

本章学习目标：

- 了解文件系统的含义。
- 掌握 Ubuntu 文件系统的结构。
- 掌握 Ubuntu 文件系统的管理方法。
- 掌握文件管理的命令。

文件和目录管理是 Linux 系统运行维护的基础工作,在 Linux 系统下用户的数据和程序都是以文件的形式保存的,所以在使用 Linux 的过程中,经常要对文件和目录进行操作。

## 5.1 文件系统概述



文件系统是操作系统最重要的组成部分之一,操作系统之所以能够找到磁盘上的文件,是因为有磁盘上的文件名与存储位置的记录。文件系统是解决如何在存储设备上存储数据的一套方法,包括存储布局、文件命名、空间管理、安全控制等,用于对磁盘进行存储管理及输入输出。Linux 操作系统支持很多现代的流行文件系统,其中,Ext2、Ext3 和 Ext4 最普遍。Ext2 文件系统是伴随着 Linux 一起发展起来的,在 Ext2 的基础上增加日志就是 Ext3,Ext4 是第 4 代扩展文件系统,是 Linux 系统下的日志文件系统,是 Ext3 文件系统的后继版本。

2008 年 12 月 25 日,Linux Kernel 2.6.28 的正式发布。随着这一新内核的发布,Ext4 文件系统也结束实验期,成为稳定版。Ext4 在功能上与 Ext3 非常相似,但支持大文件系统,提高了对碎片的抵抗力,有更高的性能以及更好的时间戳。

目前的大部分 Linux 文件系统都默认采用 Ext4 文件系统。

### 5.1.1 文件系统

#### 1. Ext2

第 2 代扩展文件系统(second extended filesystem,Ext2)是 Linux 内核所用的文件系统。它由 Rémy Card 设计,用以代替 Ext,于 1993 年 1 月加入 Linux 核心支持之中。Ext2 的经典实现为 Linux 内核中的 Ext2fs 文件系统驱动,最大可支持 2TB 的文件系统,到 Linux 核心 2.6 版时,扩展到可支持 32TB。Ext2 为 Debian、Red Hat Linux 等 Linux 发行版的默认文件系统。

Ext2 文件系统具有以下一些特点。

(1) 当创建 Ext2 文件系统时,系统管理员可以根据预期的文件平均长度来选择最佳的块大小(1024~4096B)。例如,当文件的平均长度小于几千字节时,块的大小为 1024B 是最佳的,因为这会产生较少的内部碎片——也就是文件长度与存放块的磁盘分区有较少的不匹配。另外,大的块对于大于几千字节的文件通常比较合适,因为这样的磁盘传送较少,因而减轻了系统的开销。

(2) 当创建 Ext2 文件系统时,系统管理员可以根据在给定大小的分区上预计存放的文件数来选择给该分区分配多少个索引节点。这可以有效地利用磁盘的空间。

(3) 文件系统把磁盘块分为组。每组包含存放在相邻磁道上的数据块和索引节点。正是这种结构,使得可以用较少的磁盘平均寻道时间对存放在一个单独块组中的文件并行访问。

(4) 在磁盘数据块被实际使用之前,文件系统就把这些块预分配给普通文件。因此当文件的大小增加时,因为物理上相邻的几个块已被保留,就减少了文件的碎片。

(5) 支持快速符号链接。如果符号链接表示一个短路径名(小于或等于 60 个字符),就把它存放在索引节点中而不用通过由一个数据块进行转换。

其单一文件大小与文件系统本身的容量上限与文件系统本身的簇大小有关,在一般常见的 x86 计算机系统中,簇最大为 4KB,则单一文件大小上限为 2048GB,而文件系统的容量上限为 16 384GB。

但由于目前 Linux 2.4 所能使用的单一分区最大只有 2048GB,实际上能使用的文件系统容量最多也只有 2048GB。

## 2. Ext3

第 3 代扩展文件系统(third extended filesystem, Ext3)是一个日志文件系统,常用于 Linux 操作系统。它是很多 Linux 发行版的默认文件系统。Stephen Tweedie 在 1999 年 2 月的内核邮件列表中,最早显示了他使用扩展的 Ext2,该文件系统从 2.4.15 版本的内核开始,合并到内核主线中。

Ext3 日志文件系统的特点如下。

### (1) 高可用性。

系统使用了 Ext3 文件系统后,即使在非正常关机后,系统也不需要检查文件系统。宕机发生后,恢复 Ext3 文件系统只要数十秒钟。

如果在文件系统尚未 shutdown 前就关机(如停电)时,下次重新开机后会造成文件的资料不一致,因此,需做文件系统的重整工作,将不一致与错误的地方修复。然而,此项重整工作是相当耗时的,特别是容量大的文件系统,而且也不能百分之百保证所有的资料都不会损失。

为了解决此问题,使用所谓的“日志式文件系统(Journal File System)”。此类文件系统最大的特色是会将整个磁盘的写入动作完整记录在磁盘的某个区域上,以便有需要时可以回溯追踪。

在日志式文件系统中,由于详细纪录了每个细节,故当在某个过程中被中断时,系统可以根据这些记录直接回溯并重整被中断的部分,而不必花时间去检查其他的部分,故重整的工作速度相当快,几乎不需要花时间。

### (2) 数据的完整性。

Ext3 文件系统能够极大地提高文件系统的完整性,避免了意外宕机对文件系统的破

坏。在保证数据完整性方面,Ext3 文件系统有两种模式可供选择。其中之一就是“同时保持文件系统及数据的一致性”模式。采用这种方式,用户永远不再会看到由于非正常关机而存储在磁盘上的垃圾文件。

### (3) 文件系统的速度。

尽管使用 Ext3 文件系统时,有时在存储数据时可能要多次写数据,但是从总体上来看,Ext3 比 Ext2 的性能还要好一些。这是因为 Ext3 的日志功能对磁盘的驱动器读写头进行了优化。所以,文件系统的读写性能较之 Ext2 文件系统来说并没有降低。

### (4) 数据转换。

由 Ext2 文件系统转换成 Ext3 文件系统非常容易,只要简单地输入两条命令即可完成整个转换过程,用户不用花时间备份、恢复、格式化分区等。用一个 Ext3 文件系统提供的小工具 tune2fs,可以将 Ext2 文件系统轻松转换为 Ext3 日志文件系统。另外,Ext3 文件系统可以不经任何更改,而直接加载成为 Ext2 文件系统。

### (5) 多种日志模式。

Ext3 有多种日志模式,一种工作模式是对所有的文件数据及 metadata(定义文件系统中数据的数据,即元数据)进行日志记录(data=journal 模式);另一种工作模式则是只对 metadata 记录日志,而不对数据进行日志记录,也即所谓的 data=ordered 或者 data=writeback 模式。系统管理人员可以根据系统的实际工作要求,在系统的工作速度与文件数据的一致性之间做出选择。

## 3. Ext4

第 4 代扩展文件系统(fourth extended filesystem,Ext4)是 Linux 系统下的日志文件系统,是 Ext3 文件系统的后继版本。

Ext4 是由 Ext3 的维护者 Theodore Tso 领导的开发团队实现的,并引入 Linux 2.6.19 内核中。

Ext4 的产生原因是开发人员在 Ext3 中加入了新的高级功能,但在实现的过程出现了以下几个重要问题。

- (1) 一些新功能违背向后兼容性。
- (2) 新功能使 Ext3 代码变得更加复杂并难以维护。
- (3) 新加入的更改使原来十分可靠的 Ext3 变得不可靠。

由于这些原因,从 2006 年 6 月开始,开发人员决定把 Ext4 从 Ext3 中分离出来进行独立开发。Ext4 的开发工作从那时起开始进行,但大部分 Linux 用户和管理员都没有太关注这件事情,直到 2.6.19 内核在 2006 年 11 月发布,Ext4 第一次出现在主流内核里,但是它当时还处于实验阶段,因此很多人都忽视了它。

2008 年 12 月 25 日,Linux Kernel 2.6.28 的正式发布。随着这一新内核的发布,Ext4 文件系统也结束实验期,成为稳定版。

Linux Kernel 自 2.6.28 开始正式支持新的文件系统 Ext4。Ext4 是 Ext3 的改进版,修改了 Ext3 中部分重要的数据结构,而不仅像 Ext3 对 Ext2 那样,只是增加了一个日志功能而已。Ext4 可以提供更佳的性能和可靠性,还有更为丰富的功能。

Ext4 文件系统具有以下特点。

#### (1) 更大的文件系统和更大的文件。

Ext3 文件系统最多只能支持 32TB 的文件系统和 2TB 的文件,根据使用的具体架构和系统设置,实际容量上限可能比这个数字还要低,即只能容纳 2TB 的文件系统和 16GB 的文件。而 Ext4 文件系统容量达到 1EB,文件容量则达到 16TB,这是一个非常大的数字。对一般的台式计算机和服务器而言,这可能并不重要,但对于大型磁盘阵列的用户而言,这就非常重要了。

#### (2) 更多的子目录数量。

Ext3 目前只支持 32 000 个子目录,而 Ext4 取消了这一限制,理论上支持无限数量的子目录。

#### (3) 更多的块和 i-节点数量。

Ext3 文件系统使用 32 位空间记录块数量和 i-节点数量,而 Ext4 文件系统将它们扩充到 64 位。

#### (4) 多块分配。

当数据写入 Ext3 文件系统中时,Ext3 的数据块分配器每次只能分配一个 4KB 的块,如果写一个 100MB 的文件就要调用 25 600 次数据块分配器,而 Ext4 的多块分配器 Multiblock Allocator(MBAlloc)支持一次调用分配多个数据块。

#### (5) 持久性预分配。

如果一个应用程序需要在实际使用磁盘空间之前对它进行分配,大部分文件系统都是通过向未使用的磁盘空间写入 0 来实现分配,比如 P2P 软件。为了保证下载文件有足够的空间存放,常常会预先创建一个与所下载文件大小相同的空文件,以免未来的数小时或数天之内磁盘空间不足导致下载失败。而 Ext4 在文件系统层面实现了持久预分配并提供相应的 API,比应用软件自己实现更有效率。

#### (6) 延迟分配。

Ext3 的数据块分配策略是尽快分配,而 Ext4 的策略是尽可能地延迟分配,直到文件在缓冲中写完才开始分配数据块并写入磁盘,这样就能优化整个文件的数据块分配,显著提升性能。

#### (7) 盘区结构。

Ext3 文件系统采用间接映射地址,当操作大文件时,效率极其低下。例如,一个 100MB 大小的文件,在 Ext3 中要建立 25 600 个数据块(以每个数据块大小为 4KB 为例)的映射表;而 Ext4 引入了盘区的概念,每个盘区为一组连续的数据块,上述文件可以通过盘区的方式表示为“该文件数据保存在接下来的 25 600 个数据块中”,提高了访问效率。

#### (8) 新的 i-节点结构。

Ext4 支持更大的 i-节点。之前的 Ext3 默认的 i-节点大小为 128B,Ext4 为了在 i-节点中容纳更多的扩展属性,默认 i-节点大小为 256B。另外,Ext4 还支持快速扩展属性和 i-节点保留。

#### (9) 日志校验功能。

日志是文件系统最常用的结构,日志也很容易损坏,而从损坏的日志中恢复数据会导致更多的数据损坏。Ext4 给日志数据添加了校验功能,日志校验功能可以很方便地判断日志数据是否损坏。而且 Ext4 将 Ext3 的两阶段日志机制合并成一个阶段,在增加安全性的同

时提高了性能。

(10) 支持“无日志”模式。

日志总归会占用一些开销。Ext4 允许关闭日志,以便某些有特殊需求的用户可以借此提升性能。

(11) 默认启用 Barrier。

磁盘上配有内部缓存,以便重新调整批量数据的写操作顺序,优化写入性能,因此文件系统必须在日志数据写入磁盘之后才能写 Commit 记录。若 Commit 记录写入在先,而日志有可能损坏,那么就会影响数据完整性。Ext4 文件系统默认启用 Barrier,只有当 Barrier 之前的数据全部写入磁盘,才能写 Barrier 之后的数据。

(12) 在线碎片整理。

尽管延迟分配、多块分配和盘区功能可以有效减少文件的碎片,但碎片还是不可避免会产生。Ext4 支持在线碎片整理,并提供 e4defrag 工具进行个别文件或整个文件系统的碎片整理。

(13) 支持快速 fsck。

以前的文件系统版本执行 fsck 时很慢,因为它要检查所有的 i-节点,而 Ext4 给每个块组的 i-节点表中都添加了一份未使用 i-节点的列表,所以 Ext4 文件系统做一致性检查时就可以跳过它们而只去检查那些在使用的 i-节点,从而提高了速度。

(14) 支持纳秒级时间戳。

Ext4 之前的扩展文件系统的时间戳都是以秒为单位的,这已经能够应付大多数设置,但随着处理器的速度和集成程度(多核处理器)不断提升,以及 Linux 开始向其他应用领域发展,它将时间戳的单位提升到纳秒。

Ext4 给时间范围增加了两个位,从而让时间寿命再延长 500 年。Ext4 的时间戳支持的日期到 2514 年 4 月 25 日,而 Ext3 只到 2038 年 1 月 18 日。

## 5.1.2 文件系统概念

在 Linux 系统中有一个重要的概念:一切都是文件,实现了设备无关性。其实这是 UNIX 哲学的一种体现,而 Linux 是重写 UNIX 而来,所以这个概念也就传承了下来。在 UNIX 系统中,把一切资源都看作文件,包括硬件设备。UNIX 系统把每个硬件都看成是一个文件,通常称为设备文件,这样用户就可以用读写文件的方式实现对硬件的访问,UNIX 权限模型也是围绕文件的概念来建立的,所以对设备也就可以同样处理了。

下面来详细地了解 Linux 文件系统的几个要点。

### 1. 物理磁盘到文件系统

我们知道文件最终是保存在硬盘上的。硬盘最基本的组成部分是由坚硬金属材料制成的涂以磁性介质的盘片,不同容量硬盘的盘片数不等。每个盘片有两面,都可记录信息。盘片被分成许多扇形的区域,每个区域叫一个扇区,每个扇区可存储  $128 \times 2^N$  ( $N=0,1,2,3$ ) 字节信息。在 DOS 中每扇区是  $128 \times 2^2 = 512$  字节,盘片表面上以盘片中心为圆心,不同半径的同心圆称为磁道。硬盘中,不同盘片相同半径的磁道所组成的圆柱称为柱面。磁道与柱面都表示不同半径的圆,在许多场合,磁道和柱面可以互换使用。每个磁盘有两个面,每个面都有一个磁头,人们习惯用磁头号来区分。扇区,磁道(或柱面)和磁头数构成了硬盘结

构的基本参数,通过这些参数可以得到硬盘的容量,其计算公式为:

$$\text{存储容量} = \text{磁头数} \times \text{磁道(柱面)数} \times \text{每道扇区数} \times \text{每扇区字节数}$$

要点:

- (1) 硬盘有数个盘片,每个盘片两个面,每个面一个磁头。
- (2) 盘片被划分为多个扇形区域即扇区。
- (3) 同一盘片不同半径的同心圆为磁道。
- (4) 不同盘片相同半径构成的圆柱面即柱面。
- (5) 公式: 存储容量 = 磁头数 × 磁道(柱面)数 × 每道扇区数 × 每扇区字节数。
- (6) 信息记录可表示为: ××磁道(柱面),××磁头,××扇区。

那么这些空间又是怎么管理起来的呢? UNIX/Linux 使用了一个简单的方法,如图 5.1 所示。

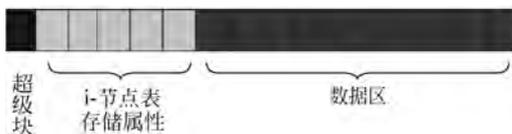


图 5.1 文件系统存储空间示意图

它将磁盘块分为以下三个部分。

(1) 超级块。文件系统中第一个块被称为超级块。这个块存放文件系统本身的结构信息。例如,超级块记录了每个区域的大小,超级块也存放未被使用的磁盘块的信息。

(2) i-点表。超级块的下一个部分就是 i-节点表。每个 i-节点就是一个对应文件/目录的结构,这个结构包含一个文件的长度、创建及修改时间、权限、所属关系、磁盘中的位置等信息。一个文件系统维护了一个索引节点的数组,每个文件或目录都与索引节点数组中的一个元素一一对应。系统给每个索引节点分配了一个号码,也就是该节点在数组中的索引号,称为索引节点号。

(3) 数据区。文件系统的第 3 个部分是数据区。文件的内容保存在这个区域。磁盘上所有块的大小都一样。如果文件包含超过一个块的内容,则文件内容会存放在多个磁盘块中。一个较大的文件很容易分布在上千个独立的磁盘块中。

## 2. 存储介质

用以存储数据的物理设备称为存储介质,如硬盘、光盘、Flash 盘、磁带、网络存储设备等。

## 3. 磁盘分区

对于容量较大的存储介质来说(通常指硬盘),在使用时需要合理地规划分区,因而牵涉到磁盘的分区。常用的 Linux 磁盘分区命令有 fdisk、cfdisk、parted 等。还有一些工具不是操作系统自带的,称为第三方工具,如 PQ 等。利用磁盘分区工具,可以将硬盘分割为大小不一的多个部分,以便规划和满足实际使用的需要。

## 4. 格式化

创建新的文件系统是一个过程,通常称为初始化或格式化,这个过程是针对存储介质进行的。一般情况下,各种操作系统都有自己的相应工具,Ubuntu 下格式化分区的命令是 mkfs,有时也可以借助第三方工具来完成此过程。而此过程是建立在磁盘分区的基础之

上,也就是说先进行磁盘分区,再进行文件系统的创建或格式化。

## 5. 挂载

在使用磁盘分区前,需要挂载该分区,这相当于激活一个文件系统。

Windows 将磁盘分为若干个逻辑分区,如 C 盘、D 盘,在各个分区中挂载文件系统。这个过程是使用其内部机制完成的,用户无法探知其过程。

Linux 系统中,没有磁盘的逻辑分区(即没有 C 盘、D 盘等),任何一个种类的文件系统被创建后都需要挂载到某个特定的目录才能使用。Linux 使用 mount 和 umount 命令来对文件系统进行挂载和卸载,挂载文件系统时需要明确挂载点。如图 5.2 所示,把 U 盘/dev/sdb1(系统识别)挂载到 /mnt/usb 下;图 5.3 中则把 U 盘卸载,注意,不能在当前目录卸载,应到上一级目录或者根目录卸载。

```
malimei@malimei-virtual-machine:/$ sudo mount /dev/sdb1 /mnt/usb
mount: /dev/sdb1 is write-protected, mounting read-only
malimei@malimei-virtual-machine:/$ cd /mnt/usb
malimei@malimei-virtual-machine:/mnt/usb$ ls
02-竞赛模型图.jpg
02-竞赛模型图.vsd
03155413g3kk.rar
03.pdf
```

图 5.2 挂载 U 盘

```
malimei@malimei-virtual-machine:/mnt/usb$ cd /
malimei@malimei-virtual-machine:/$ sudo umount /dev/sdb1
malimei@malimei-virtual-machine:/$
```

图 5.3 卸载 U 盘

### 5.1.3 文件与目录的定义

Linux 操作系统中,以文件来表示所有的逻辑实体与非逻辑实体。逻辑实体指文件与目录;非逻辑实体泛指硬盘、终端机、打印机等。一般而言,Linux 文件名由字母、标点符号、数字等构成,中间不能有空格、路径名称符号“/”或“#、\*、%、&、{ }、[]”等与 Shell 有关的特殊字符。

Linux 文件系统中,结构上以根文件系统最为重要。根文件系统是指开机时将 root partition 挂载在根目录(/),若无法挂载根目录,开机时就无法进入 Linux 系统中。根目录下有 /etc、/dev、/boot、/home、/lib、/lost+found、/mnt、/opt、/proc、/root、/bin、/sbin、/tmp、/var、/usr 等重要目录。

下面分别使用图形界面和命令终端查看各个目录,如图 5.4 和图 5.5 所示。

#### 1. /etc

本目录下存放着许多系统所需的重要配置与管理文件,如/etc/hostname 存放配置主机名字的文件,/etc/network/interfaces 存放配置修改网络接口的 IP 地址、子网掩码、网关的文件,/etc/resolv.conf 存放指定 DNS 服务器的文件等。图 5.6 显示了配置文件 hostname 和 resolv.conf 的内容。通常在修改/etc 目录下的配置文件内容后,只需重新启动相关服务,一般不用重启系统。



图 5.4 图形界面下查看文件目录

```
malimei@malimei:/home/a$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
malimei@malimei:/home/a$ cd ..
malimei@malimei:/home$ cd ..
malimei@malimei:/S$ ls
bin  cdrom  etc  initrd.img  lost+found  mnt  proc  run  srv  tmp  var
boot  dev  home  lib  media  opt  root  sbin  sys  usr  vmlinuz
malimei@malimei:/S$
```

图 5.5 Shell 终端下查看文件目录

```
malimei@malimei-virtual-machine:~$ cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
malimei@malimei-virtual-machine:~$ ifconfig eth0
eth0      Link encap:以太网 硬件地址 00:0c:29:22:51:f2
          inet 地址:192.168.3.4 广播:192.168.3.255 掩码:255.255.255.0
          inet6 地址: fe80::20c:29ff:fe22:51f2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 跃点数:1
          接收数据包:150 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:84 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:62482 (62.4 KB) 发送字节:12119 (12.1 KB)
          中断:19 基本地址:0x2000
malimei@malimei-virtual-machine:~$ hostname
malimei-virtual-machine
malimei@malimei-virtual-machine:~$ cat /etc/hostname
malimei-virtual-machine
malimei@malimei-virtual-machine:~$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 127.0.1.1
malimei@malimei-virtual-machine:~$
```

图 5.6 查看配置文件

## 2. /dev

/dev 目录中存放了 device file(装置文件),使用者可以经由核心存取系统中的硬设备,当使用装置文件时内核会辨识出输入输出请求,并传递到相应装置的驱动程序以便完成特定的动作。

该目录包含所有在 Linux 系统中使用的外部设备,每个设备在/dev 目录下均有一个相应的项目,如图 5.7 所示。注意 Linux 与 Windows/DOS 不同,不是存放外部设备的驱动程序,而是一个访问这些外部设备的端口。如/dev/cdrom 下存放光驱中的文件,/dev/u 下存放 U 盘中的文件,/dev/sda1 下一般存放的是第一块硬盘第一分区中的文件。

```
malimei@malimei:/dev$ ls
agpgart      loop-control    rtc0           tty25         tty57         ttyS3
autofs       lp0            sda           tty26         tty58         ttyS30
block        mapper         sda1          tty27         tty59         ttyS31
bsg          mcelog         sda2          tty28         tty6          ttyS4
btrfs-control mem            sda5          tty29         tty60         ttyS5
bus          midi           sg0           tty3          tty61         ttyS6
cdrom        net            sg1           tty30         tty62         ttyS7
char         network_latency sg2           tty31         tty63         ttyS8
console      network_throughput shm            tty32         tty7          ttyS9
core         null           snapshot      tty33         tty8          uhid
cpu          parport0       snd           tty34         tty9          uinput
cpu_dma_latency port           sr0           tty35         ttyprintk     urandom
cuse        ppp            sr1           tty36         ttyS0         vcs
disk        psaux          stderr         tty37         ttyS1         vcs1
dmideid     ptmx           stdin         tty38         ttyS10        vcs2
dri         pts            stdout        tty39         ttyS11        vcs3
encryptfs   ram0           tty           tty4          ttyS12        vcs4
fb0         ram1           tty0          tty40         ttyS13        vcs5
fd          ram10          tty1          tty41         ttyS14        vcs6
```

(a) 查看/dev下的所有装置文件

```
malimei@malimei:/dev/bus$ ls -R
.:
usb

./usb:
001 002

./usb/001:
001

./usb/002:
001 002 003 004
```

(b) 查看/dev/bus下的装置文件

```
malimei@malimei:/dev/input$ ls -R
.:
by-id by-path event0 event1 event2 event3 mice mouse0 mouse1

./by-id:
usb-VMware_VMware_Virtual_USB_Mouse-event-mouse
usb-VMware_VMware_Virtual_USB_Mouse-mouse

./by-path:
pci-0000:02:00.0-usb-0:1:1.0-event-mouse platform-i8042-serio-1-event-mouse
pci-0000:02:00.0-usb-0:1:1.0-mouse      platform-i8042-serio-1-mouse
platform-i8042-serio-0-event-kbd
```

(c) 查看/dev/input下的装置文件

图 5.7 查看外部设备

目录下还有一些项目是没有的装置,这通常是在安装系统时所建立的,它不一定对应到实体的硬件装置。此外还有一些虚拟的装置,不对应到任何实体装置,例如空设备的/dev/null,任何写入该设备的请求均会被执行,但被写入的资料均会如进入空设备般消失。

### 3. /boot

该目录下存放与系统激活相关的文件,是系统启动时用到的程序。如图 5.8 所示,initrd.img、vmlinuz、System.map 均为重要文件,不可任意删除。其中,initrd.img 为系统激活时最先加载的文件;vmlinuz 为 Kernel 的镜像文件;System.map 包括 Kernel 的功能及位置。top、ps 命令读此文件来显示系统目前的信息状态。

```
malimei@malimei-virtual-machine:/mnt$ cd /boot
malimei@malimei-virtual-machine:/boot$ ls
config-4.15.0-45-generic      mentest86+.elf
config-4.15.0-54-generic      mentest86+_multiboot.bin
config-4.15.0-55-generic      System.map-4.15.0-45-generic
grub                          System.map-4.15.0-54-generic
initrd.img-4.15.0-45-generic  System.map-4.15.0-55-generic
initrd.img-4.15.0-54-generic  vmlinuz-4.15.0-45-generic
initrd.img-4.15.0-55-generic  vmlinuz-4.15.0-54-generic
lost+found                    vmlinuz-4.15.0-55-generic
mentest86+.bin
malimei@malimei-virtual-machine:/boot$
```

