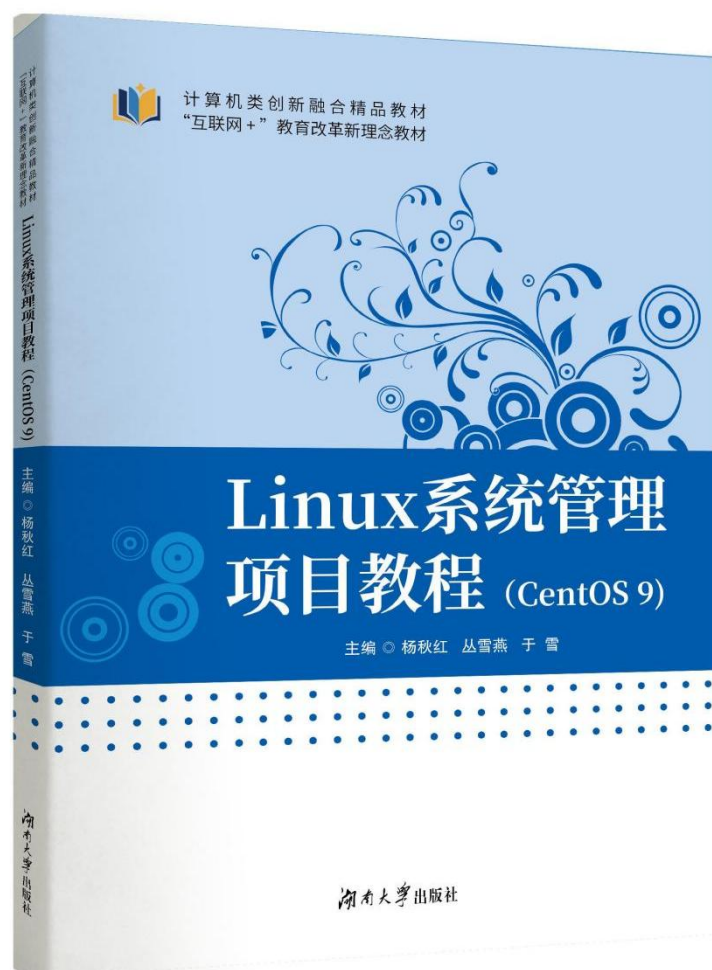


Linux 系统管理项目教程 (CentOS 9)



类目：计算机类

书名：Linux 系统管理项目教程 (CentOS 9)

主编：杨秋红 丛雪燕 于雪

出版社：湖南大学出版社

开本：大 16 开

书号：978-7-5667-3950-6

使用层次：通用

出版时间：2024 年 12 月

定价：49.80 元

印刷方式：双色

是否有资源：是



计算机类创新融合精品教材
“互联网+”教育改革新理念教材

计算机类创新融合精品教材
“互联网+”教育改革新理念教材
Linux系统管理项目教程 (CentOS 9)



Linux系统管理 项目教程 (CentOS 9)

主编◎杨秋红 丛雪燕 于雪

Linux系统管理 项目教程 (CentOS 9)

主编◎杨秋红 丛雪燕 于雪



责任编辑：胡戈特
封面设计：旗语书装



定价：49.80元

湖南大学出版社

湖南大学出版社



计算机类创新融合精品教材
“互联网+”教育改革创新理念教材



Linux 系统管理 项目教程 (CentOS 9)

主 审 © 冯雪丽

主 编 © 杨秋红 丛雪燕 于 雪

湖南大学出版社

· 长 沙 ·

图书在版编目(CIP)数据

Linux 系统管理项目教程：CentOS 9 / 杨秋红, 丛雪燕, 于雪主编. -- 长沙：湖南大学出版社, 2024.
12. -- ISBN 978-7-5667-3950-6
I. TP316. 85
中国国家版本馆 CIP 数据核字第 2024CC5556 号

Linux 系统管理项目教程 (CentOS 9)

Linux XITONG GUANLI XIANGMU JIAOCHENG (CentOS 9)

主 编：杨秋红 丛雪燕 于 雪
责任编辑：胡戈特
印 装：涿州汇美亿浓印刷有限公司
开 本：889 mm×1194 mm 1/16 印 张：14.5 字 数：460 千字
版 次：2024 年 12 月第 1 版 印 次：2024 年 12 月第 1 次印刷
书 号：ISBN 978-7-5667-3950-6
定 价：49.80 元

出 版 人：李文邦
出版发行：湖南大学出版社
社 址：湖南·长沙·岳麓山 邮 编：410082
电 话：0731-88822559 (营销部) 88821174 (编辑室) 88821006 (出版部)
传 真：0731-88822264 (总编室)
网 址：<http://press.hnu.edu.cn>
电子邮箱：xiaoshulianwenhua@163.com

版权所有，盗版必究
图书凡有印装差错，请与营销部联系

前言



Linux，全称 GNU/Linux，是一种可供免费使用和自由传播的类 UNIX 操作系统，其内核由林纳斯·托瓦兹（Linus Torvalds）于 1991 年 10 月 5 日首次发布，它主要受到 MINIX 和 UNIX 思想的启发，是一个基于 POSIX（可移植操作系统接口）的支持多用户、多任务、多线程和多 CPU 的操作系统。它支持 32 位和 64 位硬件，能运行主要的 UNIX 工具软件、应用程序和网络协议。

党的二十大报告指出，教育、科技、人才是全面建设社会主义现代化国家的基础性、战略性支撑。必须坚持科技是第一生产力、人才是第一资源、创新是第一动力，深入实施科教兴国战略、人才强国战略、创新驱动发展战略，开辟发展新领域新赛道，不断塑造发展新动能新优势。

全书共 10 个项目，包括安装与配置 Linux 操作系统、Linux 操作系统常用命令、管理 Linux 服务器的用户和组、文本管理、配置网络功能、文件系统和磁盘管理、配置与管理防火墙和 SELinux、配置与管理代理服务器、编写 shell 脚本、使用 gcc 和 make 调试程序等。

本书特色如下：

①实践导向：本书以实践为导向，重在演示如何在 CentOS 9 系统上解决实际问题 and 执行任务。提供大量实用示例，以及配套的实践指导和测试场景。

②系统全面：本书涵盖了从系统入门基础到高级管理的所有主题，包括操作系统理论、文件系统、权限管理、网络配置、系统安全、性能优化等方面。对于每个主题，我们都提供了详细的教程和实践指导。

③深度解析：本书深入解析了 Linux 系统的运作机制和各部分间的相互关系，帮助读者更深入地理解 Linux 系统的底层结构和设计原理。

④创新教学方式：本书通过清晰的图解、代码示例和详细的注解，使复杂的概念和技术变得易于理解。

本书可作为人工智能技术等相关专业的教材，也可作为 Linux 系统管理和网络管理人员的自学用书。

本书在编写过程中，参考和借鉴了不少 CentOS 9 相关的文献资料、网络资源和研究成果，在此向相关作者一并表示真诚的感谢！由于编者水平有限，书中难免有疏漏之处，敬请各位读者在使用本书时批评指正。

编 者

2024 年 8 月

目 录

项目一 安装与配置 Linux 操作系统	1
任务一 认识 Linux	2
任务二 安装 Linux 操作系统	10
任务三 VMware 虚拟机的备份	17
项目二 Linux 操作系统常用命令	23
任务一 Linux 命令行基础	24
任务二 文件目录类命令的熟练使用	41
任务三 系统信息类命令的熟练使用	61
任务四 进程管理类命令 Linux 系统管理的熟练使用	62
任务五 其他常用命令的熟练使用	81
项目三 管理 Linux 服务器的用户和组	84
任务一 用户和组管理概述	85
任务二 账户配置文件	86
任务三 用户账户管理命令	91
任务四 组账户管理命令	100
任务五 访问权限管理	105
项目四 文本管理	114
任务一 Vim 编辑器概述	115
任务二 Vim 编辑器的操作流程	116
项目五 配置网络功能	118
任务一 了解 VMware 的网络工作	119
任务二 配置网络功能	124
任务三 配置和使用 SSH 服务	130

目 录

项目六 文件系统和磁盘管理	148
任务一 了解文件系统	149
任务二 管理磁盘	154
任务三 在 Linux 中配置软 RAID	163
任务四 LVM 逻辑卷管理器	165
项目七 配置与管理防火墙和 SELinux	173
任务一 防火墙概述	174
任务二 使用 firewalld 服务	179
任务三 管理 SELinux	183
项目八 配置与管理代理服务器	194
任务一 代理服务器概述	195
任务二 安装与配置 squid 服务器	196
项目九 编写 shell 脚本	201
任务一 创建 Shell 脚本	202
任务二 条件测试与分支结构	205
任务三 循环结构	207
项目十 使用 gcc 和 make 调试程序	211
任务一 程序调试概述	212
任务二 使用传统程序语言进行编译	213
任务三 使用 make 进行宏编译	218
参考文献	224



项目一 安装与配置 Linux 操作系统

项目概述

本项目旨在为初学者提供关于 Linux 操作系统的入门和应用知识。Linux 作为一种强大的开源操作系统，具有广泛的应用领域，包括服务器管理、嵌入式系统、云计算等。通过本项目，学生将深入了解 Linux 的基本概念、特点，如何在实际环境中安装、配置以及应用 Linux 操作系统。

学习目标

知识目标

- ①了解 Linux 操作系统的发展历史，包括其起源和演变过程。
- ②能够辨别不同版本的 Linux 发行版，了解它们的特点和应用场景。
- ③熟悉 CentOS Stream 9 系统的安装过程，包括系统准备、分区设置等步骤。
- ④了解不同的 VMware 虚拟机备份方法，包括手动复制、导出为 OVF 模板、克隆和类似物理机备份的方法。

能力目标

- ①掌握 Linux 操作系统的安装，特别是针对 CentOS Stream 9 的安装。
- ②能够熟练进行手动复制虚拟机磁盘文件的操作。
- ③具备分析和解决 Linux 系统中常见问题的能力。

