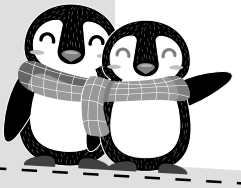

Linux

指令范例速查手册

（第3版）

黄照鹤◎编著

清華大學出版社
北京

内 容 简 介

本书是获得大量读者好评的“Linux 典藏大系”中的《Linux 指令范例速查手册》的第 3 版。本书第 1、2 版出版后获得了大量读者的好评。本书结合 653 个典型示例和 424 个经验技巧，详细介绍常见的 426 个 Linux 指令的用法，是一本编排科学、查询方便的手册。本书按照 Linux 指令的功能分章讲解，每章介绍的指令按照其重要程度和使用频率排序，每个指令除了介绍其基本语法、选项和参数外，还给出使用经验和技巧，并提供典型示例，便于读者积累丰富的实战经验。**本书提供 492 分钟教学视频、思维导图、教学 PPT 和习题参考答案等超值配套资源，帮助读者高效、直观地学习。**

本书共 25 章，分为 3 篇。第 1 篇涵盖文件与目录操作、文本编辑、文本过滤与处理、备份与压缩、Shell 内部操作、关机、打印和其他操作等 Linux 基础操作方面的 170 个常用指令；第 2 篇涵盖用户和工作组管理、硬件管理、磁盘管理、文件系统管理、进程与作业管理、性能监测与优化、内核与模块管理、X-Window 系统管理、软件包管理、系统安全管理、编程开发等 Linux 系统管理方面的 184 个常用指令；第 3 篇涵盖网络配置、网络测试、网络应用、高级网络管理、网络服务器管理、网络安全管理等 Linux 网络管理方面的 72 个常用指令。附录部分提供了按英文字母排序的 Linux 指令索引表，便于读者查询。

本书指令全面，讲解详细，查询方便，实用性强，适合 Linux 初学者、Linux 运维管理人员、Linux 系统开发人员和 Linux 爱好者作为案头查询手册。

图书在版编目（CIP）数据

Linux 指令范例速查手册 / 黄照鹤编著. -- 3 版.
北京 : 清华大学出版社, 2024. 12. -- (Linux 典藏大系).
ISBN 978-7-302-67593-8

Ⅰ. TP316.85-62

中国国家版本馆 CIP 数据核字第 2024KC7576 号

责任编辑：王中英
封面设计：欧振旭
责任校对：胡伟民
责任印制：刘　菲

出版发行：清华大学出版社
网　　址：https://www.tup.com.cn，https://www.wqxuetang.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-83470000　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：天津鑫丰华印务有限公司
经　　销：全国新华书店
开　　本：145mm×210mm　　**印　　张** 18.75　　**字　　数**：735 千字
版　　次：2011 年 2 月第 1 版　　2024 年 12 月第 3 版　　**印　　次**：2024 年 12 月第 1 次印刷
定　　价：99.80 元

产品编号：101192–01

前言

Linux 是开放源代码的类 UNIX 操作系统，具有安全性高、稳定可靠等特性。随着 Linux 操作系统日益发展壮大和功能不断增强，其市场份额逐年增长。目前，Linux 已经发展为全球第二大操作系统。就连微软公司的 Windows 系统都提供了 Linux 的子系统，以方便用户完成 Linux 系统的各项操作。

虽然 Linux 具有非常优秀的图形操作界面，但是其命令行操作方式更加灵活和强大。就连 Windows 提供的 Linux 子系统也是基于命令行的操作模式。Linux 有几百个常用指令，每个指令通常都有多个选项与参数，这无疑增大了掌握这些指令的难度。很多 Linux 初学者面对如此庞大的指令系统感到束手无策。不管是初学者，还是 Linux 专业人员，面对如此庞大的指令库，都需要一本比较全面的 Linux 指令手册作为日常工作和学习的参考书。

本书是获得大量读者好评的“Linux 典藏大系”中的《Linux 指令范例速查手册》的第 3 版。本书在第 2 版的基础上进行了全新改版，不但调整了一些指令对应的示例，而且修订了第 2 版中的一些疏漏，并移除了一些废弃的指令等，使其更加实用。本书详细介绍了 Linux 系统常用的 426 个指令的用法，涵盖 Linux 基础、系统管理和网络管理三大知识模块。本书在讲解的过程中穿插了 653 个典型示例和 424 个经验技巧提示。书中的每个指令按照基本语法、选项、参数、经验技巧和典型示例的体例编排，便于读者积累丰富的实战经验，同时还提供了大量的助记提示，帮助读者轻松记忆相关指令和选项。笔者还为每个指令录制了配套教学视频，帮助读者高效、直观地学习。另外，本书提供了按功能索引（目录）和英文字母索引（附录）两种检索方式，方便读者查询。

关于“Linux 典藏大系”

“Linux 典藏大系”是专门为 Linux 技术爱好者推出的系列图书，涵盖 Linux 技术的方方面面，可以满足不同层次和各个领域的读者学习 Linux 的需求。该系列图书自 2010 年 1 月开始陆续出版，上市后深受广大读者的好评。2014 年 1 月，作者对该系列图书进行了全面改版并增加了新品种。新版图书一上市就大受欢迎，各分册长期位居 Linux 图书销售排行榜前列。截至 2023

年 10 月底，该系列图书累计印数超过 30 万册。可以说，“Linux 典藏大系”是图书市场上的明星品牌，该系列中的一些图书多次被评为清华大学出版社“年度畅销书”，还曾获得“51CTO 读书频道”颁发的“最受读者喜爱的原创 IT 技术图书奖”，另有部分图书的中文繁体字版在中国台湾出版发行。该系列图书的出版得到了国内 Linux 知名技术社区 ChinaUnix（简称 CU）的大力支持和帮助，读者与 CU 社区中的 Linux 技术爱好者进行了广泛的交流，取得了良好的学习效果。另外，该系列图书还被国内上百所高校和培训机构选为教材，得到了广大师生的一致好评。

关于第 3 版

随着技术的发展，本书第 2 版与当前的 Linux 系统环境有所脱节，这给读者的学习带来了不便。应广大读者的要求，笔者对第 2 版图书进行了全面的升级改版，推出第 3 版。相比第 2 版图书，第 3 版在内容上的变化主要体现在以下几个方面：

- 对第 2 版中的一些疏漏进行修订，并对一些不够准确的内容重新表述；
- 对一些指令对应的示例进行调整，实用性更强；
- 移除一些已经废弃的指令，并增加新的指令；
- 为众多指令的选项增加助记提示，避免低效的死记硬背；
- 在每章后增加习题，帮助读者练习和巩固该章所学的指令；
- 新增思维导图，方便读者梳理所学的知识。

本书特色

1．指令全面，涵盖广泛

本书介绍 426 个常用的 Linux 指令，涵盖 Linux 基础、系统管理和网络管理三大知识模块的大部分常用指令，非常全面，可谓一册在手，万事无忧。

2．视频教学，高效、直观

本书特意为每个指令都配备教学视频（共 492 分钟），读者结合教学视频学习，更加高效、直观，可以取得更好的学习效果。

3．示例丰富，实用性强

本书在讲解每个指令时都给出对应的典型示例，全书示例达 653 个，这些示例可以帮助读者更好地理解相关指令的用法，而且读者也可以将其用于实际

工作中，非常实用。

4．总结大量的经验技巧

本书在介绍 Linux 指令的用法时穿插笔者总结的 424 个经验技巧，这些技巧对读者学习 Linux 指令有很好的启发，会给学习带来很大的帮助。

5．适用于大多数Linux发行版本

本书介绍的绝大多数指令适用于 Redhat、SUSE、Debian、Fedora 和 Ubuntu 等主流 Linux 发行版本及其延伸版本，只有极少数指令及其一些选项与主流发行版本存在一定的差别。读者无论使用哪个发行版本的 Linux 系统，基本都可以顺利使用本书。

6．提供两种检索方式，查询非常方便

本书不但提供按照功能检索（目录）的方式，而且还在附录中提供按照英文字母检索的方式，方便读者查询相关指令。

7．提供大量助记提示，方便记忆

Linux 指令的选项大多数采用字母缩写的形式。例如，“-v”选项表示指令采用冗余模式，会产生更多的输出信息。这类指令数量众多且晦涩难懂。为了解决这个问题，笔者在选项对应的解释中添加对应的单词，提示选项的来源。例如，“-v”选项的解释为“冗余（verbose）模式，提供更详细的输出信息”，读者根据括号中的提示单词，即可轻松记住该选项。

8．提供习题、思维导图和教学PPT

本书特意在每章后提供多道习题，用于帮助读者自测对该章指令的掌握情况，另外提供思维导图和教学 PPT 等配套资源，以方便读者学习和梳理相关知识。

本书内容

第1篇　Linux基础指令

本篇涵盖第 1～8 章，主要介绍文件与目录操作、文本编辑、文本过滤与处理、备份与压缩、Shell 内部操作、关机、打印和其他操作等 Linux 基础操作方面的 170 个常用指令的用法。

第2篇　Linux系统管理指令

本篇涵盖第 9～19 章，主要介绍用户和工作组管理、硬件管理、磁盘管

理、文件系统管理、进程与作业管理、性能监测与优化、内核与模块管理、X-Window 系统管理、软件包管理、系统安全管理和编程开发等 Linux 系统管理方面的 184 个常用指令的用法。

第3篇　Linux网络管理指令

本篇涵盖第 20～25 章，主要介绍网络配置、网络测试、网络应用、高级网络管理、网络服务器管理和网络安全管理等 Linux 网络管理方面的 72 个常用指令的用法。

体例说明

本书中的指令按照语法、功能介绍、选项说明、参数说明、经验技巧、示例和相关指令的体例进行讲解。如果某项内容未给出，则表示本指令没有相关内容。下面给出具体的解释。

【语法】：指令的语法说明；

【功能介绍】：介绍指令的常用功能；

【选项说明】：介绍指令的常用选项，如果没有出现该选项，则表示该指令没有任何选项；

【参数说明】：介绍指令的常用参数，如果没有出现该选项，则表示该指令没有任何参数；

【经验技巧】：介绍实际操作中的经验与技巧；

【示例】：介绍相关指令的上机操作示例；

【相关指令】：给出与本指令功能相关的其他指令，如果没有出现该选项，则表示该指令没有其他相关指令。

【★★★★★】：表示指令的使用频率，常见的指令都是 5 颗星。

读者对象

- Linux 初学人员；
- Linux 系统管理员和网络管理员；
- Linux 专业技术人员；
- Linux 爱好者和研究人员；
- 大中专院校的学生；
- 相关培训班的学员。

配套资源获取方式

本书涉及的配套资源如下：

- 教学视频；
- 思维导图；
- 教学 PPT；
- 习题参考答案。

上述配套资源有 3 种获取方式：关注微信公众号“方大卓越”，然后回复数字“35”自动获取下载链接；在清华大学出版社网站（www.tup.com.cn）上搜索到本书，然后在本书页面上找到“资源下载”栏目，单击“网络资源”按钮进行下载；在本书技术论坛（www.wanjuanchina.net）上的 Linux 模块进行下载。

技术支持

虽然笔者对书中所述内容都尽量予以核实，并多次进行文字校对，但是因时间所限，可能还存在疏漏和不足之处，恳请读者批评与指正。读者在阅读本书时若有疑问，可以通过以下方式获得帮助：

加入本书 QQ 交流群（群号为 302742131）进行提问；

在本书技术论坛（见上文）上留言，会有专人负责答疑；

发送电子邮件到 book@ wanjuanchina.net 或 bookservice2008@163.com 获得帮助。

黄照鹤

2024 年 11 月

目录

第 1 篇　Linux 基础指令

第 2 篇 Linux 系统管理指令

第 3 篇 Linux 网络管理指令

第 1 篇
Linux 基础指令

第 1 章　文件与目录操作

文件管理是操作系统的重要功能。在 Linux 中，所有的软硬件资源都被认为是特殊的文件。本章将介绍 Linux 的普通文件以及和目录相关的操作指令，这些指令是 Linux 管理员必须掌握的基础指令。

1.1　ls 指令：显示目录内容

【语　　法】ls [选项] [参数]　　★★★★★

【功能介绍】ls 指令用来列出（list）目录内容，在 Linux 系统中有较高的使用率。ls 指令的输出信息可以进行彩色加亮显示，以区分不同类型的文件。

【选项说明】

选　　项	功　　能
-a	显示包括隐藏文件（文件名以“.”开头）在内的所有（all）文件
-A	显示除了隐藏文件“.”和“..”以外的所有（all）文件列表
-C	多列（columns）显示输出结果。这是默认选项
--color[=WHEN]	使用不同的颜色（color）高亮显示不同类型的文件。可选值包括never、always和auto
-F	在每个输出项后追加文件（file）的类型标识符。具体含义为：“*”表示具有可执行权限的普通文件；“/”表示目录；“@”表示符号链接；“|”表示命名管道FIFO；“=”表示sockets套接字。当文件为普通文件时，不输出任何标识符
-b	将文件名中的不可输出字符以反斜线“\”加字符编码的方式输出
-B	不列出任何以~字符结束的项目
-c	与-lt选项连用时，按照文件的状态改变时间来排序并输出目录内容，排序依据是文件的索引节点中的ctime字段；与“-l”选项连用时，排序的依据是文件的状态改变时间
-d	仅显示目录（directory）名，而不显示目录下的内容列表；显示符号链接文件本身，而不显示其所指向的目录列表
-D	产生适合Emacs的dired模式使用的结果
-f	按照文件在磁盘上的存储顺序显示列表，对输出内容不进行排序操作。-f选项具有-a选项的功能，可以显示隐藏文件。不能和-f连用的选项有-l、--color和-s

续表

选　　项	功　　能
-F	加上文件（file）类型的指示符号（*、/、=、@、|其中的一个）
--file-type	与-F选项的功能相同，但是不显示*号
--format=WORD	关键字可以是across -x，commas -m，horizontal -x，long -l，single-column -1,verbose -l，vertical -c
--full-time	显示完整的日期时间，而不是使用标准的缩写。ls指令的日期时间格式与指令date的默认格式相同
-g	与-l选项的功能相同，但不列出所有者
--group-directories-first	在文件前分组目录。该选项与--sort一起使用，但是一旦使用--sort=none(-U)将禁用分组
-G	以一个长列表的形式输出，但不输出组（group）名
-h	与-l和-s一起使用，以人类（human）易于阅读的格式输出文件大小
--si	与-h选项类似，但是使用1000为基底而非1024
-H	使用命令列中的符号链接指示真正的目的地
--dereference-command-line-symlink-to-dir	跟随命令行中的指向目录的符号链接
--hide=PATTERN	不列出与Shell模式匹配的隐含条目
--hyperlink[=WHEN]	文件名使用超链接。WHEN可以是always（默认）、auto或never
--indicator-style=WORD	指定在每个项目名称后加上指示符号<方式>：none（默认）、slash(-p)、file-type(--file-type)及classify(-F)
-i	显示文件的索引节点号（inode）。一个索引节点代表一个文件
-I	忽略（ignore）任何匹配指定Shell<模式>的项目
-k	以KB（千字节）为单位显示文件大小
-l	以长（long）格式显示目录下的内容列表。输出的信息从左到右依次包括文件名、文件类型、权限模式、硬链接数、所有者、组、文件大小和文件的最后修改时间等。其与-C选项功能相反，所有输出信息以单列格式输出，不输出为多列
-L	忽略符号链接本身的信息，显示符号链接所指向的目标文件的信息
-m	以水平方式显示文件(每个文件之间用“,”和一个空格隔开)，以便每行显示尽可能多的文件数
-n	文件所属的用户和组使用用户ID号和组ID号表示。使用此选项时将自动采用长格式输出目录列表
-N	输出未经引号引起来的项目名称

续表

选　　项	功　　能
-o	类似-l选项，但不列出有关组的信息
-p	对目录附加“/”作为指示符号
-q	以“?”字符代替无法打印的字符
--show-control-chars	原样显示无法打印的字符
-Q	将条目名称加上双引号（quote）
--quoting-style=WORD	使用指定引用的方式显示条目的名称。其中，可指定的条目名称包括literal、locale、shell、shell-always、shell-escape、shell-escape-always、c、escape
-r	以文件名反序（reverse order）排列并输出目录列表；否则按照文件名升序显示目录列表
-R	递归（recursive）显示目录下的所有文件列表和子目录列表
-s	以块（1块=1024字节）为单位显示文件的大小（size）
-S	根据文件大小排序（sort）
--sort=WORD	根据关键字代替名字排序。其中，可指定的关键字有none（-U）、size（-S）、time（-t）、version（-v）、extension（-X）
--time=WORD	更改使用的修改时间的次数
--time-style=TIME_STYLE	设置显示时间的格式（time format）
-t	按照文件的最后修改时间（time）降序显示目录内容列表，最近修改过的文件显示在前面
-T	指定制表符（Tab）的宽度，而非默认的8个字符
-u	当与-lt选项一起使用时，按照访问时间排序并显示；当与-l选项一起使用时，显示访问时间并按文件名排序。其他情况，按照访问时间排序，最新的最靠前
-U	不进行排序（unsort），按照目录顺序列出项目
-v	在文本中进行数字的自然排序
-x	逐行列出项目而不是逐栏列出
-X	根据扩展名（extension）按字母顺序进行排序
-Z	输出每个文件所有安全的上下文信息
-1	每行只列出一个文件。当与-q或-b一起使用时，避免'\n'
--help	显示命令帮助信息并退出
--version	显示版本信息并退出