图 5.8 /boot 下的文件

### 4. /home

登录用户的主目录就放在此目录下,以用户的名称作为/home 目录下各个子目录的名称。如果建立一个用户,用户名是“malimei”,那么在/home 目录下就有一个对应的/home/malimei 路径,当用户 malimei 登录时,其所在的默认目录就是/home/malimei,如图 5.9 所示。

```
malimei@malimei-virtual-machine: ~
malimei@malimei-virtual-machine:~$ pwd
/home/malimei
malimei@malimei-virtual-machine:~$ ls -l
总用量 104672
-rw-rw-r-- 1 malimei malimei 10240 8月 16 09:41 11.tar
-rw-rw-r-- 1 user1 malimei 15 8月 16 11:59 11.txt
```

图 5.9 查看/home 下的用户目录

也可以在图形管理界面中查看用户主目录下的文档,如图 5.10 所示,标签页显示路径为/home,其下存放了三个用户各自的目录。双击图标 user1,进入用户 user1 的主目录,该用户的文件都存放在其中,如图 5.11 所示。



图 5.10 图形界面下查看/home 下的用户目录



图 5.11 用户 user1 的工作目录/home/user1

说明：创建用户的命令是 `adduser` 和 `useradd`，将在第 6 章介绍。

## 5. /lib

本目录存放了许多系统激活时所需要的重要的共享函数库，lib 是 library(库)的英文缩写。几乎所有的应用程序都会用到这个目录下的共享库。例如，文件名为 `library.so.version` 的共享函数库就放在 `/lib` 目录下，该函数库包含很多像 GNU C library(C 编译程序)这样的重要部分。在图 5.12 中，用命令 `ls` 查看了该目录下的库文件(该命令是在 `/lib` 目录下使用的)。

```
malimei@malimei:/lib$ ls
apparmor          libip4tc.so.0      modules
brltty            libip4tc.so.0.1.0  modules-load.d
cpp               libip6tc.so.0      plymouth
crda              libip6tc.so.0.1.0  recovery-mode
firmware          libiptc.so.0        resolvconf
hdparm            libiptc.so.0.0.0   systemd
i386-linux-gnu    libxtables.so.10    terminfo
ifupdown          libxtables.so.10.0.0  udev
init              linux-sound-base    ufw
klibc-SDKhWJaiUdo40xxZ-mvprY1CZus.so  lsb                  xtables
ld-linux.so.2     modprobe.d
```

图 5.12 查看/lib 下的库文件

Linux 下的库分为动态库和静态库，一般情况下，`.so` 为共享库，用于动态连接，`.a` 为静态库，用于静态连接。

## 6. /usr/lib

本目录下存放一些应用程序的共享函数库，例如 Netscape、X Server 等。图 5.13 中使用 `ls` 命令查看了该目录下的文件。其中，最重要的函数库为 `libc` 或 `glibc`(`glibc 2.x` 便是 `libc 6.x` 版本，标准 C 语言函数库)，几乎所有的程序都会用到 `libc` 或 `glibc`，因为这两个程序提供了对于 Linux Kernel 的标准接口。还有文件名为 `library.a` 的静态函数库，也放在 `/usr/lib` 下。

```
malimei@malimei:/$ cd usr
malimei@malimei:/usr$ ls
bin  games  include  lib  local  sbin  share  src
malimei@malimei:/usr$ cd lib
malimei@malimei:/usr/lib$ ls
2013.com.canonical.certification:checkbox
2013.com.canonical.certification:plainbox-resources
accountsservice
apg
apt
aspell
at-spi2-core
avahi
```

图 5.13 查看/usr/libs 下的共享函数库文件

## 7. /mnt

这个目录在一般情况下是空的,是系统默认的挂载点,可以临时将别的文件系统挂在这个目录下,如图 5.14 所示。如果要挂载额外的文件系统到/mnt 目录,需要在该目录下建立任一目录作为挂载目录。如新建/mnt/usb 目录,作为 USB 移动设备的挂载点。

```
malimei@malimei-virtual-machine: /mnt
malimei@malimei-virtual-machine:~$ cd /mnt
malimei@malimei-virtual-machine:/mnt$ ls
hgfs  usb
malimei@malimei-virtual-machine:/mnt$
```

图 5.14 查看/mnt 下的文件

## 8. /proc

本目录为一个虚拟文件系统,它不占用硬盘空间,该目录下的文件均放置于内存中。/proc 会记录系统正在运行的进程、硬件状态、内存使用的多少等信息,这些信息是在内存中由系统自己产生的。每当存取/proc 文件系统时,Kernel 会拦截存取动作并获取相关信息再动态地产生目录与文件内容,如图 5.15 所示。

```
malimei@malimei:/proc$ ls
1      140  168  2169  28   3041  471  asound      modules
10     141  169  22    2811  3053  472  buddyinfo  mounts
1091   142  17   222   2814  3064  473  bus         mpt
1098   143  170  2251  2816  3072  485  cgroups    mtrr
11     144  171  23    2818  3083  5    cmdline    net
1112   145  172  230   2822  3137  50   consoles   pagetypeinfo
1127   146  173  231   2823  3144  508  cpuinfo    partitions
1129   147  174  2312  2829  3158  536  crypto     sched_debug
1131   148  175  24    2830  3163  545  devices    schedstat
12     149  176  2545  2835  3170  552  diskstats  scsi
1234   15   177  2564  2857  3178  557  dma        self
124    150  178  2568  2870  3195  561  driver     slabinfo
125    151  179  26    2882  3260  647  execdomains softirqs
1253   152  18   2633  29   361   65   fb         stat
126    153  180  2644  2915  3670  670  filesystems swaps
127    1530 181  2655  2926  3677  7    fs         sys
1278   154  182  2660  2966  3678  71   interrupts sysrq-trigger
```

图 5.15 查看/proc 下的进程文件

## 9. /root

/root 是系统管理用户 root 的主目录,如果用户是以超级用户的身份登录的,这个就是超级用户的主目录,如图 5.16 所示。

```
root@malimei-virtual-machine:~# pwd
/root
root@malimei-virtual-machine:~#
```

图 5.16 超级用户的主目录/root

## 10. /bin

本目录存放一些系统启动时所需要的普通程序和系统程序,及一些经常被其他程序调用的程序,是 Linux 常用的外部命令存放的目录。例如,ls、cat、cp、mkdir、rm、su、tar 等,和外部命令相对应的还有内部命令,只要 Linux 系统启动起来,内部命令就可以应用,如 cd 等,如图 5.17 所示。

```
malimei@malimei-virtual-machine: /bin
bzip2          kbd_mode      ntfsfix       systemd-notify
bzip2recover  kill          ntfsinfo     systemd-tmpfiles
bzless        kmod          ntfsls       systemd-tty-ask-password-agent
bzmores       less          ntfsmove     tailf
cat           lessecho     ntfstruncate tar
chacl        lessfile     ntfswipe     tempfile
chgrp        lesskey      open          touch
chmod        lesspipe     openvt       true
chown        ln           pidof        udevadm
chvt         loadkeys    ping         ulockmgr_server
cp           login        pingfs       umount
cpio         loginctl     plymouth     uname
dash         lowntfs-3g  ps           uncompress
date         ls           pwd          unicode_start
dd           lsblk       rbash        vdir
df           lsmode      readlink     wdctl
dir          mkdir        red          which
dmesg        mknod       rm           whiptail
dnsdomainname mktemp      rmdir        ypdomainname
domainname   more        rnano        zcat
dumpkeys     mount       run-parts    zcmp
echo         mountpoint  sed          zdiff
ed           mt          setfacl     zegrep
efibootmgr   mt-gnu      setfont     zfgrep
egrep        mv          setupcon    zforce
false        nano        sh           zgrep
fgconsole    nc          sh.distrib  zless
fgrep        nc.openbsd  sleep       zmore
findmnt      netcat      ss          znew
fuser        netstat     static-sh
```

图 5.17 查看/bin下的程序文件

## 11. /tmp

该目录存放系统启动时产生的临时文件。有时某些应用程序执行中产生的临时文件也会暂放在此目录,如图 5.18 所示。

```
malimei@malimei:/tmp$ ls
config-err-YMNONQ          sni-qt_fcitx-qimpanel_3072-NqSv94
fcitx-qimpanel:0.pid      unity_support_test.0
fcitx-socket-:0          vmware-3
indicator-china-weather-1001.pid  VMwareDnd
qtsingleapp-chines-1a7c-3e9  vmware-root
qtsingleapp-chines-1a7c-3e9-lockfile  vmware-root-558079768
sni-qt_chinese-calendar_2978-mkQKXv
```

图 5.18 查看/tmp下的临时文件

## 12. /var

该目录存放被系统修改过的数据。在这个目录下的重要目录有 /var/log、/var/spool、/var/run 等,分别用于存放记录文件、新闻邮件、运行时信息,如图 5.19 所示。

```
malimei@malimei:/var$ ls
backups  crash  local  log  metrics  run  tmp
cache   lib    lock   mail  opt      spool
malimei@malimei:/var$
```

图 5.19 查看/var 下的文件

## 5.1.4 文件的结构、类型和属性

### 1. 文件结构

文件结构是文件存放在磁盘等存储设备上的组织方法,主要体现在对文件和目录的组织上。目录提供了管理文件的一个方便而有效的途径。Linux 使用标准的目录结构,在安装的时候,安装程序就已经为用户创建了文件系统和完整而固定的目录组成形式,并指定了每个目录的作用和其中的文件类型。

Linux 采用的是树形结构。最上层是根目录,其他的所有目录都是从根目录出发而生成的。微软的 DOS 和 Windows 也是采用树形结构,但是在 DOS 和 Windows 中这样的树形结构的根是磁盘分区的盘符,有几个分区就有几个树形结构,它们之间的关系是并列的。但是在 Linux 中,无论操作系统管理几个磁盘分区,这样的目录树只有一个。从结构上讲,各个磁盘分区上的树形目录不一定是并列的,因为 Linux 是一个多用户系统,一个固定的目录规划有助于对系统文件和不同的用户文件进行统一管理。

Linux 中对文件路径的表达有两种方法——绝对路径和相对路径。

**绝对路径:**从根目录/开始的路径。比如“/home/malimei/Documents/test1,”这一路径与当前处于哪个目录没有关系,表达式是固定的。

**相对路径:**以“.”或“..”开始的,“.”表示用户当前操作所处的位置,而“..”表示上级目录。比如“./Documents/test1”,与当前目录相关。

下面举例说明这两种路径,在 home 下存在用户 malimei 和用户 user1,当前用户为 malimei,即当前目录为/home/malimei。现有 malimei/Documents 下的 test1 文件和 user1/Documents 下的 test2 文件,使用 cat 命令查看这两个文件时,可分别使用这两种不同的路径方式。如图 5.20 所示使用的是绝对路径方式,如图 5.21 所示使用的是相对路径的方式。

**说明:**请比较一下,对于这个例子用哪种路径的方法显示文件比较好?为什么?

```
malimei@malimei-virtual-machine:~$ cd /home/malimei
malimei@malimei-virtual-machine:~$ cat /home/malimei/documents/test1
Good morning!
How are you!

malimei@malimei-virtual-machine:~$ cat /home/user1/documents/test2
Good afternoon!
How are you!
```

图 5.20 用绝对路径的方法显示文件 test1 和 test2

```
malimei@malimei-virtual-machine:~$ cd documents
malimei@malimei-virtual-machine:~/documents$ cat test1
Good morning!
How are you!

malimei@malimei-virtual-machine:~/documents$ cd ..
malimei@malimei-virtual-machine:~$ pwd
/home/malimei
malimei@malimei-virtual-machine:~$ cd ..
malimei@malimei-virtual-machine:/home$ cd user1
malimei@malimei-virtual-machine:/home/user1$ cat ./documents/test2
Good afternoon!
How are you!

malimei@malimei-virtual-machine:/home/user1$ cat documents/test2
Good afternoon!
How are you!
```

图 5.21 用相对路径的方法显示文件 test1 和 test2

## 2. 文件类型

在 Linux 系统中主要根据文件头信息来判断文件类型, Linux 系统的文件类型有以下几种。

### (1) 普通文件。

普通文件就是用户通常访问的文件, 由 `ls -l` 命令显示出来的属性中, 第一个属性为“-”。

可以使用 `ls -l` 来查看文件属性, 如图 5.22 所示, 显示了 /bin 下的各个文件, 其中的第一个“bash”就是一个普通文件, 其属性(左侧第一列)的第一位是“-”。

```
malimei@malimei:/$ cd bin
malimei@malimei:/bin$ ls -l
total 9468
-rwxr-xr-x 1 root root 986672 Oct 7 2014 bash
-rwxr-xr-x 1 root root 30240 Oct 21 2013 bunzip2
-rwxr-xr-x 1 root root 1713424 Nov 14 2013 busybox
-rwxr-xr-x 1 root root 30240 Oct 21 2013 bzipcat
lrwxrwxrwx 1 root root 6 Aug 5 20:23 bzipcmp -> bzipdiff
-rwxr-xr-x 1 root root 2140 Oct 21 2013 bzipdiff
lrwxrwxrwx 1 root root 6 Aug 5 20:23 bzegrep -> bzgrep
-rwxr-xr-x 1 root root 4877 Oct 21 2013 bzipexe
lrwxrwxrwx 1 root root 6 Aug 5 20:23 bzfgrep -> bzgrep
-rwxr-xr-x 1 root root 3642 Oct 21 2013 bzgrep
```

图 5.22 查看普通文件

### (2) 纯文本文件。

普通文件中, 有些文件内容可以直接读取, 如文本文件, 文件的内容一般是字母、数字以及一些符号等。可以使用 `cat`、`vi` 命令直接查看文件内容, 如图 5.23 所示。有些文件是为系统准备的, 如二进制文件, 可执行的文件就是这种格式, 如命令 `cat` 就是二进制文件。还有些文件是为运行中的程序准备的, 如数据格式的文件, Linux 用户在登录系统时, 会将登录数据记录在 `/var/log/wtmp` 文件内, 这个文件就是数据文件。

### (3) 目录文件。

目录文件就是目录, 相当于 Windows 中的文件夹。

可以使用 `ls -l` 命令显示文件的属性, 其中第一个属性为 `d` 的是目录文件, 如图 5.24 所示, 根目录下的 `bin`、`boot`、`dev` 等都是目录。

```
malimei@malimei:~/Documents$ ls -l
总用量 4
-rw-rw-r-- 1 malimei malimei 28 Oct 30 08:49 test1
-rw-rw-r-- 1 malimei malimei 0 Oct 30 05:28 test2
malimei@malimei:~/Documents$ cat test1
1111111111111111
1111111111111111
malimei@malimei:~/Documents$
```

图 5.23 用 cat 查看纯文本文件内容

```
malimei@malimei:/$ ls -l
total 92
drwxr-xr-x  2 root root  4096 Aug  5 21:09 bin
drwxr-xr-x  3 root root  4096 Aug  5 21:11 boot
drwxrwxr-x  2 root root  4096 Aug  5 20:29 cdrom
drwxr-xr-x 16 root root 4260 Oct 30 03:57 dev
drwxr-xr-x 130 root root 12288 Oct 30 04:05 etc
drwxr-xr-x  5 root root  4096 Oct 30 04:05 home
lrwxrwxrwx  1 root root    33 Aug  5 21:09 initrd.img -> boot/initrd.img-3.16.0-30-generic
drwxr-xr-x 23 root root  4096 Aug  5 21:09 lib
drwx----- 2 root root 16384 Aug  5 20:22 lost+found
drwxr-xr-x  3 root root  4096 Feb 18  2015 media
drwxr-xr-x  3 root root  4096 Aug  5 21:11 mnt
drwxr-xr-x  2 root root  4096 Aug  5 21:11 opt
dr-xr-xr-x 236 root root    0 Oct 30 03:56 proc
drwx----- 5 root root  4096 Oct 30 03:42 root
drwxr-xr-x 23 root root   800 Oct 30 03:58 run
```

图 5.24 查看目录文件

#### (4) 链接文件。

在 Linux 中有两种链接方式：符号链接和硬链接。符号链接相当于 Windows 中的快捷方式。可用 `ls -l` 命令查看文件属性，符号链接文件的第一个属性用 `l` 表示，只有符号链接才会显示属性 `l`。如图 5.25 所示，`bzcmp` 就是一个链接文件，其指向 `bzdiff`（5.5 节将详细介绍链接文件）。

```
malimei@malimei:/$ cd bin
malimei@malimei:/bin$ ls -l
total 9468
-rwxr-xr-x 1 root root 986672 Oct  7  2014 bash
-rwxr-xr-x 1 root root 30240 Oct 21  2013 bunzip2
-rwxr-xr-x 1 root root 1713424 Nov 14  2013 busybox
-rwxr-xr-x 1 root root 30240 Oct 21  2013 bzip2
lrwxrwxrwx 1 root root    6 Aug  5 20:23 bzcmp -> bzdiff
-rwxr-xr-x 1 root root 2140 Oct 21  2013 bzdiff
lrwxrwxrwx 1 root root    6 Aug  5 20:23 bzegrep -> bzgrep
-rwxr-xr-x 1 root root 4877 Oct 21  2013 bzeze
lrwxrwxrwx 1 root root    6 Aug  5 20:23 bzfgrep -> bzgrep
-rwxr-xr-x 1 root root 3642 Oct 21  2013 bzgrep
-rwxr-xr-x 1 root root 30240 Oct 21  2013 bzip2
-rwxr-xr-x 1 root root 9624 Oct 21  2013 bzip2recover
lrwxrwxrwx 1 root root    6 Aug  5 20:23 bzless -> bzm0re
```

图 5.25 查看链接文件

#### (5) 设备文件。

设备文件是 Linux 系统中最特殊的文件。Linux 系统为外部设备提供一种标准接口，将外部设备视为一种特殊的文件，即设备文件。它能够在系统设备初始化时动态地在 `/dev` 目录下创建好各种设备的文件节点，如图 5.26 所示，在设备卸载后自动删除 `/dev` 下对应的

```

brw-rw---- 1 root disk      8, 0 Oct 30 03:57 sda
brw-rw---- 1 root disk      8, 1 Oct 30 03:57 sda1
brw-rw---- 1 root disk      8, 2 Oct 30 03:57 sda2
brw-rw---- 1 root disk      8, 5 Oct 30 03:57 sda5
crw-rw---- 1 root disk     21, 0 Oct 30 03:57 sg0
crw-rw----+ 1 root cdrom   21, 1 Oct 30 03:57 sg1
crw-rw----+ 1 root cdrom   21, 2 Oct 30 03:57 sg2

```

图 5.26 查看各分区对应的设备文件

文件节点。在编写设备驱动的时候,不必再为设备指定主设备号,在设备注册时用 0 来动态获取可用的主设备号,然后在驱动中来实现创建和销毁设备文件。

在 Linux 系统中设备文件分为字符设备文件和块设备文件。字符设备文件是指设备发送和接收数据以字符的形式进行;而块设备文件则以整个数据缓冲区的形式进行。由 `ls -l/dev` 命令显示出来的属性中,字符设备文件的第一个属性是 c,块设备文件的第一个属性是 b。在图 5.27 中,第一行的 `agpgart` 是字符设备文件,而倒数第二行的 `fd0` 是块设备文件。

```

malimei@malimei:/S cd dev
malimei@malimei:/dev$ ls -l
total 0
crw-rw---- 1 root video    10, 175 Oct 30 03:57 agpgart
crw----- 1 root root      10, 235 Oct 30 03:57 autofs
drwxr-xr-x 2 root root      660 Oct 30 03:56 block
drwxr-xr-x 2 root root      100 Oct 30 03:56 bsg
crw----- 1 root root     10, 234 Oct 30 03:57 btrfs-control
drwxr-xr-x 3 root root      60 Oct 30 03:56 bus
lrwxrwxrwx 1 root root        3 Oct 30 03:57 cdrom -> sr0
drwxr-xr-x 2 root root    3580 Oct 30 03:57 char
crw----- 1 root root      5, 1 Oct 30 03:57 console
lrwxrwxrwx 1 root root      11 Oct 30 03:56 core -> /proc/kcore
drwxr-xr-x 2 root root      60 Oct 30 03:56 cpu
crw----- 1 root root     10, 60 Oct 30 03:57 cpu_dma_latency
crw----- 1 root root     10, 203 Oct 30 03:57 cuse
drwxr-xr-x 5 root root     100 Oct 30 03:56 disk
crw-rw----+ 1 root audio   14, 9 Oct 30 03:57 dmide
drwxr-xr-x 2 root root      80 Oct 30 03:57 drt
crw----- 1 root root     10, 61 Oct 30 03:57 ecryptfs
crw-rw---- 1 root video    29, 0 Oct 30 03:57 fb0
lrwxrwxrwx 1 root root     13 Oct 30 03:56 fd -> /proc/self/fd
brw-rw---- 1 root floppy   2, 0 Oct 30 03:57 fd0
crw-rw-rw- 1 root root      1, 7 Oct 30 03:57 full

```

图 5.27 查看设备文件

执行 `ls -l /dev/ | grep "^c"`,这条命令的意思就是在 `/dev` 目录下查找以 c 开头的文件,这里以 c 开头的文件就是字符设备文件,如图 5.28 所示。

#### (6) 套接字文件。

套接字文件通常用于网络数据连接。由 `ls -l` 命令显示出来的属性中,套接字文件的第一个属性用 s 表示,如图 5.29 所示的 `acpid.socket` 文件。

#### (7) 管道文件。

主要用来解决多个程序同时访问一个文件所造成的错误。由 `ls -l` 命令显示出来的属性中,管道文件的第一个属性用 p 表示,管道一般的权限是:所有者有读写权限,而所属组与其他用户都只有读的权限,管道文件一般都是存放在 `/dev` 目录下面,可以执行下面的命令去查看一下它的属性位: `ls -l /dev/ | grep "^p"`。这条命令的意思就是:列出 `/dev/` 目录下的文件的详细信息,然后查找以 p 开头的文件,这里的 p 就是管道文件类型,如图 5.30 所示。

```
malimei@malimei-virtual-machine:/dev$ ls -l /dev/|grep "^c"
crw----- 1 root root    10, 175 8月 15 08:23 agpgart
crw-r--r-- 1 root root    10, 235 8月 15 08:23 autofs
crw----- 1 root root    10, 234 8月 15 08:23 btrfs-control
crw----- 1 root root     5,  1 8月 15 08:24 console
crw----- 1 root root    10,  59 8月 15 08:23 cpu_dma_latency
crw----- 1 root root    10, 203 8月 15 08:23 cuse
crw-rw----+ 1 root audio  14,  9 8月 15 08:23 dmide
crw----- 1 root root    10,  61 8月 15 08:23 encryptfs
crw-rw---- 1 root video  29,  0 8月 15 08:23 fb0
crw-rw-rw- 1 root root     1,  7 8月 15 08:23 full
crw-rw-rw- 1 root root    10, 229 8月 15 08:23 fuse
crw----- 1 root root   244,  0 8月 15 08:23 hidraw0
crw----- 1 root root    10, 228 8月 15 08:23 hpet
crw----- 1 root root    10, 183 8月 15 08:23 hwrng
crw-r--r-- 1 root root     1, 11 8月 15 08:23 kmsg
crw-rw---- 1 root disk   10, 237 8月 15 08:23 loop-control
crw----- 1 root root    10, 227 8月 15 08:23 mcelog
crw-r----- 1 root kmem   1,  1 8月 15 08:23 mem
crw----- 1 root root    10,  56 8月 15 08:23 memory_bandwidth
crw-rw----+ 1 root audio  14,  2 8月 15 08:23 midi
crw----- 1 root root    10,  58 8月 15 08:23 network_latency
```

图 5.28 查找以 c 开头的字符设备文件

```
malimei@malimei:/var/run$ ls -l
total 52
-rw-r--r-- 1 root root    4 Oct 30 03:57 acpid.pid
srw-rw-rw- 1 root root    0 Oct 30 03:57 acpid.socket
drwxr-xr-x 2 root root   40 Oct 30 03:57 alsa
drwxr-xr-x 2 avahi avahi  80 Oct 30 08:00 avahi-daemon
```

图 5.29 查看套接字文件

```
malimei@malimei-virtual-machine:/dev$ ls -l *.pipe
prw-r--r-- 1 root root 0 8月 17 09:39 1.pipe
malimei@malimei-virtual-machine:/dev$ ls -l /dev/ | grep "^p"
prw-r--r-- 1 root root    0 8月 17 09:39 1.pipe
malimei@malimei-virtual-machine:/dev$
```