思政目标

- ①培养学生对 Linux 操作系统的学习兴趣和积极态度。
- ②提高学生对操作系统领域的认识和理解，培养其综合运用知识解决实际问题的能力。
- ③培养学生独立探索、分析和解决问题的能力。
- ④提升学生的团队协作和沟通表达能力。

任务一 认识 Linux

Linux 操作系统是一套可以免费使用和自由传播的类 UNIX 操作系统，其性能稳定，自 20 世纪 90 年代诞生以来，受到了用户们的广泛欢迎，应用领域不断扩大，既包括传统服务器领域，又包括新兴的云计算、大数据、人工智能等前沿科技领域，影响力长期雄踞操作系统领域榜首。本章将介绍 Linux 操作系统的发展历史、发行版和主要应用领域等内容，以使读者熟悉 Linux 操作系统的相关背景知识及安装和使用方法。



知识之窗

国产操作系统

典型的国产操作系统包括深度 (Deepin)、红旗 Linux (Red Flag Linux)、银河麒麟 (Kylin)、中标麒麟 (NeoKylin)、起点操作系统 (StartOS)、中兴新支点 (NewStart)、华为鸿蒙 (Harmony OS) 等。

一、Linux 操作系统的发展历史

Linux 操作系统由林纳斯·托瓦兹 (Linus Torvalds) 发明，并在众多网络上松散的“黑客”团队的帮助下得以发展和完善。在介绍 Linux 操作系统的发展历史之前，我们先介绍一些与 Linux 诞生和发展密不可分的因素。

(一) UNIX 操作系统的发展历史

UNIX 是一款强大的、支持多用户/多任务的操作系统，它支持多种处理器架构，属于分时操作系统。操作系统 (operating system, OS) 的概念始于 20 世纪 50 年代。当时的操作系统主要是批处理操作系统，没有配备鼠标、键盘等设备，典型的输入设备是卡片机。系统运行批处理程序，通过卡片机读取卡纸上的数据，然后将处理结果输出。20 世纪 60 年代初，分时操作系统出现，与批处理操作系统不同，它支持用户交互，还允许多个用户从不同的终端同时操作主机。

1965 年，美国贝尔实验室 (Bell Laboratory)、麻省理工学院 (Massachusetts Institute of Technology, MIT)、通用电气公司 (General Electric Company, GEC) 共同参与研发 MULTiplexed 信息与计算系统 (MULTiplexed Information and Computing System, MULTICS)。这是一个安装在大型主机上的分时操作系统，研发的目的是让大型主机同时支持 300 个以上的终端访问。MULTICS 在当时非常新颖，然而项目进展并不顺利。因进度缓慢、资金短缺，贝尔实验室选择退出该项目，最终 MULTICS 并没有取得很好的市场反响。MULTICS 项目最重要的成就是培养了很多优秀的人才，如肯·汤普森 (Ken Thompson)、丹尼斯·里奇 (Dennis Ritchie)、道格拉斯·麦克罗伊 (Douglas McIlroy) 等。

1969 年 8 月，肯·汤普森为了移植一款名为“太空旅游”的游戏，想要开发一个小的操作系统。他在一台闲置的 PDP-7 上用汇编语言写出了一组内核程序、一些内核工具程序以及一个小的文件系统。他的同事将其称为 Unics (该系统就是 UNIX 的原型)。因为汇编语言对硬件具有依赖性，Unics 只能应用于特定硬件上。如果想将其安装到不同的机器上，就需要重新编写汇编语言代码。为了提高其可移植性，肯·汤普森与丹尼斯·里奇合作，试图改用高级程序设计语言来编写 Unics。他们先后尝试过基本的组合编程语言 (basic combined programming language, BCPL)、Pascal 等语言，但是编译出来的内核性能都不是很好。

1973 年，丹尼斯·里奇在 B 语言的基础上，发明了 C 语言，因此他被人们称为“C 语言之父”。肯·汤普森与丹尼斯·里奇合作，用 C 语言重新改写 UNIX 的内核，并在改写过程中增加了许多新特征。例如，道格拉斯·麦克罗伊（Douglas Mcuroy）提出的“管道”的概念被引入 UNIX。经 C 语言改写后的 UNIX，可移植性非常好。理论上，只要获得 UNIX 的源码，针对特定主机的特性加以修改，就可以将其移植到对应的主机上。

由于 UNIX 的高度可移植性和强大的性能，加上当时并没有版权的纠纷，因此很多商业公司开始了 UNIX 操作系统的开发，研发了许多重要的 UNIX 分支。

1977 年，美国加利福尼亚大学伯克利分校的比尔·乔伊（Bill Joy）通过移植 UNIX，开发了伯克利软件套件（Berkeley software distribution, BSD）。比尔·乔伊是美国 Sun 公司（Sun Microsystems）的创始人。Sun 公司基于 BSD 内核进行了商业版本 UNIX 的开发。BSD 是 UNIX 中非常重要的一个分支，FreeBSD 就是由 BSD 改版而来的，苹果的 Mac OS X 也是从 BSD 发展而来的。

1979 年，美国电话电报公司（AT&T）推出了 System V 第 7 版 UNIX，开始支持 X86 架构的个人计算机（personal computer, PC）平台。贝尔实验室当时还属于 AT&T。AT&T 出于商业考虑，在第 7 版 System V 中特别提到了“不能对学生提供源码”的严格限制。

1984 年，因为 AT&T 规定“不能对学生提供源码”，安德鲁·坦尼鲍姆（Andrew Tanenbaum）老师以教学为目的，编写了与 UNIX 兼容的 MINIX。1989 年，安德鲁·坦尼鲍姆将 MINIX 系统移植到 X86 架构的 PC 平台。1990 年，Linux 的创始人林纳斯·托瓦兹首次接触 MINIX 系统，并立志开发一个比 MINIX 性能更好的操作系统。

（二）GNU 计划和 GPL 许可证

Linux 的诞生离不开 UNIX 操作系统和 MINIX 操作系统，而 Linux 的发展离不开 GNU 计划（GNU project）。

GNU 计划开始于 1984 年，其创始人是理查德·斯托曼（Richard Stallman）。“GNU”是“GNU's Not UNIX”的首字母缩写词，“GNU”的发音为 g'noo。GNU 计划的目的是开发一款自由、开放的类 UNIX 操作系统。类 UNIX 操作系统中用于资源分配和硬件管理的程序称为“内核”，GNU 的内核称为“Hurd”。Hurd 的开发工作始于 1990 年，但是至今仍未成熟。GNU 计划的典型产品包括 GCC、Emacs、Bash Shell 等，这些都在 Linux 中被广泛使用。

1985 年，理查德·斯托曼创立了自由软件基金会（Free Software Foundation, FSF）为 GNU 提供技术、法律以及财政支持。尽管 GNU 计划大部分时候依靠个人自愿无偿贡献，但 FSF 有时还是会聘请程序员帮助编写。当 GNU 计划开始逐渐获得成功时，一些商业公司开始介入开发并提供技术支持。其中非常著名的就是之后被 Red Hat 兼并的 Cygnus Solutions 公司。

为了避免 GNU 开发的自由软件被其他人用作专利软件，GNU 通用公共许可证（general public license, GPL）于 1985 年被提出。GPL 试图保证用户共享和修改自由软件的自由，适用于大多数 FSF 的软件。GNU 计划一共提出了 3 个许可证条款：GNU GPL、GNU 较宽松公共许可证（GNU lesser general public license, GNU LGPL）、GNU 自由文档许可证（GNU free documentation license, GNUFDL）。

FSF 中的“F”的意思是“自由”而不是“免费”，所以只要在保证用户充分自由（可以获取源码，可以修改或者重新发布）的前提下，完全可以收费。例如，Red Hat Enterprise Linux 是一款商业产品，但是它的源码是公开的。CentOS 就是在 Red Hat Enterprise Linux 的源码上，进行重新修改而形成的一个 Linux 发行版。

知识之窗

典型的开源许可证

开源许可证 (open source license) 种类繁多, 其中最有影响力的主要包括 GNU 系列、BSD 系列、Apache 系列、MIT 系列等。不同类型的开源许可证对用户权利的保护范围是不一样的, 用户需要根据自己的需求谨慎选择。

(三) Linux 操作系统的诞生和发展

1991 年年底, 林纳斯·托瓦兹公开了 0.02 版本的 Linux 内核源码, 迅速吸引了一些“黑客”关注, 这些“黑客”的加入使它很快就具有了许多吸引人的特性。1993 年, Linux 1.0 发布。1994 年, Linux 的第一个商业发行版 Slackware Linux 问世。1996 年, 美国国家标准与技术研究院 (National Institute of Standards and Technology, NIST) 确认 Linux 1.2.13 符合 POSIX 标准。同年, Linux 2.0 发布, 并将 Linux 的标志确定为企鹅。

知识之窗

Linux 命名之争

自由软件社区内对 Linux 操作系统的命名存在一定的争议。FSF 的创立者理查德·斯托曼及其支持者认为, Linux 操作系统既包括 Linux 内核, 也包括 GNU 项目的大量软件, 因此应当使用 GNU/Linux 这一名称。Linux 社区中的成员则认为使用 Linux 命名更好。

二、Linux 操作系统的发行版

根据上下文语境不同, Linux 存在两种含义: Linux 内核 (Linux kernel) 和 Linux 发行版 (Linux distribution)。

(一) Linux 内核与发行版

Linux 内核一般特指前文所提及的、由林纳斯·托瓦兹发明的 Linux。Linux 是全球最有影响力的开源项目之一。读者可以访问 Linux 官方网站免费获取 Linux 内核源码和其他资讯, 并可以在 GPL 许可证的框架内自由使用。截至 2022 年 10 月 6 日, 官方公开的 Linux 内核最新版本为 6.0, 最新的稳定版本 (Latest stable kernel) 为 5.19.14。目前可在官方获取到的最新 Linux 内核代码文件为 100 MB 的压缩包。