【参数说明】

参　　数	功　　能
目录	指定要显示列表的目录，也可以是具体的文件

【经验技巧】

- ls 指令来自 coreutils 软件包，此软件包中还包含 dir 指令，此指令与 ls 指令的功能相同，因此不再单独介绍 dir 指令。
- ls 指令的--color 选项可以使其输出内容按照文件类型用彩色加亮显示。大部分的 Linux 系统默认情况下都已经设置了命令别名 alias ls --color=tty，所以在使用 ls 指令时不必再加上此选项。
- 当结合管道符号“|”使用 ls 指令时，ls 指令的输出结果送入管道后将失去彩色加亮功能。
- 使用 ls 指令的-l 选项以长（long）格式输出文件属性，输出信息的第一列为权限信息。它们代表的含义为：r 表示读（read）权限；w 表示写（write）权限；x 表示执行（execute）权限；“-”表示没有权限。
- 当使用-l 选项时，可执行权限位可能出现 s、S、t 和 T 字母。它们代表的含义为：s 表示 setuid、setgid 或可执行权限；S 表示虽然具有 setuid 权限或 setgid 权限，但是文件没有可执行权限；t 表示 sticky 权限，同时文件还具有可执行权限；T 表示文件具有 sticky 权限，但是没有可执行权限；x 表示文件仅具有可执行权限，不具有其他的特殊权限。
- 使用 ls 指令的-i 选项可以显示文件的索引节点号，具有相同索引节点号的文件本质上是同一个文件，因此其内容完全相同。
- 默认情况下 ls 指令只能显示非隐藏文件，如果要显示所有的文件列表，则必须使用-a 选项。

【示例 1.1】显示目录列表。默认情况下 ls 指令只能显示非隐藏文件，本例使用 ls 指令显示当前工作目录的非隐藏文件列表。在命令行中输入下面的命令：

```
#显示当前目录下的非隐藏文件和目录，当前目录用“.”表示
[root@localhost ~]# ls .
```

输出信息如下：

```
anaconda-ks.cfg  command.txt  Desktop  install.log  install.
log.syslog  test
```

说明：本例中的“.”表示当前的工作目录，也可以省略。

【示例 1.2】显示当前工作目录下包括隐藏文件在内的所有文件列表。如果要显示目录下包括隐藏文件在内的全部文件，则必须使用 ls 指令的-a 选项。在命令行中输入下面的命令：

```
#显示当前工作目录下包括隐藏文件在内的所有文件列表
[root@localhost ~]# ls -a .
```

输出信息如下：

```
.              .bashrc     .eggcups   .gnome2_private  .lesshst
test
..             .chewing    .esd_auth  .gstreamer-0.10.metacity
 .Trash
......省略部分输出内容......
.bash_profile    .dmrc       .gnome2    install. log.sy-slog  .
tcshrc
```

说明：可以发现，本例的输出文件比示例 1.1 的多，包含所有以点开头的隐藏文件和隐藏目录。

【示例 1.3】输出长格式列表。默认情况下，ls 指令仅列出目录下的文件列表，并不包含文件的详细信息。使用-l 选项可以得到文件的详细信息（包括文件类型、权限、文件大小、用户和组信息等）。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l .     #以长格式显示当前目录下的内容
```

输出信息如下：

```
total 80
drwxr-xr-x 2 root root    4096 Apr 15 23:08 Desktop
......省略部分输出内容......
lrwxrwxrwx 1 root group2    11 Apr 17 06:34 test -> install.log
```

说明：在长格式的输出信息中，每列代表一个文件，第一列表示文件类型和权限（权限信息共 9 个字符，每 3 个字符为一组，分别表示文件所有者的权限、工作组的权限和其他用户的权限），第二列表示文件的链接数（这里指的是硬链接），第三列表示文件的所有者，第四列表示文件所属的工作组，第五列表示文件的大小，第六列表示文件最后一次访问的时间，第七列表示文件名称。

【示例 1.4】显示文件的 inode 信息。索引节点（index node 简称为 inode）是 Linux 中的一个特殊概念，具有相同的索引节点号的两个文件本质上是同一个文件（除了文件名不同以外），使用 ls 指令的-i 选项可以显示文件的索引节点号。下面查看两个文件的索引节点号。在命令行中输入命令如下：

```
[root@localhost ~]# ls -i -l file1 file2     #显示文件的索引节点号
```

输出信息如下：

```
64930 -rw-r--r-- 2 root root 33726 5 月 10  09:04 file1
64930 -rw-r--r-- 2 root root 33726 5 月 10  09:04 file2
```

说明：在上面的输出信息中，文件 file1 和 file2 具有相同的索引节点号，所以其内容完全相同。当删除任何一个文件时，另一个文件依然存在并且内容不受影响；当修改任何一个文件内容时，另一个文件内容同时也会发生变化。

【示例 1.5】水平输出文件列表。默认情况下，ls 指令以每行一个文件的方式输出列表，这种输出方式占用的屏幕空间较大。为了节省屏幕空间，可以使用-m 选项以水平紧凑方式显示文件列表信息。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -m          #目录列表的显示方式为水平紧凑方式
```

输出信息如下：

```
account, cache, crash, cvs, db, empty, ftp, games, gdm, lib, local,
lock, log,
lost+found, mail, named, nis, opt, preserve, racoon, run, spool,
tmp, www, yp
```

说明：在上面的输出信息中，每个文件之间使用逗号加一个空格隔开。当同时使用-m 选项与-l 选项时，-m 选项的功能将会失效。

【相关指令】 dir，vdir

1.2　cd 指令：将当前的工作目录切换为指定的目录

【语　　法】cd [选项] [参数]　　★★★★★

【功能介绍】cd 指令用来切换（change）用户的当前工作目录（directory）。默认情况下，单独使用 cd 指令将会切换到用户的宿主目录（由环境变量 HOME 定义）。

【选项说明】

选　项	功　能
-P	如果要切换的目标目录是一个符号链接，该选项可以直接切换到符号链接指向的目标目录。例如，“cd /test”（test为指向/home/test的符号链接）命令直接切换到/home/test目录
-L	与-P选项相反，如果要切换的目标目录是一个符号链接，该选项可以直接切换到符号链接名所代表的目录，而非符号链接所指向的目标目录。例如，cd /test（test为指向“/home/test”的符号链接）命令会直接切换到/test目录
-	当仅使用“-”一个选项时，当前的工作目录将会被切换到环境变量OLDPWD所表示的目录下

【参数说明】

参　数	功　能
目录	指定要切换的目标目录

【经验技巧】

- 在使用 cd 指令时，可以使用 Tab 键利用命令行的自动补齐功能加快参数的输入速度和准确度。
- 在 Linux 操作系统中，每个用户都有宿主目录（即 Home Directory），它是用户登录之后所在的默认目录。用户切换到其他目录后，如果希望快速回到宿主目录，则可以使用 cd、cd ~或者 cd $HOME 中的任何一个指令。

【示例 1.6】改变工作目录。如果用户希望从当前的工作目录切换到其他目录，则将目标目录传递给 cd 指令即可，具体步骤如下。

（1）使用 pwd 指令显示当前的工作目录。在命令行中输入下面的命令：

```
[root@localhost etc]# pwd              #显示当前的工作目录
```

输出信息如下：

```
/etc
```

说明：上面的输出信息表明当前的工作目录为/etc。

（2）将当前的工作目录切换为“/var/log”目录。在命令行中输入下面的命令：

```
#将当前的工作目录切换为/var/log 目录
[root@localhost etc]# cd /var/log
```

（3）再次使用 pwd 指令显示当前的工作目录。在命令行中输入下面的命令：

```
[root@localhost log]# pwd              #显示当前的工作目录
```

输出信息如下：

```
/var/log
```

说明：从上面的输出信息中可以看到，当前的工作目录已经切换为/var/log目录。

【示例 1.7】快速返回用户的宿主目录。如果用户希望快速返回到宿主目录，则可以使用不带任何参数和选项的 cd 指令，具体步骤如下。

（1）使用 pwd 指令显示当前的工作目录。在命令行中输入下面的命令：

```
[root@localhost httpd]# pwd            #显示当前的工作目录
```

输出信息如下：

```
/var/log/httpd
```

说明：从上面的输出信息中可以看到，当前的工作目录为/var/log/httpd。

（2）使用 cd 指令返回用户的宿主目录。在命令行中输入下面的命令：

```
[root@localhost httpd]# cd                #返回用户的宿主目录
```

说明：上面的命令没有任何输出信息。

（3）使用 pwd 指令显示当前所在的目录。在命令行中输入下面的命令：

```
[root@localhost ~]# pwd                   #显示当前工作目录
```

输出信息如下：

```
/root
```

说明：工作目录已经切换到了 root 用户的宿主目录。

【示例 1.8】-P 选项的用法。在 Linux 系统中可以使用符号链接实现类似快捷方式的功能，符号链接是一类特殊的文件，它保存了所指目录的真实的路径信息。使用 cd 指令的-P 选项可以切换到符号链接指向的实际目录，具体步骤如下。

（1）使用 pwd 指令显示当前的工作目录。在命令行中输入下面的命令：

```
[root@localhost ~]# pwd                   #显示当前的工作目录
```

输出信息如下：

```
/root
```

（2）使用 ls 指令的-l 选项显示符号链接文件所指向的实际目录。在命令行中输入下面的命令：

```
#文件 etc 为符号链接，其指向的实际目录为/etc
[root@localhost ~]# ls -l etc
```

输出信息如下：

```
lrwxrwxrwx 1 root root     5 05-13 09:03 bin -> /etc/
```

说明：从上面的输出中可以看到，符号链接文件 etc 指向的实际目录为/etc。

（3）使用 cd 指令的-P 选项切换到符号链接 etc。在命令行中输入下面的命令：

```
[root@localhost root]# cd -P etc      #切换到 etc 所指向的目录/etc
```

（4）使用 pwd 指令显示当前所在的目录。在命令行中输入下面的命令：

```
[root@localhost etc]# pwd                 #显示当前工作目录
```

输出信息如下：

```
/etc
```

说明：用户的当前工作目录变成了/etc。从本例中可以看出，使用-P 选项时切换的目录是符号链接所指向的实际目录/etc，而不是符号链接所代表的目录/root/etc。

【示例 1.9】-L 选项的用法。cd 指令的-L 选项可以使当前的工作目录切换到符号链接所代表的目录，具体步骤如下。

（1）使用 pwd 指令显示当前的工作目录。在命令行中输入下面的命令：

```
[root@localhost root]# pwd              #显示当前的工作目录
```

输出信息如下：

```
/root
```

说明：从上面的输出信息中可以看到，当前所在路径为/root。

（2）使用 ls 指令的-l 选项显示符号链接。在命令行中输入下面的命令：

```
#“etc”为符号链接，其指向的实际目录为/etc
[root@localhost root]# ls -l etc
```

输出信息如下：

```
lrwxrwxrwx    1 root   root      9  2 月 27 00:43 etc -> /etc/
```

说明：从上面的输出信息中可以看到，符号链接 etc 指向了“/etc”目录。

（3）使用 cd 指令的-L 选项切换到 etc。在命令行中输入下面的命令：

```
[root@localhost root]# cd -L etc     #切换到“/root/etc”目录
```

说明：上面的命令没有任何输出信息。

（4）使用 pwd 指令显示当前的工作目录。在命令行中输入下面的命令：

```
[root@localhost etc]# pwd              #显示当前的工作目录
```

输出信息如下：

```
/root/etc
```

说明：用户的当前工作目录变成了/root/etc。默认情况下，cd 指令切换到符号链接时与-L 选项的功能相同，所以通常可以省略-L 选项。

【相关指令】pwd

1.3　cp 指令：复制文件或目录

【语　　法】cp [选项] [参数]　　★★★★★

【功能介绍】cp 指令用来将一个或者多个源文件或者目录复制（copy）到指定的目标文件或目录中。它可以将单个源文件复制到一个指定文件名的具体文件中或者复制到一个已经存在的目录下。cp 指令还支持同时复制多个文件，当一次复制多个文件时，目标文件参数必须是一个已经存在的目录，否则将会出现错误。

【选项说明】

选　项	功　能
-a	保持源文件的所有（all）特征，如原有结构和属性，与选项-dpR的功能相同
-d	如果复制的源文件是符号链接，仅复制符号链接本身，而且保留符号链接所指向的目标文件或者目录
-f	强制（force）覆盖已经存在的目标文件，而且不提示用户进行确认。为了防止覆盖重要文件，通常不使用此选项
-i	交互（interactive）模式，在覆盖已存在的目标文件前提示用户进行确认。使用此选项可以防止覆盖重要的文件
-l	为源文件创建硬链接（link）（与ln指令的功能相同）。此选项可以节省磁盘空间，但是要求源文件和目标文件必须在同一个分区（或文件系统）上
-p	复制文件时保持（preserve）源文件的所有者、权限信息和时间属性
-R或-r	对目录进行复制操作，此选项以递归（recursive）的操作方式将指定目录及其子目录中的所有文件复制到指定的目标目录下
-s	不进行真正的复制操作，仅为源文件创建符号（signal）链接（与ln -s指令的功能相同）
-u	更新（update）模式，当目标文件不存在或者源文件比目标文件新时才进行复制操作，否则不进行复制
-S	在备份文件时，用指定的后缀SUFFIX代替文件名的默认后缀
-b	覆盖已存在的目标文件前，将目标文件备份（backup）
-v	冗余（verbose）模式，详细显示指令执行的操作

【参数说明】

参　数	功　能
源文件	指定源文件列表。默认情况下，cp指令不能复制目录，如果要复制目录，则必须使用-R选项
目标文件	指定目标文件。当“源文件”为多个文件时，要求“目标文件”为指定的目录

【经验技巧】

- cp 指令可以一次复制多个源文件，但是要求最后一个参数必须为目录。
- 通常在使用 cp 指令时，如果目标文件存在，则系统会提示是否进行覆盖操作。这是因为绝大多数 Linux 发行版都为 cp 指令指定了命令别名

“alias cp='cp -i'”，防止管理员的误操作。

- 在 Shell 脚本编程中使用 cp 指令时，为了避免-i 选项使程序必须和用户进行交互，可以使用-f 选项，以实现强制复制而不提示用户确认。
- cp 指令具备 ln 指令的功能，使用 cp 指令的-l 和-s 选项可以为源文件创建硬链接或符号链接。需要注意，当创建硬链接时，源文件和目标文件必须在同一个文件系统内（即同一个分区）。
- 由于硬链接具有相同的索引节点号，所以使用-l 选项创建硬链接时，源文件和目标文件必须在同一个文件系统下（即在同一个磁盘分区）。

【示例 1.10】复制单个文件。当使用 cp 指令复制单个文件时，第一个参数表示源文件，第二个参数表示目标文件。在命令行中输入下面的命令：

```
#复制单个文件
[root@localhost ~]# cp -v /etc/fstab /root/fstab.bak
```

输出信息如下：

```
"/etc/fstab" -> "/root/fstab.bak"
```

说明：为了便于说明，在上例中使用了-v 选项来显示 cp 指令执行的详细过程，通常可以省略-v 选项。在上例中将“/etc/fstab”文件备份为“/root/fstab.bak”文件。

【示例 1.11】复制多个文件。当使用 cp 指令复制多个文件时，最后一个参数必须是一个已经存在的目录。在命令行中输入下面的命令：

```
#复制多个源文件
[root@localhost ~]# cp -v file1 file2 file3 Desktop/
```

输出信息如下：

```
'file1' -> 'Desktop/file1'
'file2' -> 'Desktop/file2'
'file3' -> 'Desktop/file3'
```

【示例 1.12】使用通配符简化文件名的输入。上例中的源文件名有一定的规律，所以可以借助 Shell 中的通配符来简化命令的输入。在命令行中输入下面的命令：

```
#用通配符复制多个源文件
[root@localhost ~]# cp -v file[1-3] Desktop/
```

输出信息如下：

```
'file1' -> 'Desktop/file1'
'file2' -> 'Desktop/file2'
'file3' -> 'Desktop/file3'
```

说明：可以看到，使用通配符达到了简化命令行中指令输入的效果。

【示例 1.13】复制目录。在默认情况下，cp 指令只能复制普通文件。如果要进行目录的复制操作，则必须借助-R 或者-r 选项，否则将忽略目录的复制，具体步骤如下。

（1）在命令行中输入下面的命令：

```
[root@localhost ~]# cp /etc/ /root/      #复制目录，将出现错误
```

输出信息如下：

```
cp: omitting directory '/etc/'
```

说明：由于源文件/etc 是一个目录，所以 cp 指令默认情况下忽略了复制操作。

（2）使用-R 选项后则可以正常复制。在命令行中输入下面的命令：

```
#将/etc/目录复制到“/root”目录
[root@localhost ~]# cp -R /etc/ /root/
```

说明：上面的指令正常执行时将没有任何输出信息。

【示例 1.14】创建符号链接。cp 指令在复制文件时，如果指定了-s 选项，则会为源文件建立一个符号链接文件，而不进行实际的复制操作，具体步骤如下。

（1）在命令行中输入下面的命令：

```
#为源文件创建符号链接
[root@localhost ~]# cp -v -s /etc/fstab /root/myfstab
```

输出信息如下：

```
'/etc/fstab' -> '/share/myfstab'
```

（2）使用 ls 指令显示复制后的目标文件。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l /root/myfstab #显示目标文件的详细信息
```

输出信息如下：

```
lrwxrwxrwx 1 root root 10 May 14 23:53 /root/myfstab ->
/etc/fstab
```

说明：可以看到，目标文件是源文件的一个符号链接。

【示例 1.15】创建硬链接。cp 指令的选项-l 可以为源文件创建一个硬链接，具体步骤如下。

（1）在命令行中输入下面的命令：

```
#为源文件创建硬链接
[root@localhost ~]# cp -l install.log my_install.log
```

（2）使用 ls 指令查看源文件和目标文件的索引节点号。在命令行中输入

下面的命令：

```
#显示文件的索引节点号
[root@localhost ~]# ls -i install.log my_install.log
```

输出信息如下：

```
4415042 install.log  4415042 my_install.log
```

说明：从上面的输出信息中可以看出，源文件和目标文件的索引节点号都是“4415042”，表明它们是硬链接。

【示例 1.16】提高复制操作的安全性。默认情况下在使用 cp 指令复制文件时，如果目标文件已存在，则 cp 指令会自动覆盖目标文件，而且不给出任何提示信息。这种情况很容易导致错误地覆盖掉重要文件。为了提高安全性，通常在使用 cp 指令时都加上-i 选项，以便在覆盖目标前进行提示确认。在命令行中输入下面的命令：

