

文件概述

内容提要

1. 理解 Linux 系统中文件的概念
2. 掌握 Linux 系统下文件的分类
3. 熟悉 Linux 系统中的设备文件
4. 理解 Linux 系统中的链接文件

什么是文件

在 Linux 系统上，文件被看作是字节序列。这种概念使得所有的系统资源有了统一的标识，这些资源包括：普通文件或目录、磁盘设备、控制台（键盘、显示器）、打印机等。对这些资源的访问和处理都是通过字节序列的方式实现的。

文件的类型

Linux 系统下的文件类型包括：

- 普通文件 (-)
- 目录 (d)
- 符号链接 (l)
- 字符设备文件 (c)
- 块设备文件 (b)
- 套接字 (s)
- 命名管道 (p)

普通文件

普通文件仅仅就是字节序列，Linux 并没有对其内容规定任何的结构。普通文件可以是程序源代码（c、c++、python、perl等）、可执行文件（文件编辑器、数据库系统、出版工具、绘图工具等）、图片、声音、图像等。Linux 不会区别对待这些文件，只有处理这些文件的应用程序才会对根据文件的内容赋予相应的含义。

在DOS或Winodws环境中，所有的文件名的后缀就能表示该文件的类型，如：***.exe**表示可执行文件，***.bat**表示批处理文件。在Linux环境下，只要是可执行的文件并具有可执行属性它就能执行，不管其文件名后缀是什么。但是对一些数据文件一般也遵循一些文件名后缀规则。

- 系统文件
 - *.conf: 配置文件
 - *.rpm: rpm包
 - *.deb: deb包
 - *.a: 一种存档文件
 - *.lock: 一种锁定文件
 - *~: 备份文件
 - .*: 隐含文件
- 程序与脚本

- *.c: C语言源程序文件
- *.cpp: C++语言源程序文件
- *.h: C或C++的头文件
- *.o: 程序对象文件
- *.pl: perl语言源程序文件
- *.php: php语言源程序文件
- *.python: python语言源程序文件
- *.tcl: TCL脚本文件
- *.so *.lib: 库文件
- *.sql: SQL语言文件
- 其他格式文件
 - *.txt: 无格式的ASCII码文件
 - *.html *.htm: 静态Web页
 - *.ps: PostScript文件
 - *.mp3: mp3 文件
 - *.au: 一种声音文件
 - *.wav: 一种声音文件
 - *.xpm: 一种图像文件
 - *.jpg: 一种图形、图像文件
 - *.gif: 一种图形、图像文件
 - *.png: 一种图形、图像文件

目录

目录文件是由一组目录项组成，目录项可以是对其他文件的指向也可以是其下的子目录指向。

实际上，一个文件的名称是存储在他的父目录中的，而并非同文件内容本身存储在一起。

硬连接文件实际上就是在某目录中创建目录项，从而使不止一个目录可以引用到同一个文件。

硬链接将两个文件名（存储在其父目录的目录项中）指向的是硬盘上一个存储空间，对两个文件中的任何一个的内容进行修改都会影响到另一个文件，这种链接关系称为硬链接，它可以由 **ln** 命令建立。首先查看一下目录中的文件情况：

```
$ ls -l
-rwxr-xr-x    1 Mike  users      58 2006-07-01 10:05 file1
$ cat file1
This is file1.
```

使用 **ln** 命令建立文件 **file1** 的硬链接文件 **file2**：

```
$ ln file1 file2
```

该命令产生一个新的文件**file2**，它和已经存在的文件**file1**建立起硬链接关系：

```
$ cat file2
This is file1.
$ ls -l
-rwxr-xr-x    2 Mike  users      58 2006-07-01 10:05 file1
-rwxr-xr-x    2 Mike  users      58 2006-07-01 10:07 file2
```

可以看出，**file2**和**file1**的大小相同，内容相同。再看详细信息的第2列，原来**file1**的链接数是1，说明这一块硬盘存储空间只有**file1**一个文件指向它，而建立起**file1**和**file2**的硬链接关系之后，这块硬盘空间就有**file1**和**file2**两个文件同时指向它，所以**file1**和**file2**的链接数就都变为了2。因为两个文件指向一块硬盘空间，所以如果现在修改**file2**的内容为“This is file2.”，再查看**file1**的内容，就会有：

```
$ cat file1
This is file2.
```