Linux 发行版由 Linux 内核以及大量基于 Linux 的应用软件和工具软件整合而成。目前已有超过 500 个 Linux 发行版, 还有近 500 个正在开发中。典型的 Linux 发行版包括 Linux 内核、GNU 工具和库、附加软件、文档、窗口系统、窗口管理器和桌面环境、软件包管理系统等。不同的发行版由不同的团队维护, 由于定位不同, 各个发行版通常具有各自的特点, 可以满足不同类型用户的需求。

大多数 Linux 发行版包含的软件都是免费的开源软件。各个发行版中集成的软件种类和版本通常并不完全相同。大多数软件包可以在存储库中在线获得, 这些存储库通常分布在世界各地。除了一些核心组件外, 只有极少数软件是由 Linux 发行版的维护人员从头编写的。Linux 发行版通常也可能包括一些源码不公开的专有软件, 例如某些设备驱动程序所需的二进制代码。

（二）图形用户界面概述

Linux 发行版通常为用户提供了图形用户界面（graphical user interface, GUI）。需要注意的是，Linux 操作系统本身并没有 GUI。Linux 发行版的 GUI 解决方案通常基于 X Window System 实现。GUI 的引入，拓宽了 Linux 的应用场景，降低了初学者使用 Linux 操作系统的难度。而诸如排版、制图、多媒体等典型的桌面应用，更是离不开 GUI 的支持。

1. X Window System

X Window System 由麻省理工学院（MIT）于 1984 年提出，它是 UNIX 及类 UNIX 系统最流行的窗口系统之一，是一款跨网络与跨操作系统的窗口系统，可用于几乎所有的现代操作系统。需要注意的是，X Window System 与微软公司的 Windows 操作系统是不同的。后者是一种 GUI 的操作系统，图形环境与内核紧密结合，可直接访问 Windows 内核；而前者只是 Linux 操作系统上的一个可选组件。

X Window System 采用服务器/客户端（C/S）架构，能够通过网络进行 GUI 的存取。X Window System 结构如图 1-1 所示，它由 X 服务器（X Server）、X 客户端（X Client）2 个部分组成。X Client 和 X Server 并不一定位于同一台计算机，两者基于 X 协议（X Protocol）进行通信。

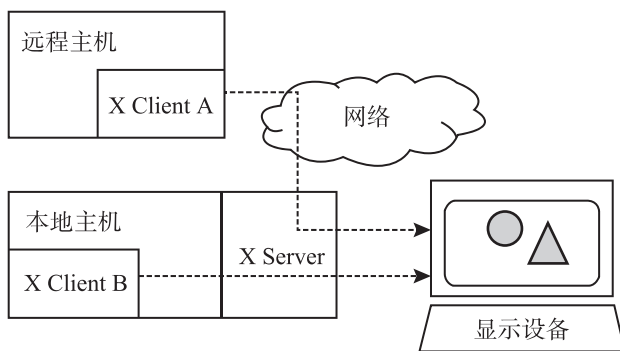


图 1-1 X Window System 结构

有一定计算机网络基础的读者，可能会对图 1-1 中 X Server 和 X Client 所处的位置产生疑惑：X Client 竟然出现在远程主机

中。在读者接触到的大多数 C/S 架构中，两者的位置应该是反过来的。注意，这里的 X Server 和 X Client 是根据图像渲染的职责来区分的。X Server 管理本地主机的大部分硬件（例如键盘、鼠标、显示器），接收用户输入并进行最终显示结果的渲染工作。X Server 收到用户输入后，将请求数据发送给相应 X Client。X Client 通过调用具体的应用来处理数据，产生结果后再将结果返回给 X Server。X Server 维护一个独立的显示控制器，通过响应 X Client 的请求，在其所管理的显示设备上，完成建立窗口、绘制图形和文字等操作。

X Window System 基于 X Protocol 完成服务器和客户端之间的通信。1987 年在 MIT 发布了 X Protocol 的第 11 版。该版协议较为完善，且被广泛应用，因此，后来 X Window System 也被称为 X11。早期 Linux 所使用的 X Window System 的核心都是由 XFree86 计划所提供的，因此许多资料习惯将 X Window System 与 XFree86 两个概念混用。XFree86 计划始于 1992 年，主要维护 X11R6（第 11 版第 6 次发行），包括对新硬件的支持以及新增功能等。X11R6 的维护工作后来由 Xorg 基金会接手。

2. KDE 和 GNOME

X Window System 提供了一个建立窗口的标准，具体的窗口形式由窗口管理器（window manager）决定。窗口管理器是 X Window System 的组成部分，它用来控制窗口的外观，并提供用户与窗口交互的方法。我们可以将窗口管理器看作一类特殊的 X Client 程序，其功能通过向 X Server 发送命令来实现。

对于具有 GUI 的操作系统的用户来说，仅有窗口管理器提供的功能是不够的。为此，开发人员在其基础上，增加了各种功能和应用程序（如会话程序、面板、登录管理器、桌面程序等），提供更完善的图形用户环境，也就是桌面环境（desktop environment）。

KDE 和 GNOME 是最常见的 Linux 桌面环境。KDE（K desktop environment），即 K 桌面环境，

由德国人 Matthias Ettrich 于 1996 年 10 月创建。KDE 中使用的 Qt 链接库早期并未采用开源协议，这限制了其应用，但也推动了 GNU 网络对象模型环境（GNU network object model environment, GNOME）的诞生。需要说明的是，目前 KDE 已经支持 GNU GPL、GNU LGPL 和 Commercial 等不同类型的授权协议。

GNOME 是 GNU 计划的正式桌面环境，也是开放源码运动的一个重要组成部分。GNOME 计划于 1997 年 8 月由米格尔·德·伊卡萨（Miguel de Icaza）和费德里科·梅纳（Federico Mena）发起，目的是取代 KDE。GNOME、KDE 都有自己的窗口管理器，GNOME 曾经使用 Metacity 作为其窗口管理器，2011 年，GNOME3 发表后，默认的窗口管理器被替换成 Mutter。KDE 使用的是 K Win，也有一些单独的窗口管理器，如 FVWM、IceWM 等。CentOS 默认提供 GNOME 与 KDE。Red Hat 默认采用 GNOME。用户可以根据自己的喜好安装并配置不同类型的桌面环境。

（三）典型 Linux 发行版

Linux 发行版类别众多，其中比较有影响力的发行版可以分为两大主流阵营：一是以 Red Hat 为首的阵营，典型的产品包括 RHEL、Fedora、CentOS、CentOS Stream 及它们的衍生品；二是以 Ubuntu 为首的阵营，典型的产品包括 Debian、Ubuntu 以及它们的衍生品。本小节先介绍后者。

1. Debian

Debian 凭借着惊人的软件数量、高度集成的软件包、良好的安全性等特性成为 Linux 领域的佼佼者。著名的 Ubuntu 操作系统就是从 Debian 发展而来的。目前大多数国产的 Linux 发行版都是基于 Debian 或 Ubuntu 发展而来的。

Debian 的发行版及其软件源有 4 个分支：旧的稳定（OldStable）分支、稳定（Stable）分支、测试（Testing）分支、不稳定（Unstable）分支等。所有开发代号均出自皮克斯动画工作室（Pixar）的电影《玩具总动员》。Debian 操作系统目前采用 Linux 内核或者 FreeBSD 内核。同时，让 Debian 支持其他内核的工作也在进行，最主要的工作就是设计 Hurd。Hurd 是由 GNU 项目所设计的自由软件。

2. Ubuntu

Ubuntu 是由南非人马克·沙特尔沃斯（Mark Shuttleworth）创办的。Ubuntu 这一名称来自非洲南部祖鲁语或豪萨语的“Ubuntu”一词，意思是“人性”“我的存在是因为大家的存在”，这是一种非洲的传统价值观。Ubuntu 的第一个正式版本于 2004 年 10 月正式推出。

Ubuntu 是基于 Debian 发行版发展而来的。早期的 Ubuntu 采用 GNOME 桌面环境。而从 11.04 版本起，Ubuntu 发行版放弃了 GNOME 桌面环境，改用 Unity。Ubuntu 早已超越桌面操作系统的范畴，成为世界领先的开源操作系统，广泛应用于 PC、智能物联网、容器、服务器和云端上。Ubuntu 拥有庞大的社区力量，用户可以方便地从社区获得帮助。

Ubuntu 更新速度快，Ubuntu 社区承诺每 6 个月发布一版本，以提供最新、最强大的软件。新版本的发布时间通常在每年的 4 月和 10 月（Ubuntu 6.06 LTS 除外）。Ubuntu 版本编号以“年份的最后一（两）位·发布月份”的格式命名。Ubuntu 的第一个版本就称为 4.10（2004.10）。除了代号之外，每个 Ubuntu 版本在开发之初还设有一开发代号。Ubuntu 开发代号比较有意思，格式为“形容词+动物名称”，且形容词和动物名称的第一个字母要一致。例如，Ubuntu 9.04 发布于 2019 年 4 月，其开发代号是 Disco Dingo。Ubuntu 官方一般每两年会发布一个长期支持（long term support, LTS）版本。LTS 版本提供的软件包可以得到更长时间的支持，稳定性和可持续性更好。例如，2022 年 4 月发布的 Ubuntu22.04 就是 LTS 版本。目前已经诞生了大量基于 Ubuntu 的 Linux 发行版，典型的包括 Elementary OS、Linux Mint、Ubuntu Ultimate Edition 等。

三、Red Hat 系列产品介绍

Red Hat（红帽）公司是 Linux 领域最成功的商业企业之一。下面对 Red Hat 系列代表产品进行介绍。

（一）Red Hat Linux

Red Hat Linux 是 Red Hat 早期版本使用的名称。Red Hat Linux 1.0 发布于 1994 年 11 月。1995 年，Red Hat（红帽）公司正式成立。Red Hat 公司是一家开源解决方案供应商，也是标准普尔 500 指数成员，总部位于美国北卡罗来纳州的罗利市。Red Hat 公司将开源社区项目产品化，使普通企业客户更容易使用开源创新技术。1999 年 8 月，Red Hat 公司上市，实现了华尔街历史上的第八大首日涨幅。