```
[root@localhost ~]# cp -i /etc/fstab /root/fstab #安全使用 cp 指令
```

输出信息如下：

```
cp: overwrite '/root/fstab'?y              #确认覆盖目标文件
```

说明：在上面的输出信息中，y 是用户输入的确认字符，表示覆盖目标文件，当不希望覆盖目标文件时可以输入 n。在多数 Linux 系统中，为 cp 指令设置的命令别名中已经包含-i 选项，所以在 Shell 中使用 cp 指令时将其省略也能够达到防止误操作的目的。

【相关指令】dd，ln

1.4　mv 指令：移动文件或改名

【语　　法】mv [选项] [参数]　　★★★★★

【功能介绍】mv 指令可以移动（move）文件或为文件改名。

【选项说明】

选　　项	功　　能
--backup=<备份模式>	指定目标文件存在时如何进行备份操作。支持的备份模式如下： none和off：关闭备份功能； number和t：以为文件追加数字后缀的方式进行备份； existing和nil：如果使用数字后缀的备份文件已存在，则覆盖已存在的备份文件； simple和never：进行简单的备份

续表

选　　项	功　　能
-b	当目标文件存在时，覆盖前为其创建一个备份（backup）
-f	在覆盖已存在的目标文件前不提示用户确认。此选项具有一定的风险，可能会导致覆盖重要文件
-i	交互（interactive）模式，在覆盖已存在的目标文件前提示用户确认，防止覆盖重要的文件
-n	不（do not）覆盖已存在的文件
--strip-trailing-slashes	删除源文件中的斜杠“/”
-S <后缀>	为备份文件指定后缀（suffix），不使用默认的后缀
--target-directory=<目录>	指定源文件要移动到的目标目录
-T	将参数中所有<目标文件>部分视为（treat）普通文件
-u	更新（update）模式，当源文件比目标文件新或者目标文件不存在时才执行移动操作
-v	冗余（verbose）模式，对正在发生的操作给出解释
-Z	将目标文件的SELinux安全上下文设置为默认类型

【参数说明】

参　　数	功　　能
源文件	源文件列表
目标文件	如果“目标文件”是文件名，则在移动文件的同时将其改名为“目标文件”；如果“目标文件”是目录名，则将源文件移动到“目标目录”下

【经验技巧】

- 在同一个文件系统（即同一个磁盘分区）中，无论移动的文件有多大，速度都是非常快的。但是，如果在两个不同的 Linux 磁盘分区间移动文件，速度将明显降低。这是因为在同一个分区移动文件时，仅需要修改文件对应的指针即可；但是在不同分区间移动文件时，必须执行复制操作，所以导致速度明显降低。
- 如果在同一个目录下利用 mv 指令移动文件，可以实现文件改名操作。
- 为了防止误操作而覆盖已经存在的文件，在使用 mv 指令时，最好加上 -i 选项。绝大多数的 Linux 发行版都为 mv 指令设置了命令别名“alias mv='mv -i'”，可以直接使用 mv 指令而无须添加-i 选项。

【示例 1.17】文件改名。使用 mv 指令将当前目录下的文件 oldfile 改名为 newfile。在命令行中输入下面的命令：

```
[root@localhost ~]# mv oldfile newfile #将文件 oldfile 改名为 newfile
```

【示例 1.18】批量移动文件。使用命令行的通配符将多个文件同时移动到指定目录下，具体步骤如下。

（1）使用 ls 指令显示当前的目录列表。在命令行中输入下面的命令：

```
[root@localhost ~]# ls                    #列目录内容
```

输出信息如下：

```
newfilea  newfileb  newfilec  newfiled  newdirectory
```

（2）使用 mv 指令结合 Shell 通配符移动文件。在命令行中输入下面的命令：

```
#将 4 个文件移动到指定目录下
[root@localhost ~]# mv newfile[a-d] newdirectory/
```

说明：上面的命令没有任何输出信息。本例中的“newfile[a-d]”为 Shell 的通配符，匹配了 4 个文件。

【相关指令】rename

1.5　pwd 指令：显示当前的工作目录

【语　　法】pwd [选项]　　★★★★★

【功能介绍】pwd 指令以绝对路径的方式显示（print）用户当前的工作目录（work directory）。

【选项说明】

参　数	功　能
--help	显示帮助信息
--version	显示版本信息

【经验技巧】

- 在使用 Linux 系统进行命令行操作时，经常需要在不同的目录间切换，使用 pwd 指令可以迅速地显示当前的工作目录。
- 在进行系统维护的 Shell 脚本开发时，可以结合 pwd 指令和反单引号在脚本内部实现一些特殊操作。

【示例 1.19】显示当前的工作目录。在命令行中输入下面的命令：

```
[root@localhost conf]# pwd                #显示当前的目录
```

输出信息如下：

```
/etc/httpd/conf
```

1.6　rm 指令：删除文件或目录

【语　　法】rm [选项] [参数]　　★★★★★

【功能介绍】rm 指令用于删除（remove）给定的文件和目录。

【选项说明】

选　　项	功　　能
-d	如果当前系统支持unlink系统调用，则使用unlink系统调用进行删除文件和目录（directory）的操作
-f	强制（force）执行删除操作，不提示用户进行确认。此选项容易造成误操作，要慎用
-i	以交互式（interactive）的方式提示用户进行确认是否删除文件。用户可以使用n和y进行回答。其中，n表示不删除，y表示确认删除。使用-i选项可以防止误删除
-I	在删除超过3个文件或者递归删除前提示一次并要求确认。该选项比-i选项的提示内容更少，但同样可以避免大多数错误的发生
--interactive[WHEN]	根据指定的<WHEN>进行确认提示，可指定的值有never、once或always
--one-file-system	递归删除一个层级时，跳过所有不符合命令行参数的文件系统中的文件
-r或-R	用递归（recursive）的方式删除目录及目录下的所有内容
--no-preserve-root	不对“/”进行特殊处理
--preserve-root	不对根目录进行递归操作
-v	冗余（verbose）模式，显示指令的详细执行过程

【参数说明】

参　　数	功　　能
文件	指定被删除的文件列表，如果参数中含有目录，则必须加上-r或者-R选项

【经验技巧】

- 默认情况下，rm 指令只能删除普通文件，当删除目录时必须使用-r 或-R 选项，以递归方式删除目录。
- 如果要删除的文件较多，可以结合 Shell 的通配符，以提高命令行的输入效率。
- 通常，在使用 rm 指令删除文件时，系统不会给出任何提示信息，这种情况很容易造成误删除。所以在使用 rm 指令删除文件时最好加上-i 选项，它在删除目标文件前会给出提示信息，询问是否进行覆盖，以防止

误操作。大多数的 Linux 发行版已经设置了带-i 选项的 rm 指令的别名“alias rm='rm -i'”，用户在使用 rm 指令时不必添加-i 选项。

- 使用 rm 指令的-f 选项时，不会给出提示信息而是直接执行删除操作，必须谨慎使用此选项。在 Shell 脚本编程时使用-f 选项可以避免 Shell 脚本和用户交互。

【示例 1.20】直接使用 rm 指令删除一个或多个普通文件。在命令行中输入下面的命令：

```
[root@localhost ~]# rm t1.sh                    #删除 t1.sh 文件
```

输出信息如下：

```
rm: remove regular file 't1.sh'? y              #确认删除操作
```

说明：在上例中，当删除文件“t1.sh”时需要用户通过 y 或 n 进行确认。

【示例 1.21】强制删除文件。如果同时删除多个文件则需要确认多次，为了提高效率，可以使用-f 选项。在命令行中输入下面的命令：

```
[root@www1 ~]# rm -v -f file1 file2 file3       #同时删除 3 个文件
```

输出信息如下：

```
removed 'file1'
removed 'file2'
removed 'file3'
```

说明：从上面的输出信息中可以看到，使用-f 选项后不会再提示用户确认；-v 选项可以显示指令的详细执行过程。

【示例 1.22】使用通配符删除文件。删除多个文件时还可以使用 Shell 通配符，以简化 Shell 命令行的输入。在命令行中输入下面的命令：

```
[root@www1 ~]# rm -v -f file[1-3]               #同时删除 3 个文件
```

输出信息如下：

```
removed 'file1'
removed 'file2'
removed 'file3'
```

说明：使用 Shell 通配符达到了同样的效果，但是简化了命令行的输入。

【示例 1.23】删除目录，具体步骤如下。

（1）使用 ls 指令显示当前的目录列表。在命令行中输入下面的命令：

```
[root@www1 demo]# ls -l                         #显示目录列表
```

输出信息如下：

```
total 4
drwxr-xr-x 2 root root 4096 May 15 14:43 dir1
```

说明：当前目录下只有目录 dir1。

（2）当使用 rm 指令删除 dir1 目录时，系统将会报错。在命令行中输入下面的命令：

```
[root@localhost demo]# rm dir1/          #不带选项，删除目录
```

输出信息如下：

```
rm: cannot remove directory 'dir1/': Is a directory
```

说明：上面的示例表明 rm 指令不能直接删除目录。

（3）可以使用 rm 指令的-R 选项，实现递归删除目录及其下的所有内容，在命令行中输入下面的命令：

```
[root@localhost demo]# rm -R dir1/       #递归删除目录下的所有内容
```

输出信息如下：

```
rm: descend into directory 'dir1/'? y    #确认删除操作
```

说明：在上面的输出信息中输入 y 确认后，dir1 目录被删除。

【示例 1.24】强制删除目录。如果目录下的文件很多，不希望显示系统的确认信息，可以使用 rm 指令的-f 选项删除目录而不显示确认信息，从而提高效率。在命令行中输入下面的命令：

```
[root@localhost demo]# rm -f -R -v dir1 #无确认信息，强制删除目录
```

输出信息如下：

```
removed 'dir1//test.o'
removed 'dir1//test.c.save'
......省略部分输出内容......
removed directory: 'dir1/'
```

说明：从上面的输出信息中可以发现，rm 指令首先删除 dir1 目录下的所有文件，然后删除 dir1 目录。

【相关指令】rmdir，mv

1.7　rmdir 指令：删除空目录

【语　　法】rmdir [选项] [参数]　　★★★★★

【功能介绍】rmdir 指令用来删除（remove）空目录（directory）。

【选项说明】

选　　项	功　　能
-p或--parents	用递归的操作方式删除指定目录路径下的所有父级目录。要求路径中出现的目录下没有普通文件，否则会出错
--ignore-fail-on-non-empty	此选项使rmdir指令忽略由于删除非空目录而导致的错误信息
-v或--verbose	显示指令的详细执行过程
--help	显示指令的帮助信息
--version	显示指令的版本信息

【参数说明】

参　　数	功　　能
目录列表	要删除的空目录列表。当删除多个空目录时，目录名之间使用空格隔开

【经验技巧】rmdir 指令的-p 选项可以递归删除指定的目录树，但是要求每个目录必须是空目录。例如指令“rmdir -p /dir1/dir2/dir3”，将依次删除目录 dir3、dir2 和 dir1。

【示例 1.25】删除空目录，具体步骤如下。

（1）使用 ls 指令显示目录 dir1 的列表。在命令行中输入下面的命令：

```
[root@localhost ~]# ls dir1/                    #显示 dir1 目录是否为空
```

说明：上面的命令没有任何输出信息，表明 dir1 是空目录。

（2）使用 rmdir 指令删除 dir1 目录。在命令行中输入下面的命令：

```
[root@localhost ~]# rmdir dir1                  #删除空目录
```

说明：上面的命令没有任何输出信息。

【示例 1.26】删除非空目录，具体步骤如下。

（1）使用 ls 指令显示目录 mydir 的列表。在命令行中输入下面的命令：

```
[root@localhost ~]# ls mydir/                   #显示 mydir 目录下的内容
```

输出信息如下：

```
anaconda-ks.cfg  install.log  install.log.syslog
```

说明：上面的输出信息表明 mydir 为非空目录。

（2）尝试使用 rmdir 指令删除 mydir 目录。在命令行中输入下面的命令：

```
[root@localhost ~]# rmdir mydir                 #尝试删除非空目录
```

输出信息如下：

```
rmdir: 'mydir': Directory not empty
```

说明：上面报错信息表明 rmdir 指令不能用来删除非空目录。

【示例 1.27】递归删除目录树，具体步骤如下。

（1）使用 mkdir 创建一个小型的目录树。在命令行中输入下面的命令：

```
[root@localhost ~]#mkdir -p -v dir1/dir2/dir3     #创建目录树
```

输出信息如下：

```
mkdir: created directory 'dir1'
mkdir: created directory 'dir1/dir2'
mkdir: created directory 'dir1/dir2/dir3'
```

说明：上面的输出信息显示了创建的目录树的结构。

（2）使用 rmdir 指令的-p 选项，在删除 dir3 目录的同时，可以将目录 dir2 和目录 dir1 一起删除。在命令行中输入下面的命令：

```
[root@localhost ~]# rmdir -p -v dir1/dir2/dir3    #删除目录树
```

输出信息如下：

```
rmdir: removing directory, dir1/dir2/dir3/
rmdir: removing directory, dir1/dir2
rmdir: removing directory, dir1
```

说明：上面的输出信息显示了递归删除空目录的详细过程。

【相关指令】mkdir，rm

1.8　chgrp 指令：改变文件所属工作组

【语　　法】chgrp [选项] [参数]　　★★★★★

【功能介绍】chgrp 指令用来改变（change）文件和目录所属的工作组（group）。如果使用--reference 选项，则按照模板文件的所属工作组来设置文件所属的工作组。

【选项说明】

选　项	功　能
-c或--changes	显示文件所属组的改变
-f或--silent或--quiet	以静默模式运行指令，不显示任何报错信息
-h或 --no-dereference	当系统支持lchown系统调用时，此选项将修改符号链接文件本身所属的工作组，而不会修改符号链接指向的文件所属的工作组
-v或--verbose	显示指令的详细执行过程

续表

选　项	功　能
-R或--recursive	用递归的方式修改指定的目录及其下所有子目录和文件所属的工作组
--dereference	不改变符号链接本身所属的工作组，而修改符号链接指向的实际文件所属的工作组
--reference=<模板文件>	把指定文件所属的工作组改为与指定的模板文件所属的工作组相同
--preserve-root	对于“/”目录，不执行递归操作

【参数说明】

参　数	功　能
组	指定新工作组的名称
文件	指定要改变所属工作组的文件列表。多个文件或者目录之间使用空格隔开

【经验技巧】

- 使用 chgrp 指令的-R 选项可以一次性改变指定的目录及其下的所有文件和子目录的所属工作组。
- 当需要有规律地改变所属工作组的文件或目录名称时，可以借助 Shell 中的通配符功能来简化命令行操作。
- 当需要改变文件或目录的所属工作组时，使用组 ID 可以达到与使用组名相同的效果。

【示例 1.28】改变文件所属的组，具体步骤如下。

（1）使用 ls 指令的“-l”选项查看文件所属的工作组信息。在命令行中输入下面的命令：

```
[root@www1 zhangsan]# ls -l                    #以长格式显示文件信息
```

输出信息如下：

```
total 4
-rw-r--r-- 1 zhangsan zhangsan 1762 May 19 16:26 index.html
```

说明：可以看出，此时文件 index.html 属于 zhangsan 工作组。

（2）使用 chgrp 指令将 index.html 文件所属的工作组改为 root 工作组。在命令行中输入下面的命令：

```
#改变文件所属工作组 ID 为 0，即 root 组
[root@localhost zhangsan]# chgrp -v 0 index.html
```

说明：本例使用组 ID“0”来表示 root 组。

输出信息如下：

```
changed group of 'index.html' to 0
```

（3）再次使用 ls 指令的-l 选项显示文件所属的工作组。在命令行中输入下面的命令：

```
[root@localhost zhangsan]# ls -l index.html  #以长格式显示文件信息
```

输出信息如下：

```
-rw-r--r-- 1 zhangsan root 1762 May 19 16:26 index.html
```

说明：上面的输出信息表明文件 index.html 所属的工作组已经被修改为 root 工作组。使用-v 选项可以显示文件所属工作组的变化情况。

【示例 1.29】用模板文件改变文件所属的工作组，具体步骤如下。

（1）使用 ls 指令的-l 选项显示文件当前所属的组信息。在命令行中输入下面的命令：

```
[root@www1 zhangsan]# ls -l          #以长格式显示文件信息
```

输出信息如下：

```
total 4
-rw-r--r-- 1 zhangsan root 1762 May 19 16:26 index.html
-rw-r--r-- 1 zhangsan zhangsan 0 May 19 16:33 template.html
```

说明：可以看出，此时文件 template.html 属于 zhangsan 工作组，文件 index.html 属于 root 工作组。

（2）使用 chgrp 指令将 index.html 文件所属的工作组改为与文件 template.html 所属的工作组相同。在命令行中输入下面的命令：

```
[root@www1 zhangsan]# chgrp -v --reference=template.html index.
html                                 #基于模板文件改变文件的工作组
```

输出信息如下：

```
changed group of 'index.html' to zhangsan
```

（3）再次使用 ls 指令的-l 选项显示当前文件所属的组信息。在命令行中输入下面的命令：