如果删除其中的一个文件（不管是哪一个），就是删除了该文件和硬盘空间的指向关系，该硬盘空间不会释放，另外一个文件的内容也不会发生改变，但是目录详细信息中的链接数会减少：

```
$ rm -f file1
$ ls -l
-rwxr-xr-x    1 Mike  users      58 2006-07-01 10:07 file2
$ cat file2
This is file2.
```

硬链接并不是一种特殊类型的文件，只是因为文件系统中允许不止一个目录项指向同一个文件。

符号链接

符号链接又称软链接，是指将一个文件指向另外一个文件的文件名。这种符号链接的关系由 **ln -s** 命令行来建立。首先查看一下目录中的文件信息：

```
$ ls -l
-rwxr-xr-x    1 Mike  users      58 2006-07-01 10:05 file1
$ cat file1
This is file1.
```

使用**ln**命令和**-s**选项建立文件**file1**的符号链接文件**file2**：

```
$ ln -s file1 file2
```

该命令产生一个新的文件**file2**，它和已经存在的文件**file1**建立起符号链接关系：

```
$ cat file2
This is file1.
$ ls -l
-rwxr-xr-x    1 Mike  users      58 2006-07-01 10:15 file1
lrwxrwxrwx    1 Mike  users       5 2006-07-01 10:17 file2 -> file1
```

可以看出**file2**这个文件很小，因为它只是记录了指向的文件名而已，请注意那个从文件**file2**指向文件**file1**的指针。

为什么**cat**命令显示的**file2**的内容与**file1**相同呢？因为**cat**命令在寻找**file2**的内容时，发现**file2**是一个符号链接文件，就根据**file2**记录的文件名找到了**file1**文件，然后将**file1**的内容显示出来。

明白了**file1**和**file2**的符号链接关系，就可以理解为什么**file1**的链接数仍然为1，这是因为**file1**指向的硬盘空间仍然只有**file1**一个文件在指向。

如果现在删除了**file2**，对**file1**并不产生任何影响；而如果删除了**file1**，那么**file2**就因无法找到文件名称为**file1**的文

件而成为死链接。

```
$ rm -f file1
$ ls -l
lrwxrwxrwx      1 Mike  users      5 2006-07-01 10:17 file2 -> file1
$ cat file2
cat: file2: No such file or directory
```

设备文件

设备是指计算机中的外围硬件装置，即除了CPU和内存以外的所有设备。通常，设备中含有数据寄存器或数据缓存器、设备控制器，它们用于完成设备同CPU或内存的数据交换。

在 Linux 下，为了屏蔽用户对设备访问的复杂性，采用了设备文件，即可以通过象访问普通文件一样的方式来对设备进行访问读写。

设备文件用来访问硬件设备，包括硬盘、光驱、打印机等。每个硬件设备至少与一个设备文件相关联。设备文件分为：字符设备（如：键盘）和块设备（如：磁盘）。Linux下设备名以文件系统中的设备文件的形式存在。所有的设备文件存放在/dev目录下。

下面对常用设备列表说明：

设备文件	说明
/dev/hd*	IDE硬盘设备，如hda1表示第一块IDE硬盘的第一个分区；hdb2表示第二块IDE硬盘的第二个分区
/dev/sd*	SCSI硬盘设备，如sda1表示第一块SCSI硬盘的第一个分区；sdb2表示第二块SCSI硬盘的第二个分区
/dev/lp*	表示并口设备，如lp0表示第一个并口设备；lp1表示第二个并口设备
/dev/tty*	终端设备
/dev/console	系统控制台
/dev/scd*	SCSI光驱设备
/dev/ppp*	ppp设备
/dev/isdn*	isdn设备
/dev/null	空设备
/dev/zero	零设备

在 /dev 目录下有许多链接文件，使用这些链接能够方便地使用系统中的设备。例如，可以通过 /dev/cdrom 而不是 /dev/hdc 来访问光驱。

套接字和命名管道

套接字和命名管道是 Linux 环境下实现进程间通信（IPC）的机制。

命名管道（FIFO）文件允许运行在同一台计算机上的两个进程之间进行通信。套接字（socket）允许运行在不同计算机上的进程之间相互通信。

套接字和命名管道通常是在进程运行时创建或删除的，一般无需系统管理员干预。

- 显示源文件
- 登录