图 5.30 管道文件

### 3. 文件属性

对于 Linux 系统的文件来说,其基本的属性有三种:读(r/4)、写(w/2)、执行(x/1)。不同用户对于文件拥有不同的读、写和执行权限。

(1) 读权限:具有读取目录结构的权限,可以查看和阅读文件,禁止对其做任何的更改操作,用 r 或 4 表示。

(2) 写权限:可以新建、删除、重命名、移动目录或文件(不过写权限受父目录权限控制),用 w 或 2 表示。

(3) 执行权限:文件拥有执行权限,才可以运行,比如二进制文件和脚本文件。有执行权限才可以进入目录文件,用 x 或 1 表示。

文件被创建时,文件所有者自动拥有对该文件的读、写和可执行权限,以便于对文件的阅读和修改。用户也可根据需要把访问权限设置为需要的任何组合。

## 5.2 文件操作命令



### 5.2.1 显示文件内容

#### 1. cat

**功能描述：**用来串接文件或显示文件的内容，也可以从标准输入设备读取数据并将其结果重定向到一个新的文件中，达到建立新文件的目的。

**语法：**cat [选项] [文件名]

**选项：**cat 命令中的常用选项如表 5.1 所示。

表 5.1 cat 命令中的常用选项

选项	作用
-n 或 -number	由 1 开始对所有输出的行数编号
-b	和-n 相似，只不过对于空白行不编号
-s	当遇到有连续两行以上的空白行时，就代换为一行空白行
-E	--show-ends, 在每行结束处显示 \$

1) 显示文件内容到屏幕

**例 5.1** 查看文件/etc/network/interfaces 的内容,带行号,如图 5.31 所示。

```
$ cd /etc          更改当前目录为/etc。
$ cd network      更改当前目录为/etc/network。
$ ls              查看/etc/network下的文件。
$ cat -n interfaces 查看该目录下文件 interfaces 的内容。
```

```
maline1@maline1-virtual-machine:/dev$ cd /etc/network
maline1@maline1-virtual-machine:/etc/network$ cat -n interfaces
 1 # interfaces(5) file used by ifup(8) and ifdown(8)
 2 auto lo
 3 iface lo inet loopback
maline1@maline1-virtual-machine:/etc/network$
```

图 5.31 查看文件内容

2) 显示文件内容到文件

cat 命令可以用于输出重定向,可以将现有文件的内容重定向到已有文件,如果目标文件不存在,则建立新文件。

格式一: cat a1.txt a2.txt > a3.txt

其中,“>”表示输出重定向,将 a1 和 a2 的内容输出到 a3。如果 a3.txt 不存在,就新建一个 a3 文件。

格式二: cat a1.txt a2.txt >> a3.txt

其中,“>>”表示追加重定向,将 a1 和 a2 的内容添加到 a3 的尾部,如果 a3 不存在,就新建一个 a3 文件。

3) 串接输入内容到文件

cat 命令也可以从标准输入设备读取数据到已有文件或新建文件。

格式一: `cat > a.txt`

格式二: `cat -> a.txt`

从标准输入设备输入内容到文件 `a.txt`, 如果 `a.txt` 不存在则新建一个 `a.txt` 文件。从屏幕输入时, 用按 `Ctrl+D` 组合键退出输入。

**例 5.2** 查看文件内容, 并将内容复制到其他文件。

```
$ cat test1
```

查看 `test1` 的内容。

```
$ cat test1 > test2
```

将 `test1` 内容重定向到 `test2` (`test2` 原来为空文本)。

```
$ cat test1 >> test2
```

将 `test1` 内容重定向到 `test2` 的尾部, 如图 5.32 所示。

```
$ cat test1 test2 >> test3
```

将 `test1` 和 `test2` 合并到新文件 `test3`, 如图 5.33 所示。

```
$ cat $ cat ->> test3
```

从键盘输入内容到 `test3` (按 `Ctrl+D` 组合键退出输入), 如图 5.34 所示。

```
malimei@malimei:~/Documents$ cat test1
11111111111111
11111111111111
malimei@malimei:~/Documents$ cat test1>test2
malimei@malimei:~/Documents$ cat test2
11111111111111
11111111111111
malimei@malimei:~/Documents$ cat test1>>test2
malimei@malimei:~/Documents$ cat test2
11111111111111
11111111111111
11111111111111
11111111111111
malimei@malimei:~/Documents$
```

图 5.32 将 `test1` 内容复制到 `test2` (或尾部)

```
malimei@malimei:~/Documents$ ls
test1 test2
malimei@malimei:~/Documents$ cat test1 test2>test3
malimei@malimei:~/Documents$ ls
test1 test2 test3
malimei@malimei:~/Documents$ cat test3
11111111111111
11111111111111
11111111111111
11111111111111
11111111111111
malimei@malimei:~/Documents$
```

图 5.33 将 `test1` 和 `test2` 内容合并复制到 `test3`

```
malimei@malimei:~/Documents$ cat ->>test3
22222222222222
33333333333333
malimei@malimei:~/Documents$ cat test3
11111111111111
11111111111111
11111111111111
11111111111111
11111111111111
11111111111111
11111111111111
22222222222222
33333333333333
```

图 5.34 从键盘输入内容到 `test3`

**例 5.3** 创建文件并为文件输入内容,遇到结束标志 EOF 退出编辑,如图 5.35 所示。

```
$ cat > textfile << EOF          创建 textfile 文件。
> This is a text file.          输入内容。
> I like it.                    输入内容。
> EOF                            退出编辑状态。
```

```
malimei@malimei:~/Documents$ cat >textfile <<EOF
> This is a textfile.
> I like it.
> EOF
malimei@malimei:~/Documents$ cat textfile
This is a textfile.
I like it.
```

图 5.35 从键盘输入到新文件

**例 5.4** 向已存在文件追加内容,遇到结束标志 EOF 退出编辑,如图 5.36 所示。

```
$ cat >> textfile << EOF        向 textfile 文件追加内容。
> really?                       所追加的内容。
> yes!                           所追加的内容。
> EOF                             退出编辑。
```

```
malimei@malimei:~/Documents$ cat >>textfile << EOF
> really?
> yes!
> EOF
malimei@malimei:~/Documents$ cat textfile
This is a textfile.
I like it.
really?
yes!
```

图 5.36 从键盘输入到已有文件

**例 5.5** 连接多个文件内容并且输出到一个文件中,如图 5.37 所示。

```
$ cat text1 text2 text3 > text0    将 text1、text2、text3 文件内容放入 text0 文件。
```

```
malimei@malimei:~/Documents$ cat text1
abcde
malimei@malimei:~/Documents$ cat text2
fghijkl
malimei@malimei:~/Documents$ cat text3
mnopqrst
malimei@malimei:~/Documents$ cat text1 text2 text3 > text0
malimei@malimei:~/Documents$ cat text0
abcde
fghijkl
mnopqrst
```

图 5.37 复制多个文件内容到 text0

在该例中,如果输出到的文件已存在则会将文件中原有内容先清空。

**例 5.6** 将一个或多个已存在的文件内容追加到一个已存在的文件中,不影响原文件内容,如图 5.38 所示。

```
$ cat text1 text2 text3 >> text4    将 text1、text2、text3 文件追加到 text4 文件中。
```

```
malimei@malimei:~/Documents$ cat text1
abcde
malimei@malimei:~/Documents$ cat text2
fghijkl
malimei@malimei:~/Documents$ cat text3
mnopqrst
malimei@malimei:~/Documents$ cat text4
uvwxyz
malimei@malimei:~/Documents$ cat text1 text2 text3 >> text4
malimei@malimei:~/Documents$ cat text4
uvwxyz
abcde
fghijkl
mnopqrst
```

图 5.38 复制多个文件内容到 text4 尾部

2. more

**功能描述:** 显示输出的内容,然后根据窗口的大小进行分页显示,在终端底部打印出“--More--”及已显示文本占全部文本的百分比。

**语法:** more [选项][文件名]

**选项:** more 命令的常用选项如表 5.2 所示。

表 5.2 more 命令的常用选项

选 项	作 用
f 或<空格>	显示下一页
<回车>	显示下一行
q 或 Q	退出 more
+num	从第 num 行开始显示
-num	定义屏幕大小为 num 行
+ /pattern	从 pattern 前两行开始显示
-c	从顶部清屏然后开始显示
-d	提示按空格键继续,按 Q 键退出,禁止响铃功能
-l	忽略换页(Ctrl+l)字符
-p	通过清除窗口而不是滚屏来对文件进行换页
-s	把连续的多个空行显示为一行
-u	把文件内容中的下画线去掉

**例 5.7** 查看文件内容,如图 5.39 所示。

```
$ more isolat1.ent
```

分页显示文件 isolat1.ent 的内容。

当文件较大时,文本内容会在屏幕上快速显示,more 命令解决了这个问题,一次只显示一屏的文本。输入命令后显示的是文本内容的第一页,按 Enter 键显示下一行,按 f 键或空格键显示下一页,按 Ctrl+B 组合键返回上一屏,按 q 键退出显示。

**例 5.8** 带选项查看文件内容,如图 5.40 所示。

```
$ more +5 test3
```

从文件的第 5 行开始显示。

```
$ more -4 test3
```

每屏只显示 4 行。

```
$ more +/2 test3
```

从文件中的第一个“2”的前两行开始显示。

```
$ more -dc test3
```

显示提示,并从终端或控制台顶部显示。

```
malimei@malimei:/usr/share/yelp/dtd$ more isolat1.ent
<!--
  File isolat1.ent produced by the XSL script entities.xml
  from input data in unicode.xml.

  Please report any errors to David Carlisle
  via the public W3C list www-math@w3.org.

  The numeric character values assigned to each entity
  (should) match the Unicode assignments in Unicode 4.0.

  Entity names in this file are derived from files carrying the
  following notice:

  (C) International Organization for Standardization 1986
  Permission to copy in any form is granted for use with
  conforming SGML systems and applications as defined in
  ISO 8879, provided this notice is included in all copies.
-->

<!--
--更多--(10%)
```

图 5.39 使用 more 命令查看文件内容

```
malimei@malimei:~/Documents$ cat test3
11111111111111
11111111111111
11111111111111
11111111111111
11111111111111
Firefox Web Browser
22222222222222
33333333333333
malimei@malimei:~/Documents$ more +5 test3
11111111111111
11111111111111
22222222222222
33333333333333
malimei@malimei:~/Documents$ more -4 test3
11111111111111
11111111111111
11111111111111
11111111111111
--More--(50%)
[3]+ Stopped more -4 test3
malimei@malimei:~/Documents$ more +/2 test3
...skipping
11111111111111
11111111111111
22222222222222
33333333333333
malimei@malimei:~/Documents$
```

图 5.40 查看文件内容

### 3. less

功能描述：显示输出的内容，然后根据窗口的大小进行分页显示。

语法：less [选项] [文件名]

选项：less 命令的常用选项如表 5.3 所示。



```
malimei@malimei:~/Documents$ head -n 3 English
The air we breathe is so freely available that we take it for granted. Yet without it we could not survive more than a few minutes. For the most part, the same air is available to everyone, and everyone needs it. Some people use the air to sustain them while they sit around and feel sorry for themselves. Others breathe in the air and use the energy it provides to make a magnificent life for themselves.
Opportunity is the same way. It is everywhere. Opportunity is so freely available that we take it for granted. Yet opportunity alone is not enough to create success. Opportunity must be seized and acted upon in order to have value. So many people are so anxious to "get in" on a "ground floor opportunity", as if the opportunity will do all the work. That's impossible.
Just as you need air to breathe, you need opportunity to succeed. It takes more than just breathing in the fresh air of opportunity, however. You must make use of that opportunity. That's not up to the opportunity. That's up to you. It doesn't matter what "floor" the opportunity is on. What matters is what you do with it.
```

图 5.42 显示文件的前 3 行/段

**例 5.11** 带选项查看文件,如图 5.43 所示。

```
$ head -2 test3          显示文件的前 2 行。
$ head -n 2 test3       显示文件的前 2 行。
$ head -n -5 test3      显示文件除后 5 行以外的所有内容。
```

```
malimei@malimei:~/Documents$ cat test3
1111111111111111
1111111111111111
1111111111111111
1111111111111111
1111111111111111
1111111111111111
1111111111111111
22222222222222
33333333333333
malimei@malimei:~/Documents$ head -2 test3
1111111111111111
1111111111111111
malimei@malimei:~/Documents$ head -n 2 test3
1111111111111111
1111111111111111
malimei@malimei:~/Documents$ head -n -5 test3
1111111111111111
1111111111111111
1111111111111111
malimei@malimei:~/Documents$
```

图 5.43 按照行查看文件

**例 5.12** 显示文件的前 n 个字节,如图 5.44 所示。

```
$ head -c 25 test3      显示文件的前 25 个字节。
$ head -c -50 test3    显示文件除了最后 50 个字节以外的内容。
```

```
malimei@malimei:~/Documents$ head -c 25 test3
1111111111111111malimei@malimei:~/Documents$ head -c 25 test3
1111111111111111
1111111111111111malimei@malimei:~/Documents$ head -c 35 test3
1111111111111111
1111111111111111
1111111111111111malimei@malimei:~/Documents$ head -c -50 test3
1111111111111111
1111111111111111
1111111111111111
1111111111111111
1111111111111111
1111111111111111
1111111111111111
malimei@malimei:~/Documents$
```

图 5.44 按照字节查看文件

## 5. tail

**功能描述：**显示文件的最后 n 行。

**语法：**tail [选项] [文件名]

**选项：**tail 命令的常用选项如表 5.5 所示。

表 5.5 tail 命令的常用选项

选 项	作 用
-n	显示文件的最后 n 行,系统默认值是 10
-f	不断读取文件的最新内容,达到实时监控的目的

**例 5.13** 查看文件内容,如图 5.45 所示。

```
$ tail -3 test3          显示文件的最后 3 行。
$ tail -n 2 test3       显示文件的最后 2 行。
$ tail -f test3         显示文件内容,并且不断刷新。按 Ctrl+Z 组合键退出实时监控。
```

```
malimei@malimei:~/Documents$ tail -3 test3
1111111111111111
2222222222222222
3333333333333333
malimei@malimei:~/Documents$ tail -n 2 test3
2222222222222222
3333333333333333
malimei@malimei:~/Documents$ tail -f test3
1111111111111111
1111111111111111
1111111111111111
1111111111111111
1111111111111111
1111111111111111
2222222222222222
3333333333333333
^Z
[1]+  Stopped                  tail -f test3
malimei@malimei:~/Documents$
```

图 5.45 使用 tail 命令查看文件内容

## 6. echo

**功能描述：**输出字符串到基本输出,通常是在显示器上输出,输出的字符串间以空白字符隔开,并在最后加上换行号。

echo 命令的功能是在显示器上显示一段文字,一般起到提示的作用。该命令在 Shell 编程中极为常用,如检查变量 value 的取值时,可以利用 echo 命令将 value 值打印到显示器上。

**语法：**echo[选项] 字符串

**选项：**echo 的常用选项如表 5.6 所示。

如图 5.46 所示,在 echo 命令中选项 n 表示输出文字后不换行;字符串能加引号,也能不加引号;用 echo 命令输出加引号的字符串时,将字符串原样输出;用 echo 命令输出不加引号的字符串时,将字符串中的各个单词作为字符串输出,各字符串之间用一个空格分隔。

表 5.6 echo 的常用命令选项

选 项	作 用	选 项	作 用
-n	不输出末尾的换行符	\b	退格
-e	启用反斜线转义	\\	反斜线
\a	发出警告声	\n	另起一行
\c	最后不加上换行符号	\r	回车
\f	换行但光标仍旧停留在原来的位置	\t	插入 Tab
\nnn	插入 nnn(八进制)所代表的 ASCII 字符	\v	垂直制表符

```
malimei@malimei-virtual-machine:~$ echo oo
oo
malimei@malimei-virtual-machine:~$ echo -n oo
oomalimei@malimei-virtual-machine:~$ echo oo nn mm
oo nn mm
malimei@malimei-virtual-machine:~$ echo "oo nn mm"
oo nn mm
malimei@malimei-virtual-machine:~$ echo oo      nn mm
oo nn mm
malimei@malimei-virtual-machine:~$ echo "oo      nn mm"
oo      nn mm
malimei@malimei-virtual-machine:~$
```

图 5.46 echo 命令输出内容

例 5.14 输出内容,如图 5.47 所示。

\$ echo -e "I \nlike\nyou!" 输入"I like you!",\n 表示换行。

```
malimei@malimei:~/Documents$ echo -e "I\nlike\nyou!"
I
like
you!
```

图 5.47 echo -e 输出内容

## 7. od

**功能描述:** od 命令用于输出文件的八进制、十六进制或其他格式编码的字节,通常用于显示或查看文件中不能直接显示在终端的字符。

**语法:** od [选项] 字符串

**选项:** od 命令的常用选项如表 5.7 所示。

表 5.7 od 命令的常用选项

选 项	作 用
-a	表示 ASCII 码的名字
-b	按照 3 个数值位的八进制数进行解释
-c	选择 ASCII 码字符或者是转义字符
-d	选择无符号两字节单位
-f	选择单精度浮点数
-l	等价于-t dI,选择十进制整型
-l	等价于-t dL,选择十进制长整型
-o	等价于-t o2,选择两字节单元并按照八进制解释
-s	等价于-t d2,选择两字节单元并按照十进制解释
-x	等价于-t x2,选择两字节单元并按照十六进制解释

例 5.15 按照八进制输出,如图 5.48 所示。

\$ od -b text1.doc                      使用单字节八进制进行输出。

```
malimei@malimei:~/Documents$ cat text1
abcde
malimei@malimei:~/Documents$ od -b text1
0000000 141 142 143 144 145 012
0000006
```

图 5.48 以八进制格式输出文件

## 5.2.2 显示目录及文件

### 1. ls

**功能描述:** 列出目录的内容,是 list 的简写形式。

**语法:** ls [选项] [文件或目录]

**选项:** ls 命令的常用选项如表 5.8 所示。

表 5.8 ls 命令的常用选项

选 项	作 用
-a	显示所有文件,包括隐藏文件(以“.”开头的文件和目录是隐藏的),还包括本级目录“.”和上一级目录“..”
-A	显示所有文件,包括隐藏文件,但不列出“.”和“..”
-b	显示当前工作目录下的目录
-l	使用长格式显示文件的详细信息,包括文件状态、权限、拥有者,以及文件大小和文件名等
-F	附加文件类别,符号在文件名最后
-d	如果参数是目录,只显示其名称而不显示其下的各个文件
-t	将文件按照建立时间的先后次序列出
-r	将文件以相反次序显示(默认按英文字母顺序排序)
-R	递归显示目录,若目录下有文件,则以下的文件也会被依序列出
-i	显示文件的 inode(索引节点)信息

例 5.16 -l 使用长格式显示文件的详细信息,包括文件状态、权限、拥有者,以及文件大小和文件名等,如图 5.49 所示。

\$ ls -l work                      将 work 目录下的详细信息列出来。

```
malimei@malimei:~/Documents$ ls -l work
total 32
-rw-rw-r-- 1 malimei malimei 5743 Nov 10 05:45 English
-rw-rw-r-- 1 malimei malimei 23 Nov 10 08:22 text0
-rw-rw-r-- 1 malimei malimei 6 Nov 9 16:11 text1
-rw-rw-r-- 1 malimei malimei 8 Nov 9 16:12 text2
-rw-rw-r-- 1 malimei malimei 9 Nov 9 16:12 text3
-rw-rw-r-- 1 malimei malimei 30 Nov 10 08:24 text4
-rw-rw-r-- 1 malimei malimei 44 Nov 10 08:11 textfile
```

图 5.49 查看详细信息

选项-l 会显示文件的详细信息,各列的含义如下。

第 1 列表示是文件还是目录(d 开头的为目录)。

第 2 列表示如果是目录,则是目录下的子目录和文件数目;如果是文件,则是文件的数目或文件的链接数。

第 3 列表示文件的所有者名字。

第 4 列表示所属的组名字。

第 5 列表示文件的字节数。

第 6~8 列表示上一次修改的时间。

第 9 列表示文件名。

**例 5.17** 带参数显示目录下的内容,如图 5.50 所示。

<code>\$ ls -a</code>	列出当前目录下的所有文件和目录,包括隐藏文件。
<code>\$ ls -A</code>	列出当前目录下的所有文件和目录,包括隐藏文件,不包括.和..。
<code>\$ ls -R</code>	递归显示,若目录下还有目录,则其下的文件也会被依序列出。
<code>\$ ls -R -l</code>	以递归的形式显示当前目录下文件或目录的详细信息。
<code>\$ ls Documents</code>	列出/Documents 下的文件和目录。
<code>\$ ls -F Documents</code>	列出/Documents 下的文件和目录,在目录后加斜线以区分。

```
malimei@malimei:~$ ls
a.txt Documents
malimei@malimei:~$ ls -a
.  .. a.txt .bash_history .bash_logout .bashrc Documents .profile
malimei@malimei:~$ ls -A
a.txt .bash_history .bash_logout .bashrc Documents .profile
malimei@malimei:~$ ls -l
总用量 4
-rw-rw-r-- 1 malimei malimei  0 Oct 31 05:13 a.txt
drwxrwxr-x 2 malimei malimei 4096 Oct 31 03:14 Documents
malimei@malimei:~$ ls -R
.:
a.txt Documents
./Documents:
test1 test2 test3
malimei@malimei:~$ ls -R -l
.:
总用量 4
-rw-rw-r-- 1 malimei malimei  0 Oct 31 05:13 a.txt
drwxrwxr-x 2 malimei malimei 4096 Oct 31 03:14 Documents
./Documents:
总用量 12
-rw-rw-r-- 1 malimei malimei 28 Oct 30 08:49 test1
-rw-rw-r-- 1 malimei malimei 56 Oct 31 03:11 test2
-rw-rw-r-- 1 malimei malimei 112 Oct 31 03:15 test3
malimei@malimei:~$ ls Documents
English~ text1~ text2~ text3~ text4~ textfile~ work
malimei@malimei:~$ ls -F Documents
English~ text1~ text2~ text3~ text4~ textfile~ work/
```

图 5.50 带参数查看当前目录下的内容

**例 5.18** /bin 目录下显示的是 Linux 的命令,命令包括内部命令和外部命令。在/bin 下显示文件名的是外部命令,没有显示的是内部命令。如图 5.51 所示,pwd、ls、touch 是外部命令,cd 是内部命令。

```

dbus-uuidgen      lsblk             plymouth          whiptail
dd                lsmod            plymouth-upstart-bridge  ypdomainname
df               mkdir            ps                zcat
dir              mknod           pwd               zcmp
dmesg            mktemp          rbash            zdiff
dnsdomainname    more            readlink         zegrep
domainname       mount           red              zfgrep
dumpkeys         mountpoint      rm               zforce
echo             mt              rmdir            zgrep
ed              mt-gnu          rnano            zless
egrep            mv              running-in-container  zmore
false           nano            run-parts        znew
malimei@malimei-virtual-machine:/bin$ ls pwd
pwd
malimei@malimei-virtual-machine:/bin$ ls cd
ls: 无法访问cd: 没有那个文件或目录
malimei@malimei-virtual-machine:/bin$ ls ls
ls
malimei@malimei-virtual-machine:/bin$ ls touch
touch
malimei@malimei-virtual-machine:/bin$ cd -
malimei@malimei-virtual-machine:~$ pwd
/home/malimei
malimei@malimei-virtual-machine:~$

```

图 5.51 查看/bin 下的命令文件

例 5.19 显示文件的 inode 索引节点的信息,如图 5.52 所示。

\$ ls -il 显示当前目录下每个文件的索引节点号。

```

malimei@malimei-virtual-machine:~$ ls -il
总用量 76
134885 -rw-rw-r-- 1 malimei malimei 4 12月 17 16:33 a
134896 lrwxrwxrwx 1 malimei malimei 1 12月 17 16:47 a1 -> a
134909 drwxrwxr-x 2 malimei malimei 4096 3月 31 15:20 c1
134923 -rw-rw-r-- 1 malimei malimei 344 3月 31 15:50 di.tar
131421 drwxrwxr-x 5 malimei malimei 4096 3月 31 16:13 d3
129962 -rw-rw-r-- 1 malimei malimei 0 7月 19 2019 dd
129794 -rw-r--r-- 1 malimei malimei 8980 7月 19 2019 examples.desktop
134891 lrwxrwxrwx 1 malimei malimei 1 12月 17 16:38 rss -> a
130286 drwxrwxrwx 2 malimei malimei 4096 12月 10 16:21 
134890 drwxrwxr-x 2 malimei malimei 4096 12月 17 16:48 ss
130027 drwxrwxr-x 3 malimei malimei 4096 7月 20 2019 vmware
134924 -rw-rw-r-- 1 malimei malimei 1774 3月 31 15:51 zdi.zip
129807 drwxr-xr-x 2 malimei malimei 4096 7月 19 2019 公共的
129806 drwxr-xr-x 2 malimei malimei 4096 7月 19 2019 模板
129811 drwxr-xr-x 2 malimei malimei 4096 7月 19 2019 视频
129810 drwxr-xr-x 2 malimei malimei 4096 4月 3 16:49 图片
129808 drwxr-xr-x 2 malimei malimei 4096 7月 19 2019 文档
129805 drwxr-xr-x 2 malimei malimei 4096 7月 19 2019 下载
129809 drwxr-xr-x 2 malimei malimei 4096 7月 19 2019 音乐
129804 drwxr-xr-x 2 malimei malimei 4096 7月 21 2019 桌面

```

图 5.52 查看文件的信息及索引节点号

这里所说的 inode,中文译名为“索引节点”,与文件的存储有关。

文件存储在硬盘上,硬盘的最小存储单位叫作“扇区”(sector)。每个扇区存储 512B(相当于 0.5KB)。操作系统读取硬盘的时候,不会一个个扇区地读取,这样效率太低,而是一次性连续读取多个扇区,即一次性读取一个“块”(block)。这种由多个扇区组成的“块”,是