```
[root@www1 zhangsan]# ls -l          #以长格式显示文件信息
```

输出信息如下：

```
total 4
-rw-r--r-- 1 zhangsan zhangsan 1762 May 19 16:26 index.html
-rw-r--r-- 1 zhangsan zhangsan 0 May 19 16:33 template.html
```

说明：上面的输出信息表明文件 index.html 所属的工作组已经被改成与文件 template.html 相同的工作组，即都是 zhangsan 工作组。

【相关指令】chown

1.9　chmod 指令：改变文件访问权限

【语　　法】chmod [选项] [参数]　　★★★★★

【功能介绍】chmod 指令用来改变指定文件的权限。在 chmod 指令中，权限支持字符标记法和数字标记法两种。

数字标记法表示的权限模式是 4 个八进制数，每个数由位权为 4、2、1 的 3 个二进制数相加得到。如果对应位的数字被省略，则将此位置的数字默认设置为 0。数字所代表的权限的含义为：0 表示没有权限，在第一个八进制数中，1 表示粘滞位；2 表示 sgid 权限；4 表示 suid 权限，然后将这 3 个数字相加所得到的数字即为最终权限。在第 2～4 个八进制数中，1 表示可执行权限；2 表示可写权限；4 表示可读权限，然后将这 3 个数字相加所得到的数字即为最终权限。第 2 位数字代表文件所有者（u）的权限。第 3 位数字代表文件所属组的用户（g）的权限。第 4 位数字代表其他用户（o）的权限。

字符标记法表示权限模式的语法格式为“[ugoa][[+-=] [rwxXstugo]]”。其中，“[ugoa]”表示对哪类用户设置权限，具体的含义为：u 表示 user，即文件或目录的所有者；g 表示 group，表示文件所属的组内的用户；o 表示 others，即除了 u 和 g 所代表的用户之外的其他用户；a 表示 all，即所有用户，涵盖 u、g 和 o 表示的用户。

字符标记法中的“[+-=]”表示权限的操作符，其具体含义为：“+”表示在文件原来权限的基础上添加指定的权限；“-”表示在文件原来权限的基础上去除指定的权限；“=”表示不考虑文件原来的权限，将文件的权限设置为指定的权限。

字符标记法中的“[rwxXstugo]”表示具体的权限，可以进行任意组合，它们的含义为：r 表示 read，即读权限；w 表示 write，即写权限；x 表示 execute，即执行权限；X 表示只有当目标文件对用户是可执行的或该目标文件是目录时才设置 X 权限；s 表示设置 suid（set-uid）权限和 sgid（set-gid）权限，其只能和 u、g 连用，如 u+s、g-s；t 表示 sticky，即粘滞位；u 表示将指定类别的权限设置成与文件所有者（user）的权限相同；g 表示将指定类别的权限设置成与文件所属工作组（group）的权限相同；o 表示将指定类别的权限设置成与其他用户（others）的权限相同。

【选项说明】

选　　项	功　　能
-c	显示文件权限的变化（change）情况
--silent或--quiet	安静模式，不显示任何错误信息
-v	冗余（verbose）模式，显示指令执行的详细信息
-R	以递归（recursive）的操作方式改变指定目录及其下所有子目录和文件的权限
--reference=<模板文件>	将指定文件的权限改为与指定模板文件的权限相同

【参数说明】

参　　数	功　　能
权限模式	指定文件的权限模式
文件	要改变权限的文件

【经验技巧】

- 系统管理员（root 用户）可以对具有执行权限（x）的文件设置 suid 权限，此时运行此文件的用户将临时具有与文件所有者相同的权限，如果可执行文件的所有者是 root 用户，则运行此文件的用户将临时具有 root 用户的权限。这种方式使权限的设置更加灵活，但很多情况下其被认为是一个安全隐患，黑客入侵系统后经常会设置一些具有 suid 权限的文件以便留下后门。要使用此功能必须具有管理员权限。
- 在使用 ls -l 指令显示不可执行文件的 suid 和 sgid 权限时，对应的权限位显示为大写的 S。在使用 ls -l 指令显示可执行文件的 suid 和 sgid 权限时，对应的权限位显示为小写的 s。这是因为 suid 和 sgid 权限仅对可执行文件有作用，对于不可执行文件没有意义，所以当看到对应的权限位显示为大写的 S 时将会忽略相应的权限。
- 在 chmod 指令中采用数字标识法表示权限时，权限模式通常采用 3 个数字来表示，此时特殊权限 suid、sgid 和 sticky 将被忽略。当使用 4 个数字表示权限模式时，第一个数字表示的是 3 个特殊权限位的组合。
- 目录的读权限表示可以用 ls 指令显示目录列表，目录的执行权限表示可以用 cd 指令进入目录，目录的写权限表示可以在目录下创建新文件或新的子目录。因此，只有目录的执行权限是没有太大作用的。
- 当使用 chmod 指令改变符号链接的权限时，实际改变的是符号链接所指向的文件的权限。而且在使用 chmod 指令的-R 选项进行递归方式操作时，chmod 指令将会忽略遇到的符号链接。

【示例 1.30】使用“+”和“-”设置权限。本例演示使用“+”和“-”设置文件的权限的方法，即在原有的权限基础上添加或者删除指定的权限，具体步骤如下。

（1）使用指令 ls -l 显示文件的原始权限。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l myfile          #以长格式显示文件信息
```

输出信息如下：

```
-rw-r--r--  1 root root 0 Mar 29 22:38 myfile
```

（2）使用“+”或“-”的组合方式在文件原有权限基础上修改文件的权限。在命令行中输入下面的命令：

```
#所有用户去除读权限，为文件所有者添加读权限
[root@localhost ~]# chmod a-r,u+x myfile
```

（3）再次使用指令 ls -l 显示文件权限。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l myfile          #以长格式显示文件信息
```

输出信息如下：

```
--wx------  1 root root 0 Mar 29 22:38 myfile
```

【示例 1.31】使用“=”设置权限。本例使用“=”为文件赋予全新的权限，具体步骤如下。

（1）使用指令 ls -l 显示文件的原始权限。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l myfile          #以长格式显示文件信息
```

输出信息如下：

```
-rwxr-x---  1 root root 0 Mar 29 22:38 myfile
```

（2）使用“=”为文件赋予全新的权限，在命令行中输入下面的命令：

```
#设置所有用户对 myfile 文件具有读、写和执行权限
[root@localhost ~]# chmod a=rwx myfile
```

（3）再次使用指令 ls -l 显示文件权限，在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l myfile          #以长格式显示文件信息
```

输出信息如下：

```
-rwxrwxrwx  1 root root 0 Mar 29 22:38 myfile
```

说明：从上面的输出信息中可以看出，使用“=”将所有用户的权限都设置为具有读、写和执行权限。

【示例 1.32】使用数字方式设置权限，具体步骤如下。

（1）使用指令 ls -l 显示文件的原始权限。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l myfile          #以长格式显示文件信息
```

输出信息如下：

```
-r--r--r--  1 root root 0 Mar 29 22:38 myfile
```

（2）使用数字方式为文件设置新的权限。在命令行中输入下面的命令：

```
#文件所有者具有读、写和执行权限，其他用户没有任何权限
[root@localhost ~]# chmod 700 myfile
```

（3）再次使用指令 ls -l 显示文件权限。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l myfile          #以长格式显示文件信息
```

输出信息如下：

```
-rwx------  1 root root 0 Mar 29 22:38 myfile
```

【示例 1.33】特殊权限位 suid 的应用。本示例演示特殊权限位 suid 的应用，使普通用户临时具有超级用户的权限，具体步骤如下。

（1）使用 useradd 指令创建普通用户 user100。在命令行中输入下面的命令：

```
[root@localhost ~]# useradd user100                    #创建普通用户
```

（2）将 rm 指令复制到 user100 用户的宿主目录下。在命令行中输入下面的命令：

```
[root@localhost ~]# cp /bin/rm /home/user100/     #复制 rm 指令
```

（3）使用 ls 指令的-l 选项显示 rm 指令的权限。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l /home/user100/rm   #以长格式显示文件信息
```

输出信息如下：

```
-rwxr-xr-x 1 root root 43740 May 20 10:15 /home/user100/rm
```

（4）使用 su 指令切换到 user100 用户身份，试图删除 root 用户的文件，在命令行中输入下面的命令：

```
[root@www1 ~]# su user100          #切换到 user100 用户身份
[user100@www1 root]$ cd            #切换到 user100 的宿主目录
[user100@www1 ~]$
#使用宿主目录下的 rm 指令删除 root 用户的文件
[user100@www1 ~]$ ./rm /root/install.log
#输入 y 确认
./rm: remove write-protected regular file '/root/install.log'? y
#提示权限不够
./rm: cannot remove '/root/install.log': Permission denied
```

说明：上面的输出信息表明 user100 用户无法使用 rm 指令删除 root 用户的文件，因为其权限不够。

（5）使用 exit 指令切换到 root 用户身份，使用 chmod 指令为 rm 指令设置 suid 权限。在命令行中输入下面的命令：

```
[user100@www1 ~]$ exit                          #切换到 root 用户身份
exit
[root@www1 ~]# chmod u+s /home/user100/rm #为 rm 指令设置 suid 权限
```

（6）使用 ls 指令的-l 选项显示 rm 指令的权限。在命令行中输入下面的命令：

```
[root@www1 ~]# ls -l /home/user100/rm          #以长格式显示文件信息
```

输出信息如下：

```
-rwsr-xr-x 1 root root 43740 May 20 10:15 /home/user100/rm
```

说明：在上面的输出中，对应的执行权限位上的显示信息由 x 变成了 s。

（7）再次使用 su 指令切换到 user100 用户身份，使用 rm 指令删除 root 的文件。在命令行中输入下面的命令：

```
[root@www1 ~]# su user100        #切换到 user100 用户身份
[user100@www1 root]$ cd          #切换到 user100 的宿主目录
[user100@www1 ~]$
#使用宿主目录下的 rm 指令删除 root 用户文件
[user100@www1 ~]$ ./rm /root/install.log
```

说明：调用 rm 指令后没有任何输出信息。

（8）使用 ls 指令查看上面删除的文件。在命令行中输入下面的命令：

```
[user100@www1 ~]$ ls /root/install.log        #显示文件
```

输出信息如下：

```
ls: /root/install.log: No such file or directory
```

说明：上面的输出信息表明 install.log 文件已经被 user100 用户使用具有 suid 权限的 rm 指令删除。

【示例 1.34】演示不可执行文件的特殊权限 suid，具体步骤如下。

（1）使用指令 ls -l 显示文件的原始权限。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l myfile     #以长格式显示文件信息
```

输出信息如下：

```
-rw-rw-rw-  1 root root 0 Mar 29 22:38 myfile
```

（2）为文件加上 suid 权限。在命令行中输入下面的命令：

```
[root@localhost ~]# chmod u+s myfile
#为文件 myfile 加上 suid 权限，但是对于不可执行文件 suid 权限不起作用（在下
#面显示 S）
```

说明：上面的命令没有任何输出信息。

（3）使用 ls 指令的-l 选项显示改变后的文件权限。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l myfile          #以长格式显示文件信息
```

输出信息如下：

```
-rwSrw-rw-  1 root root 0 Mar 29 22:38 myfile
```

说明：在上面的输出信息中，S 表示文件的原有执行权限位为不可执行，suid 特殊权限对于不可执行文件没有任何意义。

【示例 1.35】用 4 位数字修改特殊权限位，具体步骤如下。

（1）使用指令 ls -l 显示文件的原始权限。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l myfile          #以长格式显示文件信息
```

输出信息如下：

```
-rwxr-xr-x  1 root root 0 Mar 29 22:38 myfile
```

（2）为文件同时添加 suid 和 sgid 特殊权限。在命令行中输入下面的命令：

```
#为文件 myfile 加上 suid、sgid 和 sticky，不会改变文件的基本权限
[root@localhost ~]# chmod 6755 myfile
```

说明：在本例中，数字权限是 4 位，第 1 位表示特殊权限。上面的命令没有任何输出信息。

（3）再次使用指令 ls -l 显示文件权限。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l myfile          #以长格式显示文件信息
```

输出信息如下：

```
-rwsr-sr-x  1 root root 0 Mar 29 22:38 myfile
```

【相关指令】chown，chgrp

1.10　chown 指令：改变文件的所有者和所属工作组

【语　　法】chown [选项] [参数]　　★★★★★

【功能介绍】chown 指令用来改变（change）文件的所有者（own）和所属的工作组。如果参数中只提供用户名，那么文件所属的工作组不会发生任何变化；如果同时提供用户名和所属的工作组，用户名和所属的工作组之间使用冒

号或者点隔开，那么文件的所属用户和所属工作组将会同时改变。

【选项说明】

选　　项	功　　能
-c或--changes	显示文件的所有者或所属工作组的详细变化情况
-f或--silent或--quiet	忽略任何错误信息
-h或--no-dereference	当系统中提供了lchown系统调用时，不改变符号链接所指向的文件的所有者和所属工作组，只是改变符号链接本身的所有者和所属的工作组
-v或--verbose	显示指令执行的详细过程
-R或--recursive	递归操作，依次修改指定目录及其下所有内容的所有者和所属的工作组
--dereference	修改符号链接指向的实际文件的所有者和所属的工作组，符号链接文件本身不发生变化
--reference=<模版文件>	把文件的所有者和所属工作组改为与模版文件相同

【参数说明】

参　　数	功　　能
用户:组	指定所有者和所属的工作组。当省略“:组”时，仅改变文件所有者
文件	指定要改变所有者和工作组的文件列表。支持多个文件和目录，支持Shell通配符

【经验技巧】

- 要同时改变文件的所有者和所属工作组，参数可以使用“用户:组”或者“用户.组”的方式。
- 当需要改变的所有者和工作组的文件在同一目录下时，使用-R 选项可以递归地完成对所有文件的修改。
- 当要修改的文件名有一定规律时，使用 Shell 通配符可以简化操作。
- 可以使用用户 ID 和工作组 ID 来代替 chown 指令中使用的用户名和工作组名称。

【示例 1.36】使用 chown 指令改变文件的所有者。具体步骤介绍如下。

在命令行中输入下面的命令：

```
#将 newfile 文件的所有者改为 root 用户
[root@localhost ~]# chown -v root newfile
```

输出信息如下：

```
changed ownership of 'newfile' to root
```

【示例 1.37】改变文件的所有者和所属工作组。使用 chown 指令可以同时修改文件的所有者和所属工作组。在命令行中输入下面的命令：

```
#将 newfle 文件所有者改为 user100，所属工作组改为 user100
[root@localhost ~]# chown -v user100:user100 newfile
```

输出信息如下：

```
changed ownership of 'newfile to user100:user100
```

【示例 1.38】递归改变目录下所有文件的所有者。使用 chown 指令的-R 选项以递归操作方式改变整个目录下所有内容的所有权。在命令行中输入下面的命令：

```
#递归改变给定目录下的所有内容
[root@localhost ~]# chown -R -v user100 dir1/
```

输出信息如下：

```
changed ownership of 'dir1/fstab.bak' to user100
......省略部分输出内容......
changed ownership of 'dir1/' to user100
```

【示例 1.39】使用通配符改变文件的所有者。chown 指令支持通配符操作，可以一次修改多个文件的所有者。在命令行中输入下面的命令：

```
#将 file1~file6 文件的所有者全部改为 user100
[root@localhost ~]# chown -v user100 file[1-6]
```

输出信息如下：

```
changed ownership of 'file1' to user100
......省略部分输出内容......
changed ownership of 'file6' to user100
```

说明：本例中，同时将 6 个文件的所有者修改设置为 user100。

【示例 1.40】使用模版文件改变文件的所有者和所属工作组。使用 chown 指令的--reference 选项可以把文件的所有者设置成和参考文件相同。在命令行中输入下面的命令：

```
#使用模板文件修改文件的所有者
[root@localhost ~]# chown -v --reference=template file1
```

输出信息如下：

```
changed ownership of 'file1' to user100:user100
```

【相关指令】chgrp

1.11　find 指令：查找文件并执行指定的操作

【语　　法】find [选项] [参数]　　★★★★★

【功能介绍】find 指令在指定目录下查找文件。find 指令还能够对查找到

的文件执行指定的操作，这种功能是通过调用其他 Linux 指令来实现的。使用 find 指令时必须指定一个查找的起始目录，find 指令将从指定目录向下递归地遍历其各个子目录，将满足查找条件的文件显示在标准输出设备（通常是显示终端）上或者对这些文件采取指定的操作。

【选项说明】

选　　项	功　　能
-name <查找模式>	按照指定的文件名查找模式查找文件
-lname <查找模式>	按照指定的文件名查找模式查找符号链接
-gid <组ID>	查找属于指定组（group）ID的所有文件
-uid <用户ID>	查找属于指定用户（user）ID的所有文件
-group <组名>	查找属于指定组名的所有文件
-user <用户名>	查找属于指定用户名的所有文件
-empty	查找文件大小为0的目录或文件
-path <查找模式>	按照指定的路径查找模式查找文件
-perm <权限模式>	按照指定的权限模式查找文件和目录
-size <文件大小>	按照指定的文件大小查找文件。“文件大小”的默认单位为块（每块为512字节）
-type <文件类型>	按照指定的文件类型查找文件，支持的文件类型如下： b 块设备文件（block device）、c 字符设备文件（character device）、d 目录（directory）、p 命名管道（FIFO）、f 普通文件、l 符号链接文件（symbolic links）、s 网络套接字文件（socket）
-xtype <类型>	仅查找符号链接文件，其他功能与-type选项相同
-amin <分钟数>	查找指定“分钟数”以前被访问过的所有文件
-atime <天数>	查找指定“天数”以前被访问过的所有文件
-cmin <分钟数>	查找指定“分钟数”以前被修改过文件状态的所有文件
-ctime <天数>	查找指定“天数”以前被修改过文件状态的所有文件
-mmin <分钟数>	查找指定“分钟数”以前被修改过文件内容的所有文件
-mtime <天数>	查找指定“天数”以前被修改过文件内容的所有文件
-exec 指令名称 {} \;	用指定的Linux指令操作查找到的文件。“{}”表示将查找到的文件作为Linux指令的参数；“\;”是固定字符，放在find指令的最后
-ok 指令名称 {} \;	用指定的Linux指令操作查找到的文件，语法与-exec选项相同。直接执行操作而不提示用户进行确认
-ls	详细列出找到的文件
-fprintf <文件名>	不在终端显示查找到的文件信息，而是将其保存到指定的文件中
-print	在标准输出设备上显示查找到的文件信息，这是默认选项，可以省略
-printf <格式>	指定显示查找结果的格式，与C语言的printf函数格式的输出语法相似