2003 年 4 月，Red Hat Linux 9.0 发布后，Red Hat 公司将全部力量集中在服务器版，也就是 Red Hat Enterprise Linux 版的开发上。2004 年 4 月 30 日，Red Hat 公司正式停止对 Red Hat Linux 9.0 的支持，标志着 Red Hat Linux 的正式完结。原本的桌面版 Red Hat Linux 发行套件则与来自开源社区的 Fedora 计划合并，成为 Fedora Core（第 7 版起，改为 Fedora）。

（二）Fedora Linux

Fedora Linux 由 Fedora 社区开发、Red Hat 公司赞助，目标是创建一套新颖、多功能且自由（开放源代码）的操作系统。Fedora 是商业化的 Red Hat Enterprise Linux 发行版的上游源码。Fedora 被 Red Hat 公司定位为新技术的测试平台，许多新的技术都会在 Fedora 中检验，为 Red Hat Enterprise Linux 的发布奠定基础。

Fedora Core 1 发布于 2003 年末，定位于桌面用户。最早 Fedora Linux 社区的目标是为 Red Hat Linux 制作并发布第三方的软件包。当 Red Hat Linux 停止发行后，Fedora 社区便集成到 Red Hat 赞助的 Fedora Project，目标是开发出由社区支持的操作系统。Red Hat Enterprise Linux 则取代 Red Hat Linux 成为官方支持的系统版本。

Fedora 大约每 6 个月发布新版本。Fedora Core 1 发布于 2003 年 11 月，前 6 版都采用 Fedora Core 命名，第 7 版起开始使用 Fedora。Fedora 7 发布于 2007 年 5 月。2021 年 11 月正式推出 Fedora 35。Fedora Project 每个版本的维护，通常会持续到其下下个版本发布后一个月，也就是每个版本大约维护 13 个月。Fedora 的版本更新频繁，性能和稳定性得不到保证，因此，一般在服务器上不推荐采用 Fedora Core。

（三）Red Hat Enterprise Linux

Red Hat Enterprise Linux，缩写为 RHEL。RHEL 是由 Red Hat 公司提供收费技术支持和更新的 Linux 发行版。Red Hat 现在主要做服务器版的 Linux 开发，在版本上注重性能和稳定性，以及对硬件的支持。由于企业版操作系统的开发周期较长，注重性能、稳定性和服务器端软件支持，因此版本更新相对较缓慢。RHEL 是从其他版本中更改过来的，并没有第 1 版。RHEL 2.1 基于 Red Hat Linux 7.2 开发，发布于 2002 年 3 月 23 日。由于 Red Hat Linux 停止发行，RHEL 4 开始基于 Fedora Core（已经更名为 Fedora）进行开发。2005 年 2 月开始发布的 RHEL 4 基于 Fedora Core 3 开发。2019 年 5 月 7 日，RHEL 8 发布（基于 Fedora 28）。RHEL 又分为 AS、ES、WS 等分支。AS 是 Advanced Server 的简称；ES 是 Enterprise Server 的简称，是 AS 的精简版本；WS 是 Workstation Server 的简称，是 ES 的进一步简化的版本。

Red Hat 的 Fedora Linux 和 Enterprise Linux 都需要遵循 GNU 协议，即需要发布自己的源码。关于免费的 Fedora Linux，我们既可以下载编译后的 ISO 镜像，也可以下载软件包源码。关于收费的 Red Hat Enterprise Linux 系列，我们可以获得 AS/ES/WS 系列的软件包源码 ISO 文件，但由于其是一款商业产品，我们需要购买正式授权方可使用编译后 ISO 镜像。

(四) CentOS

社区企业操作系统 (community enterprise operating system, CentOS) 作为基于 RHEL 源码的社区重新发布版, 在 Linux 发行版中有相当大的影响力, 特别适合需要相对稳定的开发环境且无商业支持需求的开发者。

Red Hat 发布 Red Hat 9.0 后, 全面转向 RHEL 的开发。RHEL 要求用户购买正式授权, 它的二进制代码不再允许免费下载。然而, 由于 RHEL 依然需要遵循 GNU 协议, 即需要发布自己的源码, 这些文件可以被自由地下载、修改代码、重新编译和使用, 从而诞生了众多的 RHEL 的副本, CentOS 是其中表现最为突出的一员。在某种意义上, CentOS 可以认为是 RHEL 的免费版本。CentOS 移除了不能自由使用的 RHEL 商标和一些闭源软件。使用 CentOS, 可以获得与 RHEL 相同的性能和体验感。许多要求高度稳定性的服务器以 CentOS 替代商业版的 RHEL 使用。

CentOS 已经被 Red Hat 公司彻底改变了原有的发展方向。2014 年年初, Red Hat 和 CentOS 社区宣布将开始合作, 将 CentOS 打造成全方位整合开源社区资源的稳定社区发行版。2020 年 12 月 8 日, CentOS 社区在官方博客发布 “CentOS Project shifts focus to CentOS Stream” 和关于该问题的维基百科说明。该博文的发布标志着 CentOS Linux 版本的终结, 同时大幅缩短了 CentOS Linux 8 的支持和维护时间。官方网站的下载页显示, CentOS Linux 8 的支持和维护时间已经变更为至 2021 年 12 月 31 日截止 (原计划 2029 年截止)。CentOS Linux 从 2020 年 12 月以后不会再有 CentOS Linux 9 及之后的版本, 仅有 CentOS Stream 版本。

(五) CentOS Stream

作为 CentOS 的替代, CentOS Stream 不再是 RHEL 原生代码的重新编译版。传统的 “Fedora→RHEL→CentOS” 路径已经成为历史, 取而代之的将是 “Fedora→CentOS Stream→RHEL”。CentOS Stream 是一个持续交付的发行版, 位于 Fedora Linux 之后、RHEL 之前。Red Hat 已经将工作重点从重建 RHEL 的 CentOS Linux 转移到 CentOS Stream, 后者将在当前 RHEL 发布之前进行跟踪。为了实现 CentOS Stream 的稳定性, CentOS Stream 的每个主要版本都从一个稳定的 Fedora Linux 发展而来。CentOS Stream 9 从 Fedora 34 的分支发展而来, RHEL 9 的不同子版本, 将从 CentOS Stream 9 发展而来。随着更新的软件包通过测试并满足稳定性标准, 它们将被合并到 CentOS Stream 9 以及每日构建的 RHEL 9 中。发布到 CentOS Stream 9 的更新与发布到 RHEL 9 子版本的更新是相同的。CentOS Stream 9 现在的形态就是将来 RHEL 9 的样子。

四、Linux 操作系统的主要应用领域

(一) 传统企业级服务器领域

常见的服务器类型包括万维网 (world wide web, WWW) 服务器、数据库服务器、负载均衡服务器、邮件服务器、域名系统 (domain name system, DNS) 服务器、代理服务器等。Linux 因其具备稳定、开源、安全、高效、免费等特点, 被广泛应用于各类传统企业级服务器中。Linux 在为企业提供具有高稳定性和高可靠性的业务支撑的同时, 还有效降低了企业运营成本, 避免了出现商业软件的版权纠纷问题。

(二) 智能手机、平板电脑、上网本等移动终端

随着移动通信技术的发展, 移动终端进入智能化时代。智能手机、平板电脑、上网本等移动终端随处可见。大家对 Android (安卓) 系统肯定不会陌生, 它就是一种基于 Linux 的操作系统, 目前已广泛应用于各类移动终端。2019 年, 华为正式推出 HarmonyOS (鸿蒙) 系统, 该系统一经问世便受到产业界高度关注, 充分反映出大家对优秀国产操作系统的期盼。

（三）物联网、车联网等应用场景

Linux 操作系统在嵌入式应用的领域里广受欢迎。Linux 操作系统开放源码、功能强大、可靠、稳定性强，它支持各类微处理体系结构、硬件设备和通信协议。Linux 操作系统在传统 Linux 内核基础上，经过裁减，就可以移植到嵌入式系统上运行。经过多年的发展，Linux 操作系统已经成功地跻身于主流嵌入式开发平台。

很多开源组织和商业公司对 Linux 进行了一番改造，使其更符合嵌入式系统或物联网应用的需求，例如改为实时操作系统。Brillo 是一种联网底层操作系统，它源于 Android，是对 Android 底层的细化，得到了 Android 的全部支持，例如蓝牙、Wi-Fi 等技术，并且能耗很低，安全性很高，任何设备制造商都可以直接使用。

在车联网方面，Linux Foundation 联合英特尔（Intel）、丰田（Toyota）、三星（SAMSUNG）、英伟达（NVIDIA）等十多家合作伙伴，推出了汽车端的开源车联网系统——Automotive Grade Linux，简称 AGL。作为由 Linux Foundation 牵头的开源项目，AGL 旨在为车联网提供坚实的开源基础，AGL 提供的第一批开源程序包括主屏、仪表盘、空调系统等，未来 AGL 会提供并更新车联网中更多部件的开源系统。

（四）面向日常办公、休闲娱乐等的桌面应用场景

个人桌面操作系统主要面向日常办公和休闲娱乐等场景。绝大多数 Linux 发行版都可以满足这类需求。Windows 平台中常见的软件在 Linux 环境下都有替代品，许多知名软件本身就提供了针对 Linux 平台的版本，例如浏览器（如 Firefox 等）、办公软件（如 OpenOffice、WPS for Linux 等）、即时通信工具（如 QQ for Linux 等）、软件开发工具（如 Eclipse、Visual Studio Code 等）等。尽管如此，Linux 在桌面应用场景中的市场占有率仍然远远不及 Windows 操作系统。其中的问题可能不在于 Linux 桌面系统产品本身，而在于用户的使用理念、操作习惯和应用技能，以及曾经在 Windows 操作系统上开发的软件的移植问题。