文件存取的最小单位。“块”的大小,最常见的是 4KB,即连续 8 个 sector 组成一个 block。文件数据都存储在“块”中,那么很显然,我们还必须找到一个地方存储文件的元信息,比如文件的创建者、文件的创建日期、文件的大小等。这种存储文件元信息的区域就叫作 inode, ls -li 则显示了文件的这一信息。

inode 包含文件的元信息,具体来说有以下内容。

- (1) 文件的字节数。
- (2) 文件拥有者的 User ID。
- (3) 文件的 Group ID。
- (4) 文件的读、写、执行权限。
- (5) 文件的时间,共有三个: ctime 指 inode 上一次变动的时间,mtime 指文件内容上一次变动的时间,atime 指文件上一次打开的时间。
- (6) 链接数,即有多少文件名指向这个 inode。
- (7) 文件数据 block 的位置。

可以用 stat 命令,查看某个文件的 inode 信息,如图 5.53 所示。总之,除了文件名以外的所有文件信息,都存在 inode 之中。

```
malimei@malimei:~/Documents$ stat test3
文件: "test3"
大小: 112          块: 8          IO 块: 4096   普通文件
设备: 801h/2049d   Inode: 264254  硬链接: 1
权限: (0664/-rw-rw-r--) Uid: ( 1000/ malimei)  Gid: ( 1002/ malimei)
最近访问: 2015-10-31 03:18:46.620786704 -0700
最近更改: 2015-10-31 03:15:28.252795191 -0700
最近改动: 2015-10-31 03:15:28.252795191 -0700
创建时间: -
malimei@malimei:~/Documents$
```

图 5.53 查看文件的 indoe 信息

## 2. pwd

**功能描述:** 显示当前工作目录的完整路径。

**语法:** pwd

**选项:** pwd 的常用选项如表 5.9 所示。

表 5.9 pwd 的常用命令选项

选 项	作 用
-P	如果目录是链接时,显示出实际路径,而非使用链接(link)路径

**例 5.20** 查看当前工作路径,如图 5.54 所示。

```
$ pwd          查看默认工作目录的完整路径。
```

## 3. cd

**功能描述:** 改变当前工作目录。把希望进入的目录名称作为参数,从而在目录间进行移动,目录名称可以是工作目录下的子目录名称,也可以是系统中任何目录的全路径名。想要回到主目录,只需要直接输入 cd 或 cd~。

**语法:** cd [目录]

```
malimei@malimei-virtual-machine:~$ pwd
/home/malimei
malimei@malimei-virtual-machine:~$ cd ..
malimei@malimei-virtual-machine:/home$ pwd
/home
malimei@malimei-virtual-machine:/home$ cd /home/user1
malimei@malimei-virtual-machine:/home/user1$ pwd
/home/user1
malimei@malimei-virtual-machine:/home/user1$
```

图 5.54 查看当前目录的路径

**例 5.21** 回到上一级目录,如图 5.55 所示。

`$ cd ..` 返回上一级路径。

```
malimei@malimei:~$ pwd
/home/malimei
malimei@malimei:~$ cd ..
malimei@malimei:/home$ pwd
/home
malimei@malimei:/home$ cd ..
malimei@malimei:/$ pwd
/
```

图 5.55 返回上一级路径

**例 5.22** 切换到用户的主目录,如图 5.56 所示。

`$ cd ~` 回到用户的主目录。

```
malimei@malimei-virtual-machine:/home$ cd /home/user1
malimei@malimei-virtual-machine:/home/user1$ pwd
/home/user1
malimei@malimei-virtual-machine:/home/user1$ cd ~
malimei@malimei-virtual-machine:~$ pwd
/home/malimei
malimei@malimei-virtual-machine:~$
```

图 5.56 切换当前目录

### 5.2.3 文件创建、删除命令

#### 1. touch

**功能描述:** 生成空文件和修改文件存取时间。当执行了 touch 命令后,文件的创建时间或修改时间会更新为当前系统的时间,如果文件不存在,就会自动添加一个空文件。

**语法:** touch [选项] [文件名]

**选项:** touch 命令的常用选项如表 5.10 所示。

表 5.10 touch 命令的常用选项

选 项	作 用
-d	以 yyyyymmdd 的形式给出要修改的时间,而非现在的时间
-a	只更改存取时间
-c	不建立任何文档
-f	此参数将忽略不予处理,仅负责解决 BSD 版本指令的兼容性问题
-m	只更改变动时间
-r	把指定文档或目录的日期时间设成参考文档或目录的日期时间

例 5.23 创建新的空文件,如图 5.57 所示。

\$ touch t1 t2 创建新的空文件 t1 和 t2。

```
malimei@malimei-virtual-machine:~/ss$ touch t1 t2
malimei@malimei-virtual-machine:~/ss$ ls -l
总用量 4
-rw-rw-r-- 1 malimei malimei 4 12月 17 16:48 a
lrwxrwxrwx 1 malimei malimei 1 12月 17 16:48 a1 -> a
-rw-rw-r-- 1 malimei malimei 0 5月 13 16:10 t1
-rw-rw-r-- 1 malimei malimei 0 5月 13 16:10 t2
malimei@malimei-virtual-machine:~/ss$
```

图 5.57 创建新的空文件

例 5.24 修改文件的时间,如图 5.58 所示。

\$ touch -r a t1 将 t1 的文件时间更改为和文件 a 一样的文件时间。

```
malimei@malimei-virtual-machine:~/ss$ touch -r a t1
malimei@malimei-virtual-machine:~/ss$ ls -l
总用量 4
-rw-rw-r-- 1 malimei malimei 4 12月 17 16:48 a
lrwxrwxrwx 1 malimei malimei 1 12月 17 16:48 a1 -> a
-rw-rw-r-- 1 malimei malimei 0 12月 17 16:48 t1
-rw-rw-r-- 1 malimei malimei 0 5月 13 16:10 t2
malimei@malimei-virtual-machine:~/ss$
```

图 5.58 修改文件的时间

例 5.25 指定修改文件的时间,如图 5.59 所示。

\$ touch -d time filename 将 filename 的文件时间更改为指定的时间。

```
malimei@malimei-virtual-machine:~$ ls --full-time a
-rw-rw-r-- 1 malimei malimei 4 2019-12-17 16:33:57.681220390 +0800 a
malimei@malimei-virtual-machine:~$ touch -d "2020-08-20 20:25:30" a
malimei@malimei-virtual-machine:~$ ls --full-time a
-rw-rw-r-- 1 malimei malimei 4 2020-08-20 20:25:30.000000000 +0800 a
malimei@malimei-virtual-machine:~$
```

图 5.59 指定修改文件的时间

## 2. rm

**功能描述:** 删除一个目录中的若干个文件或子目录。在默认情况下,rm 命令只能删除指定的文件,而不能删除目录,如果删除目录必须加参数-r。

**注意:** 一旦用命令删除文件,不容易恢复。

**语法:** rm [选项] [文件或目录]

**选项:** rm 命令的常用选项如表 5.11 所示。

表 5.11 rm 命令的常用选项

选 项	作 用
-f	强制删除。忽略不存在的文件,不提示确认
-i	在删除前会有提示,需要确认

选 项	作 用
-I	在删除超过 3 个文件时或在递归删除前需要确认
-r(R)	递归删除目录及其内容(无该选项时只删除文件)

例 5.26 删除文件,删除前确认,如图 5.60 所示。

\$ rm -i \*.doc                      删除所有.doc 文件,执行前系统会先询问是否删除。

```
malimei@malimei:~/Documents/test$ ll
total 8
drwxrwxr-x 2 malimei malimei 4096 Nov 10 17:48 ./
drwxr-xr-x 4 malimei malimei 4096 Nov 10 17:44 ../
-rw-rw-r-- 1 malimei malimei 0 Nov 11 2015 touch
-rw-rw-r-- 1 malimei malimei 0 Nov 10 17:45 touch1.doc
-rw-rw-r-- 1 malimei malimei 0 Nov 10 17:48 touch2.doc
malimei@malimei:~/Documents/test$ rm -i *.doc
rm: remove regular empty file 'touch1.doc'? y
rm: remove regular empty file 'touch2.doc'? y
malimei@malimei:~/Documents/test$ ll
total 8
drwxrwxr-x 2 malimei malimei 4096 Nov 10 17:51 ./
drwxr-xr-x 4 malimei malimei 4096 Nov 10 17:44 ../
-rw-rw-r-- 1 malimei malimei 0 Nov 11 2015 touch
```

图 5.60 删除文件前确认

例 5.27 删除目录及其下的子目录和文件,如图 5.61 所示。

\$ rm -ri documents                      删除目录 documents 及其下的子目录 d1 和 d1 目录中的文件。

```
malimet@malimei-virtual-machine:~$ cd /home/malimei/documents
malimei@malimei-virtual-machine:~/documents$ ls -l
总用量 8
drwxrwxr-x 2 malimei malimei 4096 8月 15 17:24 d1
-rw-rw-r-- 1 malimei malimei 29 7月 20 00:00 test1
malimet@malimei-virtual-machine:~/documents$ cd d1
malimet@malimei-virtual-machine:~/documents/d1$ ls -l
总用量 0
-rw-rw-r-- 1 malimei malimei 0 8月 15 17:24 file1
malimet@malimei-virtual-machine:~/documents/d1$ cd -
malimet@malimei-virtual-machine:~$ rm -ri documents
rm: 是否进入目录'documents'? y
rm: 是否进入目录'documents/d1'? y
rm: 是否删除普通空文件'documents/d1/file1'? y
rm: 是否删除目录'documents/d1'? y
rm: 是否删除普通文件'documents/test1'? y
rm: 是否删除目录'documents'? y
malimet@malimei-virtual-machine:~$ cd documents
bash: cd: documents: 没有那个文件或目录
malimei@malimei-virtual-machine:~$
```

图 5.61 删除目录及其下子目录和文件

例 5.28 I 和 i 的区别。I 删除子目录前一次性确认,i 删除子目录前逐个确认,如图 5.62 所示。

\$ rm -ri cc                              逐个确认删除 cc 子目录。  
 \$ rm -rI cc                              一次性确认删除 cc 子目录。

```

malimei@malimei-virtual-machine:~$ cd cc
malimei@malimei-virtual-machine:~/cc$ ls
cc
malimei@malimei-virtual-machine:~/cc$ cd ..
malimei@malimei-virtual-machine:~$ rm -ri cc
rm: 是否进入目录"cc"? y
rm: 是否进入目录"cc/cc"? y
rm: 是否删除目录"cc/cc/[c1,c2]"? y
rm: 是否删除目录"cc/cc"? y
rm: 是否删除目录"cc"? y
malimei@malimei-virtual-machine:~$ cd ww
malimei@malimei-virtual-machine:~/ww$ ls
malimei@malimei-virtual-machine:~/ww$ mkdir bb
malimei@malimei-virtual-machine:~/ww$ cd bb
malimei@malimei-virtual-machine:~/ww/bb$ mkdir cc
malimei@malimei-virtual-machine:~/ww/bb$ cd ..
malimei@malimei-virtual-machine:~/ww$ cd ..
malimei@malimei-virtual-machine:~$ rm -rI ww
rm: 递归删除所有参数? y
malimei@malimei-virtual-machine:~$

```

图 5.62 I 和 i 的区别



## 5.2.4 目录创建、删除命令

### 1. mkdir

**功能描述：**mkdir 命令用来创建指定名称的目录，要求创建目录的用户在当前目录中具有写权限，并且指定的目录名不能是当前目录中已有的目录。

**语法：**mkdir [选项] [目录名]

**选项：**mkdir 命令的常用选项如表 5.12 所示。

表 5.12 mkdir 命令的常用选项

选 项	作 用
-p	依次创建目录,需要时创建目标目录的上级目录
-m	设置权限模式,在建立目录时按模式指定设置目录权限
-v	每次创建新目录都显示执行过程信息

其中的-m 选项用来设置目录的权限。对目录的读权限是 4、写权限是 2、执行权限是 1，这三个数字的和表达了对该目录的权限，如 7 代表同时具有读、写和执行权限，6 代表具有读和写的权限，4 则代表只有读的权限。

-m 的格式为 mkdir -m [参数] [目录名]，这里的参数由三位如上所述的数字组成，分别代表目录所有者的权限、组中其他人对目录的权限和系统中其他人对目录的权限。常用的组合如表 5.13 所示。

表 5.13 -m 参数的含义

参 数	含 义
600	只有所有者有读和写的权限
644	所有者有读和写的权限,组用户只有读的权限
666	每个人都有读和写的权限
700	只有所有者有读和写以及执行的权限
777	每个人都有读和写以及执行的权限

例 5.29 创建目录,如图 5.63 所示。

\$ mkdir c-language 在当前目录下创建子目录 c-language。

126

```
malimei@malimei:~$ ls
a.txt  Desktop  Downloads  Pictures  Templates
malimei@malimei:~$ mkdir c-language
malimei@malimei:~$ ls
a.txt  Desktop  Downloads  Pictures  Templates
c-language  Documents  Music  Public  Videos
malimei@malimei:~$ cd c-language
malimei@malimei:~/c-language$
```

图 5.63 创建目录

例 5.30 依次创建目录,如果上级目录不存在,则同时创建上级目录,如图 5.64 所示。

\$ mkdir -p aaa/test 创建子目录 aaa,并在其下建立子目录 test。

```
malimei@malimei:~$ ls
a.txt  Desktop  Downloads  Pictures  Templates
c-language  Documents  Music  Public  Videos
malimei@malimei:~$ mkdir aaa/test
mkdir: cannot create directory 'aaa/test': No such file or directory
malimei@malimei:~$ mkdir -p aaa/test
malimei@malimei:~$ ls
aaa  c-language  Documents  Music  Public  Videos
a.txt  Desktop  Downloads  Pictures  Templates
malimei@malimei:~$ cd aaa
malimei@malimei:~/aaa$ ls
test
```

图 5.64 创建目标及父目录

例 5.31 同时创建多个目录,如图 5.65 所示。

\$ mkdir -vp scf/{lib/,bin/,doc/{info,product}}

创建目录 scf; scf 下创建目录 lib,bin,doc; doc 下创建目录 info,product,并显示过程。

```
malimei@malimei-virtual-machine:~$ mkdir -vp scf/{lib/,bin/,doc/{info,product}}
mkdir: 已创建目录 'scf'
mkdir: 已创建目录 'scf/lib/'
mkdir: 已创建目录 'scf/bin/'
mkdir: 已创建目录 'scf/doc'
mkdir: 已创建目录 'scf/doc/info'
mkdir: 已创建目录 'scf/doc/product'
malimei@malimei-virtual-machine:~$ ls -R
.:
dd          scf      公共的  视频  文档  音乐
examples.desktop  vmware  模板   图片  下载  桌面

./scf:
bin doc lib

./scf/bin:

./scf/doc:
info product

./scf/doc/info:

./scf/doc/product:
```

图 5.65 逐层创建、查看创建的多级目录

例 5.32 创建新目录,同时设置访问权限,如图 5.66 所示。

\$ mkdir -m 777 test4 创建目录 test4,每个人对该目录都有读、写和执行的权限。

```
malimei@malimei:~/Documents/test$ mkdir -m 777 test4
malimei@malimei:~/Documents/test$ ll
total 20
drwxrwxr-x 5 malimei malimei 4096 Nov 10 17:55 ./
drwxr-xr-x 4 malimei malimei 4096 Nov 10 17:44 ../
drwxrwxr-x 2 malimei malimei 4096 Nov 10 17:53 test1/
drwxrwxr-x 3 malimei malimei 4096 Nov 10 17:54 test2/
drwxrwxrwx 2 malimei malimei 4096 Nov 10 17:55 test4/
-rw-rw-r-- 1 malimei malimei 0 Nov 11 2015 touch
```

图 5.66 创建新目录并设置权限

例 5.33 只能在自己的工作目录下建立子目录,否则必须具有超级用户权限,如图 5.67 所示。

\$ mkdir -m 700 d11 在根目录下创建子目录 d11,只有超级用户才可以建立。

```
malimei@malimei-virtual-machine:~$ pwd
/home/malimei
malimei@malimei-virtual-machine:~$ mkdir -m 777 d11
malimei@malimei-virtual-machine:~$ cd /
malimei@malimei-virtual-machine:/$ mkdir -m 700 d11
mkdir: 无法创建目录"d11": 权限不够
malimei@malimei-virtual-machine:/$ sudo mkdir -m 700 d11
[sudo] malimei 的密码:
malimei@malimei-virtual-machine:/$
```

图 5.67 root 权限创建目录

## 2. rmdir

**功能描述:** 删除空目录。在操作系统中,有时会出现比较多的空目录,这是可以使用目录删除命令 rmdir 将它们都删除。rmdir 命令只能删除空目录,如果有文件需要先删除文件。

**语法:** rmdir [选项] [目录列表]

**选项:** rmdir 命令的常用选项如表 5.14 所示。

表 5.14 rmdir 命令的常用选项

选 项	作 用
-p	当子目录被删除后其父目录为空目录时,也一起被删除
-v	显示详细的进行步骤

可使用空格来分隔多个目录名(称为目录列表),同时删除多个目录。注意:要删除的目录必须为空,如果其下有文件,需要先将文件删除。

例 5.34 删除目录,如图 5.68 所示。

\$ rmdir c - language 删除当前目录下的子目录 c - language。

例 5.35 删除目标目录,删除后如果上级目录成为空目录,则同时删除,如图 5.69 所示。

\$ rmdir -p aaa/test 删除目标目录 test 和上级目录 aaa(aaa 下只有 test 一个子目录)。

```
malimei@malimei:~$ ls
aaa  c-language  Documents  Music      Public     Videos
a.txt Desktop     Downloads  Pictures   Templates
malimei@malimei:~$ rmdir c-language
malimei@malimei:~$ ls
aaa  Desktop     Downloads  Pictures   Templates
a.txt Documents  Music      Public     Videos
malimei@malimei:~$
```

图 5.68 删除目录

```
malimei@malimei:~$ ls
aaa  Desktop     Downloads  Pictures   Templates
a.txt Documents  Music      Public     Videos
malimei@malimei:~$ cd aaa
malimei@malimei:~/aaa$ ls
test
malimei@malimei:~/aaa$ cd
malimei@malimei:~$ rmdir -p aaa/test
malimei@malimei:~$ ls
a.txt  Documents  Music      Public     Videos
Desktop Downloads  Pictures   Templates
malimei@malimei:~$
```

图 5.69 删除目标及上级目录

**例 5.36** 删除带文件目录,如图 5.70 所示。

```
$ rm *           删除目录 test2 下的所有文件。
$ rmdir test2   删除目录 test2。
```

```
malimei@malimei-virtual-machine:/home$ sudo rmdir test2
rmdir: 删除 "test2" 失败: 目录非空
malimei@malimei-virtual-machine:/home$ cd test2
malimei@malimei-virtual-machine:/home/test2$ ls
wj  wj1  wj1~  wj21
malimei@malimei-virtual-machine:/home/test2$ sudo rm *.*
rm: 无法删除"*.*": 没有那个文件或目录
malimei@malimei-virtual-machine:/home/test2$ sudo rm *
malimei@malimei-virtual-machine:/home/test2$ ls
malimei@malimei-virtual-machine:/home/test2$ cd ..
malimei@malimei-virtual-machine:/home$ sudo rmdir test2
malimei@malimei-virtual-machine:/home$
```

图 5.70 先删文件后删目录

从图 5.70 中可以看出,rmdir 只能删除空白目录,如果目录不空(如 test2),则不能删除,需要先删除文件再删除目录。

**例 5.37** 显示删除的详细过程,如图 5.71 所示。

```
malimei@malimei:~/Documents/test$ rmdir -v test2
rmdir: removing directory, 'test2'
rmdir: failed to remove 'test2': Directory not empty
malimei@malimei:~/Documents/test$ rmdir -v test1
rmdir: removing directory, 'test1'
malimei@malimei:~/Documents/test$ ll
total 20
drwxrwxr-x 5 malimei malimei 4096 Nov 10 18:02 ./
drwxr-xr-x 4 malimei malimei 4096 Nov 10 17:44 ../
drwxrwxr-x 5 malimei malimei 4096 Nov 10 18:00 scf/
drwxrwxr-x 3 malimei malimei 4096 Nov 10 17:54 test2/
drwxrwxrwx 2 malimei malimei 4096 Nov 10 17:55 [redacted]/
-rw-rw-r-- 1 malimei malimei 0 Nov 11 2015 touch
```

图 5.71 删除目录并显示过程

```
$ rmdir -v test2      删除目录 test2 并显示过程(test2 不是空白目录不能删除)
$ rmdir -v test1      删除目录 test1 并显示详细步骤
```

## 5.2.5 复制、移动命令

### 1. cp

**功能描述：**将文件或目录复制到另一个文件或目录中。如同时指定两个以上的文件或目录，且最后的目的地是一个已经存在的目录，则它会把前面指定的文件或目录复制到此目录中。若同时指定多个文件或目录，而最后的目的地并非一个已存在的目录，则会出现错误信息。

**语法：**cp [选项] [源文件或目录] [目的文件或目录]

cp [选项] 源文件组 目标目录

cp 命令可以复制多个文件，将要复制的多个文件用空格分隔，所形成的列表称为源文件组。

**选项：**cp 命令的常用选项如表 5.15 所示。

表 5.15 cp 命令的常用选项

选 项	作 用
-b	将要覆盖的文件做备份，但不接受参数递归时特殊文本的副本内容
-i	覆盖前查询，提示是否覆盖已存在的目标文件
-f	强制复制文件，若目标文件无法打开则将其移除并重试
-p	保留源文件或目录的属性，如日期
-R	复制所有文件及目录
-a	不进行文件数据复制，只对每一个现有目标文件的属性进行备份
-H	跟踪源文件中的命令行符号链接
-l	链接文件而不复制
-L	总是跟随源文件中的符号链接
-n	不要覆盖已存在的文件
-P	不跟随源文件中的符号链接
-s	只创建符号链接而不复制文件
-t	将所有参数指定的源文件/目录复制到目标目录下
-T	将目标目录视为普通文件
-u	只在源文件比目标文件新或目标文件不存在时才进行复制
-v	显示详细的进行步骤
-x	不跨越文件系统进行操作

**例 5.38** 复制文件，如图 5.72 所示。

```
$ cp test1/file1 test2      将 test1 文件夹下的 file1 文件复制到目录 test2 中。
```

**例 5.39** 复制并备份已有文件，如图 5.73 所示。

```
$ cp -i a1 a2      复制文件 a1 为 a2, 如果文件 a2 存在, 则询问是否覆盖 a2。
$ cp -b a1 a2      复制文件 a1 为 a2, 若 a2 存在, 则将 a2 备份为 a2~。
```



表 5.16 mv 命令的常用选项

选 项	作 用
-f	禁止交互模式,本选项会使 mv 命令执行移动而不给出提示(在权限足够的情况下直接执行;如果目标文件存在但用户没有写权限时,mv 会给出提示)
-i	交互模式,当移动的目录已存在同名的目标文件名时,用覆盖方式写文件,但在写入之前系统会询问用户是否重写,要求用户回答 y 或者 n,这样可以避免误覆盖文件
-n	不要覆盖已存在的文件
-u	只在源文件比目标文件新或者目标文件不存在时才进行移动
-v	显示详细的进行步骤

例 5.41 移动文件,如图 5.75 所示。

\$ mv -v test1/file2 test2 将 test1 目录中的 file2 文件移动到目录 test2 中。

```
malimei@malimei:~/Documents/test$ ls test1
file1 file2
malimei@malimei:~/Documents/test$ ls test2
file1 test3
malimei@malimei:~/Documents/test$ mv -v test1/file2 test2
'test1/file2' -> 'test2/file2'
malimei@malimei:~/Documents/test$ ls test1
file1
malimei@malimei:~/Documents/test$ ls test2
file1 file2 test3
```

图 5.75 移动文件

例 5.42 更改文件名字,如图 5.76 所示。

\$ mv 1.txt 11.txt 将文件 1.txt 更名为 11.txt。  
 \$ mv 2.txt 11.txt 将文件 2.txt 更名为 11.txt,原 11.txt 被覆盖。  
 \$ mv -i 3.txt 11.txt 将文件 3.txt 更名为 11.txt,覆盖原 11.txt 之前询问。

```
malimei@malimei-virtual-machine:~$ touch 1.txt
malimei@malimei-virtual-machine:~$ touch 2.txt
malimei@malimei-virtual-machine:~$ touch 3.txt
malimei@malimei-virtual-machine:~$ mv 1.txt 11.txt
malimei@malimei-virtual-machine:~$ mv 2.txt 11.txt
malimei@malimei-virtual-machine:~$ mv -i 3.txt 11.txt
mv: 是否覆盖'11.txt'? y
malimei@malimei-virtual-machine:~$ ls -l
总用量 64
-rw-rw-r-- 1 malimei malimei 0 8月 15 18:12 11.txt
drwx----- 2 malimei malimei 4096 8月 15 17:52 cc
```