【参数说明】

参　数	功　能
起始目录	查找文件的起始目录

【经验技巧】

- find 指令支持逻辑运算符与（and）、或（or）和非（not）组成的复合查询条件。选项-a 为默认选项。逻辑与表示当所有给定的条件都满足时才符合查找条件；逻辑或表示只要所给的条件中有一个满足就符合查找条件；逻辑非表示查找与所给条件相反的文件。
- find 指令的-exec 选项可以通过外部 Linux 指令对找到的文件进行操作。如果找到的文件较多，则有可能出现“参数太长”或者“参数溢出”的错误。可以使用 xargs 指令每次只读取一部分查找到的文件，等处理完毕后再读取一部分查找到的文件，以此类推，直到所有的文件都被处理完毕。
- 为了缩短 find 指令的执行时间，要尽量地缩小查找的起始目录。因为 find 指令使用递归方式遍历目录，所以起始目录的范围较大，会导致 find 指令的运行过程过长。
- 不带任何选项和参数的 find 指令可以显示当前目录下的所有内容，包括所有子目录下的文件列表。

【示例 1.41】列表显示目录及子目录的内容。不带任何参数的 find 指令可以递归地列表显示当前目录及其下所有子目录的内容。在命令行中输入下面的命令：

```
[root@localhost test]# find                    #显示目录列表
```

输出信息如下：

```
./.mozilla
......省略部分输出内容......
./.kde/Autostart/.directory
```

【示例 1.42】按文件名查找。find 指令的-name 选项以文件名为依据进行查找。在命令行中输入下面的命令：

```
[root@hn ~]# find /etc -name httpd          #按文件名查找文件并显示
```

输出信息如下：

```
/etc/httpd
/etc/logrotate.d/httpd
/etc/rc.d/init.d/httpd
/etc/sysconfig/httpd
```

说明：find 指令输出查找到的文件的绝对路径，每行一个文件。

【示例 1.43】查找文件并执行相关的操作。利用 find 指令提供的-exec 选项可以调用外部指令完成对查找到的文件的操作。例如，利用 find 指令查找内核的 core 文件并将其删除。在命令行中输入下面的命令：

```
#查找并删除内核输出的 core 文件
[root@localhost ~]# find / -name core -print -exec rm -f {} \;
```

说明：灵活地利用-exec 选项还可以调用其他 Linux 指令完成更加复杂的任务。

【相关指令】locate，updatedb

1.12　ln 指令：为文件创建链接

【语　　法】ln [选项] [参数]　　★★★★★

【功能介绍】ln 指令用来为文件创建链接。链接类型分为硬链接（hard link）和符号链接（symbolic link）两种，默认的链接类型是硬链接。如果要创建符号链接，则必须使用-s 选项。

【选项说明】

选　　项	功　　能
-b	为每个存在的文件创建备份（backup）文件
-d或-F或 --directory	默认情况下不允许对目录创建硬链接，此选项允许root用户建立目录的硬链接。受系统设置的影响，此选项可能会导致命令执行失败
-f	强制（force）创建链接，即使目标文件已经存在。目标文件会被强制覆盖
-n或 --no-dereference	把指向目录的符号链接目标当作一个普通文件
-i或 --interactive	创建链接时，如果目标文件已经存在，则提示用户确认覆盖已存在的目标文件
-L或 --logical	如果目标为符号链接，本次创建链接时将取消引用
-s或 --symbolic	创建符号链接。如果系统不支持符号链接，则命令会出错
-S或 --suffix=SUFFIX	指定备份文件的后缀
-t或 --target-directory=DIRECTORY	在指定<目录>下创建链接
-T或 --no-target-directory	总是将给定的<链接名>当作普通文件
-v或--verbose	详细信息模式，输出指令的详细执行过程

【参数说明】

参　数	功　能
源文件	指定链接的源文件。如果使用-s选项创建符号链接，则“源文件”参数可以是文件或者目录。创建硬链接时，则“源文件”参数只能是文件
目标文件	指定源文件的目标链接文件

【经验技巧】

- ln 指令默认创建的链接为硬链接，所以不能对目录建立链接。要为目录建立链接必须使用-s 选项，指明创建的链接类型为符号链接。
- 只能为普通文件创建硬链接，不能为目录创建硬链接，而符号链接则没有任何限制。
- 互为硬链接的两个文件（源文件和目标文件）等同于一个文件，所不同的仅是文件名。可以使用 ls -i 指令查看文件的索引节点，互为硬链接的文件的索引节点（inode：index node）号相同。删除互为硬链接的两个文件中的任何一个文件，另一个文件的内容不受任何影响。而编辑或者修改其中任何一个文件，另一个文件的内容也会发生同样的变化。
- 创建硬链接时，源文件和目标文件必须在同一个磁盘分区，不能跨越不同的分区；而创建符号链接时，源文件和目标文件可以在任何磁盘分区。因为符号链接文件本身只记录源文件的路径信息，而硬链接要创建一个具有相同索引节点的链接文件，索引节点在不同的分区中是自成体系的，不同分区中的索引节点不能混用，所以硬链接只能在同一个磁盘分区。
- 符号链接文件中保存的是源文件的路径信息，所以删除源文件后，符号链接文件将失去意义。符号链接类似于“快捷方式”，可以简化文件或目录的访问路径。可以为路径很深或书写不方便的文件或目录创建符号链接，以提高访问效率。

【示例 1.44】为文件和目录创建链接。ln 命令默认创建的是硬链接。下面举例说明硬链接的创建步骤。

（1）在命令行中输入下面的命令：

```
#为源文件/etcfstab 创建硬链接 myfstab
[root@localhost ~]# ln /etc/fstab ./myfstab
```

说明：上面的命令没有任何输出信息。

（2）使用 ls 指令的-i 选项，显示源文件和硬链接文件的索引节点信息。在命令行中输入下面的命令：

```
#创建互为硬链接文件的索引节点号
[root@localhost ~]# ls -i /etc/fstab ./myfstab
```

输出信息如下：

```
1393895 ./myfstab  1393895 /etc/fstab
```

说明：可以看出/etc/fstab 文件和“./myfstab”文件的索引节点号是相同的，因此除了文件名不同外，其他的完全相同。

（3）硬链接仅对文件起作用，如果要建立目录的硬链接将导致出错。在命令行中输入下面的命令：

```
[root@localhost ~]# ln mydir demolink    #试图对目录创建硬链接
```

输出信息如下：

```
ln: 'mydir: hard link not allowed for directory
```

说明：错误信息说明目录不允许创建硬链接。

（4）可以使用 ln 指令的-s 选项创建目录的符号链接。在命令行中输入下面的命令：

```
#为目录 mydir 创建符号链接 demolink
[root@localhost ~]# ln -s mydir demolink
```

（5）使用 ls 指令查看链接文件的详细信息。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l                #显示文件详细信息
```

输出信息如下：

```
total 84
drwxr-xr-x 2 root root  4096 May 14 15:16 Desktop
-rw------- 1 root root  1495 May 12 23:31 anaconda-ks.cfg
lrwxrwxrwx 1 root root     5 May 14 17:25 demolink -> mydir
drwxr-xr-x 2 root root  4096 May 14 17:25 mydir
```

说明：从上面的输出信息 demolink->mydir 中可以看出，链接文件 demolink 是 mydir 目录的符号链接。

1.13　mkdir 指令：创建目录

【语　　法】mkdir [选项] [参数]　　★★★★★

【功能介绍】mkdir 指令用来创建目录。

【选项说明】

选　项	功　能
-Z	设置安全上下文，当使用SELinux时有效
-m <权限>或 --mode=<权限>	设置新创建的目录的默认权限。如果不设置此选项，则新创建的目录的权限=rwxrwxrwx减去umask指令设置的权限
-p或--parents	创建给定路径中缺少的中间目录
--verbose	详细信息模式，显示创建目录的详细过程

【参数说明】

参　数	功　能
目录	指定要创建的目录列表，多个目录之间用空格隔开

【经验技巧】

- 使用 mkdir 指令的-p 选项可以创建目录路径中所有不存在的目录。例如，创建 dir5 目录（路径为“/dir1/dir2/dir3/dir4/dir5”），当 dir1、dir2、dir3 和 dir4 目录都不存在时使用指令“mkdir –p /dir1/dir2/dir3/dir4/dir5”自动创建中间路径。
- 可以使用-m 选项指定新创建的目录的默认权限，从而使新建的目录权限不受 umask 指令设置的影响。
- 当 mkdir 指令与适当的 Shell 通配符搭配使用时，可以一次性地创建大量的目录。请参看典型示例。

【示例 1.45】创建目录，具体步骤如下。

（1）在命令行中输入下面的命令：

```
[root@localhost ~]# mkdir mydir1                    #创建目录
```

说明：上面的命令没有任何输出信息。

（2）当使用 mkdir 指令创建的目录缺少中间目录时，系统会报错。在命令行中输入下面的命令：

```
#使用不带-p 选项的 mkdir 指令创建 dir5 目录
[root@localhost ~]# mkdir dir1/dir2/dir3/dir4/dir5
```

输出信息如下：

```
mkdir: cannot create directory ' dir1/dir2/dir3/dir4/dir5': No
such file or directory
```

说明：上面的输出信息表明缺少中间目录，可以使用-p 选项在创建目录的同时创建中间缺少的目录。

（3）在命令行中输入下面的命令：

```
[root@localhost ~]     # mkdir -p --verbose dir1/dir2/dir3/
dir4/dir5                          #使用“-p”选项创建缺少的中间目录
```

输出信息如下：

```
mkdir: created directory 'dir1'
......省略部分输出内容......
mkdir: created directory 'dir1/dir2/dir3/dir4/dir5'
```

说明：上面的输出信息显示了 mkdir 指令创建目录的详细过程。

【示例 1.46】指定新建目录的权限，具体步骤如下。

（1）使用 umask 指令设置的权限掩码。在命令行中输入下面的命令：

```
[root@localhost ~]# umask                          #显示权限掩码
```

输出信息如下：

```
022
```

说明：上面的输出信息表明，新建立的目录的权限应该是“755”（即 777–022 的结果）。

（2）创建目录 mydir。在命令行中输入下面的命令：

```
[root@localhost ~]# mkdir mydir                    #创建目录
```

（3）使用 ls -l -d 指令查看新建目录的权限。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l -d mydir/                #显示目录的详细信息
```

输出信息如下：

```
drwxr-xr-x 2 root root 4096 May 14 17:43 mydir/
```

说明：在上面的输出信息中，mydir 目录的权限 rwxr-xr-x 对应的数字为 755，这个权限受到了 uamsk 指令的影响。

（4）使用 mkdir 指令的-m 选项为创建的目录指定默认权限。在命令行中输入下面的命令：

```
#为新创建的目录指定默认的权限
[root@localhost ~]# mkdir -m 000 demodir
```

（5）使用 ls -l -d 指令查看新建目录的权限。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l -d demodir/              #显示目录的详细信息
```

输出信息如下：

```
d--------- 2 root root 4096 May 14 17:49 demodir/
```

说明：在上面的输出信息中，mydir 目录的权限“---------”对应的数字为 000，这个权限是通过 mkdir 指令的-m 选项指定的默认权限，没有受到 umask 指令的影响。

【示例 1.47】大批量地创建目录。利用 mkdir 指令可以批量创建大量的目录。在命令行中输入下面的命令：

```
[root@localhost ~]#mkdir mydir_{1,2,3,4,5,6,7,8,9}  #批量创建目录
```

说明：上例使用 mkdir 指令一次性创建了 mydir_1 到 mydir_9 共 9 个目录。

【相关指令】rmkdir

1.14　whereis 指令：显示指令及相关文件的路径

【语　　法】whereis [选项] [参数]　　★★★★★

【功能介绍】whereis 指令用来定位指令的二进制程序、源代码文件和 man 手册页等相关文件的路径。

【选项说明】

选　项	功　能
-b	仅查找二进制（binary）程序或命令
-B <目录>	仅从指定目录下查找二进制（binary）程序或命令
-m	仅查找man手册文件
-M <目录>	仅从指定目录下查找man手册文件
-s	只查找源代码（source）文件
-S <目录>	仅从指定目录下查找源代码（source）文件
-f	终止<目录>参数列表
-u	搜索不常见的记录
-l	输出有效查找路径列表（list）

【参数说明】

参　数	功　能
指令名	要查找的二进制程序、源文件和man手册页的指令名

【经验技巧】使用 whereis 指令可以显示与给出指令相关的文件路径，但是 whereis 通常只对指令执行查找其二进制程序、源代码文件和 man 手册页等

相关文件的操作。其他的普通文件使用 locate 指令进行定位。要仅显示指令的绝对路径则使用 which 指令。

【示例 1.48】定位指令及相关文件。显示 rm 指令的程序和 man 手册页的位置，具体步骤如下。

（1）在命令行中输入下面的命令：

```
#显示 rm 指令的程序路径和 man 手册页路径
[root@localhost ~]# whereis rm
```

输出信息如下：

```
rm: /usr/bin/rm /usr/share/man/man1p/rm.1p.gz /usr/share/
man/man1/rm.1.gz
```

说明：上面的输出信息不但包含 rm 指令的二进制程序的路径，而且包含 rm 指令的 man 手册的路径。

（2）使用-b 选项仅查找二进制程序信息。在命令行中输入下面的命令：

```
#显示 make 指令的二进制程序的路径
[root@localhost ~]# whereis -b make
```

输出信息如下：

```
Make: /usr/bin/make
```

说明：上面的输出信息仅包含 make 指令的二进制程序的路径，没有包含指令的其他相关文件。

（3）使用-m 选项仅查找 man 手册信息。在命令行中输入下面的命令：

```
#显示 make 指令的 man 手册页路径
[root@localhost ~]# whereis -m make
```

输出信息如下：

```
make: /usr/share/man/man1p/make.1p.gz /usr/share/man/
man1/make.1.gz
```

【相关指令】locate，which

1.15　which 指令：显示指令的绝对路径

【语　　法】which [选项] [参数]　　★★★★★

【功能介绍】which 指令用于查找并显示给定指令的绝对路径，在环境变量 PATH 中保存了查找指令时需要遍历的目录。

【选项说明】

选　项	功　能
-a或--all	显示查找到的所有文件的路径信息。默认情况下仅显示第一个
--read-functions	从标准输入读取Shell函数的定义，将查找到的函数送到标准输出设备进行显示
--skip-tilde	忽略环境变量PATH中以波浪线开头的目录
--skip-dot	忽略环境变量PATH中以点开头的隐藏目录
--show-dot	如果一个PATH中的目录以“.”开头且为该路径找到了一个匹配的可执行文件，则显示“./programname”而不是显示完整路径
--show-tilde	当一个目录匹配HOME目录时，输出一个波浪号。如果以root用户执行which，则忽略该选项
--tty-only	如果不在tty上，则停止处理选项
--help	显示帮助信息
--version	显示版本信息

【参数说明】

参　数	功　能
指令名	指令名列表

【经验技巧】

- which 指令基于环境变量 PATH 查找路径信息，如果 PATH 设置有问题，则会出现指令找不到的错误信息。
- which 指令仅能显示指令的绝对路径，使用 whereis 指令可以显示指令的源代码文件和 man 手册的绝对路径。
- 使用 which 指令还可以显示 Linux 系统中定义的与所给指令同名的命令别名。

【示例 1.49】显示指令的绝对路径。使用 which 指令显示给定指令的绝对路径，具体步骤如下。

（1）在命令行中输入下面的命令：

```
[root@localhost ~]# which halt          #显示给定指令的绝对路径
```

输出信息如下：

```
/usr/sbin/halt
```

（2）如果给定的指令设置了命令别名，则 which 指令还可以显示命令别名的设置情况。在命令行中输入下面的命令：

```
[root@localhost ~]# which ls            #显示 ls 指令的绝对路径和命令别名
```

输出信息如下：

```
    ls='ls --color=auto'
    /usr/bin/ls
```

说明：在上面的输出信息中，第一行是与 cp 指令命令别名而非存在于磁盘上的一个二进制程序。

【相关指令】whereis

1.16　file 指令：探测文件类型

【语　　法】file [选项] [参数]　　★★★★★

【功能介绍】file 指令用来探测给定文件的类型。file 指令对文件的检查分为文件系统检查、魔幻数检查和语言检查 3 个过程。

如果文件系统检查成功，则输出文件类型。输出的文件类型如下：

文件类型	说 明 信 息
text	文件中只有ASCII码字符，可以在字符终端显示文件内容
executable	文件是可以运行的
data	其他类型文件。此类型的文件一般是二进制文件或者不能在字符终端上直接显示的文件

file 指令能够判断一些在 Linux 中常用的包含二进制数据的文件格式（如内核的 core 文件）。

魔幻数检查是指检查文件中是否含有特殊的固定格式的数据，以此来判断文件类型。例如，使用 C 语言编译器编译生成的二进制文件 a.out，在文件开始部分的特殊位置保存有一个“魔幻数”，此魔幻数告诉操作系统此文件是二进制可执行文件。其他类型的文件的检查方法与此类似。Linux 中的所有魔幻数信息都保存在文件“/usr/share/magic”中，file 指令通过读取该文件的内容来完成对文件类型的判断。

如果文件是 ASCII 码文件，则 file 指令会进一步尝试检查文件的编写语言。由于语言检查不一定精确，所以放在最后执行。

【选项说明】

选　　项	功　　能
-b	输出信息使用精简格式，不输出文件名
-c	显示详细的指令执行过程，便于排错或分析程序执行的情形
-f <文件>	从指定文件（file）中读取需要检查文件类型的所有文件列表。文件中的每行代表一个文件。当文件参数为“-”时，表示从标准输入读取文件列表

续表

选　项	功　能
-F	使用指定分隔符号替换输出文件名后默认的“:”分隔符
-i	输出MIME类型字符串
-m <文件列表>	指定魔幻数文件（magic file）。如果是多个文件，则文件之间使用冒号分隔
-z	试图查看压缩文件的内部信息
-Z	仅显示压缩文件的内容
-L	显示符号链接指向的源文件
-n	当file指令检查完一个文件时就强制刷新标准输出。仅在检查一组文件时有效。一般在将file指令输出的文件类型输出到管道时使用此选项

【参数说明】

参　数	功　能
文件	要确定类型的文件列表，多个文件之间使用空格隔开，可以使用Shell通配符匹配多个文件

【经验技巧】

- file 指令默认的魔幻数文件通过环境变量 MAGIC 指定，默认值为“/usr/share/magic”。
- file 指令还可以显示指定的设备文件的类型。输出信息中包含设备的主设备号和子设备号。

【示例 1.50】探测单个文件类型。使用 file 指令探测单个文件类型，具体步骤如下。

（1）在命令行中输入下面的命令：

