4列	文件的用户组名
第5列	文件所占的字节数
第6~8列	文件上一次的修改时间
第9列	文件名

【例5.6】 显示/var目录下文件目录简单信息。

```
[root@PC-LINUX ~]# ls /var
```

```
account cvs  ftp  gopher lock named preserve tmp  
adm      db   games lib   log  nis  run   www  
cache    empty gdm  local mail opt  spool yp
```


【例5.7】 显示/root目录下所有文件目录的详细信息，包括隐藏文件。

```
[root@PC-LINUX ~]# ls -al /root
```

总用量 404

```
dr-xr-x---. 24 root root 4096 6月 3 06:00 .
dr-xr-xr-x. 18 root root 4096 6月 2 23:05 ..
drwxr-xr-x. 2 root root 4096 6月 3 05:29 .abrt
-rw-----. 1 root root 10670 6月 3 01:17 anaconda-ks.cfg
-rw-----. 1 root root 551 6月 3 05:55 .bash_history
-rw-r--r--. 1 root root 18 1月 15 05:25 .bash_logout
-rw-r--r--. 1 root root 176 1月 15 05:25 .bash_profile
-rw-r--r--. 1 root root 176 1月 15 05:25 .bashrc
drwx-----. 8 root root 4096 6月 3 05:30 .cache
drwx-----. 9 root root 4096 6月 3 05:30 .config
-rw-r--r--. 1 root root 100 1月 15 05:25 .cshrc
drwx-----. 3 root root 4096 6月 3 04:58 .dbus
-rw-----. 1 root root 16 6月 3 05:29 .esd_auth
drwx-----. 3 root root 4096 6月 3 05:34 .gconf
drwxr-xr-x. 2 root root 4096 6月 3 05:30 .gstreamer-0.10
-rw-r--r--. 1 root root 160 6月 3 05:34 .gtk-bookmarks
```

【例5.8】 显示当前目录下的文件目录信息，
用标记标出文件类型。

```
[root@PC-LINUX ~]# ls -F  
a*  abc/  anaconda-ks.cfg  
install.log.syslog
```

//其中abc是目录文件，a是可执行文件，其他的是一般文件

5.3.4 touch：创建空文件、更改文件或目录时间

使用touch命令可以创建空文件以及更改文件或目录的时间。

命令语法：

```
touch [-acfm] [-d <日期时间>] [-r <参考  
文件或目录>] [-t <日期时间>] [文件]
```

【例5.9】 创建空文件file, file1和file2。

```
[root@PC-LINUX ~]# touch file1
[root@PC-LINUX ~]# touch file2 file3
[root@PC-LINUX ~]# ls -l file1 file2 file3
-rw-r--r--. 1 root root 0 6月  3 05:45 file1
-rw-r--r--. 1 root root 0 6月  3 05:45 file2
-rw-r--r--. 1 root root 0 6月  3 05:45 file3
```

//file1, file2, file3这3个都是空文件，文件内没有任何数据

【例5.10】 将文件file1的时间记录改为6月7日19点30分，时间格式为MMDDHHmm。

```
[root@PC-LINUX ~]# ls -l file1
```

```
-rw-r--r--. 1 root root 0 6月  3 05:45 file1
```

//空文件file1其创建日期为6月3日5:45

```
[root@PC-LINUX ~]# touch -c -t 06071930 file1
```

```
[root@PC-LINUX ~]# ls -l file1
```

```
-rw-r--r--. 1 root root 0 6月  7 19:30 file1
```

//可以看到文件file1现在的时间已经改为6月7日19点30分

5.3.5 mkdir: 创建目录

使用mkdir命令可以在Linux系统中创建目录。

命令语法:

mkdir [选项] [目录名]

mkdir -p:可以创建多层目录

【例5.11】 创建目录newdir1，其默认权限为755。

```
[root@PC-LINUX ~]# mkdir newdir1
```

```
[root@PC-LINUX ~]# ls -l
```