知识之窗

WINE

WINE（Wine is Not an Emulator）是一个能够在多种 POSIX-compliant 操作系统（如 Linux、Mac OS X 及 BSD 等）上运行 Windows 应用的兼容层。WINE 可以工作在绝大多数的 UNIX 版本之上。借助 WINE，用户可以在非 Windows 操作系统中运行 Windows 程序。

（五）云计算、区块链、大数据、深度学习等应用场景

随着云计算、区块链、大数据、深度学习等技术迅速发展，作为一个开源平台，Linux 有着独特的核心优势。据 Linux 基金会研究，超过 86% 的企业已经使用 Linux 操作系统进行云计算、大数据平台的构建，目前，Linux 已开始取代 UNIX 成为极受青睐的云计算、大数据平台操作系统。

课内思考

为什么 Linux 在操作系统领域变得如此重要？它相对于其他操作系统的优势是什么？

任务二 安装 Linux 操作系统

一、CentOS Stream 9 安装案例概述

本案例将以 CentOS Stream 9 为例，介绍 Linux 发行版的安装和使用。CentOS 官网免费提供 CentOS Stream 9 镜像文件，绝大多数用户适合选择包含“x86_64”字样的版本，例如编者下载的镜像文件名称为 CentOS-Stream-9-latest-x86_64-dvd.iso。镜像文件大小超过 8 GB。下载时间与网速有关，请耐心等待。部分用户的硬盘或者 U 盘如果采用 FAT/FAT32 格式的文件系统，可能无法存放这么大的文件，此时通过修改格式为 NTFS 可以解决这一问题。Linux 初学者若遇到此类问题，建议寻找专业人士来获取帮助，以免数据丢失。

对于初学者，直接在计算机上安装 Linux 发行版并不可取。不论是单独安装 Linux 发行版还是安装 Linux 和 Windows 双操作系统，都不建议初学者尝试；否则，操作稍有不当，就可能导致硬盘上的数据丢失。通过虚拟机安装和使用 Linux 发行版是最为常见的解决方案之一。虚拟机是指通过软件模拟的、具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统。典型的虚拟机软件有 VM Workstation 和 VirtualBox。Windows 用户也可以使用微软（Microsoft）公司提供的 Hyper-V。

本案例中，编者将以 VM Workstation 为基础，演示 CentOS Stream 9 的安装过程。在 VirtualBox 和 Hyper-V 中安装 CentOS Stream 9 的过程基本类似，有兴趣的读者可以自行尝试。完成 CentOS Stream 9 的安装后，我们将简要介绍其 GUI。需要注意的是，对于 Linux 用户，使用 GUI 的机会极其有限，因此这部分内容并不是学习的重点，之后的章节中也将不再涉及 GUI 相关内容。此外，我们还将介绍 VM Workstation 提供的快照功能。初学者容易遇到各类问题，快照有助于快速备份和恢复。

课内思考

在安装 CentOS Stream 9 时，为什么分区设置是一个关键步骤？你认为合理的分区设置有哪些考虑因素？

二、CentOS Stream 9 的安装和使用案例详解

1. 安装准备

首先，创建 VM Workstation 虚拟机。关于 VM Workstation 的安装、虚拟机的创建和配置方法，在此不做详细介绍。然后，在 VM Workstation 主界面左侧的列表中，选择刚才创建的虚拟机，并在界面右侧中，单击“编辑虚拟机设置”按钮，打开“虚拟机设置”对话框，执行效果如图 1-2 所示。图 1-2 中显示编者为该虚拟机分配的内存是 4 GB，编者宿主机内存是 8 GB。读者应当在内存允许的情况下尽可能为虚拟机分配更大内存。过多为虚拟机分配内存会降低宿主机性能，读者应当综合考虑。图 1-2 中显示编者为该虚拟机分配的硬盘空间为 20 GB，但实际上 20 GB 并不够用。建议读者在宿主机硬盘空间允许的情况下尽量设置一个更大的空间（至少 40 GB），否则后期可能出现硬盘空间不够的情况，需要自行清理多余文件，而这需要丰富的经验。VM Workstation 默认动态分配空间，即使设置一个很大的硬盘空间值也不会在一开始就真正占用指定大小的空间。



图 1-2 选择安装镜像文件

在“虚拟机设置”对话框的硬件列表中，选择“CD/DVD（IDE）”，然后在右侧的“连接”选项组中选择“使用 ISO 映像文件”，单击“浏览”按钮，选择之前下载的 CentOS Stream 9 安装镜像文件，最后单击“确定”按钮保存设置。至此，虚拟机配置完成。

2. 安装过程

接下来我们开启虚拟机。单击菜单栏上绿色的三角形按钮，虚拟机启动后，屏幕显示 GRUB 菜单，并默认选中第 1 项，执行效果如图 1-3 所示。

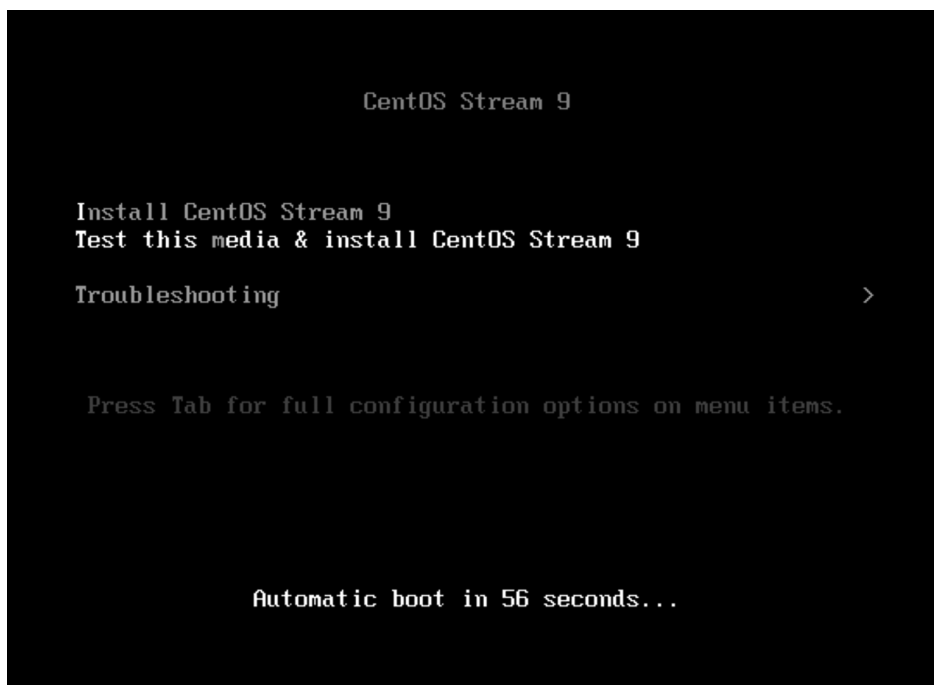


图 1-3 GRUB 菜单

此时，编者选择默认的第 1 项，按“Enter”键后，进入语言选择界面，执行效果如图 1-4 所示。读者如果选择图 1-3 中第 2 项，计算机将直接加载 CentOS Stream 9 系统，并不需要用户进行过多设置。试用时的运行效果与安装后再启动 CentOS Stream 9 的效果基本类似，有兴趣的读者可以自行尝试。

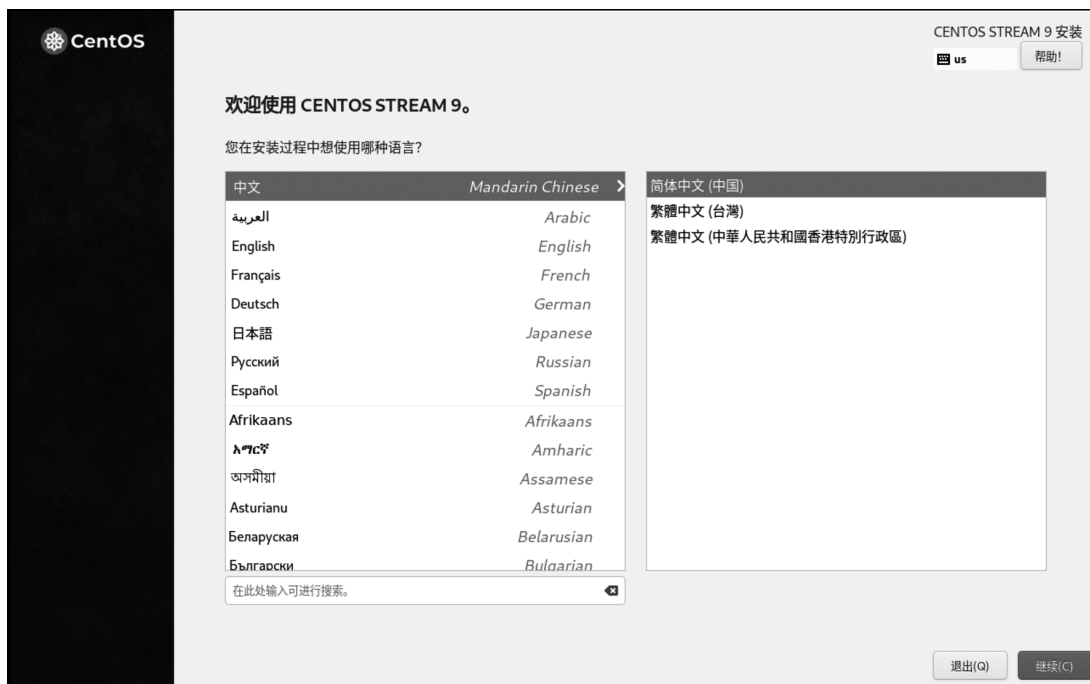


图 1-4 选择语言

在左侧的列表中选择“中文”，然后在右侧的列表中选择“简体中文（中国）”，单击“继续”按钮，系统将显示“安装信息摘要”界面。执行效果如图 1-5 所示。在该界面的最下面，显示一条带感叹号的提示信息，“请先完成带有此图标标记的内容再进行下一步”。