```
[root@linuxsrv bin]# file /usr/bin/ls          #探测文件 ls 的类型
```

输出信息如下：

```
/usr/bin/ls: ELF 64-bit LSB pie executable, x86-64, version 1
(SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-
64.so.2, BuildID[sha1]=49c2fad65d0c2df70025644c9bc7485b28bab899,
for GNU/Linux 3.2.0, stripped
```

说明：从上面的输出信息中可以看出，文件 ls 是一个 64 位的二进制可执行程序，工作的 Linux 内核版本是 3.2.0。

（2）如果文件为文本文件，则使用 file 指令可以探测文件的编写语言。在命令行中输入下面的命令：

```
#探测文件 table.c 的类型
[root@hn proftpd-1.3.2a]# file src/table.c
```

输出信息如下：

```
src/table.c: ASCII C program text
```

说明：从上面的输出信息中可以看出，文件 table.c 是一个 C 程序源代码文件。

（3）当使用 file 指令探测设备文件时，还可以探测设备文件的类型和主次设备号。在命令行中输入下面的命令：

```
[root@localhost ~]# file /dev/tty0          #探测设备文件的类型
```

输出信息如下：

```
/dev/tty0: character special (4/0)
```

说明：从上面的输出信息中可以看出，文件 tty0 是字符设备文件，并且其主设备号为 4，子设备号为 0。

【示例 1.51】批量探测文件的类型。file 指令支持批量探测多个文件的类型，可以将要探测的文件保存在一个文件中，通过-f 选项传递给 file 指令，具体步骤如下。

（1）使用 cat 指令显示包含待探测内容的文件。在命令行中输入下面的命令：

```
[root@hn ~]# cat files                      #显示文本文件的内容
```

输出信息如下：

```
/usr/sbin/halt

/dev/nvme0
/etc
```

（2）使用 file 指令探测 files 中的文件。在命令行中输入下面的命令：

```
[root@hn ~]# file -f files                  #批量探测文件类型
```

输出信息如下：

```
/usr/sbin/halt:   symbolic link to ../bin/systemctl
/dev/nvme0:       character special (240/0)
/etc:             directory
```

（3）在命令行中指定多个文件或者使用 Shell 通配符也可以实现批量探测文件类型。下面举例说明通配符的应用，在命令行中输入下面的命令：

```
[root@localhost ~]# file /usr/bin/q*        #探测以 q 开头的文件类型
```

输出信息如下：

```
/usr/bin/qemu-ga:           ELF 64-bit LSB pie executable, x86-64,
version 1 (SYSV), dynamically linked, interpreter /lib64/ld-
linux-x86-64.so.2, BuildID[sha1]=799236c6ddbfb8e4754c0b2ad20344
c6f23deca2, for GNU/Linux 3.2.0, stripped, too many notes (256)
```

```
/usr/bin/qmicli:            ELF 64-bit LSB pie executable, x86-64,
version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-
x86-64.so.2, BuildID[sha1]=84eaf1409333ea54705f9d120250d3dfd45
385f7, for GNU/Linux 3.2.0, stripped
/usr/bin/qmi-firmware-update: ELF 64-bit LSB pie executable,
x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/
ld-linux-x86-64.so.2, BuildID[sha1]=8aac8c706c0b65a793a357b1a8
2375fb94f70152, for GNU/Linux 3.2.0, stripped
/usr/bin/qmi-network:        a /usr/bin/sh script, ASCII text
executable
/usr/bin/quota:             ELF 64-bit LSB pie executable, x86-64,
version 1 (SYSV), dynamically linked, interpreter /lib64/ld-
linux-x86-64.so.2, BuildID[sha1]=646e2f66a5aa231391322fd39c085
01421c1287c, for GNU/Linux 3.2.0, stripped
/usr/bin/quotasync:         ELF 64-bit LSB pie executable, x86-64,
version 1 (SYSV), dynamically linked, interpreter /lib64/ld-
linux-x86-64.so.2, BuildID[sha1]=24a1226e49a9b262b8962e01d0a1
5683b5063ea7, for GNU/Linux 3.2.0, stripped
```

说明：本例中测试了“/usr/bin/”目录下所有以 q 开头的文件类型。从输出信息中可以看出，文件 qemu-ga、qmicli、qmi-firmaware-update、quota 和 quotasync 是二进制可执行程序，而文件 qmi-network 则是 Bash 脚本程序。

1.17　touch 指令：设置文件的时间属性

【语　　法】touch [选项] [参数]　　★★★★★

【功能介绍】touch 指令有两个功能：一是用于改变文件的时间属性，它将文件的最后访问时间和最后修改时间设置为系统的当前时间；二是用于创建新的空文件。

【选项说明】

选　　项	功　　能
-r <模板文件>或--reference=<模板文件>	将指定文件的时间属性设置为与指定的模板文件的时间属性相同
-t <时间>	指定文件的时间格式。指定的时间格式为MMDDhhmm[[CC]YY][.ss]，其含义从左到右分别表示月、日、小时、分钟、世纪、年和秒
-a <时间>	将指定文件的最后访问（access）时间设置为当前系统时间，其他时间属性不变
-c或--no-create	如果指定的文件不存在，则不创建这些不存在的文件
-m <时间>	仅将文件的最后修改（modify）时间设置为当前系统时间
-d <字符串>或--date=<字符串>	使用字符串所代表的时间来设置文件的时间属性

【参数说明】

参　数	功　能
文件	指定要设置时间属性的文件列表

【经验技巧】

- Linux 中没有单独的指令用来创建新的空文件，使用 touch 指令可以创建新的空文件，并且新创建的空文件的最后访问时间和最后修改时间均为当前的系统时间。
- 使用 touch 指令可以一次性创建大量的空文件。请参看典型示例。

【示例 1.52】设置文件的时间属性。使用 stat 指令可以显示文件的时间属性，具体步骤如下。

（1）在命令行中输入下面的命令：

```
[root@localhost ~]# stat myfile      #显示 myfile 的时间属性
```

输出信息如下：

```
......省略部分输出内容......
最近访问：2023-05-16 17:58:50.845391606 +0800
最近更改：2023-05-16 17:58:50.845391606 +0800
最近改动：2023-05-16 17:59:09.181506731 +0800
创建时间：2023-05-16 17:58:50.845391606 +0800
```

（2）使用 touch 指令修改文件 myfile 的时间属性为当前系统时间。在命令行中输入下面的命令：

```
[root@localhost ~]# touch myfile     #设置文件的时间为当前系统时间
```

说明：上面的命令没有任何输出信息。

（3）再次使用 stat 指令显示 myfile 文件的时间属性。在命令行中输入下面的命令：

```
[root@localhost ~]# stat myfile      #显示文件的时间属性
```

输出信息如下：

```
......省略部分输出内容......
最近访问：2023-05-17 11:34:20.732465271 +0800
最近更改：2023-05-17 11:34:20.732465271 +0800
最近改动：2023-05-17 11:34:20.732465271 +0800
创建时间：2023-05-16 17:58:50.845391606 +0800
```

说明：从上面的输出信息中可以看出使用 touch 指令前后文件 myfile 的时间属性发生了变化。

【示例 1.53】创建空文件。使用 touch 指令创建空文件 newfile，具体步骤如下。

在命令行中输入下面的命令：

```
[root@localhost ~]# touch newfile          #创建空文件 newfile
```

说明：上面的命令没有任何输出信息。

【示例 1.54】大批量地创建空文件。利用 touch 指令可以批量创建大量的空文件，具体步骤如下。

（1）在命令行中输入下面的命令：

```
[root@hn demo]#touch file_{1,2,3,4,5,6,7,8,9}     #批量创建空文件
```

说明：在上例中使用 touch 指令一次性创建了 file_1 到 file_9 共 9 个空文件。

（2）使用 ls 指令显示创建的空文件列表。在命令行中输入下面的命令：

```
[root@hn demo]# ls                                  #显示文件列表
```

输出信息如下：

```
file_1  file_2  file_3  file_4  file_5  file_6  file_7  file_8
file_9
```

1.18　locate/slocate 指令：快速定位文件的路径

【语　　法】locate [选项] [参数]　　★★★★☆

【功能介绍】locate 指令用于快速查找文件系统中某个文件或目录的位置。该指令不是实时搜索整个文件系统，而是通过搜索一个预先建立的数据库（通常是/var/lib/plocate/plocate.db）实现快速查找的功能。为了保证查询结果的准确度，管理员必须定期更新 locate 数据库。

【选项说明】

选　　项	功　　能
-c或--count	不将文件名输出到终端，而是只显示符合条件的文件数目
-d <目录>或--database=<目录>	指定存放locate数据库的目录
-i	忽略（ignore）文件名大小写
-q	忽略错误信息
-r <正则表达式>或--regexp=<正则表达式>	进行查找匹配时，使用基本的POSIX正则表达式

【参数说明】

参　　数	功　　能
查找字符串	要查找的文件名中包含的字符串

【经验技巧】

- 由于 locate 指令基于数据库进行查询，所以第一次运行前必须使用 updatedb 指令创建 locate 数据库。
- locate 指令的数据库需要定期更新，以提高 locate 指令的准确性。大多数的 Linux 发行版都设置了自动调用 updatedb 指令来更新数据库。
- locate 指令与 slocate 指令的功能相同。slocate 指令是 GNU locate 指令的安全增强版。在大多数的 Linux 发行版中，考虑到与 UNIX 系统的兼容性，locate 指令实际是 slocate 指令的符号链接。

【示例 1.55】查找文件路径。在命令行中输入下面的命令：

```
#查找文件名中包含 rmdir 关键字的所有文件的绝对路径
[root@localhost ~]# locate rmdir
```

输出信息如下：

```
warning: locate: warning: database /var/lib/slocate/slocate.db'
is more than 8 days old
/bin/rmdir
......省略部分输出内容......
/usr/share/man/man3p/rmdir.3p.gz
```

说明：输出的第一行信息“warning: locate: warning: database /var/lib/slocate/slocate.db' is more than 8 days old”表示 locate 数据库存放的位置为“/var/lib/slocate/”目录，但是此数据库已经太老了，查询到的结果并不准确，需要手动使用 updatedb 指令更新 locate 数据库。

【示例 1.56】统计符合条件的文件数。使用 locate 指令的-c 选项可以统计符合条件的文件总数。在命令行中输入下面的命令：

```
[root@localhost ~]# locate -c gcc        #统计匹配的文件数目
```

输出信息如下：

```
229
```

【相关指令】whereis，updatedb

1.19　dd 指令：复制文件并进行内容转换

【语　　法】dd [选项]　　★★★★☆

【功能介绍】dd 指令用于复制文件并对原文件的内容进行转换和格式化处理。

【选项说明】

选　　项	功　　能
if=<输入文件>	指定输入文件（input file）。如果不指定if选项，则从标准输入设备读入信息
of=<输出文件>	指定输出文件（output file），否则输出到标准输出设备
ibs=<字节数>	指定每次输入（input）的字节数（bytes）。默认值是512字节
obs=<字节数>	指定每次输出（output）的字节数（bytes）。默认值是512字节
bs=<字节数>	设定每次读写的字节数（bytes）。此选项将覆盖ibs和obs选项
cbs=<字节数>	为块转换和非块转换指定转换（convert）块的字节数（bytes）
skip=<块数>	跳过输入文件最初指定的块数，块大小由ibs选项指定
seek=<块数>	跳过输出文件前面的指定块数开始写入，块大小由ibs选项指定
count=<块数>	只复制输入文件前面指定的块数（块大小由ibs选项指定）
conv=<关键字,关键字...>	将文件按指定关键字的方式转换（注意在“,”前后没有空格）。支持的转换方式如下： ascii：将EBCDIC码转换成ASCII码； ebcdic：将ASCII码转换成EBCDIC码； ibm：将ASCII码转换成可变EBCDIC码； block：每行输入信息无论其长短，输出都是选项cbs指定的字节数，并且其中的“换行符”用空格替换。如果有必要，行尾填充空格； unblock：用“换行符”替换每个输入块（由选项cbs设定字节数）末尾的空格； lcase：将大写字母转换成小写字母； ucase：将小写字母转换成大写字母； swab：交换每对输入字节。如果读入的字节数是奇数，则最后一个字节只是简单地复制到输出； noerror：当读取信息发生错误时，仍然继续执行； notrunc：对输出文件不进行截断操作； sync：用0填充每个输入块的末尾，使其大小为选项ibs的值

【经验技巧】

- 使用 dd 指令可以在复制文件的同时对文件内容进行转换或格式化处理。
- 使用 dd 指令可以用来制作光盘的映像文件。制作光盘的 ISO 映像文件的指令格式为“dd　if=/dev/cdrom　/path/cdrom.iso”。

【示例 1.57】复制文件并转换文件内容。使用 dd 指令可以在复制文件的同时将文件中的小写字母转换为大写，具体步骤如下。

（1）使用 cat 指令显示原始文件的内容。在命令行中输入下面的命令：

```
[root@localhost ~]# cat test.sh                    #显示文本文件的内容
```

输出信息如下：

```
#!/bin/bash
for i in 1 2 3 4 5 6 7
do
       echo $i
done
```

说明：此时的文件内容全部是小写字母。

（2）使用 dd 命令复制文件并将文件中的小写字母全部转换成大写字母，同时使用 if、of 和 conv 选项。在命令行中输入下面的命令：

```
#复制文件的同时将文件中的小写字母全部转换成大写字母
[root@localhost ~]# dd if=test.sh conv=ucase of=newtest.sh
```

输出信息如下：

```
0+1 records in
0+1 records out
53 bytes (53 B) copied, 0.000146282 seconds, 362 kB/s
```

（3）再次使用 cat 指令显示复制生成的新文件的内容，在命令行中输入下面的命令：

```
[root@localhost ~]# cat newtest.sh              #显示文本文件的内容
```

输出信息如下：

```
#!/BIN/BASH
FOR I IN 1 2 3 4 5 6 7
DO
       ECHO $I
DONE
```

说明：可以发现文件的内容已经由小写字母转换成大写字母。

【示例 1.58】制作光盘 ISO 映像文件。把光盘的设备文件作为 dd 指令的输入文件（if），将要生成的 ISO 映像文件作为 dd 指令的输出文件（of），dd 指令自动完成转换工作，具体步骤如下。

在命令行中输入下面的命令：

```
#制作光盘映像文件
[root@localhost ~]#dd  if=/dev/cdrom  of=/path/cdrom.iso
```

说明：生成的光盘 ISO 映像文件为 cdrom.iso，格式为 ISO9660，可以被用来刻录光盘。

【示例 1.59】备份/dev/sda 磁盘数据到指定路径的 image 文件中。把磁盘/dev/sda 文件作为 dd 指令的输入文件（if），将要生成的 image 文件作为 dd 指令的输出文件（of），dd 指令自动完成转换工作。具体步骤介绍如下。

在命令行中输入下面的命令：

```
#备份磁盘中的文件
[root@localhost ~]#dd  if=/dev/sda  of=/path/image
```

说明：生成的备份文件为 image。当用户的/dev/sda 磁盘中的数据丢失时，可以通过 image 文件恢复。

【相关指令】cp

1.20　updatedb 指令：创建或更新 slocate 数据库

【语　　法】updatedb [选项]　　★★★★★

【功能介绍】updatedb 指令用来创建或更新 slocate 指令必备的数据库文件。updatedb 指令的执行过程较长，因为在执行时它会遍历整个系统的目录树，并将所有的文件信息写入 slocate 数据库文件。

【选项说明】

选　　项	功　　能
-o <文件>	忽略默认的数据库文件，输出（output）到指定的slocate数据库文件
-U <目录>	更新（update）指定目录的slocate数据库
-v	冗余（verbose）模式，显示指令执行的详细过程

【经验技巧】

- 第一次使用 updatedb 指令时，其运行速度比较慢，这是由于要新创建当前操作系统中所有文件信息的数据库。第二次使用 updatedb 指令时仅执行数据库的更新操作，因此速度比较快。
- 在使用-U 选项时，必须使用绝对路径。

【示例 1.60】更新 slocate 数据库。可以直接使用 updatedb 指令更新 slocate 数据库。在命令行中输入下面的命令：

```
[root@localhost ~]# updatedb                    #更新 slocate 数据库
```

说明：如果是第一次执行上面的指令，则执行时间较长。

【示例 1.61】更新指定目录的 slocate 数据库。使用 updatedb 指令的-U 选项可以指定要更新的 slocate 数据库的目录。在命令行中输入下面的命令：

```
#仅更新指定目录的 slocate 数据库
[root@localhost ~]# updatedb -U /usr/local/
```

说明：上例中仅更新目录/usr/local/的 slocate 数据库记录。

【相关指令】locate，slocate

1.21　dirname 指令：去除文件名中的非目录部分

【语　　法】dirname [选项] [参数]　　★★★★☆

【功能介绍】dirname 指令用于去除文件名中的非目录部分，仅显示与目录有关的内容。

【选项说明】

选　　项	功　　能
--help	显示帮助
--version	显示版本号

【参数说明】

参　　数	功　　能
文件	带目录的文件名，如/var/log/message

【经验技巧】dirname 指令通常应用在 Shell 脚本程序设计中，以得到文件名中的目录信息。

【示例 1.62】仅显示文件的目录信息。用 dirname 指令仅显示文件名中的目录信息。在命令行中输入下面的命令：

```
#显示目录信息
[root@localhost ~]# dirname /var/log/httpd/access_log
```

输出信息如下：

```
/var/log/httpd
```

【相关指令】basename

1.22　pathchk 指令：检查文件路径名的有效性和可移植性

【语　　法】pathchk [选项] [参数]　　★★★★☆

【功能介绍】pathchk 指令用来检查文件名中不可移植的部分。

【选项说明】

选　　项	功　　能
-p	检查大多数的POSIX系统
-P	检查空名字和以“-”开头的文件

续表

选　项	功　能
--portability	检查所有POSIX系统的规范，等同于“-p -P”选项
--help	显示帮助
--version	显示版本号

【参数说明】

参　数	功　能
文件	带路径信息的文件，如/var/log/message
后缀	可选参数，指定要去除的文件后缀字符串

【经验技巧】pathchk 指令仅用于测试路径的可移植性，其参数可以是并不存在的路径。

【示例 1.63】检查路径名的有效性。使用 pathchk 指令检查系统中“/etc/httpd/conf/httpd.conf”路径名称的有效性和可移植性。在命令行中输入下面的命令：