总用量 272

```
-rw-----. 1 root root 10670 6月 3 01:17
```

```
anaconda-ks.cfg
```

```
-rw-r--r--. 1 root root 155641 6月 3 01:16 install.log
```

```
-rw-r--r--. 1 root root 65450 6月 3 01:15
```

```
install.log.syslog
```

```
drwxr-xr-x. 2 root root 4096 6月 3 05:46 newdir1
```

//目录newdir1的权限为rwxr-xr-x（755）

【例5.12】 创建目录newdir2，其权限为777。

```
root@PC-LINUX ~]# mkdir -m 777 newdir2
```

```
[root@PC-LINUX ~]# ls -l
```

总用量 276

```
-rw-----. 1 root root 10670 6月 3 01:17 anaconda-ks.cfg
```

```
-rw-r--r--. 1 root root 155641 6月 3 01:16 install.log
```

```
-rw-r--r--. 1 root root 65450 6月 3 01:15 install.log.syslog
```

```
drwxr-xr-x. 2 root root 4096 6月 3 05:46 newdir1
```

```
drwxrwxrwx. 2 root root 4096 6月 3 05:47 newdir2
```

.....

//目录newdir2的权限为rwxrwxrwx（777）

5.3.6 rmdir：删除空目录

rmdir命令功能：删除空目录。

命令语法：

rmdir [选项] [目录名]

【例5.13】 删除空目录newdir1。

```
[root@PC-LINUX ~]# rmdir newdir1
```

【例5.14】 同时删除/root/dir1，
/root/dir1/dir2这两个空目录。

```
[root@PC-LINUX ~]# mkdir /root/dir1
```

```
[root@PC-LINUX ~]# mkdir /root/dir1/dir2
```

//创建目录/root/dir1和/root/dir1/dir2

```
[root@PC-LINUX ~]# rmdir -p /root/dir1/dir2
```

rmdir: 删除目录 "/root" 失败: 目录非空

//这个命令将删除空目录/root/dir1， /root/dir1/dir2，
但由于/root目录下不是空的，所以/root目录无法删除，
而其他两个目录已经删除

5.3.7 cp：复制文件和目录

使用cp命令可以复制文件和目录到其他目录中。

命令语法：

cp [选项] [源文件或目录] [目标文件或目录]

拷贝目录要加-r

【例5.15】 将文件/etc/named.conf复制到/root目录下，并改名为named.c。

```
[root@PC-LINUX ~]# cp /etc/named.conf /root/named.c
```

【例5.16】 将文件/etc/named.conf复制到/root目录下。

```
[root@PC-LINUX ~]# cp /etc/named.conf /root
```

【例5.17】 将/boot目录中的所有文件及其子目录复制到目录/root中。

```
[root@PC-LINUX ~]# cp -r /boot /root
```

5.3.8 mv：移动或更名现有文件或目录

使用mv命令可以对文件和目录更改名称以及移动文件和目录的路径。

命令语法：

mv [选项] [源文件或目录] [目标文件或目录]

【例5.18】 将/root/picture目录下所有的后缀名为“.png”的文件移到/usr/local/share/picture目录下。

```
[root@PC-LINUX ~]# mv -f /root/picture/*.png  
/usr/local/share/picture
```

【例5.19】 把/root/picture目录下的文件kdepic.png改名为life.png。

```
[root@PC-LINUX ~]# mv /root/picture/kdepic.png  
/root/picture/life.png
```

【例5.20】 把目录/root/pitcure名称更改为/root/mypicture。

```
[root@PC-LINUX ~]# mv /root/picture /root/mypicture
```

5.3.9 rm：删除文件或目录

使用rm命令可以删除系统中的文件或目录。

命令语法：

rm [选项] [文件或目录]