图 5.76 移动并覆盖同名文件

## 5.2.6 压缩、备份命令

### 1. tar 压缩、解压缩

**功能描述:** tar 命令是在 Ubuntu 中广泛应用的压缩解压命令,可以把许多文件打包成为一个归档文件或者把它们写入备份文件。tar 可以对文件和目录进行打包,能支持的格式为 tar、gz 等。

**语法:** tar [选项] [目标文件名] [源文件名]

选项: tar 命令的选项如表 5.17 所示。

表 5.17 tar 命令的选项

选 项	作 用
-z	使用 gzip 或者 gunzip 压缩格式处理备份文件。如果配合选项 c 使用是压缩,配合选项 x 使用是解压缩
-c	创建一个新的压缩文件,格式为 .tar
-v	显示过程
-f	指定压缩后的文件名
-x	从压缩文件中还原文件
-u	仅转换比压缩文件新的内容
-r	新增文件至已存在的压缩文件中结尾部分

例 5.43 压缩小文件,如图 5.77 所示。

\$ tar -czf 111.tar 11.txt 将 11.txt 文件压缩成 tar 格式,并命名为 111.tar。

```
malimei@malimei-virtual-machine:~/d1$ tar -czf 111.tar 11.txt
malimei@malimei-virtual-machine:~/d1$ ls -l
总用量 24
-rw-rw-r-- 1 malimei malimei 147 8月 16 09:45 111.tar
-rw-rw-r-- 1 malimei malimei 10240 8月 16 09:43 11.tar
-rw-rw-r-- 1 malimei malimei 84 8月 16 09:42 11.txt
drwxrwxr-x 2 malimei malimei 4096 8月 15 17:53 ss
```

图 5.77 压缩小文件

对于很小的文件,在进行 tar 打包并压缩或者 zip 压缩时,其占用的磁盘空间会比源文件大很多(11.txt 大小为 84KB 压缩后 111.tar 为 147KB)。

例 5.44 压缩多个文件,并显示过程,如图 5.78 所示。

\$ tar -czvf doc.tar.gz \*.doc 将目录中所有 doc 文件打包成 doc.tar 后并用 gzip 压缩,生成一个 gzip 压缩过的包,命名为 doc.tar.gz,显示过程。

```
malimei@malimei:~/Documents$ tar -czvf doc.tar.gz *.doc
tar1.doc
tar2.doc
malimei@malimei:~/Documents$ ll
total 36
drwxr-xr-x 4 malimei malimei 4096 Nov 10 18:22 ./
drwxr-xr-x 16 malimei malimei 4096 Nov 10 17:33 ../
-rw-rw-r-- 1 malimei malimei 132 Nov 10 18:22 doc.tar.gz
-rw-rw-r-- 1 malimei malimei 5746 Nov 10 05:42 English-
-rw-rw-r-- 1 malimei malimei 0 Nov 10 18:21 tar1.doc
-rw-rw-r-- 1 malimei malimei 0 Nov 10 18:21 tar2.doc
```

图 5.78 压缩多个文件

例 5.45 压缩大于 500MB 的文件,如图 5.79 所示。

\$ tar -czf VM.tar VMware-workstation-full-15.1.0-13591040.exe 将大文件压缩。

压缩大文件(500MB 以上)时,压缩比例在 10%左右。

例 5.46 压缩、打包目录,如图 5.80 所示。

```
malimei@malimei-virtual-machine:~/cc$ tar -czvf VM.tar VMware-workstation-full-15.1.0-13591040.exe
VMware-workstation-full-15.1.0-13591040.exe
malimei@malimei-virtual-machine:~/cc$ ls -l
总用量 1212664
-rw-rw-r-- 1 malimei malimei      25 8月 16 09:32 cc1
drwxrwxr-x 2 malimei malimei    4096 8月 16 10:23 d1
-rw-rw-r-- 1 malimei malimei 107070761 8月 16 10:25 d11.tar
-rw-r--r-- 1 malimei malimei     8980 8月 16 09:31 examples.desktop
-rw-rw-r-- 1 malimei malimei  1584023 8月 16 10:42 u.tar
-rw-rw-r-- 1 malimei malimei 485456668 8月 16 10:59 VM.tar
-rw-rw-r-- 1 malimei malimei 107704942 8月 16 09:37 VMWARETO.TGZ
-rwxr-xr-x 1 malimei malimei 538195976 8月 16 10:54 VMware-workstation-full-15.1.0-13591040.exe
```

图 5.79 压缩大于 500MB 的文件

```
malimei@malimei-virtual-machine:~/cc$ cd d1
malimei@malimei-virtual-machine:~/cc/d1$ ls -l
总用量 105196
-rw-r--r-- 1 malimei malimei     8980 8月 16 10:23 examples.desktop
-rw-rw-r-- 1 malimei malimei 107704942 8月 16 10:23 VMWARETO.TGZ
malimei@malimei-virtual-machine:~/cc/d1$ cd ..
malimei@malimei-virtual-machine:~/cc$ ls -l
总用量 105204
-rw-rw-r-- 1 malimei malimei      25 8月 16 09:32 cc1
drwxrwxr-x 2 malimei malimei    4096 8月 16 10:23 d1
-rw-r--r-- 1 malimei malimei     8980 8月 16 09:31 examples.desktop
-rw-rw-r-- 1 malimei malimei 107704942 8月 16 09:37 VMWARETO.TGZ
malimei@malimei-virtual-machine:~/cc$ tar -czvf d11.tar d1
d1/
d1/VMWARETO.TGZ
d1/examples.desktop
malimei@malimei-virtual-machine:~/cc$ ls -l
总用量 209768
-rw-rw-r-- 1 malimei malimei      25 8月 16 09:32 cc1
drwxrwxr-x 2 malimei malimei    4096 8月 16 10:23 d1
-rw-rw-r-- 1 malimei malimei 107070761 8月 16 10:25 d11.tar
-rw-r--r-- 1 malimei malimei     8980 8月 16 09:31 examples.desktop
-rw-rw-r-- 1 malimei malimei 107704942 8月 16 09:37 VMWARETO.TGZ
malimei@malimei-virtual-machine:~/cc$
```

图 5.80 压缩、打包目录

\$ tar -czvf d11.tar d1 将目录 d1 打包压缩成 d11.tar 文件。

**例 5.47** 解压缩文件,如图 5.81 所示。

\$ tar -xvf doc.tar.gz 将压缩包 doc.tar.gz 解压缩到当前目录。

```
malimei@malimei:~/Documents$ tar -xvf doc.tar.gz
tar1.doc
tar2.doc
```

图 5.81 解压缩文件

这里需要注意,解压缩时必须先进入目标目录,然后再解压缩。如果当前目录不是目标目录,而是把目标目录写到了命令中,会出现错误,如图 5.82 所示。

一般情况下,小于 0.5MB 的文件,在进行 tar 打包并压缩或者 zip 压缩时,其占用的磁盘空间会比源文件大很多;大于 0.5MB 的文件,在进行 tar 打包并压缩或者 zip 压缩时,其占用的磁盘空间大小不变;中等文件(100MB 左右的 PDF 文件),在进行 tar 打包并压缩或

```

drwxrwxr-x 2 malimei malimei 4096 8月 16 11:14 d11
drwxrwxr-x 2 malimei malimei 4096 8月 16 11:15 d111
-rw-rw-r-- 1 malimei malimei 107070761 8月 16 10:25 d11.tar
-rw-r--r-- 1 malimei malimei 8980 8月 16 09:31 examples.desktop
-rw-rw-r-- 1 malimei malimei 1584023 8月 16 10:42 u.tar
-rw-rw-r-- 1 malimei malimei 485456668 8月 16 10:59 VM.tar
-rw-r--r-- 1 malimei malimei 1719632 8月 16 10:42 下载ubuntu.docx
malimei@malimei-virtual-machine:~/cc$ tar -xzvf ./d11 d11.tar
tar (child): ./d11: 无法 read: 是一个目录
tar (child): 处于磁带的起点, 现在退出
tar (child): Error is not recoverable: exiting now

gzip: stdin: unexpected end of file
tar: Child returned status 2
tar: Error is not recoverable: exiting now
malimei@malimei-virtual-machine:~/cc$ cd d111
malimei@malimei-virtual-machine:~/cc/d111$ tar -xzvf ../d11.tar
d1/
d1/VMWARETO.TGZ
d1/examples.desktop

```

图 5.82 加目标路径后解压缩出错

者 zip 压缩时,大约节省 20%~30%的空间;对于大文件(500MB),大约节省 10%空间,也会由于文件类型不同,压缩比例有所不同。

## 2. gzip 只压缩不打包

**功能描述:** gzip 是在 Linux 系统中经常使用的一个对文件进行压缩和解压缩的命令,用 Lempel- Ziv coding(LZ77)技术压缩文件,压缩后文件格式为 .gz,只压缩不打包。gzip 不仅可以用来压缩大的、较少使用的文件以节省磁盘空间,还可以和 tar 命令一起构成 Linux 操作系统中比较流行的压缩文件格式。据统计,gzip 命令对文本文件有 60%~70%的压缩率。

**语法:** gzip [选项] 要压缩的源文件名

**选项:** gzip 命令的常用选项如表 5.18 所示。

表 5.18 gzip 命令的常用选项

选 项	作 用
-1	数字 1,表示快速压缩
-9	9 代表最佳状况压缩,读音 nine 约等于 nice
-r	递归式地查找指定目录并压缩其中的所有文件或者是解压缩
-c	压缩结果写入标准输入,源文件保持不变
-v	对每一个压缩和解压的文件,显示文件名和压缩比
-d	解压缩指定文件
-t	测试压缩文件的完整性
-l	对每个压缩文件,显示压缩文件的大小、未压缩文件的大小、压缩比、未压缩文件的名字等详细信息

**例 5.48** 压缩、解压缩文件,如图 5.83 所示。

```
$ gzip -9 a.txt
```

以最佳压缩比压缩文件 a.txt,生成 a.txt.gz。

```
$ gzip -d a.txt.gz
```

将压缩包 a.txt.gz 解压缩到当前目录。

```
malimei@malimei:~$ ls
a.txt  Documents  Music      Public    Templates
Desktop Downloads Pictures tar.tar Videos
malimei@malimei:~$ gzip -9 a.txt
malimei@malimei:~$ ls
a.txt.gz Documents Music      Public    Templates
Desktop Downloads Pictures tar.tar Videos
malimei@malimei:~$ gzip -d a.txt.gz
malimei@malimei:~$ ls
a.txt  Documents Music      Public    Templates
Desktop Downloads Pictures tar.tar Videos
```

图 5.83 压缩、解压缩文件

例 5.49 压缩多个文件,如图 5.84 所示。

\$ gzip \* 压缩当前目录下的所有文件为.gz 文件。

```
malimei@malimei:~/Documents/work$ gzip *
malimei@malimei:~/Documents/work$ ll
total 36
drwxrwxr-x 2 malimei malimei 4096 Nov 10 18:27 ./
drwxr-xr-x 4 malimei malimei 4096 Nov 10 18:25 ../
-rw-rw-r-- 1 malimei malimei 2534 Nov 10 05:45 English.gz
-rw-rw-r-- 1 malimei malimei 49 Nov 10 08:22 text0.gz
-rw-rw-r-- 1 malimei malimei 32 Nov 9 16:11 text1.gz
-rw-rw-r-- 1 malimei malimei 34 Nov 9 16:12 text2.gz
-rw-rw-r-- 1 malimei malimei 35 Nov 9 16:12 text3.gz
-rw-rw-r-- 1 malimei malimei 56 Nov 10 08:24 text4.gz
-rw-rw-r-- 1 malimei malimei 71 Nov 10 08:11 textfile.gz
```

图 5.84 压缩多个文件

例 5.50 压缩目标目录下的文件并解压缩,如图 5.85 所示。

\$ gzip -r Documents 压缩 Documents 下的所有文件为.gz 文件。  
\$ gzip -dr Documents 解压缩 Documents 下的所有文件。

```
malimei@malimei:~$ ls Documents
a.txt scf test1 test2 test3
malimei@malimei:~$ gzip -r Documents
malimei@malimei:~$ ls
a.txt  Documents Music      Public    Templates
Desktop Downloads Pictures tar.tar Videos
malimei@malimei:~$ ls Documents
a.txt.gz scf test1.gz test2.gz test3.gz
malimei@malimei:~$ gzip -d Documents
gzip: Documents is a directory -- ignored
malimei@malimei:~$ gzip -dr Documents
malimei@malimei:~$ ls Documents
a.txt scf test1 test2 test3
```

图 5.85 解压缩目标目录下的文件

### 3. gunzip

功能描述: 解压缩以 gzip 压缩的.gz 文件。

语法: gunzip [选项] [文件或目录]

选项: gunzip 命令的常用选项如表 5.19 所示。

表 5.19 gunzip 命令的常用选项

选 项	作 用
-a	使用 ASCII 文字模式
-d	解压文件
-c	把解压后的文件输出到标准输出设备
-f	强行解压缩文件,不理睬文件名称或硬链接是否存在
-h	在线帮助
-l	列出压缩文件的相关信息
-L	显示版本与版权信息
-n	解压文件时,若压缩文件内容含有原来的文件名称及时间戳记,则将其忽略不予处理
-p	不显示警告信息
-r	递归处理,将指定目录下的所有文件及子目录一并处理
-S	更改压缩字尾字符串
-t	测试压缩文件是否正确无误
-v	显示指令执行过程

**例 5.51** 解压缩文件,如图 5.86 所示。

\$ gunzip -dr Documents                      解压缩 Documents 下的所有 .gz 文件。

```
malimei@malimei:~$ ls Documents
a.txt scf test1 test2 test3
malimei@malimei:~$ gzip -r Documents
malimei@malimei:~$ ls Documents
a.txt.gz scf test1.gz test2.gz test3.gz
malimei@malimei:~$ gunzip -d Documents
gzip: Documents is a directory -- ignored
malimei@malimei:~$ gunzip -dr Documents
malimei@malimei:~$ ls Documents
a.txt scf test1 test2 test3
```

图 5.86 解压缩指定目录下的 .gz 文件

对于文件,如果不加任何参数,gzip 是压缩,gunzip 是解压缩。

#### 4. zip 压缩打包

**功能描述:** zip 是一个压缩和归档工具,压缩文件时使用 zip 命令。会创建一个带 .zip 扩展名的 zip 文件,如果没有指定文件,则 zip 会将压缩数据输出到标准输出。

**语法:** zip [选项] 压缩后生成的目标文件名 源文件名

**选项:** zip 命令的常用选项如表 5.20 所示。

表 5.20 zip 命令的常用选项

选 项	作 用
-f	以新文件取代现有文件
-u	只更新改变过的文件和新文件
-d	从 zip 文件中移出一个文件
-m	将特定文件移入 zip 文件中,并且删除特定文件
-r	递归压缩子目录下的所有文件,包括子目录

选 项	作 用
-j	只存储文件的名称,不含目录
-l	最快压缩,压缩率最差
-9	表示最慢速度的压缩(最佳压缩),预设值为-6
-q	安静模式,不会显示相关信息和提示
-v	显示版本资讯或详细信息

**例 5.52** 压缩单个文件,如图 5.87 所示。

# zip test2.zip test2                    将文件 test2 压缩为 test2.zip。

```

root@malimei:/home/malimei/Documents/test# zip test2.zip test2
  adding: test2/ (stored 0%)
root@malimei:/home/malimei/Documents/test# ll
total 28
drwxrwxr-x 6 malimei malimei 4096 Nov 10 19:01 ./
drwxr-xr-x 4 malimei malimei 4096 Nov 10 19:00 ../
drwxrwxr-x 5 malimei malimei 4096 Nov 10 18:00 scf/
drwxrwxr-x 2 malimei malimei 4096 Nov 10 18:14 test1/
drwxrwxr-x 3 malimei malimei 4096 Nov 10 18:14 test2/
-rw-r--r-- 1 root    root      162 Nov 10 19:01 test2.zip
drwxrwxrwx 2 malimei malimei 4096 Nov 10 17:55 /
-rw-rw-r-- 1 malimei malimei   0 Nov 11 2015 touch

```

图 5.87 压缩单个文件

**例 5.53** 递归压缩,如图 5.88 所示。

# zip -r test.zip ./ \*                    将当前目录下的所有子目录和文件递归压缩为 test.zip。

```

root@malimei:/home/malimei/Documents/test# zip -r test.zip ./ *
  adding: test1/ (stored 0%)
  adding: test1/file1 (stored 0%)
  adding: test2/ (stored 0%)
  adding: test2/test3/ (stored 0%)
  adding: test2/file2 (stored 0%)
  adding: test2/file1 (stored 0%)
  adding: touch (stored 0%)
root@malimei:/home/malimei/Documents/test# ll
total 20
drwxrwxr-x 4 malimei malimei 4096 Nov 10 19:05 ./
drwxr-xr-x 4 malimei malimei 4096 Nov 10 19:00 ../
drwxrwxr-x 2 malimei malimei 4096 Nov 10 18:14 test1/
drwxrwxr-x 3 malimei malimei 4096 Nov 10 18:14 test2/
-rw-r--r-- 1 root    root     1042 Nov 10 19:05 test.zip
-rw-rw-r-- 1 malimei malimei   0 Nov 11 2015 touch

```

图 5.88 压缩当前目录下的所有文件和文件夹

**例 5.54** 删除压缩文件中的部分文件,如图 5.89 所示。

# zip -d test.zip touch                    删除压缩文件 test.zip 中的文件 touch。  
# zip -d test.zip test2/file2              删除压缩文件 test.zip 中的 test2 目录下的 file2 文件。

**例 5.55** 向压缩文件中添加文件,如图 5.90 所示。

# zip -m test.zip touch                    向压缩文件 test.zip 中添加 touch 文件。

```
root@malimei:/home/malimei/Documents/test# zip -d test.zip touch
deleting: touch
root@malimei:/home/malimei/Documents/test# zip -d test.zip test2/file2
deleting: test2/file2
```

图 5.89 删除压缩包中的部分文件

```
root@malimei:/home/malimei/Documents/test# zip -m test.zip touch
adding: touch (stored 0%)
```

图 5.90 添加文件到压缩包

**例 5.56** 压缩文件时排除某个文件,如图 5.91 所示。

# zip test.zip /\* -x test2/file1 压缩当前目录下所有文件,除了 test2/file1 文件。

```
root@malimei:/home/malimei/Documents/test# zip test.zip /* -x test2/file1
adding: test1/ (stored 0%)
adding: test2/ (stored 0%)
```

图 5.91 设置压缩范围

### 5. unzip

**功能描述:** 解压缩 zip 文件。

**语法:** unzip [选项] 压缩文件名

**选项:** unzip 命令的常用选项见表 5.21。

表 5.21 unzip 命令的常用选项

选 项	作 用
-x	“文件列表”解压文件,但不包含文件列表中指定的文件
-t	测试压缩文件有无损坏,并不解压
-v	查看压缩文件的详细信息,具体包括压缩文件中包含的文件大小、文件名和压缩比等,并不解压
-n	解压时不覆盖已经存在的文件
-o	解压时覆盖已经存在的文件,并且不要求用户确认
-d	按目录名把压缩文件解压到指定目录下

**例 5.57** 解压缩文件,如图 5.92 所示。

# unzip test.zip 将 test.zip 压缩文件直接解压到当前目录。

```
root@malimei:/home/malimei/Documents/test# unzip test.zip
Archive: test.zip
replace test2/file1? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
extracting: test2/file1
```

图 5.92 解压缩文件

**例 5.58** 解压缩文件到指定目录,如图 5.93 所示。

# unzip -n test.zip -d /home/malimei/Documents 将 test.zip 压缩文件解压到目录/home/malimei/Documents 下。

```
root@malimei:/home/malimei/Documents/test# unzip -n test.zip -d /home/malimei/Do
cuments
Archive: test.zip
  extracting: /home/malimei/Documents/test2/file1
   creating: /home/malimei/Documents/test1/
```

图 5.93 解压缩文件到指定目录

与 tar 命令不同,在 unzip 命令后面添加-d 就可以指定目标目录,而不需要必须进入目标目录再解压缩。

**例 5.59** 解压缩并覆盖已有文件,如图 5.94 所示。

\$ unzip -o test.zip -d /home/malimei/Documents 将压缩文件 test.zip 解压到目录/home/malimei/Documents 下,如有相同文件则覆盖。

```
root@malimei:/home/malimei/Documents/test# unzip -o test.zip -d /home/malimei/Do
cuments
Archive: test.zip
  extracting: /home/malimei/Documents/test2/file1
```

图 5.94 解压缩并覆盖已有文件

**例 5.60** 查看压缩文件,如图 5.95 所示。

\$ unzip -v cc1.zip 查看压缩文件 cc1.zip 的详细信息。

```
malimei@malimei-virtual-machine:~/cc$ unzip -v cc1.zip
Archive: cc1.zip
 Length Method      Size Cmpr   Date   Time   CRC-32   Name
-----
  25 Defl:N          24   4% 2019-08-16 09:32 6361b56e cc1
-----
  25          24   4%
-----
                                1 file
malimei@malimei-virtual-machine:~/cc$
```

图 5.95 查看压缩文件

## 5.2.7 权限管理命令

### 1. chgrp

**功能描述:** 改变文件或目录的所属组。在 Linux 系统中,文件或者目录的权限由拥有者和所属群组来管理,采用群组名称或者群组识别码来标记不同权限,超级用户拥有最大权限。chgrp 命令是 change group 的缩写,要被改变的组名必须在/etc/group 文件内存在才可以,默认情况下只有 root 权限才能执行。

**语法:** chgrp [选项] [群组] [文件或目录]

**选项:** chgrp 命令的常用选项见表 5.22。

表 5.22 chgrp 命令的常用选项

选 项	作 用
-R	处理指定目录及其子目录下的所有文件
-c	当发生改变时输出调试信息
-f	不显示错误信息



续表

选 项	作 用
-v	运行时显示详细的处理信息
-dereference	作用于符号链接的指向,而不是符号链接本身
--no-dereference	作用于符号链接本身
--reference	=文件 1,文件 2,改变文件 2 所属群组,使其与文件 1 相同

**例 5.61** 改变文件的群组属性并显示过程,如图 5.96 所示。

\$ chgrp -v bin tar1.doc 将 tar1.doc 文件由 malimei 群组改为 bin。

```

root@malimei:/home/malimei/Documents/test# ll
total 16
drwxrwxr-x 4 malimei malimei 4096 Nov 11 00:09 ./
drwxr-xr-x 5 malimei malimei 4096 Nov 11 00:09 ../
-rw-rw-r-- 1 malimei malimei 0 Nov 10 18:21 tar1.doc
-rw-rw-r-- 1 malimei malimei 0 Nov 10 18:21 tar2.doc
drwxrwxr-x 2 root root 4096 Nov 10 18:14 test1/
drwxrwxr-x 3 root root 4096 Nov 10 19:58 test2/
root@malimei:/home/malimei/Documents/test# chgrp -v bin tar1.doc
changed group of 'tar1.doc' from malimei to bin
root@malimei:/home/malimei/Documents/test# ll
total 16
drwxrwxr-x 4 malimei malimei 4096 Nov 11 00:09 ./
drwxr-xr-x 5 malimei malimei 4096 Nov 11 00:09 ../
-rw-rw-r-- 1 malimei bin 0 Nov 10 18:21 tar1.doc
-rw-rw-r-- 1 malimei malimei 0 Nov 10 18:21 tar2.doc
drwxrwxr-x 2 root root 4096 Nov 10 18:14 test1/
drwxrwxr-x 3 root root 4096 Nov 10 19:58 test2/

```

图 5.96 更改文件所属的群组

**例 5.62** 根据指定文件改变文件的群组属性,如图 5.97 所示。

\$ chgrp --reference=tar2.doc tar1.doc 改变 tar1.doc 文件所属群组,使其与 tar2.doc 相同。

```

root@malimei:/home/malimei/Documents/test# ll
total 16
drwxrwxr-x 4 malimei malimei 4096 Nov 11 00:09 ./
drwxr-xr-x 5 malimei malimei 4096 Nov 11 00:09 ../
-rw-rw-r-- 1 malimei bin 0 Nov 10 18:21 tar1.doc
-rw-rw-r-- 1 malimei malimei 0 Nov 10 18:21 tar2.doc
drwxrwxr-x 2 root root 4096 Nov 10 18:14 test1/
drwxrwxr-x 3 root root 4096 Nov 10 19:58 test2/
root@malimei:/home/malimei/Documents/test# chgrp --reference=tar2.doc tar1.doc
root@malimei:/home/malimei/Documents/test# ll
total 16
drwxrwxr-x 4 malimei malimei 4096 Nov 11 00:09 ./
drwxr-xr-x 5 malimei malimei 4096 Nov 11 00:09 ../
-rw-rw-r-- 1 malimei malimei 0 Nov 10 18:21 tar1.doc
-rw-rw-r-- 1 malimei malimei 0 Nov 10 18:21 tar2.doc
drwxrwxr-x 2 root root 4096 Nov 10 18:14 test1/
drwxrwxr-x 3 root root 4096 Nov 10 19:58 test2/