```
#检查路径名的可移植性
[root@localhost ~]# pathchk /etc/httpd/conf/httpd.conf
```

1.23　unlink 指令：调用 unlink()函数删除指定的文件

【语　　法】unlink [选项] [参数]　　★★★★☆

【功能介绍】unlink 指令使用系统调用函数 unlink()删除指定的文件。

【选项说明】

选　项	功　能
--help	显示帮助
--version	显示版本号

【参数说明】

参　数	功　能
文件	指定要删除的文件

【经验技巧】

- unlink 指令仅能删除普通文件，不能删除目录。
- unlink 指令没有 rm 指令中的-i 选项，所以无法防止误删除操作，在使

用时要特别小心。

【示例 1.64】删除文件。使用 unlink 指令删除普通的文件。在命令行中输入下面的命令：

```
[root@localhost ~]# unlink myfile100          #删除普通的文件
```

说明：unlink 指令执行成功后没有任何输出信息。

【示例 1.65】删除目录。当使用 unlink 删除目录时将会出现错误。在命令行中输入下面的命令：

```
[root@localhost ~]# unlink mydir              #删除目录
```

输出信息如下：

```
unlink: cannot unlink 'mydir': Is a directory
```

【相关指令】rm

1.24　basename 指令：去掉文件名中的路径和扩展名

【语　　法】basename [选项] [参数]　　★★★★☆

【功能介绍】basename 指令用于显示去掉路径信息和文件扩展名之后的文件名。

【选项说明】

选　项	功　能
--help	显示帮助
--version	显示版本号

【参数说明】

参　数	功　能
文件	带路径信息的文件，如/var/log/message
后缀	可选参数，指定要去除的文件后缀字符串

【经验技巧】

- basename 指令的第二个参数为可选项，如果省略此选项，则仅去掉路径信息。
- basename 指令通常应用在 Shell 脚本程序设计中，以获得文件名中需要的部分字符串。

【示例 1.66】去掉文件名中的路径信息。使用 basename 指令去掉给定的绝对路径的文件名中包含的路径信息。在命令行中输入下面的命令：

```
#去掉路径信息仅显示文件名
[root@localhost ~]# basename /var/log/message
```

输出信息如下：

```
message
```

说明：上例中仅显示去除路径信息后的文件名 message。

【示例 1.67】去掉文件的路径信息和后缀。如果为 basename 指令指定第二个参数，则 basename 指令在去掉路径信息的同时将文件的后缀也去除，仅显示不带后缀的文件名。在命令行中输入下面的命令：

```
#显示去掉路径和后缀的文件名
[root@localhost ~]# basename /etc/updatedb.conf .conf
```

输出信息如下：

```
Updatedb
```

【相关指令】dirname

1.25　rename 指令：批量为文件改名

【语　　法】rename [参数]　　★★★★★

【功能介绍】rename 指令用字符串替换的方式批量改变文件名。

【参数说明】

参　数	功　能
原字符串	需要替换文件名的字符串
目标字符串	将文件名中的原字符串替换成目标字符串
文件	指定要改变文件名的文件列表

【经验技巧】如果文件名有一定的规律，则可以用 rename 指令批量改变文件名。rename 指令的本质是采用替换的方式将文件名中的指定字符串替换为目标字符串，在进行替换时需要使用 Shell 通配符来匹配文件名。

【示例 1.68】批量重命名文件，具体步骤如下。

（1）使用 ls 指令显示当前目录下的文件列表。在命令行中输入下面的命令：

```
[root@localhost ~]# ls -l                    #显示目录列表
```

输出信息如下：

```
total 64
drwxr-xr-x 2 root root  4096 May 14 15:16 Desktop
-rw------- 1 root root  1495 May 12 23:31 anaconda-ks.cfg
-rw-r--r-- 1 root root     0 May 22 14:59 file_0
```

```
-rw-r--r-- 1 root root     0 May 22 14:59 file_1
……省略部分输出内容……
-rw-r--r-- 1 root root     0 May 22 14:59 file_8
-rw-r--r-- 1 root root     0 May 22 14:59 file_9
-rw-r--r-- 1 root root 42568 May 12 23:30 install.log
-rw-r--r-- 1 root root     0 May 12 22:44 install.log.syslog
```

（2）使用 rename 指令将文件名中的字符串“file_”替换为“linux_”。在命令行中输入下面的命令：

```
[root@localhost ~]# rename file_ linux_ file *    #批量重命名文件
```

（3）再次使用 ls 指令显示当前目录下的文件列表。在命令行中输入下面的命令：

```
[root@localhost ~]# ls                                    #显示目录列表
```

输出信息如下：

```
total 64
drwxr-xr-x 2 root root  4096 May 14 15:16 Desktop
-rw------- 1 root root  1495 May 12 23:31 anaconda-ks.cfg
-rw-r--r-- 1 root root 42568 May 12 23:30 install.log
-rw-r--r-- 1 root root     0 May 12 22:44 install.log.syslog
-rw-r--r-- 1 root root     0 May 22 14:59 linux_0
-rw-r--r-- 1 root root     0 May 22 14:59 linux_1
……省略部分输出内容……
-rw-r--r-- 1 root root     0 May 22 14:59 linux_8
-rw-r--r-- 1 root root     0 May 22 14:59 linux_9
```

说明：上面的输出信息表明批量修改文件名成功。

【相关指令】mv

1.26　习　　题

一、填空题

1．文件管理是操作系统的重要功能。在 Linux 中，所有的软硬件资源都被认为是_____。

2．file 指令对文件的检查分为三个过程，分别为_____、_____和_____。

3．touch 指令有两个功能，一是_____；二是_____。

二、选择题

1．下面的（　　）指令用来显示当前的工作目录。

A．ls　　B．cd　　C．pwd　　D．find

2．使用 find 指令查找文件，（　　）选项依据文件的名称进行查找。

A．-name　　B．-size　　C．-type　　D．-user

3．下面的（　　）命令可以改变文件的相关属性。

A．chgrp　　B．chmod　　C．chown　　D．ls

三、判断题

1．创建硬链接时，源文件和目标文件必须在同一个磁盘分区，不能跨越不同的分区。而创建符号链接时，源文件和目标文件可以在任何磁盘分区。（　　）

2．使用 dd 指令可以在复制文件的同时对文件内容进行转换或格式化处理。（　　）

3．rmdir 命令和 rm 命令的作用一样，都是删除整个目录。（　　）

四、操作题

1．创建一个目录 test 并使用 ls 命令查看其属性。

2．切换到 test 目录并查看当前目录，然后删除 test 目录。

3．创建一个名为 test.txt 的文件。

第 2 章　文 本 编 辑

在操作系统中，信息以文件的方式保存在存储介质上，而文本文件则是最常使用的文件格式。文本编辑是系统管理员最常见的操作任务之一。Linux 系统提供了众多优秀的文本编辑工具，熟练掌握这些文本编辑工具，可以极大地提高管理员的工作效率。本章将介绍 Linux 最常用的文本编辑指令。

2.1　vi 指令：全屏纯文本编辑器

【语　　法】vi [选项] [参数]　　★★★★★

【功能介绍】vi 是 UNIX 操作系统和类 UNIX 操作系统中最通用的全屏幕纯文本编辑器。Linux 中的 vi 编辑器叫 vim，它是 vi 的增强版（Vi IMproved），与 vi 编辑器完全兼容，而且实现了很多增强功能。

vi 编辑器支持编辑模式和命令模式。在编辑模式下可以完成文本的编辑功能；在命令模式下可以完成对文件的操作。要正确使用 vi 编辑器，就必须熟练掌握这两种切换模式。默认情况下，打开 vi 编辑器后自动进入命令模式。从编辑模式切换到命令模式使用 Esc 键，从命令模式切换到编辑模式使用 A、a、O、o、I 和 i 键（功能描述请看下面的内置命令列表）。

vi 编辑器提供了丰富的内置命令，有些内置命令使用键盘上的组合键即可完成，有些内置命令则需要以冒号“:”作为开头进行输入。常用的内置命令如下。

内置命令	功　　能
Esc	从编辑模式切换到命令模式
ZZ	在命令模式下保存当前文件所做的修改后退出vi
Ctrl+d	将显示内容向下滚动半屏
Ctrl+u	将显示内容向上滚动半屏
Ctrl+f	将显示内容向下滚动一屏
Ctrl+b	将显示内容向上滚动一屏
:行号	光标跳转到指定行的行首
:$	光标跳转到最后一行的行首
x	删除当前光标所在位置的字符
X	删除当前光标所在位置的前一个字符

续表

内置命令	功　能
D	删除从当前光标到光标所在行尾的全部字符
dd	删除光标行的整行内容
ndd	删除当前光标所在行的后（包括当前光标所在行）*n*（*n*为数字）行内容
Y	复制当前光标所在行的全部内容，复制的内容放到内存缓冲区备用
nyy	复制当前光标所在行的后（包括当前光标所在行）*n*（*n*为数字）行内容，复制的内容放到内存缓冲区备用
p	粘贴文本操作，用于将缓存区的内容粘贴到当前光标所在位置的下方
P	粘贴文本操作，用于将缓存区的内容粘贴到当前光标所在位置的上方
/字符串	文本查找操作，用于从当前光标所在位置开始向文件尾部查找指定字符串的内容，查找到的字符串会被加亮显示
?name	文本查找操作，用于从当前光标所在位置开始向文件头部查找指定字符串的内容，查找到的字符串会被加亮显示
a,b s/F/T	替换文本操作，用于在第a行到第b行之间，将F字符串换成T字符串。其中，“s/”表示进行替换操作
a	从命令模式切换到编辑模式，并且从当前光标所在位置之后开始输入内容
A	从命令模式切换到编辑模式，并且从当前光标所在行的行末开始输入内容
i	从命令模式切换到编辑模式，并且从当前光标所在位置开始插入文本内容
I	从命令模式切换到编辑模式，并且从当前光标所在行的行首开始插入文本内容
o	从命令模式切换到编辑模式，并且在当前光标所在行的下方新建一个空行开始插入文本
O	从命令模式切换到编辑模式，并且在当前光标所在行的上方新建一个空行开始插入文本
:wq	在命令模式下执行存盘写入（write）并退出（quit）操作
:w	在命令模式下执行存盘写入（write）操作
:w!	在命令模式下执行强制存盘写入（write）操作（即使文件是只读的）
:w 文件名	在命令模式下将当前文件名另存为指定的文件名
:q	在命令模式下执行退出（quit）vi操作（如果文件内容发生改变但是尚未保存，则提示是否保存）
:q!	在命令模式下执行强制退出vi操作（无论文件是否保存）
:e 文件名	在命令模式下打开并编辑（edit）指定名称的文件
:n	在命令模式下如果同时打开了多个文件，则继续编辑下一个（next）文件
:f	在命令模式下显示当前的文件名、光标所在行的行号及显示比例
:set number	在命令模式下在最左端显示行号，可用简写方式:set nu
:set nonumber	在命令模式下取消在最左端显示的行号，可用简写方式:set nonu

【选项说明】

选　项	功　能
+<行号>	从指定行号的行开始显示文本内容
-b	以二进制模式打开文件，用于编辑二进制文件和可执行文件
-c <指令>	在完成对第一个文件的编辑任务后，执行给出的指令
-d	以diff模式打开文件。当进行多文件编辑时，显示文件的差异部分
-l	使用Lisp模式。打开lisp和showmatch选项
-m	取消写文件功能，重置write选项，仅允许编辑缓冲区中的内容，但是不允许将改变写入磁盘文件
-M	关闭修改文件功能，取消write和modifiable选项的设置，所以不允许执行修改文件和写文件操作
-n	不使用缓存功能，不会产生.swap的交换文件
-o<文件数目>	同时打开（open）指定数目的文件
-R	以只读（read-only）方式打开文件。使用该选项可以设置readonly选项
-s	安静（silence）模式，不显示指令的任何错误信息

【参数说明】

参　数	功　能
文件列表	指定要编辑的文件列表，多个文件之间使用空格分隔开

【经验技巧】

- 在编辑文件时，可以在命令模式下使用:set nu 和:set nonu 来显示和取消行号。
- 在使用 vi 编辑器编辑文件时，可以通过多按几次 Esc 键以确认切换到编辑模式。
- 默认情况下，vi 编辑器为了提高运行效率使用了缓存功能。在编辑文件时，会在文件所在目录下创建一个形如.filename.swp 的交换文件。当退出 vi 时，交换文件将被删除。如果没有正常退出 vi 或者同一个文件被打开两次，则会出现警告信息。

【示例 2.1】显示文件行号，具体步骤如下。

（1）在编辑文件时（特别是程序源代码），可以通过:set nu 显示文件中的行号，以增强可读性。在命令行中输入下面的命令：

```
[root@hn ~]# vi /etc/rc.d/rc.local  #编辑 Shell 脚本文件 rc.local
```

输出信息如下：

```
#! /bin/bash
#
......省略部分输出内容......
```

```
touch /var/lock/subsys/local
"rc.local" 14L, 474B
```

（2）vi 编辑器会自动进入命令模式，直接输入“:set number”指令以显示行号。输出信息如下：

```
     1 #! /bin/bash
     2 #
......省略部分输出内容......
     13 touch /var/lock/subsys/local
:set number                                    #输入显示行号的指令
```

说明：在上面的输出信息中，每一行的开头都显示了行号，如果要取消显示行号，可以使用:set nonumber 指令。需要注意，显示行号仅是为了方便阅读，并非文件的正文。

2.2　emacs 指令：全屏文本编辑器

【语　　法】emacs [选项] [参数]　　★★★★★

【功能介绍】emacs 指令是由 GNU 组织的创始人 Richard Stallman 开发的一个功能强大的全屏文本编辑器，它支持多种编程语言，具有很多优良的特性。有众多的系统管理员和软件开发者都在使用 emacs。

【选项说明】

选　　项	功　　能
+<行号>	启动emacs编辑器并将光标移动到指定行号的行中
-q	启动emacs编辑器但不加载初始化文件
-u <用户>	启动emacs编辑器时加载指定用户（user）的初始化文件
-t <文件>	启动emacs编辑器时把指定的文件作为终端（terminal），不使用标准输入（stdin）与标准输出（stdout）
-f <函数>	执行指定的Lisp语言函数（function）
-l <lisp代码文件>	加载指定的Lisp代码文件
-batch	以批处理模式运行emacs编辑器

【参数说明】

参　　数	功　　能
文件	指定要编辑的文本文件

【经验技巧】emacs 编辑器的内置指令和相关操作及其扩展功能相当丰富，初学者不太容易掌握。读者可以先从最基本的操作入手，逐步掌握 emacs 的使用方法。详细的指令和功能介绍请参考相关的书籍。

【示例 2.2】启动 emacs 编辑器。

可以在命令行中将待编辑的文件传递给 emacs 指令。在命令行中输入下面的命令：

```
#启动 emacs 编辑文件“/etc/fstab”
[root@hn ~]# emacs /etc/fstab
```

emacs 指令的输出信息会占满整个终端屏幕，为了节省篇幅，此处省略。

说明：可以在 emacs 编辑器运行界面按 F1 键，获取 emacs 指令的帮助信息。

2.3　ed 指令：行文本编辑器

【语　　法】ed [选项] [参数]　　★★★★☆

【功能介绍】ed 指令是单行纯文本编辑器，它有命令模式（command mode）和输入模式（input mode）两种工作模式，默认的工作模式为命令模式。当由命令模式切换到输入模式时，使用 a、c 或 i 命令（具体功能描述参看下面的内置命令列表）中的任何一个命令即可；当由输入模式切换为命令模式时，在新的空行中输入“.”后按 Enter 键即可。ed 指令支持多个内置命令。常见的内置命令如下。

内置命令	命令功能描述
A	切换到输入模式，在文件的最后一行之后输入新的内容
C	切换到输入模式，用输入的内容替换最后一行的内容
i	切换到输入模式，在当前行之前加入一个新的空行用来输入内容
d	用于删除（delete）最后一行文本内容
n	用于显示最后一行的行号与内容
w <文件名>	以给定的文件名保存当前正在编辑的文件
q	退出（quit）ed编辑器

【选项说明】

选　　项	功　　能
-G或--traditional	强制使用向后兼容模式
-p <提示符>或--prompt= <提示符>	设置命令模式下的命令提示符
-s或--quiet或--silent	打开文件时不执行检查功能，常用在脚本中

【参数说明】

参　　数	功　　能
文件	待编辑的文件

【经验技巧】

- 使用 ed 编辑器时，如果指定了要编辑的文件，则将该文件读入 ed 指令的缓冲区中。对文件所做的修改不会直接改变磁盘中的文件，而是仅影响缓冲区中的文件内容。如果 ed 异常退出，则将丢失对文件所做的修改。
- 在 Linux 中经常出现一些超大规模的文件（超过 2GB），如果直接使用 vi 等全屏幕的文本编辑器，则可能导致内存问题，此时可以利用 ed 编辑器轻松地编辑这些超大规模的文件。
- 在 Linux 中，经常使用 ed 指令在 Shell 脚本中完成对文件的编辑功能。

【示例 2.3】以行为单位编辑文本文件，具体步骤如下。

（1）使用 cat 指令显示文本文件的内容。在命令行中输入下面的命令：

```
[root@localhost test]# cat /etc/fstab.bak    #显示文本文件内容
```

输出信息如下：

```
#
# /etc/fstab
.......//省略部分输出内容......
/dev/mapper/rhel-root        /          xfs     defaults    0 0
UUID=beb1b7b8-e7f6-498e-     /boot      xfs     defaults    0 0
8ab3-ce90265b0e48
/dev/mapper/rhel-home        /home      xfs     defaults    0 0
/dev/mapper/rhel-swap        none       swap    defaults    0 0
```

（2）使用 ed 编辑器编辑文件/etc/fstab.bak。在命令行中输入下面的命令：