【例5.21】 删除当前目录下的file4文件。

```
[root@PC-LINUX ~]#rm -i file4
```

rm: 是否删除普通文件 “file4” ? y //输入y确认删除该文件

【例5.22】 连同文件/root/ab/a和目录/root/ab一起删除。

```
[root@PC-LINUX ~]# mkdir /root/ab
```

```
[root@PC-LINUX ~]# touch /root/ab/a
```

//创建目录/root/ab和文件/root/ab/a

```
[root@PC-LINUX ~]# rm -rf /root/ab
```

//连同文件/root/ab/a和目录/root/ab一起删除

-f : 交互时不需要用户确认

5.3.10 wc：统计文件行数、单词数和字节数

使用wc命令可以统计指定文件中的行数、单词数或字节数，并将统计结果显示输出到屏幕。

命令语法：

wc [选项] [文件]

【例5.23】 统计文件file1，file2的行数、单词数和字符数。

```
[root@PC-LINUX ~]#wc -lwc file1 file2
```


5.4 链接文件

在Linux系统中，内核为每一个新创建的文件分配一个inode（索引节点）号，文件属性保存在索引节点里，在访问文件时，索引节点被复制到内存里，从而实现文件的快速访问。

5.4.1 链接文件简介

链接是一种在共享文件和访问它的用户的若干目录项之间建立联系的方法。Linux系统中包括两种链接：硬链接（Hard Link）和软链接（Soft Link），软链接又称为符号链接。

`ls -li`:展示文件和文件编号（唯一）

1. 硬链接

硬链接是一个指针，指向文件索引节点，系统并不为它重新分配inode，可以用ln命令来建立硬链接。硬链接节省空间，也是Linux系统整合文件系统的传统方式。

硬链接文件有两个限制：

(1) 不允许给目录创建硬链接；

(2) 只有在同一文件系统中的文件之间才能创建链接。

2. 软链接

软链接又称为符号链接，这个文件包含了另一个文件的路径名。

3. 硬链接和软链接的区别

硬链接记录的是目标的inode，软链接记录的是目标的路径。软链接就像是快捷方式，而硬链接就像是备份。软链接可以做跨分区的链接，而硬链接由于inode的缘故，只能在本分区中做链接。所以，软链接的使用频率要高得多。

5.4.2 链接文件的使用

本节主要以案例的形式讲述在Linux系统下如何创建、使用硬链接文件和软链接文件。使用ln命令可以创建链接文件（包括软链接文件和硬链接文件）。

命令语法：

ln [选项] [源文件名] [链接文件名]

【例5.24】 硬链接文件的使用。

```
[root@PC-LINUX ~]#echo hello > a
```

```
[root@PC-LINUX ~]#ln a b
```

【例5.25】 软链接文件的使用。

```
[root@PC-LINUX ~]#echo hello > a
```

```
[root@PC-LINUX ~]#ln -s a b
```


小 结

在Linux系统下除了一般文件之外，所有的目录和设备（光驱、硬盘等）都是以文件的形式存在。Linux文件类型和Linux文件的文件名所代表的意义是两个不同的概念。Linux文件类型常见的有：普通文件、目录文件、字符设备文件、块设备文件和符号链接文件等。

小 结

Linux系统都有根文件系统，它包含系统引导和使其他文件系统得以挂载所必要的文件，Linux系统的目录结构是分层的树形结构，都是挂载在根文件系统“/”下。

在Linux系统下对文件和目录进行操作的命令主要有pwd、cd、ls、touch、mkdir、rmdir、cp、mv、rm以及wc等。

小 结

在Linux系统中，内核为每一个新创建的文件分配一个inode（索引节点）号，文件属性保存在索引节点里，在访问文件时，索引节点被复制到内存里，从而实现文件的快速访问。

小 结

在Linux系统中包括硬链接和软链接两种链接。硬链接是一个指针，指向文件索引节点，系统并不为它重新分配inode。硬链接节省空间，也是Linux系统整合文件系统的传统方式。软链接又叫符号链接，这个文件包含了另一个文件的路径名。可以是任意文件或目录，可以链接不同文件系统的文件。