图 1-5 安装信息摘要

首先，在图 1-5 所示界面中，单击“root 密码（R）”，进入“ROOT 密码”界面，并为 root 用户设置密码。编者习惯使用 SSH 登录，因此勾选了“允许 root 用户使用密码进行 SSH 登录”，建议读者跟编者的设置保持一致。执行效果如图 1-6 所示。单击左上角的“完成”按钮可以返回到图 1-5 所示界面。



图 1-6 “ROOT 密码”界面

然后，在图 1-5 所示界面中，单击“安装目的地（D）”，进入“安装目标位置”界面，存储配置默认选择“自动”，初学者不需要进行更改。执行效果如图 1-7 所示。单击左上角的“完成”按钮可以返回到图 1-5 所示界面。

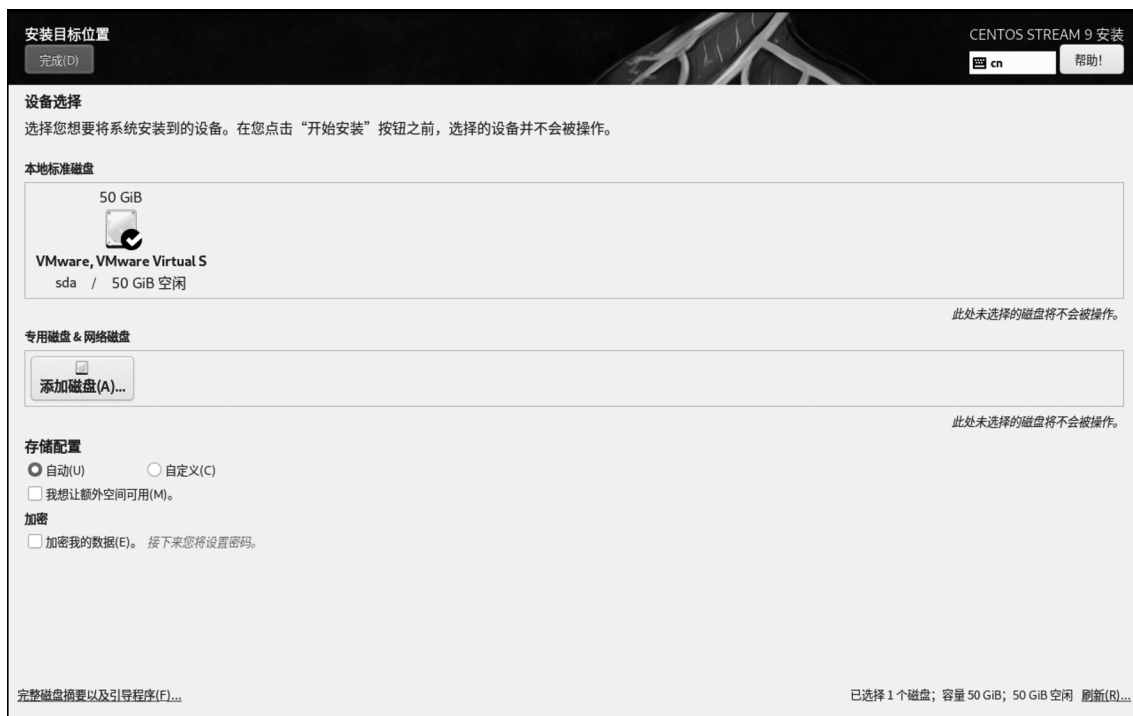


图 1-7 “安装目标位置”界面

完成上述操作后，图 1-5 中的红色提示信息消失，并且界面右下角的“开始安装”按钮变成蓝色。单击“开始安装”按钮，进入“安装进度”界面，执行效果如图 1-8 所示。安装过程耗时较长，大约 20 分钟。安装完成后，图 1-8 界面右下角的“重启系统”按钮将变为蓝色。单击“重

启系统”按钮启动系统。

首次启动系统，将显示“欢迎”界面，执行效果如图 1-9 所示。



图 1-8 “安装进度”界面

单击图 1-9 所示界面下方的“开始配置”按钮，将依次显示“隐私”界面和“在线账号”界面。读者依次单击界面右上方的“前进”按钮和“跳过”按钮，此时将出现“关于您”界面，执行效果如图 1-10 所示。



图 1-9 首次启动系统时显示“欢迎”界面



图 1-10 “关于您”界面

读者在图 1-10 所示界面中的“全名”后面的文本框中输入内容，下方“用户名”后面的文本框中将自动填充同样的内容。本书后续实例中经常要用到用户名“zp”，并且许多路径和变量都与用户名“zp”直接相关。为避免犯错，建议读者与编者保持一致，也使用“zp”作为用户名。如果读者坚持要使用其他用户名，建议尽量设置一个简单的用户名，并且确保在英文半角状态下输入。设置完用户名后，单击界面右上角的“前进”按钮，进入“设置密码”界面，如图 1-11 所示。



图 1-11 “设置密码”界面

设置密码后，单击界面右上角的“前进”按钮，进入“配置完成”界面，执行效果如图 1-12 所示。



图 1-12 “配置完成”界面

单击“开始使用 CentOS Stream”按钮，显示“开始导览”界面，执行效果如图 1-13 所示。



图 1-13 “开始导览”界面

3. 界面简介

读者可以直接单击“开始导览”界面左上角的“关闭”按钮，完成系统安装过程，再正常启动系统。执行效果如图 1-14 所示。

单击图 1-14 所示界面左上角的“活动”按钮，可以查找需要启动的应用程序。执行效果如图 1-15 所示。

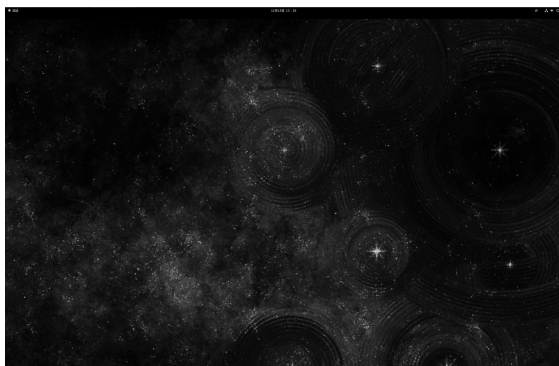


图 1-14 系统正常启动完成界面

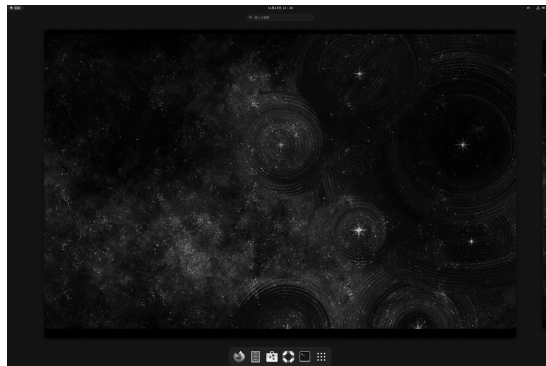


图 1-15 “活动”界面

读者也可以在界面正上方的搜索框中输入内容，查找所需要的应用程序。界面下方显示了常用的应用程序列表。读者将鼠标指针停留在各个按钮上，可以查看相应的提示信息。单击列表右侧的九宫格按钮，可以显示更多内容。执行效果如图 1-16 所示。系统基本设置、网络设置、开关机等相关功能入口主要位于界面的右上角。界面右上角依次是输入法、网络设置、声音、开关机等相关功能图标。单击右上角的图标（以声音图标为例），可以显示更详细的选项。执行效果如图 1-17 所示。

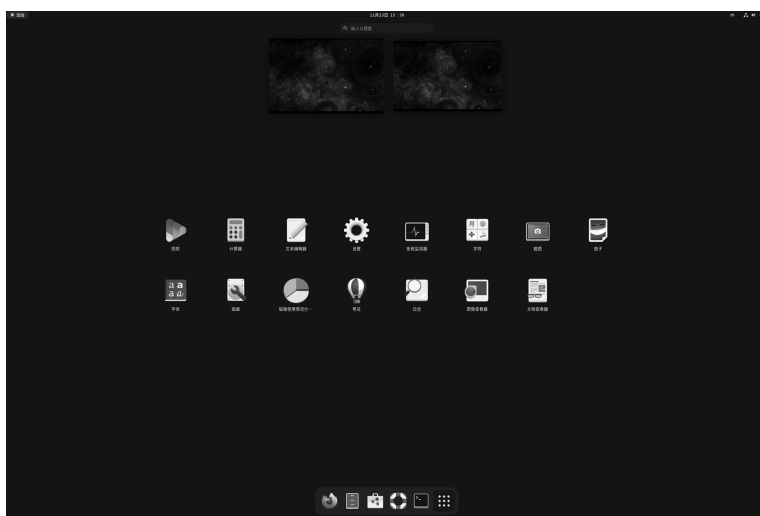


图 1-16 显示更多内容

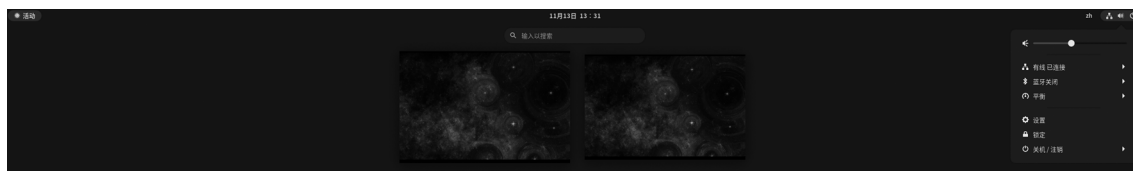


图 1-17 右上角的菜单内容

4. 创建快照

安装完成后，建议读者做好备份。VM Workstation 提供了一种称为“快照”的轻量级备份工具。读者可以依次单击“虚拟机 (M)”→“快照 (N)”→“拍摄快照 (T)...”使用该工具，如图 1-18 所示。