```
#编辑文件/etc/fstab.bak
[root@localhost test]# ed /etc/fstab.bak
```

输出信息如下：

```
654                                 #显示文件当前的字节数
1                                   #显示第一行的内容
#
i                                   #进入输入模式，在文件开头加入新内容
hello! This is a Demo for ed!       #输入的新文本
.                                   #切换到命令模式
,s/xfs/TTT/g                        #将文件中的 xfs 全部替换为 TTT
w                                   #保存所做的修改
684                                 #显示文件当前的字节数
q                                   #退出 ed 编辑器
[root@hn ~]#
```

说明：本例中黑体部分的内容需要从键盘输入。

（3）显示使用 ed 编辑器编辑过的文件内容。在命令行中输入下面的命令：

```
[root@localhost test]# cat /etc/fstab.bak    #显示文本文件的内容
```

输出信息如下：

```
hello! This is a Demo for ed!
#
# /etc/fstab
# Created by anaconda on Sat Dec  3 04:40:30 2022
......省略部分输出内容......
/dev/mapper/rhel-home      /home    TTT      defaults      0 0
/dev/mapper/rhel-swap      none     swap     defaults      0 0
```

说明：在上面的输出信息中，第一行的内容 hello! This is a Demo for ed!是 ed 编辑器插入的文本。文件中的字符串 xfs 被替换成了 TTT。

【相关指令】sed

2.4　ex 指令：以 Ex 模式运行 vi 指令

【语　　法】ex [参数]　　★★★★☆

【功能介绍】ex 指令以 Ex 模式（单行模式）启动 vi 编辑器。它与指令 vi -E 的运行效果等同。

【参数说明】

参　数	功　能
文件	指定待编辑的文件

【经验技巧】ex 指令是 vi 的单行编辑模式，当 vi 指令进入 Ex 模式时，输入 visual 即可恢复全屏编辑模式，此时将具有 vi 编辑器的全部功能。

【示例 2.4】使用 vi 的 Ex 模式编辑文件，具体步骤如下。

（1）ex 指令是 vi 编辑器的单行编辑模式。在命令行中输入下面的命令：

```
#用 Ex 模式编辑文件/etc/passwd
[root@localhost test]# ex /etc/passwd
```

输出信息如下：

```
"/etc/passwd" 46L, 2186C
Entering Ex mode.  Type "visual" to go to Normal mode.
:
```

说明：上面的输出信息中，可以在冒号后面输入相关的操作命令。

（2）在冒号提示符下输入行号，可以显示指定行号的内容。在命令行中输入下面的命令：

```
:3                                  #显示第 3 行的内容
```

```
#文件/etc/passwd 第 3 行的内容
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

（3）在冒号提示符下输入 q 命令，退出 ex 编辑器。在命令行中输入下面的命令：

```
:q                                    #退出 ex 编辑器
[root@localhost test]#                #回到 Shell 提示符状态
```

【相关指令】vi

2.5　jed 指令：程序员的文本编辑器

【语　　法】jed[选项][参数]　　★★★★☆

【功能介绍】jed 指令是由 Slang 开发的，其主要用途是编辑程序的源代码。它支持彩色语法加亮显示，可以模拟 Emacs、EDT、Wordstar 和 Brief 编辑器。

【选项说明】

选　　项	功　　能
-n	不（do not）加载配置文件.jedrc
-2	将jed运行窗口分隔为上下两个编辑区
-batch	以批处理方式运行jed，这是一种非交互式的操作方式
-f <函数>	执行指定的函数（function）
-g <行号>	打开文件并将光标移动到指定的行
-i <文件>	将指定的文件插入（insert）当前缓冲区
-s <字符串>	查找（search）指定的字符串

【参数说明】

参　　数	功　　能
文件	指定待编辑的文件列表

【经验技巧】jed 指令是专门为程序设计人员准备的文本编辑器，支持多种编程语言的语法加亮显示，可以模拟多种编辑器。

【示例 2.5】编辑 Shell 脚本文件。

jed 指令可以给程序员提供友好的显示和操作界面，便于程序员编辑程序源代码。例如，编辑 Shell 脚本文件，在命令行中输入下面的命令：

```
[root@hn ~]# jed /etc/rc.d/rc          #使用 jed 编辑 Shell 脚本
```

jed 指令的输出信息会占满整个终端屏幕，为了节省篇幅，此处省略。

说明：jed 指令的输出信息将会以彩色加亮显示，使用 F10 键可以激活屏幕最上方的菜单命令。

【相关指令】vi，ed

2.6　nano 指令：文本编辑器

【语　　法】nano [选项] [参数]　　★★★☆☆

【功能介绍】nano 是 UNIX 和类 UNIX 系统中的一个文本编辑器，是 Pico 的替代品。它操作简单，提供了丰富的快捷键。常用的快捷键如下。

快捷键（^代表Ctrl键）	功 能 描 述
^G	获得nano的帮助信息
^O	保存文件内容。如果是新文件，则需要输入文件名
^R	在当前光标位置插入指定的一个文本文件中的内容
^Y	向前翻页
^V	向后翻页
^W	对文件进行搜索
^K	剪切当前行到粘贴缓冲区
^U	粘贴缓冲区中的内容到当前光标所在位置
^C	显示当前光标位置
^T	调用拼写检查功能，对文档进行拼写检查（仅限英文）
^J	段落重排
^X	退出，当文件内容发生改变时，提示是否保存修改

【选项说明】

选　　项	功　　能
-h	显示帮助信息（help）
-j	使用切换功能
-k	当使用剪切命令时，把光标所在行的内容全部删除
-m	激活鼠标（mouse）选择对应命令的功能
-n	不要（do not）读取文件，仅写入
-o <工作目录>	设置指令的操作（operating）目录
-s <拼写检查器>	指定进行指令拼写检查的拼写（spell）检查器
-t	当使用^X退出时，保存更改后的缓冲区而不提示
-v	以只读方式查看（view）文件内容

续表

选　项	功　能
-w	关闭自动换行功能，以便编辑长内容
-x	关闭屏幕下方的命令帮助信息
-z	支持Ctrl+z键中止程序的运行，将其放到后台作业
+<行号>	当进入编辑模式时，将光标移动到指定的行号上

【参数说明】

参　数	功　能
文件	指定要编辑的文件

【示例 2.6】用 nano 编辑指定的文本文件。在命令行中输入下面的命令：

```
[root@localhost ~]# nano  /etc/fstab          #编辑文本文件
```

输出信息如下：

```
  GNU nano 5.6.1           /etc/fstab
......省略部分输出内容......
/dev/mapper/rhel-home   /home     xfs       defaults       0 0
/dev/mapper/rhel-swap   none      swap      defaults       0 0


^G 帮助    ^O 写入    ^W 搜索    ^K 剪切    ^T 执行命令     ^C 位置
M-U 撤销    M-A 设置标记
^X 离开    ^R 读档    ^\ 替换    ^U 粘贴    ^J 对齐         ^_ 跳行
M-E 重做    M-6 复制
```

说明：使用 nano 指令对文本文件进行操作时，只要注意查看屏幕下方的快捷键帮助即可方便地完成所有操作步骤。

【相关指令】vi

2.7　sed 指令：用于文本过滤和转换的流式编辑器

【语　　法】sed [选项] [参数]　　★★★★★

【功能介绍】sed 指令是一个流（stream）式文本编辑器（editor），用于在输入流（可以是一个文本文件或者从命令管道送来的文本内容）上处理基本的文本转换。sed 指令还具有强大的文本过滤功能。

sed 指令在工作时首先将文本文件中的一行内容读取到称为“模式空间”（pattern space）的临时缓冲区中，然后对文本进行处理，处理完成后将缓冲区中的文本显示到标准输出设备（显示终端）上，然后处理下一行文本，重复此

过程，直到文件结束。

sed 指令支持丰富的内部命令，常用的命令有 d（删除指定的行）、s（替换指定的文本）、i（插入文本）。

【选项说明】

选　　项	功　　能
-n或--quiet或--silent	禁止模式空间（pattern space）自动显示到标准输出设备上，除非显式地要求显示模式空间的内容
-e <脚本>或--expression=<脚本>	在命令行中添加脚本并执行
-f <脚本文件>或--file=<脚本文件>	在命令行中添加脚本文件并执行
-i<后缀>或--in-place=<后缀>	在适当的位置编辑文件。如果提供后缀，则执行备份操作
-c或--copy	当使用-i选项移动文件时，使用复制操作代替重命名操作（避免改变输入文件的所有权）
-l <数字>或 --line-length=<数字>	指定行的最大字符长度。当超过此值时将自动换行
--posix	关闭所有的GNU扩展功能
-r或--regexp-extended	在脚本中使用扩展的规则表达式
-s或--separate	将每个文件看作单独的文本，而不是将所有文件看作一个长的文本流
-u或--unbuffered	从文件中加载最少的数据量，增加清空输出缓冲区的频率

【参数说明】

参　　数	功　　能
文件	指定待处理的文本文件列表

【经验技巧】

- sed 指令是单行文本流式编辑器，它一次处理一行内容。当 sed 指令工作时，首先将当前行保存在称为“模式空间”（pattern space）的临时缓冲区，然后使用内部命令处理缓冲区中的内容，最后将处理完的“模式空间”的内容在显示终端上显示。处理完一行内容后接着处理下一行，直到文件结束。
- 使用 sed 指令处理文本文件时，原来的文本文件的内容是不发生改变的。除非使用 Shell 的重定向功能保存输出的内容。
- sed 指令可以自动编辑或者处理一个或多个文本文件，这样可以极大地简化对文本文件的反复操作和对文件内容的转换等。

- 如果文本文件很大（例如一个文本文件达到 2GB），直接使用 vi 之类的编辑器的效率是很低的，也有可能根本打不开文件。这种情况下，使用 sed 进行文本处理是非常合适的。
- sed 指令的内部命令最好使用单引号引起来，以防止 Shell 扩展一些特殊的字符时影响程序的执行。

【示例 2.7】删除指定的行，具体步骤如下。

（1）使用 sed 指令的内部命令 d 可以删除指定的行。例如，删除文件的第一行，在命令行中输入下面的命令：

```
#删除文件 fstab 的第一行
[root@localhost ~]# sed -e '1d' /etc/fstab
```

输出信息如下：

```
# /etc/fstab
# Created by anaconda on Sat Dec  3 04:40:30 2022
......省略部分输出内容......
/dev/mapper/rhel-swap   none          swap    defaults        0 0
```

说明：上面显示的内容是将文件“/etc/fstab”第一行删除之后的结果。

（2）显示文件“/etc/fastab”的原始内容（注意，源文件的内容是不发生变化的）并与上面的输出信息进行对比。在命令行中输入下面的命令：

```
[root@hn ~]# cat /etc/fstab                  #显示文本文件的内容
```

输出信息如下：

```
#
# /etc/fstab
# Created by anaconda on Sat Dec  3 04:40:30 2022
......省略部分输出内容......
/dev/mapper/rhel-swap   none          swap    defaults      0 0
```

（3）使用 d 命令还可以删除多行内容。在命令行中输入下面的命令：

```
#删除文件 fstab 的 1～3 行
[root@localhost ~]# sed -e '1,3d' /etc/fstab
```

输出信息如下：

```
#
# Accessible filesystems, by reference, are maintained under
'/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for
more info.
......省略部分输出内容......
/dev/mapper/rhel-swap   none    swap    defaults        0 0
```

说明：在本例中，逗号前后的数字分别表示要删除的起始行和结束行。

【示例 2.8】删除文件中以“#”开头的行，具体步骤如下。

（1）sed 指令支持规则表达式，对符合规则表达式匹配规则的内容执行相应的操作。在命令行中输入下面的命令：

```
#删除文件/etc/vsftpd/vsftpd.conf 中以#开头的行
[root@localhost ~]#sed -e '/^#/d' /etc/vsftpd/vsftpd.conf
```

输出信息如下：

```
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=NO
listen_ipv6=YES

pam_service_name=vsftpd
userlist_enable=YES
```

说明：文件“/etc/vsftpd/vsftpd.conf”中含有很多以“#”开头的注释内容。在上面的输出信息中已经将以“#”开头的注释内容删除。

（2）显示文件“/etc/vsftpd/vsftpd.conf”的原始内容并与上面的输出信息进行对比。在命令行中输入下面的命令：

```
#显示文本文件 vsftpd.conf 的内容
[root@hn ~]# cat /etc/vsftpd/vsftpd.conf
```

输出信息如下：

```
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample
file
......省略部分输出内容......
pam_service_name=vsftpd
userlist_enable=YES
```

【示例 2.9】替换指定的内容，具体步骤如下。

（1）使用 sed 指令的内部命令 s 可以替换指定的内容，例如，将文件/etc/fstab 中的 defaults 替换为 hello。在命令行中输入下面的命令：

```
#将文件“/etc/fstab”中的 defaults 替换为 hello
[root@hn ~]# sed -e 's/defaults/hello/g' /etc/fstab
```

输出信息如下：

```
......省略部分输出内容......
/dev/mapper/rhel-root       /         xfs       hello      0 0
```

```
UUID=beb1b7b8-e7f6-498e-    /boot     xfs       hello      0 0
8ab3-ce90265b0e48
/dev/mapper/rhel-home       /home     xfs       hello      0 0
/dev/mapper/rhel-swap       none      swap      hello      0 0
```

（2）输出文件“/etc/fstab”的原始内容并与上面的输出信息进行对比。在命令行中输入下面的命令：

```
[root@hn ~]#cat /etc/fstab                          #显示文本文件的内容
```

输出信息如下：

```
......省略部分输出内容......
/dev/mapper/rhel-root       /         xfs       defaults   0 0
UUID=beb1b7b8-e7f6-498e-    /boot     xfs       defaults   0 0
8ab3-ce90265b0e48
/dev/mapper/rhel-home       /home     xfs       defaults   0 0
/dev/mapper/rhel-swap       none      swap      defaults   0 0
```

【相关指令】ed

2.8　joe 指令：全屏文本编辑器

【语　　法】joe [选项] [参数]　　★★★☆☆

【功能介绍】joe 指令是一款功能强大的纯文本编辑器，拥有众多编写程序和文本的优良特性。

【选项说明】

选　　项	功　　能
-force	当保存文件时，强制在文件的最后一行添加换行符
-lines <数字>	指定屏幕显示的行数
-lightoff	执行块命令后，取消块的加亮显示
-autoindent	自动缩进，对于编写代码很有帮助

【参数说明】

参　　数	功　　能
文件	指定要编辑的文件

【经验技巧】joe 指令内置了众多操作指令，可以在打开 joe 编辑器后，按组合键 Ctrl+K+H 显示这些操作指令。

【示例 2.10】使用 joe 编辑文本文件，具体步骤如下。

（1）使用 joe 指令打开要编辑的文本文件。在命令行中输入下面的命令：

```
[root@hn ~]# joe /etc/fstab          #使用 joe 打开文本文件 fstab
```

输出信息如下：

```
  I      /etc/fstab                                  Row 1    Col 1

......省略部分输出内容......
/dev/mapper/rhel-root       /         xfs       defaults    0 0
UUID=beb1b7b8-e7f6-498e-    /boot     xfs       defaults    0 0
8ab3-ce90265b0e48
/dev/mapper/rhel-home       /home     xfs       defaults    0 0
/dev/mapper/rhel-swap       none      swap      defaults    0 0
```

（2）在 joe 指令的界面中按组合键 Ctrl+K+H 显示操作指令的含义。输出信息如下：

```
    REGION    GO TO    GO TO    DELETE   EXIT     SEARCH
    ^Arrow Select ^Z Prev. word^U/^V PgUp/PgDn   ^D Char.
^KX Save ^KF Find
    ^KB Begin          ^X Next word MISC          ^Y Line  ^C
Abort     ^L  Next
    ^KK End   ^KU Top of file   ^KJ Paragraph^W >Word ^KQ All
HELP
    ^KC Copy  ^KV End of file   ^KA Center line   ^O Word< FILE
Esc . Next
    ^KM Move  ^A Beg. of line   ^K Space Status   ^J >Line ^KE
Edit Esc , Prev
    ^KW File  ^E End of line    SPELL    ^[O Line<     ^KR Insert
^KH Off
    ^KY Delete    ^KL To line no.   Esc N Word    ^_ Undo  ^KD
Save ^T  Menu
    ^K/ Filter    ^G  Matching (    Esc L File    ^^ Redo  ^K`
Revert
  I      /etc/fstab                                  Row 1    Col 1

......省略部分输出内容......
/dev/mapper/rhel-home      /home    xfs    defaults       0 0
/dev/mapper/rhel-swap  none     swap    defaults       0 0
```

说明：上面的输出信息中，上半部分为快捷键的帮助信息，下半部分为文件的正文。

2.9　习　　题

一、填空题

1. 在操作系统中信息以_____的方式保存在存储介质上，而_____则是最常使用的文件格式。

2．vi 编辑器支持_____模式和_____模式。

3．sed 是一个_____文本编辑器，用于在输入流上处理基本的文本转换。

二、选择题

1．下面的（　　）编辑器是全屏文本编辑器。

A．vi　　B．emacs　　C．ed　　D．joe

2．使用 vi 编辑器编辑文本时，按（　　）键进入编辑模式。

A．A　　B．I　　C．O　　D．Y

3．使用 vi 编辑器时，（　　）命令用来保存并退出文本操作。

A．:w　　B．wq　　C．:q　　D．:q!

三、判断题

1．使用 vi 编辑器编辑文件时，可以在命令模式下使用“:set nu”和“:set nonu”来显示和取消行号。（　　）

2．ed 指令是一个行文本编辑器，包括命令和输入两种模式。（　　）

3．sed 指令是单行文本流式编辑器，它一次只能处理一行内容。（　　）

四、操作题

1．使用 vi 编辑器创建一个文本文件 test 并输入如下内容：

```
Hello World!
This is a text document.
```

2．为 test 文件的内容添加行号。

3．保存并退出 test 文本编辑模式。