```

图 5.97 改变文件群组属性与指定文件相同

**例 5.63** 递归改变多个文件的群组属性,如图 5.98 所示。

\$ chgrp -R malimei test1 递归改变目录 test1 及其下文件的所属群组为 malimei。

```

root@malimei-virtual-machine:/home/test1# cd ..
root@malimei-virtual-machine:/home# chgrp -R malimei test1
root@malimei-virtual-machine:/home# ls -l
总用量 105216
drwx----- 2 root    root      4096 10月  7 10:53 bc
drwx----- 2 root    root     16384 8月 29 17:29 lost+found
drwxr-xr-x 18 malimei malimei   4096 10月  7 16:11 malimei
drwxr-xr-x  2 root    malimei   4096 10月  7 12:06 test1
drwxr-xr-x  2 root    root      4096 10月  7 11:17 test2
-rw-r--r--  1 root    malimei    0 10月  7 10:46 text
-rw-r--r--  1 root    root     107704942 9月 19 09:52 VMWARETO.TGZ
root@malimei-virtual-machine:/home# cd test1
root@malimei-virtual-machine:/home/test1# ls -l
总用量 16
-rw-r--r--  1 root malimei  0 10月  7 00:00 ax
-rw-r--r--  1 root malimei 25 10月  7 11:16 wj1.gz
-rw-r--r--  1 root malimei 24 10月  7 11:16 wj1.gz
-rw-r--r--  1 root malimei 25 10月  7 11:15 wj21.gz
-rw-r--r--  1 root malimei 23 10月  7 10:57 wj.gz
root@malimei-virtual-machine:/home/test1#

```

图 5.98 递归改变目录及其下文件的群组属性

从该例可以看出, 添加了参数-R 后, test1 目录及目录中的文件所属的组都改变为 malimei 组, 这是一种递归改变。如果不添加参数-R, 仅改变目录 test1 的组, 但是目录中的文件所属的组没变, 如图 5.99 所示。

```

drwxr-xr-x  2 root    root      4096 10月  7 12:06 test1
drwxr-xr-x  2 root    root      4096 10月  7 11:17 test2
-rw-r--r--  1 root    malimei    0 10月  7 10:46 text
-rw-r--r--  1 root    root     107704942 9月 19 09:52 VMWARETO.TGZ
root@malimei-virtual-machine:/home# chgrp malimei test1
root@malimei-virtual-machine:/home# ls -l
总用量 105216
drwx----- 2 root    root      4096 10月  7 10:53 bc
drwx----- 2 root    root     16384 8月 29 17:29 lost+found
drwxr-xr-x 18 malimei malimei   4096 10月  7 16:11 malimei
drwxr-xr-x  2 root    malimei   4096 10月  7 12:06 test1
drwxr-xr-x  2 root    root      4096 10月  7 11:17 test2
-rw-r--r--  1 root    malimei    0 10月  7 10:46 text
-rw-r--r--  1 root    root     107704942 9月 19 09:52 VMWARETO.TGZ
root@malimei-virtual-machine:/home# cd test1
root@malimei-virtual-machine:/home/test1# ls
ax wj1.gz wj1.gz wj21.gz wj.gz
root@malimei-virtual-machine:/home/test1# ls -l
总用量 16
-rw-r--r--  1 root root  0 10月  7 00:00 ax
-rw-r--r--  1 root root 25 10月  7 11:16 wj1.gz
-rw-r--r--  1 root root 24 10月  7 11:16 wj1.gz
-rw-r--r--  1 root root 25 10月  7 11:15 wj21.gz
-rw-r--r--  1 root root 23 10月  7 10:57 wj.gz

```

图 5.99 仅改变目录的群组属性

## 2. chown

**功能描述:** chown 是 change owner 的简写, 将文件或目录的所有者改变为指定用户, 还可以修改文件所属组群。如果需要将某一目录下的所有文件都改变其拥有者, 可以使用-R 参数。

**语法:** chown [选项] [用户[: 群组]] [文件或目录]

**选项:** chown 命令的常用选项见表 5.23。

表 5.23 chown 命令的常用选项

选 项	作 用
-c	显示更改的部分信息
-f	忽略错误信息
-R	处理指定目录及其子目录下的所有文件,递归式地改变指定目录及其下的所有子目录和文件的拥有者
-v	显示详细的处理信息
-reference=<目录或文件>	把指定的目录/文件作为参考,把操作的目录/文件设置成参考文件/目录相同所有者和群组

例 5.64 改变文件所有者及所属组,如图 5.100 所示。

\$ chown root:bin file1 将 file1 文件的所有者改为 root,所属群组为 bin。

```

root@malimei:/home/malimei/Documents/file# ll
total 12
drwxrwxr-x 2 malimei malimei 4096 Nov 11 18:19 ./
drwxr-xr-x 6 malimei malimei 4096 Nov 11 18:12 ../
-rw-rw-r-- 1 malimei bin      0 Nov 11 18:12 file1
-rw-rw-r-- 1 malimei malimei  6 Nov  9 16:11 text1
root@malimei:/home/malimei/Documents/file# chown root:bin file1
root@malimei:/home/malimei/Documents/file# ll
total 12
drwxrwxr-x 2 malimei malimei 4096 Nov 11 18:19 ./
drwxr-xr-x 6 malimei malimei 4096 Nov 11 18:12 ../
-rw-rw-r-- 1 root    bin      0 Nov 11 18:12 file1
-rw-rw-r-- 1 malimei malimei  6 Nov  9 16:11 text1
    
```

图 5.100 改变文件所有者及所属组

例 5.65 更改目录及子目录所有者,如图 5.101 所示。

\$ chown -R malimei test1 将 test1 目录及其下文件的所有者更改为 malimei。

```

drwxr-xr-x 18 malimei malimei 4096 10月 7 16:11 malimei
drwxr-xr-x  2 root    malimei 4096 10月 7 12:06 test1
drwxr-xr-x  2 root    root    4096 10月 7 11:17 test2
-rw-r--r--  1 root    malimei  8 10月 7 10:46 text
-rw-r--r--  1 root    root    107704942 9月 19 09:52 VMWARETO.TGZ
root@malimei-virtual-machine:/home# cd test1
root@malimei-virtual-machine:/home/test1# ls -l
总用量 16
-rw-r--r-- 1 root malimei 0 10月 7 00:00 ax
-rw-r--r-- 1 root malimei 25 10月 7 11:16 wj1.gz
-rw-r--r-- 1 root malimei 24 10月 7 11:16 wj1.gz
-rw-r--r-- 1 root malimei 25 10月 7 11:15 wj21.gz
-rw-r--r-- 1 root malimei 23 10月 7 10:57 wj.gz
root@malimei-virtual-machine:/home/test1# cd ..
root@malimei-virtual-machine:/home# chown -R malimei test1
root@malimei-virtual-machine:/home# cd test1
root@malimei-virtual-machine:/home/test1# ls -l
总用量 16
-rw-r--r-- 1 malimei malimei 0 10月 7 00:00 ax
-rw-r--r-- 1 malimei malimei 25 10月 7 11:16 wj1.gz
-rw-r--r-- 1 malimei malimei 24 10月 7 11:16 wj1.gz
-rw-r--r-- 1 malimei malimei 25 10月 7 11:15 wj21.gz
-rw-r--r-- 1 malimei malimei 23 10月 7 10:57 wj.gz
root@malimei-virtual-machine:/home/test1#
    
```

图 5.101 更改目录及子目录所有者

从例 5.65 可以看出, chown 命令是在超级用户下执行的, 需要 root 权限。

### 例 5.66 更改所有者实例。

文件 11.txt 的所有者是 malimei, 更改所有者为 user1, 转到 user1 用户下, user1 用户就可以对 11.txt 文件进行编辑、修改, 如图 5.102 所示。如果不是这个文件的所有者, 也不是同组人, 就是其他人, 那么就不能修改文件, 如 user2 用户就不能修改 11.txt 文件, 如图 5.103 所示。

```
malimei@malimei-virtual-machine:~$ sudo chown user1 11.txt
malimei@malimei-virtual-machine:~$ su user1
密码:
user1@malimei-virtual-machine:/home/malimei$ nano 11.txt
user1@malimei-virtual-machine:/home/malimei$ ls -l 11.txt
-rw-rw-r-- 1 user1 malimei 27 5月 14 11:19 11.txt
user1@malimei-virtual-machine:/home/malimei$
```

图 5.102 更改文件的所有者

```
user2@malimei-virtual-machine: /home/malimei
GNU nano 2.5.3 文件: 11.txt
this is test!
Read 2 lines (Warning: No write permission)
^G 求助      ^O Write Out ^W 搜索      ^K 剪切文字  ^J 对齐      ^C 游标位置
^X 离开      ^R 读档      ^\ 替换      ^U Uncut Text ^T 拼写检查  ^_ 跳行
```

图 5.103 其他人不能修改文件

## 3. chmod

**功能描述:** 改变文件或目录的访问权限。

在 Linux 系统中, 用户设定文件权限控制使其他用户不能访问、修改。但在系统应用中, 有时需要让其他用户使用某个原来其不能访问的文件或目录, 这时就需要重新设置文件的权限, 使用的命令是 chmod 命令。并不是谁都可改变文件和目录的访问权限, 只有文件和目录的所有者才有权限修改其权限。另外, 超级用户可对所有文件或目录进行权限设置。

文件或目录的访问权限分为只读、只写和可执行三种。文件所有者拥有对该文件的读、写和可执行权限, 用户也可根据需把访问权限设置为需要的任何组合。访问文件的用户有三种类型: 文件所有者、组成员用户和普通用户, 他们都有各自的文件访问方式。

**语法:** chmod[选项][模式]文件  
 chmod 命令有两种模式:符号模式和绝对模式。  
**选项:** chmod 命令的常用选项如表 5.24 所示。

表 5.24 chmod 命令的常用选项

选 项	作 用
-v	运行时显示详细的处理信息
-c	显示改变部分的命令执行过程
-f	不显示错误信息
-R	将指定目录下的所有文件和子目录做递归处理
-reference=<目录或者文件>	设置成与指定目录或者文件具有相同的权限

下面分别介绍该命令的两种不同模式。

### 1) 符号模式

chmod [选项] [who] operator [permission] files

其中,who、operator 和 permission 的选项如表 5.25~表 5.27 所示。

表 5.25 chmod 命令的 who 选项

选 项	作 用
-a	所有用户均具有的权限
-o	除了目录或者文件的当前用户或群组以外的用户或者群组
-u	文件或目录的当前所有者
-g	文件或者目录的当前群组

表 5.26 chmod 命令的 operator 选项

选 项	作 用
+	增加权限
-	取消权限
=	设定权限

表 5.27 chmod 命令的 permission 选项

选 项	作 用
r	读权限
w	写权限
x	执行权限

### 2) 绝对模式

chmod [选项] mode files

其中,mode 代表权限等级,由三个八进制数表示。

这三位数的每一位都表示一个用户类型的权限设置,取值是 0~7,即二进制的[000]~[111]。这个三位二进制数的每一位分别表示读、写、执行权限,如 000 表示三项权限均无,100 表示只读。这样,就有了下面的对应。

- 0 [000]: 无任何权限。
- 1 [001]: 执行权限。
- 2 [010]: 写权限。
- 3 [011]: 写、执行权限。

- 4 [100]: 只读权限。
- 5 [101]: 读、执行权限。
- 6 [110]: 读、写权限。
- 7 [111]: 读、写、执行权限。

三个如上所示的二进制字符串([000]~[111])构成了模式,第一位表示所有者的权限,第二位表示组用户的权限,第三位表示其他用户的权限。常用的模式如下。

- 600: 只有所有者有读和写的权限。
- 644: 所有者有读和写的权限,组用户只有读的权限。
- 700: 只有所有者有读和写以及执行的权限。
- 666: 每个人都有读和写的权限。
- 777: 每个人都有读、写以及执行的权限。

**例 5.67** 查看文件的权限,如图 5.104 所示。

`$ ls -l` 查看当前目录下所有文件及子目录的详细信息。

```
malimei@malimei:~/Documents$ ll
total 24
drwxrwxr-x  3 malimei malimei 4096 Nov 28 00:36 ./
drwxr-xr-x 15 malimei malimei 4096 Nov 28 00:14 ../
-rw-rw-r--  1 malimei malimei   0 Oct 31 05:13 a.txt
drwxrwxr-x  5 malimei malimei 4096 Nov 27 19:10 scf/
-rw-rw-r--  1 malimei malimei   28 Oct 30 08:49 test1
-rw-rw-r--  1 malimei malimei   56 Oct 31 03:11 test2
-rw-rw-r--  1 malimei malimei  112 Oct 31 03:15 test3
```

图 5.104 查看文件的权限

在图 5.104 中,ls 命令显示了文件或目录的详细信息,其中最左边一列(第一个字母除外)为文件的访问权限。具体如下。

- r: 表示文件可以被读(read)。
- w: 表示文件可以被写(write)。
- x: 表示文件可以被执行(如果它是程序的话)。
- : 表示相应的权限还没有被授予。

**例 5.68** 符号模式下添加可执行权限,如图 5.105 所示。

`# chmod a+x file1` 给 file1 文件所有用户增加可执行权限。

```
root@malimei:/home/malimei/Documents/file# ls -al file1
-rw-rw-r-- 1 root bin 0 Nov 11 18:12 file1
root@malimei:/home/malimei/Documents/file# chmod a+x file1
root@malimei:/home/malimei/Documents/file# ls -al file1
-rwxrwxr-x 1 root bin 0 Nov 11 18:12 file1
```

图 5.105 设置文件权限为所有用户可执行

**例 5.69** 符号模式下设置文件仅可执行,如图 5.106 所示。

`# chmod u=x file1` 设置文件 file1 所有者的权限为可执行。

从图 5.106 可以看到,使用“=x”选项时,文件 file1 所有者的权限从“rwx”变为“-x”,即原有的权限被撤销,重新设置为仅可执行。这与“+”选项不同。

```

root@malimei:/home/malimei/Documents/file# ls -al file1
-rwxrwxr-- 1 root bin 0 Nov 11 18:12 file1
root@malimei:/home/malimei/Documents/file# chmod u=x file1
root@malimei:/home/malimei/Documents/file# ls -al file1
---xrwxr-- 1 root bin 0 Nov 11 18:12 file1

```

图 5.106 设置文件权限为只可读

**例 5.70** 符号模式下设置文件的多重权限,如图 5.107 所示。

# chmod ug+w, o-x file1 给 file1 文件的所有者和文件属群增加写权限,删除其他用户的执行权限。

```

root@malimei:/home/malimei/Documents/file# ls -al file1
-r-xr-xr-x 1 root bin 0 Nov 11 18:12 file1
root@malimei:/home/malimei/Documents/file# chmod ug+w,o-x file1
root@malimei:/home/malimei/Documents/file# ls -al file1
-rwxrwxr-- 1 root bin 0 Nov 11 18:12 file1

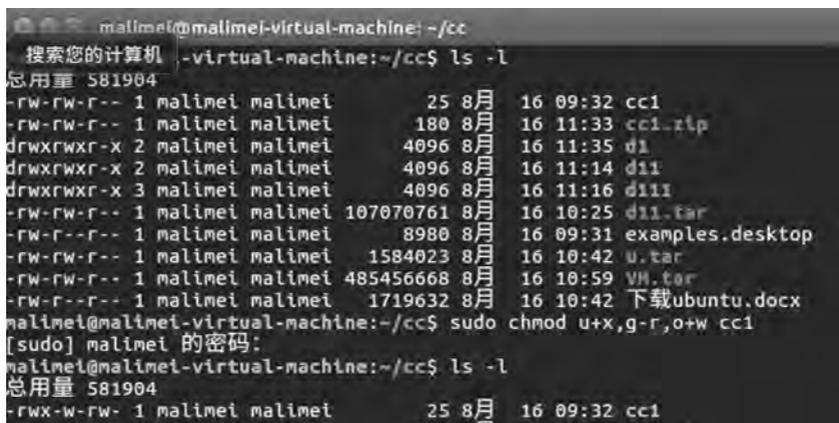
```

图 5.107 设置文件的多重权限

从例 5.70 可以看出,在符号模式下可以使用“,”来连接多个选项,为所有者、所属群和其他用户分别设置不同的权限。

**例 5.71** 符号模式下设置文件的多重权限,如图 5.108 所示。

\$ sudo chmod u+x,g-r,o+w cc1 为所有者添加执行权限,同组人去掉读权限,其他人(普通人)加上写的权限。



```

malimei@malimei-virtual-machine: ~/cc
搜索您的计算机 -virtual-machine:~/cc$ ls -l
总用量 581904
-rw-rw-r-- 1 malimei malimei 25 8月 16 09:32 cc1
-rw-rw-r-- 1 malimei malimei 180 8月 16 11:33 cc1.zip
drwxrwxr-x 2 malimei malimei 4096 8月 16 11:35 d1
drwxrwxr-x 2 malimei malimei 4096 8月 16 11:14 d11
drwxrwxr-x 3 malimei malimei 4096 8月 16 11:16 d111
-rw-rw-r-- 1 malimei malimei 107070761 8月 16 10:25 d11.tar
-rw-r--r-- 1 malimei malimei 8980 8月 16 09:31 examples.desktop
-rw-rw-r-- 1 malimei malimei 1584023 8月 16 10:42 u.tar
-rw-rw-r-- 1 malimei malimei 485456668 8月 16 10:59 VH.tar
-rw-r--r-- 1 malimei malimei 1719632 8月 16 10:42 下载ubuntu.docx
malimei@malimei-virtual-machine:~/cc$ sudo chmod u+x,g-r,o+w cc1
[sudo] malimei 的密码:
malimei@malimei-virtual-machine:~/cc$ ls -l
总用量 581904
-rwx-w-rw- 1 malimei malimei 25 8月 16 09:32 cc1

```

图 5.108 设置文件的多重权限

**例 5.72** 绝对模式下设置对文件的权限,如图 5.109 所示。

\$ chmod 712 cc1 设置 a 的权限:所有者具有读、写和执行权限,同组人具有可执行权限,其他人具有写权限。

从例 5.72 中可以看出,文件的所有者不使用 sudo 命令,可以更改文件的权限。

**注意:** 符号模式和绝对模式不能混着用,要遵循各自的格式。如图 5.110 所示的命令“chmod u-w g+7 o+1 a”,既然使用了符号模式,那么 permission 选项只能使用“r”“w”“x”,不能采用绝对模式中用数字表示权限的方法。

```
malimei@malimei-virtual-machine:~/cc$ ls -l
总用量 581904
-rw-rw-r-- 1 malimei malimei 25 8月 16 09:32 cc1
-rwxrwxr-x 2 malimei malimei 180 8月 16 11:33 cci.zip
drwxrwxr-x 2 malimei malimei 4096 8月 16 11:35 d1
drwxrwxr-x 2 malimei malimei 4096 8月 16 11:14 d11
drwxrwxr-x 3 malimei malimei 4096 8月 16 11:16 d111
-rw-rw-r-- 1 malimei malimei 107070761 8月 16 10:25 d11.tar
-rw-r--r-- 1 malimei malimei 8980 8月 16 09:31 examples.desktop
-rw-rw-r-- 1 malimei malimei 1584023 8月 16 10:42 u.tar
-rw-rw-r-- 1 malimei malimei 485456668 8月 16 10:59 VM.tar
-rw-r--r-- 1 malimei malimei 1719632 8月 16 10:42 下载ubuntu.docx
malimei@malimei-virtual-machine:~/cc$ chmod 712 cc1
malimei@malimei-virtual-machine:~/cc$ ls -l
总用量 581904
-rwx--x-w- 1 malimei malimei 25 8月 16 09:32 cc1
```

图 5.109 绝对模式下设置文件的权限

```
malimei@malimei-virtual-machine:~$ chmod u-w g+7 o+1 a
chmod: 无法访问"g+7": 没有那个文件或目录
chmod: 无法访问"o+1": 没有那个文件或目录
malimei@malimei-virtual-machine:~$ chmod u-w g+1 o+2 a
chmod: 无法访问"g+1": 没有那个文件或目录
chmod: 无法访问"o+2": 没有那个文件或目录
malimei@malimei-virtual-machine:~$ chmod u-w 1 2 a
chmod: 无法访问"1": 没有那个文件或目录
chmod: 无法访问"2": 没有那个文件或目录
```

图 5.110 符号模式和绝对模式不能混淆

**例 5.73** 文件 lx 属于 malimei 用户,我们看到同组人对文件 lx 有 rw 的权限,把 user2 用户加到 malimei 组里,就是同组人,如图 5.111 所示。user2 就可以修改文件 malimei 用户的文件了,如图 5.112 所示。

```
malimei@malimei-virtual-machine:~/cc/d11$ ls -l
总用量 4
-rw-rw-r-- 1 malimei malimei 16 8月 16 12:19 lx
malimei@malimei-virtual-machine:~/cc/d11$ sudo gpasswd -a user2 malimei
正在将用户"user2"加入到"malimei"组中
malimei@malimei-virtual-machine:~/cc/d11$ id user2
uid=1002(user2) gid=1002(user2) 组=1002(user2),1000(malimei)
malimei@malimei-virtual-machine:~/cc/d11$
```

图 5.111 加入同组人

```
user2@malimei-virtual-machine: /home/malimei
GNU nano 2.5.3 文件: a

THIS IS TEST!

LS(1) User Commands LS(1)
NAME
ls - list directory contents
SYNOPSIS
ls [OPTION]... [FILE]...
DESCRIPTION
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
fied.

Mandatory arguments to long options are mandatory for short options
too.

已写入237行
^G 求助 ^O Write Out ^W 搜索 ^K 剪切文字 ^J 对齐 ^G 光标位置
^X 离开 ^R 读档 ^V 替换 ^U Uncut Text ^I 拼写检查 ^_ 换行
```

图 5.112 同组用户 user2 修改属于 malimei 用户的文件 a

## 5.2.8 文件查找命令

文件查找命令有 `whereis`、`find` 和 `locate`。和 `find` 相比, `whereis` 查找的速度非常快, 当使用 `whereis` 和 `locate` 时会从 Linux 的数据库中查找数据, 但是该数据库并不是实时更新的, 默认情况下是一星期更新一次。因此, 在用 `whereis` 和 `locate` 查找文件时, 有时会找到已经被删除的数据, 或者刚刚建立文件却无法查找到, 原因就是数据库文件没有被更新。

### 1. whereis

**功能描述:** 寻找命令的二进制文件, 同时也会找到其帮助文件。

这个程序的主要功能是寻找一个命令所在的位置, 例如, 最常用的 `ls` 命令, 它是在 `/bin` 这个目录下的。如果希望知道某个命令存在哪一个目录下, 可以用 `whereis` 命令来查询。但是 `whereis` 命令只能用于程序名的搜索, 而且只搜索二进制文件(参数 `-b`)、帮助文件(参数 `-m`)和源代码文件(参数 `-s`)。如果省略参数, 则返回所有信息。

**语法:** `whereis` [选项] [文件名]

**选项:** `whereis` 命令的常用选项如表 5.28 所示。

表 5.28 `whereis` 命令的常用选项

选 项	作 用
<code>-b</code>	定位可执行文件
<code>-m</code>	定位帮助文件
<code>-s</code>	定位源代码文件
<code>-u</code>	搜索默认路径下除可执行文件、源代码文件、帮助文件以外的其他文件
<code>-B</code>	指定搜索可执行文件的路径
<code>-M</code>	指定搜索帮助文件的路径

**例 5.74** 搜索命令, 如图 5.113 所示。

```
$ whereis ls           搜索 ls 命令的路径。
$ whereis find        搜索 find 命令的路径。
$ whereis tar         搜索 tar 命令的路径。
```

```
malimei@malimei:~$ whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz
malimei@malimei:~$ whereis find
find: /usr/bin/find /usr/bin/X11/find /usr/share/man/man1/find.1.gz
malimei@malimei:~$ whereis tar
tar: /bin/tar /usr/lib/tar /usr/include/tar.h /usr/share/man/man1/tar.1.gz
malimei@malimei:~$
```

图 5.113 搜索命令的路径

**例 5.75** 搜索命令的帮助文件, 如图 5.114 所示。

```
$ whereis -m ls       搜索 ls 的帮助文件。
$ whereis -m find     搜索 find 的帮助文件。
$ whereis -m tar      搜索 tar 的帮助文件。
```



```

malimei@malimei:~$ help -s cd
cd: cd [-L[[-P [-e]] [-@]] [dir]
malimei@malimei:~$ help -d cd
cd - Change the shell working directory.
malimei@malimei:~$ help cd
cd: cd [-L[[-P [-e]] [-@]] [dir]
    Change the shell working directory.

    Change the current directory to DIR.  The default DIR is the value of the
    HOME shell variable.

    The variable CDPATH defines the search path for the directory containing
    DIR.  Alternative directory names in CDPATH are separated by a colon (:).
    A null directory name is the same as the current directory.  If DIR begins
    with a slash (/), then CDPATH is not used.

    If the directory is not found, and the shell option 'cdable_vars' is set,
    the word is assumed to be a variable name.  If that variable has a value,
    its value is used for DIR.

Options:
  -L      force symbolic links to be followed: resolve symbolic links in
          DIR after processing instances of '..'
  -P      use the physical directory structure without following symbolic

```

图 5.116 带参数查看命令 cd 的帮助

**例 5.78** 查看命令 ls 的帮助信息,如图 5.117 所示。

```
$ ls --help
```

查看 ls 命令的帮助信息,给出了用法和各个选项。

```

malimei@malimei:~$ help ls
bash: help: no help topics match 'ls'. Try 'help help' or 'man -k ls' or 'info
ls'.
malimei@malimei:~$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
-a, --all                do not ignore entries starting with .
-A, --almost-all        do not list implied . and ..
--author                 with -l, print the author of each file
-b, --escape              print C-style escapes for nongraphic characters

```

图 5.117 查看 ls 命令的帮助信息

**注意:** 使用 help 查看命令的帮助信息时需要区分是内部命令还是外部命令: 对于内部命令格式为 help <命令>, 如前面的例 5.76、例 5.77; 而外部命令需要使用 <命令> --help 格式, 如例 5.78。

### 3. man

**功能描述:** 查看命令的帮助手册。

查找命令的帮助信息更常用的是 man 命令。man 用来查看帮助手册, 通常使用者只要在命令 man 后输入想要获取的帮助命令的名称(例如 ls), man 就会列出一份完整的说明, 其内容包括命令语法、各选项的意义以及相关命令等。