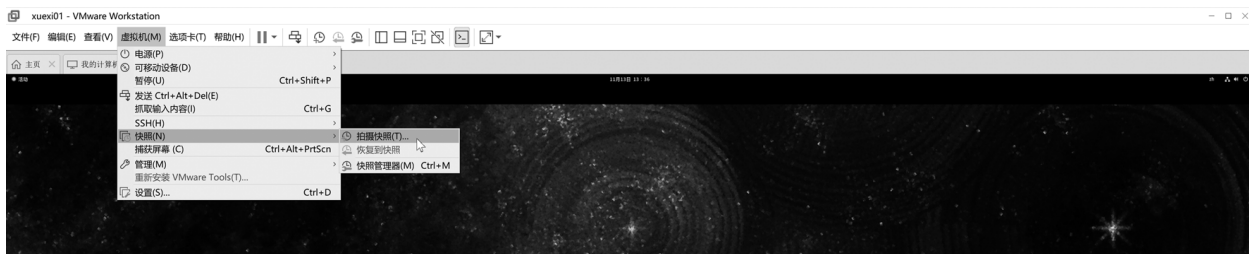


图 1-18 拍摄快照

执行效果如图 1-19 所示。读者可以对快照进行命名，并添加描述信息，以方便管理快照。

读者可以不定期创建和维护快照。后续如因操作不当损坏系统，可以使用特定的快照进行快速恢复。



图 1-19 “拍摄快照”对话框

任务三 VMware 虚拟机的备份

一、手动复制虚拟机磁盘文件

使用 VMware Workstation 复制文件就是一个可行的方案，我们可以通过复制存放虚拟机磁盘文件和配置文件的文件夹的方式来达成备份目的。

步骤 1：打开 VMware Workstation，在主界面左侧的虚拟机列表中找到自己想要备份的虚拟机，鼠标选中它并将鼠标停留在该虚拟机名称上，会发现一个悬浮窗，里面显示了此虚拟机的文件位置，如图 1-20 所示。

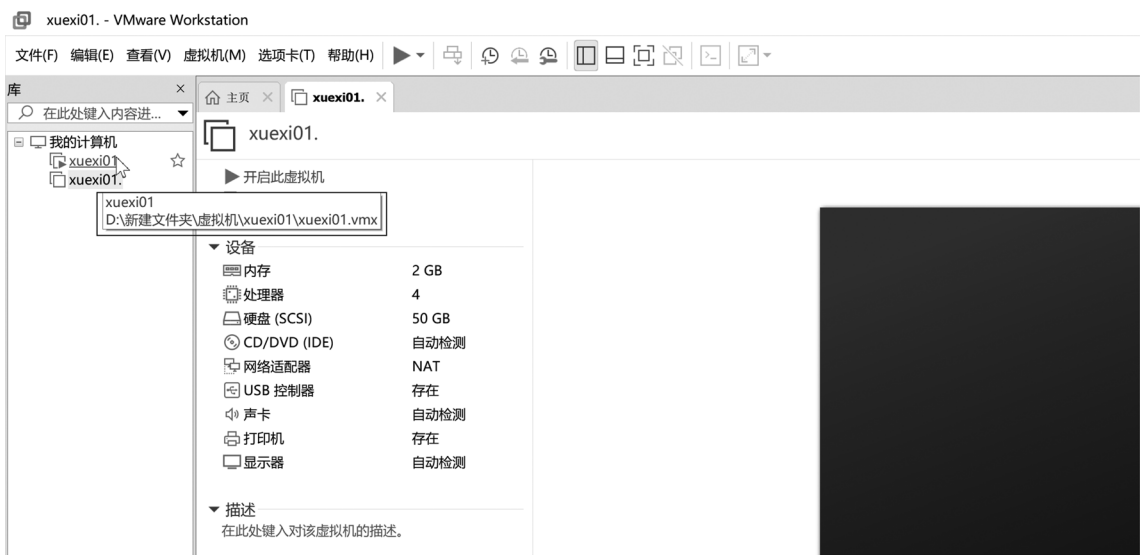


图 1-20 找到备份的虚拟机

步骤 2：找到该文件夹位置，将整个文件夹复制到指定的位置即可完成虚拟机备份任务，如图 1-21 所示。



图 1-21 找到文件夹位置

二、将虚拟机导出为 OVF 模板

导出为 OVF 模板也是一个比较常用的 VMware Workstation 备份虚拟机方法。

步骤 1：打开 VMware Workstation，在左侧列表中选中自己想要备份的虚拟机，然后单击左上角的“文件 (F)” → “导出为 OVF (E)...”，如图 1-22 所示。



图 1-22 打开 VMware Workstation

步骤 2：在弹出窗口中选择一个目标位置，然后单击“保存”，如图 1-23 所示。

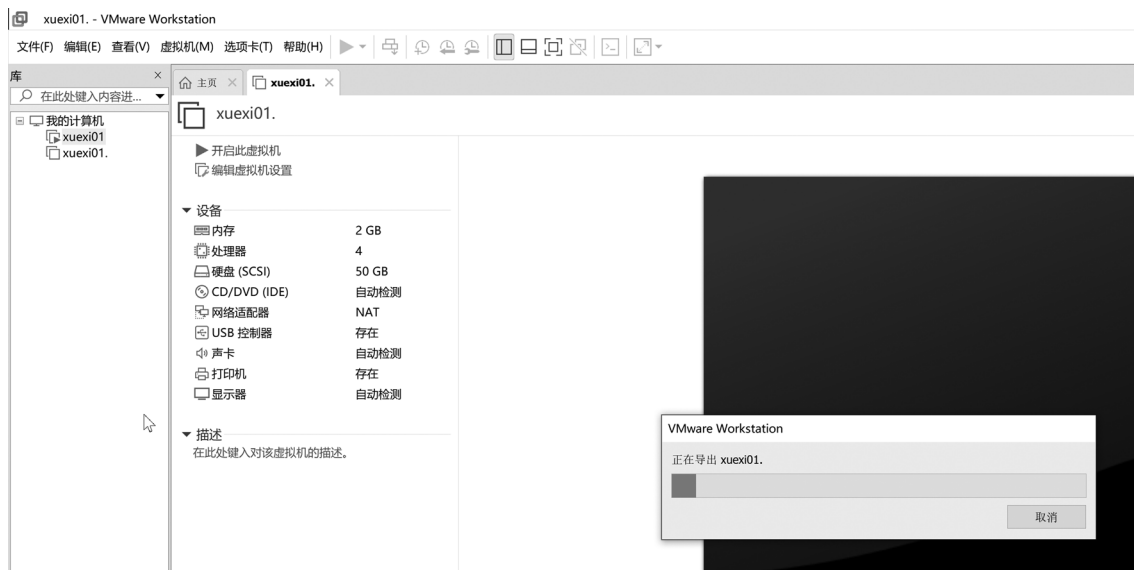


图 1-23 单击“保存”

步骤 3：等待 VMware Workstation 导出虚拟机完成之后，在目标位置将找到 3 个文件，分别是：清单文件 .mf，虚拟机完整规范文件 .ovf，虚拟机磁盘文件 .vmdk。

三、克隆虚拟机

克隆其实也是属于备份的一种类型，在 VMware Workstation 也是如此。

步骤 1：打开 VMware Workstation，在左侧列表中选中自己想要备份的虚拟机，点击“虚拟机 (M)” → “管理 (M)” → “克隆 (C)...” 然后在弹出窗口中点击“下一步”，如图 1-24 所示。



图 1-24 打开 VMware Workstation

步骤 2: 指定一个克隆源, 可以选择“虚拟机中的当前状态 (C)”或者“现有快照 (仅限关闭的虚拟机) (S)”, 然后单击“下一步”, 如图 1-25 所示。