**语法:** man [选项] 命令名称

**选项:** man 命令的常用选项见表 5.30。

表 5.30 man 命令的常用选项

选项	作用
-s	根据章节显示,具体见后面的说明
-f	只显示出命令的功能而不显示其中详细的说明文件
-w	不显示手册页,只显示将被格式化和显示的文件所在位置
-a	显示所有的手册页,而不是只显示第一个
-E	在每行的末尾显示 \$ 符号

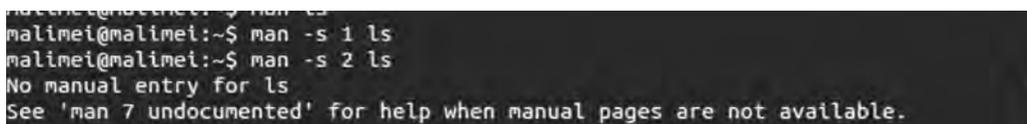
其中,选项-s 是根据章节显示帮助,常用的章节选项见表 5.31。

表 5.31 选项-s 的章节参数

章节参数	作用
1	一般使用者的命令
2	系统调用的命令
3	C 语言函数库的命令
4	有关驱动程序和系统设备的解释
5	配置文件的解释
6	游戏程序的命令
7	其他的软件或程序的命令和有关系统维护的命令

**例 5.79** 查看 ls 命令的帮助手册,如图 5.118 所示。

```
$ man -s 1 ls          查看 ls 命令的帮助手册
```



```
malimei@malimei:~$ man -s 1 ls
malimei@malimei:~$ man -s 2 ls
No manual entry for ls
See 'man 7 undocumented' for help when manual pages are not available.
```

图 5.118 查看 ls 命令的帮助手册

从例 5.79 中可以看出,ls 是一般使用者的命令,加-s 参数时用“1”选项。如果用其他章节参数会提示错误。

使用 man 命令后会显示所查看命令的 man 文件。如图 5.119 所示,可以使用鼠标上下滑动来翻页,按 Q 键退出帮助手册返回命令界面。

#### 4. find

**功能描述:** 寻找文件或目录的位置。

如果有大量的文件保存在许多不同的目录中,可能需要搜索它们,以便能找出某种类型的一个或者多个文件,这就需要 find 命令。find 命令可以按照文件名、类型、所有者甚至最后更新的时间来搜索文件。

**语法:** find [搜索路径] [搜寻关键字] [文件或目录]

**选项:** find 命令的常用选项如表 5.32 所示。

```

LS(1)                                User Commands                                LS(1)
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
  fied.

  Mandatory arguments to long options are mandatory for short options
  too.

  -a, --all
      do not ignore entries starting with .

  -A, --almost-all
      do not list implied . and ..

  --author
Manual page ls(1) line 1 (press h for help or q to quit)
    
```

图 5.119 命令 ls 的帮助手册

表 5.32 find 命令的常用选项

选 项	作 用
-type	查找某一类型的文件,具体参数见后面说明
-name	按照文件名查找文件
-group	按照文件所属的组来查找文件
-user	按照文件所有者来查找文件
-print	find 命令将匹配的文件输出到标准输出
-link	按照文件的链接数来查找文件
-size n : [c]	查找文件长度为 n 块的文件,带有 c 时表示文件长度以字节计
-newer file1 ! file2	查找更改时间比文件 file1 新,但比文件 file2 旧的文件
-perm	按照文件权限来查找文件
-depth	在查找文件时,首先查找当前目录中的文件,然后在其子目录中查找
-prune	不在指定的目录中查找,如同时使用-depth 选项,-prune 将被忽略
-nogroup	查找无有效所属组的文件,即该文件所属的组在/etc/groups 中不存在
-nouser	查找无有效属主的文件,即该文件的属主在/etc/passwd 中不存在

其中,选项-type 表示按照文件类型查找文件,具体的参数见表 5.33。

表 5.33 -type 选项参数

type 参数	作 用	type 参数	作 用
b	块设备文件	p	管道文件
d	目录	l	符号链接文件
c	字符设备文件	f	普通文件

另外,find 命令还可以利用时间特征来查找文件,其参数见表 5.34。

表 5.34 时间特征参数

时间参数	作用
amin n	查找 n 分钟以内被访问过的所有文件
atime n	查找 n 天以内被访问过的所有文件
cmin n	查找 n 分钟以内文件状态被修改过的所有文件
ctime n	查找 n 天以内文件状态被修改过的所有文件
mmin n	查找 n 分钟以内文件内容被修改过的所有文件
mtime n	查找 n 天以内文件内容被修改过的所有文件

下面详细说明 find 的用法。

#### (1) 通过文件名查找。

知道了某个文件的文件名,却不知道它存于哪个目录下,此时可通过查找命令找到该文件,命令如下。

```
# find / -name httpd.conf -print
```

#### (2) 根据部分文件名查找。

当要查找某个文件时,不知道该文件的全名,只知道这个文件包含几个特定的字母,此时用查找命令也可找到相应文件。这时在查找文件名时可使用通配符“\*”“?”。例如,还是查找文件 httpd.conf,但仅记得该文件名包含“http”字符串,可使用如下命令查找。

```
# find / -name *http* -print
```

#### (3) 根据文件的特征查询。

如果仅知道某个文件的大小、修改日期等特征也可使用 find 命令把该文件查找出来。例如,知道一个文件尺寸为 2500B,可使用如下命令查找。

```
# find /etc -size -2500c -print
```

**例 5.80** 按照文件名查找文件,如图 5.120 所示。

```
$ find ~ -name "*.doc" -print 查找当前目录及子目录中扩展名为.doc的文件并显示。
$ find . -name "[A-F]*" -print 查找以大写字母 A~F 开头的文件并显示。
$ find /etc -iname 'f????' 查找/etc下所有以 f 开头后面有四个字符的文件。
```

**例 5.81** 按照文件权限模式查找文件,如图 5.121 所示。

```
$ find . -perm 777 -print 在当前目录下查找文件权限为 777 的文件,即查找每个人都有可读写可执行权限的文件。
```

**例 5.82** 忽略某个子目录查找文件,如图 5.122 所示。

```
$ find work -path "cc/d1" -prune -o -print 查找 cc 文件夹及子文件夹的文件,忽略子文件夹 d1。
```

**例 5.83** 按文件所有者、用户组等查找文件,如图 5.123 所示。

```
$ find work -user malimei -print 在 work 文件夹下查找所有者为 malimei 的文件并输出。
$ find /home/malimei/Documents -group malimei -print 在 Documents 文件夹下查找属于 malimei 用户组的文件。
```

```
malimei@malimei:~$ find -name "*.doc" -print
/home/malimei/Documents/test/tar1.doc
/home/malimei/Documents/test/tar2.doc
/home/malimei/Documents/tar1.doc
/home/malimei/Documents/tar2.doc
malimei@malimei:~$ find . -name "[A-F]*" -print
./Desktop
./Desktop/vmware-tools-distrib/FILES
./Downloads
./Documents
./Documents/English-
./Documents/English
./gconf/apps/gnome-terminal/profiles/Default
./local/share/Trash/info/English.zip.trashinfo
./local/share/Trash/files/English.zip
malimei@malimei:~$ find /etc -iname 'f????'
find: '/etc/polkit-1/localauthority': Permission denied
/etc/fstab
find: '/etc/ppp/peers': Permission denied
find: '/etc/ssl/private': Permission denied
find: '/etc/chatscripts': Permission denied
/etc/X11/xinit/xinput.d/fcitx
/etc/X11/fonts
find: '/etc/cups/ssl': Permission denied
/etc/apparmor.d/abstractions/fonts
/etc/fonts
```

图 5.120 按照文件名查找文件

```
root@malimei:/home/malimei# find . -perm 777 -print
./Desktop/vmware-tools-distrib/INSTALL
./Desktop/vmware-tools-distrib/vmware-install.pl
./local/share/Trash/files/test4
```

图 5.121 按照文件权限模式查找文件

```
malimei@malimei-virtual-machine:~$ ls -l
总用量 581908
-rwx-x-w- 1 malimei malimei 25 8月 16 09:32 cc1
-rw-rw-r-- 1 malimei malimei 180 8月 16 11:33 cc1.zip
drwxrwxr-x 2 malimei malimei 4096 8月 16 11:35 d1
drwxrwxr-x 2 malimei malimei 4096 8月 16 12:23 d11
drwxrwxr-x 3 malimei malimei 4096 8月 16 11:16 d111
-rw-rw-r-- 1 malimei malimei 107070761 8月 16 10:25 d11.tar
-rw-r--r-- 1 malimei malimei 8980 8月 16 09:31 examples.desktop
-rw-rw-r-- 1 malimei malimei 16 8月 16 12:18 lx
-rw-rw-r-- 1 malimei malimei 1584023 8月 16 10:42 u.tar
-rw-rw-r-- 1 malimei malimei 485456668 8月 16 10:59 VM.tar
-rw-r--r-- 1 malimei malimei 1719632 8月 16 10:42 下载ubuntu.docx
malimei@malimei-virtual-machine:~/cc$ cd ..
malimei@malimei-virtual-machine:~$ find cc -path "cc/d1" -prune -o -print
cc
cc/d11.tar
cc/lx
cc/d111
cc/d111/d1
cc/d111/d1/VMWARETO.TGZ
cc/d111/d1/examples.desktop
cc/cc1
cc/examples.desktop
cc/cc1.zip
cc/VM.tar
cc/下载ubuntu.docx
cc/d11
cc/d11/lx
cc/d11/.lx.swp
cc/u.tar
malimei@malimei-virtual-machine:~$
```

图 5.122 忽略某个子目录查找文件

```
malimei@malimei:~/Documents$ find work -user malimei -print
work
work/text3
work/text1
work/text2
work/file
work/file/text1
malimei@malimei:~$ find /home/malimei/Documents -group malimei -print
/home/malimei/Documents
/home/malimei/Documents/English-
/home/malimei/Documents/text3-
/home/malimei/Documents/work
/home/malimei/Documents/work/text3
/home/malimei/Documents/work/text1
/home/malimei/Documents/work/text2
/home/malimei/Documents/work/file2
/home/malimei/Documents/work/file1
/home/malimei/Documents/work/file1/text1
```

图 5.123 按照所有者、用户组等查找文件

例 5.84 按照时间查找文件,如图 5.124 所示。

`$ find /home/malimei/Documents/work -mtime -5 -print` 查找更改时间在 5 日内的/home/malimei/Documents/work 文件。

```
malimei@malimei:~$ find /home/malimei/Documents/work -mtime -5 -print
/home/malimei/Documents/work
/home/malimei/Documents/work/text3
/home/malimei/Documents/work/text1
/home/malimei/Documents/work/text2
/home/malimei/Documents/work/file2
/home/malimei/Documents/work/file1
/home/malimei/Documents/work/file1/text1
/home/malimei/Documents/work/file1/file1
```

图 5.124 按照时间查找文件

例 5.85 按照文件类型查找文件,如图 5.125 所示。

`$ find /home/malimei/Documents -type d -print` 查找指定目录下所有的目录文件并显示。

```
malimei@malimei:~$ find /home/malimei/Documents -type d -print
/home/malimei/Documents
/home/malimei/Documents/work
/home/malimei/Documents/work/file2
/home/malimei/Documents/work/file1
/home/malimei/Documents/test2
/home/malimei/Documents/test
/home/malimei/Documents/test/test1
/home/malimei/Documents/test/test2
/home/malimei/Documents/test/test2/test3
```

图 5.125 按照文件类型查找文件

例 5.86 按照文件长度查找文件,如图 5.126 所示。

`$ find. -size +1000000c -print` 在当前目录下查找文件长度大于 1MB 的文件。

## 5. locate

功能描述: 寻找文件或目录。

locate 用于在文件系统内通过搜寻数据库查找指定文件,相对 find 命令查找速度快。

```
malimei@malimei:~$ find . -size +1000000c -print
./cache/software-center/software-center-agent.db/spelling.DB
./cache/software-center/software-center-agent.db/term1ist.DB
./cache/software-center/software-center-agent.db/post1ist.DB
./cache/software-center/piston-helper/software-center.ubuntu.com,api,2.0,applic
ations,en,ubuntu,trusty,i386,.08aef60ad6cfa3b66fb77f2e75041241
./cache/software-center/piston-helper/reviews.ubuntu.com,reviews,api,1.0,review
-stats,_any,_any,..862d34545f176252d7fc8e5907b8b5d1
```

图 5.126 按照文件长度查找文件

数据库由 updatedb 程序来更新,updatedb 是由 crondaemon 周期性建立的,locate 命令在搜寻数据库时比由整个硬盘资料搜寻来得快,但若所找到的文件是最近才建立或刚更名的,locate 命令可能会找不到。updatedb 每天会更新一次,可以由修改 crontab 来更新设定值。

**语法:** locate [选项] [搜索关键字]

**选项:** locate 命令的常用选项见表 5.35。

表 5.35 locate 命令的常用选项

选 项	作 用	选 项	作 用
-a	输出所有匹配模式的文件	-h	显示辅助信息
-d	指定文件库的路径	-q	安静模式,不会显示任何错误信息
-e	将排除在寻找的范围之外	-V	显示程式的版本信息

**例 5.87** 查找文件名中包含“x”的文件,如图 5.127 所示。

\$ locate x|more 查找所有和“x”相关的文件,并用 more 命令显示。

```
malimei@malimei-virtual-machine:/$ locate x|more
/bin/busybox
/bin/bzexe
/bin/gzexe
/bin/ntfsfix
/boot/grub/gfxblacklist.txt
/boot/grub/i386-pc/effat.mod
/boot/grub/i386-pc/efctest.mod
/boot/grub/i386-pc/ext2.mod
/boot/grub/i386-pc/extcmd.mod
/boot/grub/i386-pc/gettext.mod
/boot/grub/i386-pc/gfxmenu.mod
/boot/grub/i386-pc/gfxterm.mod
/boot/grub/i386-pc/gfxterm_background.mod
```

图 5.127 查找文件名中包含“x”的文件

**例 5.88** 查找指定目录下以“t”开头的文件,如图 5.128 所示。

# locate /file1/t 查找 file1 文件夹下以“t”开头的文件

```
root@malimei:/home/malimei/Documents/work# locate /file1/t
/home/malimei/Documents/work/file1/text1
```

图 5.128 查找“t”开头的文件

相关命令还有:

whatis 命令: 查询命令的功能。

which 命令: 显示可执行命令的路径。

例 5.89 查询命令的功能和可执行路径,如图 5.129 所示。

```
$ whatis ls          查询 ls 命令的功能。
$ which ls          查询 ls 命令的可执行路径。
```

```
malimei@malimei-virtual-machine:/home$ whatis ls
ls (1)                - list directory contents
malimei@malimei-virtual-machine:/home$ which ls
/bin/ls
```

图 5.129 查询命令的功能和可执行路径

## 6. grep

**功能描述:** 使用正则表达式查找文件内容。

**语法:** `grep [选项] 匹配字符串 文件列表`

**选项:** `grep` 命令的常用选项如表 5.36 所示。

表 5.36 `grep` 命令的常用选项

选 项	作 用
-v	列出不匹配串或正则表达式的行,即显示不包含匹配文本的所有行
-c	对匹配的行计数
-l	只显示包含匹配的文件的文件名
-h	查询多文件时不显示文件名,抑制包含匹配文件的文件名的显示
-n	每个匹配行只按照相对的行号显示
-i	产生不区分大小写的匹配,默认状态是区分大小写

正则表达式的参数如下。

`\`: 忽略正则表达式中特殊字符的原有含义。

`^x`: 匹配正则表达式的开始行,匹配一个字符 `x`。

`$`: 匹配正则表达式的结束行。

`\<`: 从匹配正则表达式的行开始。

`\>`: 到匹配正则表达式的行结束。

`[ ]`: 单个字符,如`[A]`即 `A` 符合要求。

`[ - ]`: 范围,如`[A-Z]`,即 `A`、`B`、`C` 一直到 `Z` 都符合要求。

`[^x]`: 匹配一个字符,这个字符是除了 `x` 以外的所有字符。

`*`: 有字符,长度可以为 0。

例 5.90 搜索文件中包含“s”的内容,如图 5.130 所示。

```
$ grep -n "s" 1.txt          搜索文件 1.txt 中包含“s”的行,并显示行号。
```

```
malimei@malimei-virtual-machine:~$ grep -n "s" 1.txt
1: ss
2: s
3: s
```

图 5.130 搜索文件中包含“s”的行

例 5.91 搜索文件的内容,如图 5.131 所示。

```
$ grep -n '-' a          搜索文件 a 中包含“-”的行,并显示行号。
```

```
malimei@malimei:~/Documents/test$ grep -n '-' a
3:Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
6: -a, --all do not ignore entries starting with .
7: -A, --almost-all do not list implied . and ..
8: --author with -l, print the author of each file
9: -b, --escape print C-style escapes for nongraphic characters
10: --block-size=SIZE scale sizes by SIZE before printing them. E.g.,
11: '--block-size=M' prints sizes in units of
13: -B, --ignore-backups do not list implied entries ending with ~
14: -c with -lt: sort by, and show, ctime (time of last
16: with -l: show ctime and sort by name
malimei@malimei:~/Documents/test$ grep -vn '-' a
1:Usage: ls [OPTION]... [FILE]...
2:List information about the FILES (the current directory by default).
4:
5:Mandatory arguments to long options are mandatory for short options too.
12: 1,048,576 bytes. See SIZE format below.
15: modification of file status information)
17: otherwise: sort by ctime, newest first
20: or can be 'never' or 'auto'. More info below
22: and do not dereference symbolic links
malimei@malimei:~/Documents/test$ grep -n '*' a
25: -F, --classify append indicator (one of */=>|) to entries
26: --file-type likewise, except do not append '*'
101:SIZE is an integer and optional unit (example: 10M is 10*1024*1024). Units
malimei@malimei:~/Documents/test$ grep -n '[a-z]\{14\}' a
3:Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
70: -R, --recursive list subdirectories recursively
95: -X sort @alphabetically by entry extension
```

图 5.131 搜索文件的内容

- \$ grep -vn '-' a                    搜索文件 a 中不包含“-”的行,并显示行号。
- \$ grep -n '\*' a                    搜索文件 a 中包含“\*”的行,并显示行号。
- \$ grep -n'[a-z]\{14\}' a        搜索文件 a 中等于 14 个小写字母的字符串。

例 5.92 在文件中搜索包含“li”的行,如图 5.132 所示。

- \$ grep -n 'li' c                    搜索当前目录下文件 c 中包含“li”的行。

```
malimei@malimei-virtual-machine:~$ ls -l > c
malimei@malimei-virtual-machine:~$ cat c
总用量 104652
-rw-rw-r-- 1 malimei malimei 10240 8月 16 09:41 11.tar
-rw-rw-r-- 1 user1 malimei 15 8月 16 11:59 11.txt
-rw-rw-r-- 1 malimei malimei 0 8月 16 17:10 c
drwx----- 5 malimei malimei 4096 8月 16 12:18 cc
-rw-rw-r-- 1 malimei malimei 107070810 8月 16 09:39 cc1.tar
-rw-rw-r-- 1 malimei malimei 4203 8月 16 09:34 cc.tar
drwxrwxrwx 3 malimei malimei 4096 8月 16 09:45 d1
drwxrwxrwx 2 malimei malimei 4096 8月 15 17:54 d11
-rw-rw-r-- 1 malimei malimei 0 7月 19 18:34 dd
-rw-r--r-- 1 malimei malimei 8980 7月 19 17:59 examples.desktop
drwxrwxr-x 5 malimei malimei 4096 8月 15 17:43 scf
drwxrwxr-x 3 malimei malimei 4096 7月 20 11:27 vmware
drwxr-xr-x 2 malimei malimei 4096 7月 19 18:25 公共的
drwxr-xr-x 2 malimei malimei 4096 7月 19 18:25 模板
drwxr-xr-x 2 malimei malimei 4096 7月 19 18:25 视频
drwxr-xr-x 2 malimei malimei 4096 8月 12 17:50 图片
drwxr-xr-x 2 malimei malimei 4096 7月 19 18:25 文档
drwxr-xr-x 2 malimei malimei 4096 7月 19 18:25 下载
drwxr-xr-x 2 malimei malimei 4096 7月 19 18:25 音乐
drwxr-xr-x 2 malimei malimei 4096 7月 21 10:32 桌面
malimei@malimei-virtual-machine:~$ grep -n 'li' c
2:-rw-rw-r-- 1 malimei malimei 10240 8月 16 09:41 11.tar
3:-rw-rw-r-- 1 user1 malimei 15 8月 16 11:59 11.txt
4:-rw-rw-r-- 1 malimei malimei 0 8月 16 17:10 c
5:drwx----- 5 malimei malimei 4096 8月 16 12:18 cc
6:-rw-rw-r-- 1 malimei malimei 107070810 8月 16 09:39 cc1.tar
7:-rw-rw-r-- 1 malimei malimei 4203 8月 16 09:34 cc.tar
8:drwxrwxrwx 3 malimei malimei 4096 8月 16 09:45 d1
9:drwxrwxrwx 2 malimei malimei 4096 8月 15 17:54 d11
```

图 5.132 在文件中搜索字符串

**例 5.93** 搜索当前目录下的文件中包含“any”的行,如图 5.133 所示。

\$ grep 'any' \* 搜索当前目录下的所有文件中包含“any”的行。

```
malinei@malinei-virtual-machine:~$ grep 'any' *
*: can be augmented with a --sort option, but any use of
*: print any security context of each file
```

图 5.133 搜索多个文件的内容

**例 5.94** 搜索指定文件中第一个字符不是 d,后面是 bc 的字符串,如图 5.134 所示。

\$ grep -n '[^d]bc' tmp 搜索 tmp 文件中第一个字符不是 d,后面是 bc 的字符串。

```
malinei@malinei-virtual-machine:~$ cat tmp
abcccdbc
dbcnnnabc

malinei@malinei-virtual-machine:~$ grep -n '[^d]bc' tmp
1:abcccdbc
2:dbcnnnabc
malinei@malinei-virtual-machine:~$
```

图 5.134 搜索指定的字符串

[^]匹配一个不在指定范围内的字符,例如,'[^A-R T-Z]rep'匹配一个不包含 A~R 和 T~Z 的字母开头,紧跟 rep 的行。

### 5.2.9 统计命令 wc

**功能描述:** 统计指定文件中的字节数、字数、行数,并将统计结果显示输出。

该命令统计给定文件中的字节数、单词数、行数。如果没有给出文件名,则从标准输入设备读取,字符数包括空格和回车键。

**语法:** wc [选项] 文件列表

**选项:** wc 命令的常用选项见表 5.37。

表 5.37 wc 命令的常用选项

选 项	含 义
-c	统计字节数
-w	统计字数,一个字被定义为由空白、跳格或换行字符分隔开的字符串
-l	统计行数
-L	统计最长行的长度
-m	统计字符数,不能与-c 一起使用

**例 5.95** 统计行数、字节数和字数,如图 5.135 所示。

\$ wc -lcw 1.txt 统计 1.txt 文件的行数、字节数、字数。

**例 5.96** 统计文件的字数等信息,如图 5.136 所示。

```
$ wc -c test1 统计文件 test1 字节数。
$ wc -w test1 统计文件 test1 字数。
$ wc -l test1 统计文件 test1 行数。
$ wc -L test1 统计文件 test1 最长行长度。
```

```
malimei@malimei-virtual-machine:~$ wc -lcw 1.txt
4 4 9 1.txt
malimei@malimei-virtual-machine:~$ cat 1.txt
ss
s
s
d
```

图 5.135 统计行数、字节数和字数

```
malimei@malimei:~/Documents$ cat test1
1111111111111111
1111111111111111
malimei@malimei:~/Documents$ wc -c test1
28 test1
malimei@malimei:~/Documents$ wc -m test1
28 test1
malimei@malimei:~/Documents$ wc -w test1
2 test1
malimei@malimei:~/Documents$ wc -l test1
2 test1
malimei@malimei:~/Documents$ wc -L test1
13 test1
```

图 5.136 统计文件的字数等信息

注意：一个字被定义为由空白、跳格或换行字符分隔的字符串，如图 5.136 中 test1 的字数为 2，而不是 28。

## 5.3 输入、输出重定向

### 5.3.1 标准输入、输出

执行一个 Shell 命令行时通常会自动打开三个标准文档：标准输入文档(stdin)，标准输出文档(stdout)和标准错误输出文档(stderr)。其中，标准输入对应终端的键盘，标准输出和标准错误输出对应终端的屏幕。进程将从标准输入文档中得到输入数据，将正常输出数据输出到标准输出文档，而将错误信息送到标准错误文档中。

以 cat 命令为例，如果命令“\$ cat”中不带参数，就会从标准输入中读取数据，并将其送到标准输出上。

**例 5.97** 从键盘输入数据到显示屏，如图 5.137 所示。

\$ cat	从标准输入到标准输出。
Hello!	在键盘上输入“Hello!”。
Welcome to Ubuntu!	在键盘上输入“Welcome to Ubuntu!”。

```
malimei@malimei:~$ cat
Hello!
Hello!
Welcome to Ubuntu!
Welcome to Ubuntu!

^Z
[1]+  Stopped                  cat
malimei@malimei:~$
```

图 5.137 标准输入输出

注意：图 5.137 中第一行“Hello!”是通过键盘输入时的显示，第二行才是输出。按 Ctrl+Z 组合键退出输入模式。

### 5.3.2 输入重定向

从标准输入录入数据时，输入的数据系统没有保存，只能使用一次，下次再想使用这些数据就得重新输入。而且在终端输入时，如果输入错误修改起来也不方便。正是因为使用标准输入很不方便，我们需要将输入从标准输入重新定个方向转到其他位置，这就是输入重定向。

Linux 支持输入重定向，可以把命令的标准输入重定向到指定的文件中，也就是输入的数据不是来自键盘，而是来自一个指定的文件。也就是说，输入重定向主要用于改变一个命令的输入源，特别是那些需要大量数据输入的输入源。

在输入时，使用符号“<”和“<<”分别表示“输入”与“结束输入”。

**例 5.98** 将文档内容作为输入，如图 5.138 所示。

\$ wc < /etc/passwd 输入重定向，将/etc/passwd 文档内容传给 wc 命令。

```
malimei@malimei-virtual-machine:~$ wc < /etc/passwd
41  66 2196
```

图 5.138 输入重定向文件

**例 5.99** 使用 <<结束输入，如图 5.139 所示。

\$ cat <<end 从控制台输入字符串，当输入为“end”时结束输入。

```
malimei@malimei:~$ cat <<end
> Hello
> Welcome to Ubuntu
> Good luck
> end
Hello
Welcome to Ubuntu
Good luck
malimei@malimei:~$
```

图 5.139 从键盘输入<<结束

从图 5.139 中可以看到，使用符号“<<”后，输入过程与仅使用 cat 不同：屏幕上显示的是键盘输入内容，直至输入 end，才结束输入开始显示输出。

**例 5.100** 将标准输入保存至文件，如图 5.140 所示。

\$ cat > ss.txt <<eof 从控制台输入字符串，当输入“eof”时结束输入，并把内容保存到文件 ss.txt 中。

```
root@malimei-virtual-machine:/home# cat > ss.txt <<eof
> das
> das
> eof
root@malimei-virtual-machine:/home# cat ss.txt
das
das
root@malimei-virtual-machine:/home#
```

图 5.140 标准输入并保存至文件

### 5.3.3 输出重定向

因为输出到终端屏幕上的数据只能看,不能进行更多的处理,所以需要把输出从标准输出或标准错误输出重新定向到指定的文件,这就是输出重定向。Linux 支持输出重定向,可以将输出写入指定文件,而不是在屏幕上显示。用符号“>”表示替换,符号“>>”表示追加。

输出重定向有很多应用,如保存某个命令的输出,或者某个命令的输出内容很多,不能显示在屏幕的一页内,也可以将输出重定向到一个文档内,然后用文本编辑器打开这个文档查看输出信息。输出重定向还能够把一个命令的输出作为另一个命令的输入,这就是 5.4 节要讲到的管道命令。

**例 5.101** 从键盘输入信息保存到文件,如图 5.141 所示。

\$ cat >c 将屏幕输入的信息保存到文件 c(c 初始为空文件),按 Ctrl+Z 组合键退出输入。

```
malimei@malimei:~/Documents/test$ cat c
malimei@malimei:~/Documents/test$ cat >c
Hello
Welcome
bye
^Z
[3]+ Stopped          cat > c
malimei@malimei:~/Documents/test$ cat c
Hello
Welcome
bye
```

图 5.141 标准输入并保存到文件 c

**例 5.102** 将命令的输出保存到文件,如图 5.142 所示。

\$ ls -l >>c 将命令 ls 的输出追加到文件 c,即保存到文件的尾部,原内容不变。

\$ ls -l >c 将命令 ls 的输出保存到文件 c,如 c 不空,则覆盖原内容。

```
malimei@malimei-virtual-machine:~/ss$ ls -l>>c
malimei@malimei-virtual-machine:~/ss$ cat c
Hello
Welcome
bye
总用量 8
-rw-rw-r-- 1 malimei malimei 4 12月 17 16:48 a
lrwxrwxrwx 1 malimei malimei 1 12月 17 16:48 a1 -> a
-rw-rw-r-- 1 malimei malimei 18 5月 13 16:25 c
-rw-rw-r-- 1 malimei malimei 0 12月 17 16:48 t1
-rw-rw-r-- 1 malimei malimei 0 5月 13 16:10 t2
malimei@malimei-virtual-machine:~/ss$ ls -l>c
malimei@malimei-virtual-machine:~/ss$ cat c
总用量 4
-rw-rw-r-- 1 malimei malimei 4 12月 17 16:48 a
lrwxrwxrwx 1 malimei malimei 1 12月 17 16:48 a1 -> a
-rw-rw-r-- 1 malimei malimei 0 5月 13 16:26 c
-rw-rw-r-- 1 malimei malimei 0 12月 17 16:48 t1
-rw-rw-r-- 1 malimei malimei 0 5月 13 16:10 t2
malimei@malimei-virtual-machine:~/ss$
```

图 5.142 命令输出重定向到文件 c

**例 5.103** 将文件内容输出到另一个文件,如图 5.143 所示。

```
$ cat tmp > tmp1
```

 将文件 tmp 中的内容复制到文件 tmp1 中。

```
malimei@malimei-virtual-machine:~$ cat tmp
abccdbc
dbcnnnabc

malimei@malimei-virtual-machine:~$ cat tmp>tmp1
malimei@malimei-virtual-machine:~$ cat tmp1
abccdbc
dbcnnnabc

malimei@malimei-virtual-machine:~$
```

图 5.143 文件内容输出重定向至文件

## 5.4 管道

管道(Pipeline): 一个由标准输入输出链接起来的进程集合,是一个连接两个进程的连接  
器,如图 5.144 所示。管道的命令操作符是“|”,它将操作符左侧命令的输出信息(STDOUT)  
作为操作符右侧命令的输入信息(STDIN)。可以用下面的图形示意,Command1 正确输出,  
作为 Command2 的输入,然后 Command2 的输出作为 Command3 的输入,Command3  
输出则会直接显示在屏幕上。

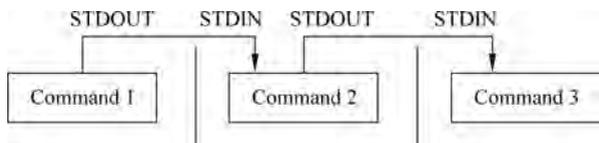


图 5.144 管道示意图

从功能上说,管道类似于输入输出重定向,但是管道触发的是两个子进程,分别执行“|”  
两边的程序,而重定向执行的是一个进程。一般如果是命令间传递参数,还是管道的好,如  
果处理输出结果需要重定向到文件,还是用输出重定向比较好。

使用管道时需要注意以下几点。

- (1) 管道是单向的,一端只能输入,另一端只能用于输出,遵循“先进先出”原则。
- (2) 管道命令只处理前一个命令的正确输出,如果输出的是错误信息,则不处理。
- (3) 管道操作符右侧命令,必须能够接收标准输入流命令。

管道分为普通管道和命名管道两种。这里所讲的管道是普通管道。

**例 5.104** 查找文件内容并显示、统计,如图 5.145 所示。

```
$ cat a1.txt|grep "a"          查找文件 a1.txt 中包含 a 的字符串并显示。
$ cat a1.txt|grep "a"|wc -l    查找文件 a1.txt 中包含 a 的字符串并统计行数,随后显示统计结果。
```

**例 5.105** 查找文件内容进行统计后显示,如图 5.146 所示。

```
$ cat 1.txt|grep "s"|wc -l    统计文件 1.txt 中包含 s 的字符串行数,并显示。
$ cat 1.txt|grep "s"|wc -w    统计文件 1.txt 中包含 s 的字符串字数,并显示。
```

```
malimei@malimei-virtual-machine:~$ cat a1.txt|grep "a"
asdfg
axdce
malimei@malimei-virtual-machine:~$ cat a1.txt|grep "a"|wc -l
2
```

图 5.145 查找、统计并显示文件内容

```
malimei@malimei-virtual-machine:~$ wc -lcw 1.txt
4 4 9 1.txt
malimei@malimei-virtual-machine:~$ cat 1.txt
ss
s
s
d
malimei@malimei-virtual-machine:~$ cat 1.txt|grep "s"|wc -l
3
malimei@malimei-virtual-machine:~$ cat 1.txt|grep "s"|wc -w
3
malimei@malimei-virtual-machine:~$ cat 1.txt|grep "s"|wc -c
7
```

图 5.146 查找、统计并显示

`$ cat 1.txt|grep "s"|wc -c` 统计文件 1.txt 中包含 s 的字节数,并显示。

**例 5.106** 查找命令的输出,如图 5.147 所示。

`$ ls -l | grep '^d'` 通过管道过滤 `ls -l` 输出的内容,只显示以 d 开头的行(即只显示当前目录中的目录/子目录)。

```
malimei@malimei-virtual-machine:~$ ls -l |grep "^d"
drwxr-xr-x 3 root root 4096 10月 12 10:44 bb
drwxrwxr-x 2 malimei malimei 4096 9月 12 11:22 deja-dup
drwxr-xr-x 2 malimei malimei 4096 9月 3 19:27 公共的
drwxr-xr-x 2 malimei malimei 4096 9月 3 19:27 模板
drwxr-xr-x 2 malimei malimei 4096 9月 3 19:27 视频
drwxr-xr-x 2 malimei malimei 4096 9月 3 19:27 图片
drwxr-xr-x 2 malimei malimei 4096 9月 3 19:27 文档
drwxr-xr-x 2 malimei malimei 4096 9月 3 19:27 下载
drwxr-xr-x 2 malimei malimei 4096 9月 3 19:27 音乐
drwxr-xr-x 2 malimei malimei 4096 9月 3 19:27 桌面
```

图 5.147 查找 ls 命令的输出内容

## 5.5 链 接

### 5.5.1 什么是链接

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。例如,当我们需要在不同的目录用到相同的文件时,不需要在每一个需要的目录下都放一个必须相同的文件,只要在某个固定的目录下存放该文件,然后在其他的目录下链接它就可以,不必重复地占用磁盘空间。

### 5.5.2 索引节点

要了解链接,首先得了解一个概念,即索引节点(inode)。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),每个文件都有一个唯一的 inode 号。我们可以将 inode 简单理解成一个指针,它永远指向本文件的具体存储位置。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。系统是通过索引节点(而不是文件名)来定位每一个文件的。

### 5.5.3 两种链接

Linux 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(Symbolic Link)。

(1) 硬链接:硬链接就是一个指针,指向文件索引节点,但系统并不为它重新分配 inode,不占用实际空间。硬链接不能链接到目录和不同文件系统的文件。

ln 命令用来建立硬链接。

(2) 软链接:软链接又叫符号链接,这个文件包含另一个文件的路径名,系统会为其重新分配 inode,类似于 Windows 中的快捷方式。软链接可以是任意文件或目录,包括不同文件系统的文件和不存在的文件名。

ln -s 命令用来建立软链接。

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,所以软链接使用更多。

### 5.5.4 链接命令 ln

**功能描述:**为某一个文件在另外一个位置建立一个同步的链接。ln 命令会保持每一处链接文件的同步性,也就是说,不论改动了哪一处,其他的文件都会发生相同的变化。

**语法:**ln [参数][源文件或目录][目标文件或目录]

**选项:**ln 命令的常用选项如表 5.38 所示。

表 5.38 ln 命令的常用选项

选 项	作 用
-s	软链接(符号链接)
-b	删除,覆盖以前建立的链接
-d	允许超级用户制作目录的硬链接
-f	强制执行
-i	交互模式,文件存在则提示用户是否覆盖
-n	把符号链接视为一般目录
-v	显示详细的处理过程
-S	"-S<字尾备份字符串>"或"--suffix=<字尾备份字符串>"
-V	"-V<备份方式>"或"--version-control=<备份方式>"

**例 5.107** 建立文件符号链接和硬链接,如图 5.148 所示。

```
$ ln -s x.txt xx.txt      建立 x.txt 文件的符号链接 xx。  
$ ln x.txt xxy.txt       建立 x.txt 文件的硬链接 xxy.txt。
```

```
malimei@malimei-virtual-machine: ~/cc
总用量 581912
-rwx--x-w- 1 malimei malimei      25 8月 16 09:32 cc1
-rw-rw-r-- 1 malimei malimei      180 8月 16 11:33 cc1.zip
drwxrwxr-x 2 malimei malimei     4096 8月 16 11:35 d1
drwxrwxr-x 2 malimei malimei     4096 8月 16 12:23 d11
drwxrwxr-x 3 malimei malimei     4096 8月 16 11:16 d111
-rw-rw-r-- 1 malimei malimei 107070761 8月 16 10:25 d11.tar
-rw-r--r-- 1 malimei malimei      8980 8月 16 09:31 examples.desktop
-rw-rw-r-- 1 malimei malimei        16 8月 16 12:18 lx
-rw-rw-r-- 1 malimei malimei    1584023 8月 16 10:42 u.tar
-rw-rw-r-- 1 malimei malimei 485456668 8月 16 10:59 VM.tar
-rw-rw-r-- 1 malimei malimei        16 8月 16 17:55 x.txt
-rw-r--r-- 1 malimei malimei    1719632 8月 16 10:42 下载ubuntu.docx
malimei@malimei-virtual-machine:~/cc$ ln -s x.txt xx.txt
malimei@malimei-virtual-machine:~/cc$ ln x.txt xxy.txt
malimei@malimei-virtual-machine:~/cc$ ls -il
总用量 581916
135643 -rwx--x-w- 1 malimei malimei      25 8月 16 09:32 cc1
135661 -rw-rw-r-- 1 malimei malimei      180 8月 16 11:33 cc1.zip
135265 drwxrwxr-x 2 malimei malimei     4096 8月 16 11:35 d1
135638 drwxrwxr-x 2 malimei malimei     4096 8月 16 12:23 d11
135644 drwxrwxr-x 3 malimei malimei     4096 8月 16 11:16 d111
135652 -rw-rw-r-- 1 malimei malimei 107070761 8月 16 10:25 d11.tar
134827 -rw-r--r-- 1 malimei malimei      8980 8月 16 09:31 examples.desktop
135665 -rw-rw-r-- 1 malimei malimei        16 8月 16 12:18 lx
135655 -rw-rw-r-- 1 malimei malimei    1584023 8月 16 10:42 u.tar
135656 -rw-rw-r-- 1 malimei malimei 485456668 8月 16 10:59 VM.tar
135672 -rw-rw-r-- 2 malimei malimei        16 8月 16 17:55 x.txt
135673 lrwxrwxrwx 1 malimei malimei         5 8月 16 17:55 xx.txt -> x.txt
135672 -rw-rw-r-- 2 malimei malimei        16 8月 16 17:55 xxy.txt
135654 -rw-r--r-- 1 malimei malimei    1719632 8月 16 10:42 下载ubuntu.docx
malimei@malimei-virtual-machine:~/cc$
```

图 5.148 建立文件符号链接和硬链接

从图 5.148 可以看到建立符号链接的索引节点号(inode)的改变,原文件索引节点号为 135672,链接文件的索引节点号为 135673,属性链接数目变为 2,链接文件权限的第一位用 l 表示。建立硬链接的索引节点号(inode)不变,仍为 135672,链接数目变为 2。

**例 5.108** 建立文件-目录的符号链接,如图 5.149 所示。

\$ ln -s s2 aa 在 aa 目录下建立 s2 文件的符号链接,默认命名为 s2。

```
malimei@malimei-virtual-machine:~$ ln -s s2 aa
malimei@malimei-virtual-machine:~$ ls -l
总用量 36
drwxrwxr-x 2 malimei malimei 4096 10月 14 11:34 aa
-rw-rw-r-- 2 malimei malimei 0 10月 14 10:34 s1
-rw-rw-r-- 2 malimei malimei 0 10月 14 10:34 s2
lrwxrwxrwx 1 malimei malimei 2 10月 14 11:33 s3 -> s2
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 公共的
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 模板
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 视频
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 图片
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 文档
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 下载
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 音乐
drwxr-xr-x 2 malimei malimei 4096 10月 7 12:14 桌面
malimei@malimei-virtual-machine:~$ cd aa
malimei@malimei-virtual-machine:~/aa$ ls
s2
malimei@malimei-virtual-machine:~/aa$ ls -l
总用量 0
lrwxrwxrwx 1 malimei malimei 2 10月 14 11:34 s3 -> s2
malimei@malimei-virtual-machine:~/aa$
```

图 5.149 在目标目录下建立符号链接

例 5.109 建立文件-目录的硬链接,如图 5.150 所示。

\$ ln s1 aa 在 aa 目录下建立文件 s1 的硬链接。

```
malimei@malimei-virtual-machine:~$ ln s1 aa
malimei@malimei-virtual-machine:~$ ls -l
总用量 36
drwxrwxr-x 2 malimei malimei 4096 10月 14 11:38 aa
-rw-rw-r-- 3 malimei malimei 0 10月 14 10:34 s1
-rw-rw-r-- 3 malimei malimei 0 10月 14 10:34 s2
lrwxrwxrwx 1 malimei malimei 2 10月 14 11:33 s3 -> s2
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 公共的
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 模板
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 视频
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 图片
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 文档
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 下载
drwxr-xr-x 2 malimei malimei 4096 8月 29 18:26 音乐
drwxr-xr-x 2 malimei malimei 4096 10月 7 12:14 桌面
malimei@malimei-virtual-machine:~$ cd aa
malimei@malimei-virtual-machine:~/aa$ ls -l
总用量 0
-rw-rw-r-- 3 malimei malimei 0 10月 14 10:34 s1
lrwxrwxrwx 1 malimei malimei 2 10月 14 11:34 s3 -> s1
malimei@malimei-virtual-machine:~/aa$
```

图 5.150 在目标目录下建立硬链接

例 5.110 建立目录的符号链接和硬链接,将看到不允许给目录建立硬链接,如图 5.151 所示。

```
malimei@malimei-virtual-machine:~/cc$ ls -l
总用量 581916
-rwx--x-w- 1 malimei malimei 25 8月 16 09:32 cc1
-rw-rw-r-- 1 malimei malimei 180 8月 16 11:33 cc1.rtf
drwxrwxr-x 2 malimei malimei 4096 8月 16 11:35 d1
drwxrwxr-x 2 malimei malimei 4096 8月 16 12:23 d11
drwxrwxr-x 3 malimei malimei 4096 8月 16 11:16 d111
-rw-rw-r-- 1 malimei malimei 107070761 8月 16 10:25 d11.tar
-rw-r--r-- 1 malimei malimei 8980 8月 16 09:31 examples.desktop
-rw-rw-r-- 2 malimei malimei 16 8月 16 12:18 lx
-rw-rw-r-- 1 malimei malimei 1584023 8月 16 10:42 u.tar
-rw-rw-r-- 1 malimei malimei 485456668 8月 16 10:59 VM.tar
-rw-rw-r-- 2 malimei malimei 16 8月 16 17:55 x.txt
lrwxrwxrwx 1 malimei malimei 5 8月 16 17:55 xx.txt -> x.txt
-rw-rw-r-- 2 malimei malimei 16 8月 16 17:55 xxy.txt
-rw-r--r-- 1 malimei malimei 1719632 8月 16 10:42 下载ubuntu.docx
malimei@malimei-virtual-machine:~/cc$ ln -s d1 ddi
malimei@malimei-virtual-machine:~/cc$ ln d1 dying
ln: d1: 不允许将硬链接指向目录
malimei@malimei-virtual-machine:~/cc$
```

图 5.151 目录不能建立硬链接

尽管硬链接节省空间,也是 Linux 系统整合文件系统的传统方式,但是存在一些不足之处。

(1) 不允许给目录创建硬链接。

(2) 不可以在不同文件系统的文件间建立硬链接。因为 inode 是这个文件在当前分区中的索引值,是相对于这个分区的,因此不能跨越文件系统了。

软链接克服了硬链接的不足,没有任何文件系统的限制,任何用户可以创建指向目录的符号链接。因而现在使用更为广泛,它具有更大的灵活性,甚至可以跨越不同机器、不同网络对文件进行链接。

硬链接和软链接的区别如下。

(1) 硬链接原文件/链接文件公用一个 inode 号,说明它们是同一个文件,而软链接原文件/链接文件拥有不同的 inode 号,表明它们是两个不同的文件。

(2) 在文件属性上软链接明确写出了是链接文件,而硬链接没有写出来,因为在本质上硬链接文件和原文件是完全平等关系。

(3) 链接数目是不一样的,软链接的链接数目不会增加。

(4) 文件大小是不一样的,硬链接文件显示的大小是跟原文件一样的,而这里软链接显示的大小与原文件就不同。

(5) 软链接没有任何文件系统的限制,任何用户可以创建指向目录的符号链接。

总之,建立软链接就是建立了一个新文件。当访问链接文件时,系统就会发现它是个链接文件,它读取链接文件找到真正要访问的文件。

当然软链接也有硬链接没有的缺点:因为链接文件包含原文件的路径信息,所以当原文件从一个目录下移到其他目录中,再访问链接文件,系统就找不到了;而硬链接就没有这个缺陷,想怎么移就怎么移。软链接要系统分配额外的空间用于建立新的索引节点和保存原文件的路径。

## 习 题

### 1. 填空题

- (1) Linux 操作系统支持很多现代的流行文件系统,其中\_\_\_\_\_文件系统使用最广泛。
- (2) Linux 系统中,没有磁盘的逻辑分区(即没有 C 盘、D 盘等),任何一个种类的文件系统被创建后都需要\_\_\_\_\_到某个特定的目录才能使用,这相当于激活一个文件系统。
- (3) Linux 采用的是\_\_\_\_\_拓扑结构,最上层是根目录。
- (4) 当前用户为 ma,则登录后进入的主目录为\_\_\_\_\_。
- (5) 查看文件的内容常用命令有:\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- (6) cp 命令可以复制多个文件,将要复制的多个文件由\_\_\_\_\_分隔开。
- (7) 使用 touch 命令,创建一个\_\_\_\_\_文件。
- (8) rm 命令只能删除文件,不能删除目录,如果删除目录必须加参数\_\_\_\_\_。
- (9) 管道的命令就是将操作符左侧命令的输出信息作为操作符右侧命令的\_\_\_\_\_。
- (10) 命令 \$ cd ~ 是切换到\_\_\_\_\_。

### 2. 问答题

- (1) Ubuntu 根目录下有哪些重要的目录? 各存放了什么信息?
- (2) Ubuntu 下有哪些文件类型?
- (3) 使用 ls -l 命令可以查看文件的详细属性,说明图 5.152 中各列信息的含义。

```
-rw-rw-r-- 1 malimei malimei      0 8月 20 2020 dd
-rw-r--r-- 1 malimei malimei 8980 7月 19 2019 examples.desktop
lrwxrwxrwx 1 malimei malimei      1 12月 17 16:38 rss -> a
drwxrwxrwx 2 malimei malimei 4096 12月 10 16:21 share
drwxrwxr-x 2 malimei malimei 4096 5月 13 16:25 ss
```

图 5.152 查看文件的详细属性

(4) 使用 `chmod` 命令设置文件权限的两种模式是什么？它们分别采用什么方法来描述权限？

(5) `find` 命令查找文件有哪些方式？

(6) 什么是输入输出重定向？如何将命令输出保存到新文件？

(7) 什么是管道？分析其与重定向的异同。

(8) Ubuntu 中两种链接方式是什么？并分析其不同。

### 3. 实验题

(1) 在你的用户下面，建立空文件 `file1`、`file2`、`file3`、`file4`，建立目录 `dir1`、`dir2`。

(2) 建立目录 `dir3`，其权限为 `442`。

(3) 建立 `file1` 文件的符号链接，自定文件名；建立 `file1` 文件的硬链接，自定文件名；用 `ls` 命令加参数显示索引节点号，比较硬链接和符号链接的不同。

(4) 在目录下建立链接：建立源文件为 `file2`，目标为 `dir1` 目录的符号链接，自定文件名；建立源文件为 `file2`，目标为 `dir1` 目录的硬链接，自定文件名。

(5) 跨目录建立链接：建立源文件为 `file2`，目标为 `/home` 目录的符号链接，自定文件名；建立源文件为 `file2`，目标为 `/home` 目录的硬链接，自定文件名，是否可以？

(6) 把 `file3` 文件复制到 `dir1` 目录下。

(7) 用 `tar` 压缩 `dir1` 目录，自定名字；用 `gzip` 压缩 `dir1` 目录，比较不同点。

(8) 把 `file4` 文件移动到 `dir2` 目录下。

(9) 更改组和所有者：更改 `dir2` 目录和目录中的文件的组和所有者为你登录的用户名。

(10) 改变文件和目录的权限：用符号模式更改 `dir2` 目录和目录中的文件权限为所有者具有全权，同组人具有读和写的权限，其他人只有执行的权限。

(11) 改变文件和目录的权限：用绝对模式更改 `dir1` 目录和目录中的文件权限为 `421`。

(12) 查找文件：查找根目录下所有以 `“.conf”` 为扩展名的文件。

(13) 查找根目录及其子目录下所有最近 20 天内访问过的文件。

(14) 在 `dir1` 目录下查找小于 10B 的文件。

(15) 查找文件内容：在自己的目录下查找以 `f` 开头的文件名，并在这些文件中搜索包含 `“is”` 的行。

(16) 删除 `dir1`、`dir2`、`dir3` 目录。