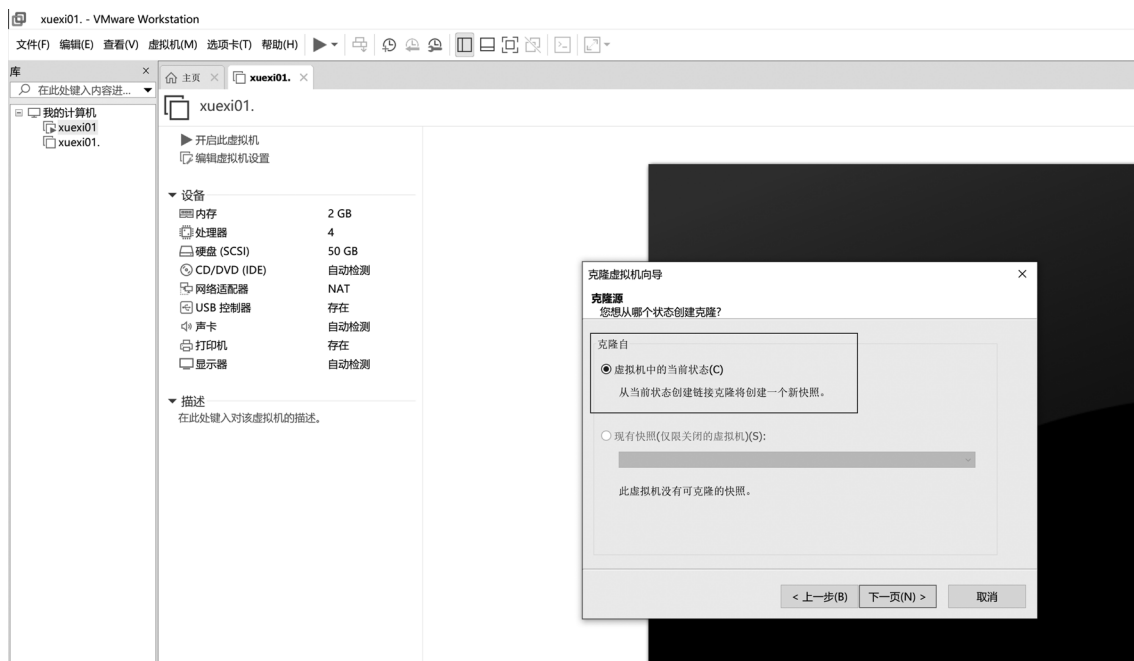


图 1-25 指定一个克隆源

步骤 3: 指定克隆类型, 可以选择“创建链接克隆 (L)”或者“创建完整克隆 (F)”, 然后单击“下一步”, 如图 1-26 所示。

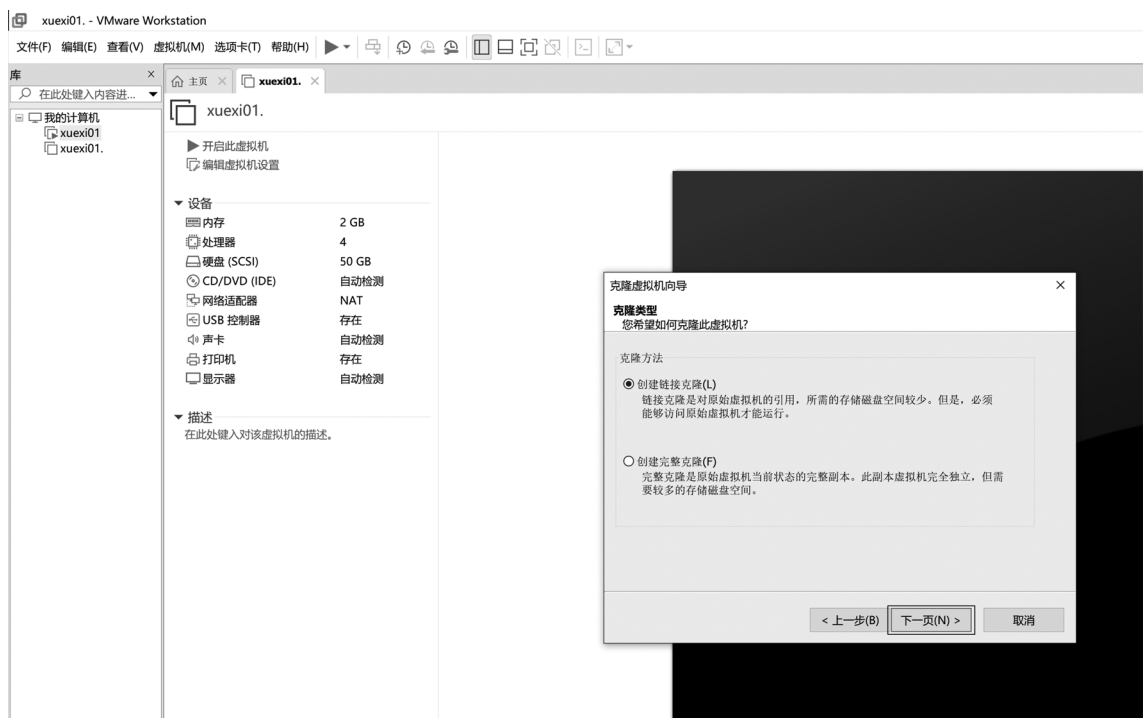


图 1-26 指定克隆类型

步骤 4：填写克隆任务的名称和存储路径，然后点击“完成”即可开始执行克隆任务。

课内思考

在备份虚拟机时，手动复制和克隆分别适用于什么样的场景？有何优缺点？

四、像物理机一样备份虚拟机

通过在 VMware Workstation 虚拟机的来宾操作系统（OS）上安装备份代理，可以将其作为物理计算机进行备份。这意味着用户可以从备份中排除不重要的应用程序和数据，以减少大小，并在用户只需要一些特定文件时执行选择性文件还原任务。

通过这种方式，还可以将一台虚拟机或物理机上的全部数据迁移到另一台，例如，VMware 转 Hyper-V 或 Hyper-V 转 VMware。

项目实训

教师评语	教师签字	日期	成绩	
学生姓名		学号		
实训名称	Linux 发行版安装与基本界面环境熟悉			
实训准备	(1) 提前准备好具有虚拟化支持的物理机或虚拟机环境 (2) 下载并准备选定 Linux 发行版的安装镜像文件			

实训目标	学生将通过实际操作，掌握在物理机上安装 Linux 发行版（CentOS Stream、RHEL 或其他最新版），并熟悉基本的界面环境
实训步骤	
步骤一：Linux 发行版的安装 1. 介绍： 简要介绍 Linux 发行版的种类，特别是 CentOS Stream、RHEL 或其他选择的发行版 2. 安装环境准备： 指导学生检查硬件要求，确保物理机或虚拟机满足安装条件 3. 安装过程： 指导学生通过启动设备，选择安装选项，并按照提示完成发行版的安装过程 4. 分区设置： 解释分区设置的基本概念，然后引导学生选择合适的分区方案 5. 网络配置： 指导学生在安装过程中配置网络，确保系统能够联网 6. 用户账户： 指导学生创建至少一个用户账户，讲解账户管理的重要性 步骤二：基本界面环境熟悉 1. 登录系统： 指导学生登录新安装的 Linux 系统，介绍登录凭证的重要性 2. 图形界面和命令行界面： 演示 Linux 系统的图形用户界面（GUI）和命令行界面（CLI），让学生理解两者之间的关系 3. 基本命令： 教授学生一些基本的 Linux 命令，如“ls”“cd”“mkdir”等，以便他们能够在命令行环境中导航和操作 4. 软件包管理： 介绍使用包管理器安装、更新和删除软件包的方法 5. 系统配置文件： 解释 Linux 系统的基本配置文件，指导学生编辑这些文件以修改系统行为。	
实训总结	让学生总结他们在实训中遇到的挑战和解决方案，鼓励提问并进行讨论。强调实际操作的重要性，并鼓励学生在日常学习中继续深入了解和使用 Linux 系统



项目二 Linux 操作系统常用命令

项目概述

随着信息技术的发展，Linux 操作系统在服务器端和嵌入式系统等领域的应用日益广泛。为提高 IT 从业人员对 Linux 系统的熟练度，本项目旨在通过实操与命令行技能培训，帮助学生深入了解 Linux 操作系统，并掌握关键的命令行技能。

学习目标

知识目标

- ①了解 Linux 命令行的基本概念，了解 Linux 命令行的基本操作和高级技巧。
- ②掌握文件和目录操作命令的基本功能和使用方法，理解文件权限和所有者的概念。
- ③理解系统信息类命令的作用，能够获取关键系统信息。
- ④了解进程管理类命令，包括进程状态监测、进程状态控制、进程启动与作业控制。
- ⑤熟悉其他常用命令，如“echo”“date”“chmod”等。

能力目标

- ①能够灵活运用 Linux 命令行进行系统管理和日常任务。
- ②熟练使用文件和目录操作命令进行文件和目录的管理，包括创建、编辑、复制、移动和删除。
- ③能够通过系统信息类命令获取关键的系统性能和资源使用信息。
- ④具备使用进程管理类命令监测和控制系统的进程的能力。
- ⑤熟练使用其他常用命令，提高在命令行环境下的工作效率。

思政目标

- ①培养学生对 Linux 命令行的实际操作兴趣，激发主动学习和探索新命令的动力。
- ②提高学生对系统细节的关注，培养细致入微的学习与工作态度。
- ③鼓励学生在实际操作中注重安全性，培养对系统权限和所有权的重视。
- ④强调团队协作，通过分享实际工作中的经验，促进团队内的知识传递和协作能力的提升。

任务一 Linux 命令行基础

尽管面向桌面应用场景的 Linux 发行版都提供了 GUI，但命令行交互依然是进行 Linux 操作系统配置和管理的首选模式。在大多数真实的应用场景中，Linux 操作系统通常位于远程服务器、嵌入式设备之中，并没有为管理人员或者开发人员提供 GUI。熟练掌握一定的命令行知识是使用 Linux 操作系统的基本要求。本章将介绍 Linux 命令行基础知识，以帮助读者掌握 Linux 命令行的基本用法。

知识之窗

“卡脖子”技术

《科技日报》曾推出系列文章，报道了制约我国工业发展的 35 项“卡脖子”技术，主要包括光刻机、芯片、操作系统、航空发动机短舱、触觉传感器、真空蒸镀机、手机射频器件、重型燃气轮机、激光雷达、适航标准、核心工业软件、核心算法、航空钢材、铣刀、高端轴承钢、高压柱塞泵、航空设计软件、光刻胶、高压共轨系统、透射式电镜、掘进机主轴承、微球、水下连接器、燃料电池关键材料、高端焊接电源、锂电池隔膜、医学影像设备元器件、超精密抛光工艺、环氧树脂、高强度不锈钢、数据库管理系统、扫描电镜、ICLIP 技术、ITO 靶材、高端电容电阻等。

一、Linux 命令行概述

说到 Linux 命令行，通常会提到 Shell 和 Bash 等相关概念。

(一) 什么是 Shell

Shell 接收用户命令，并协助用户完成与系统内核的交互，以完成命令的执行。我们可以从以下 3 个方面来理解 Linux 中 Shell 的功能和含义。



什么是 Shell

(1) Shell 是 Linux 操作系统的用户界面。Shell 提供了用户与系统内核进行交互操作的接口。与目前广泛使用的 GUI 交互方式不同，Shell 提供的是一种命令行交互方式。Shell 本身不是 Linux 内核的一部分，但是它调用了系统内核的大部分功能来执行程序。Shell 支持个性化的用户环境配置，这种配置通常由 Shell 初始化配置文件实现。

(2) Shell 是一个命令解释程序，它能解释用户在命令行界面输入的命令。Shell 拥有自己内建的命令集，它能执行的命令包括内部命令和外部命令。典型的 Shell 解释程序包括 Bourne Shell、C Shell、Korn Shell、POSIX Shell 及 Bourne Again Shell (Bash) 等。大多数 Linux 发行版默认使用 Bash 作为 Shell 解释程序。Bash 基于 Bourne Shell，它吸收了 C Shell 和 Korn Shell 的一些特性。Bash 提供了几百个内部命令，尽管这些命令的功能不同，但它们的使用方式和规则是统一的。

(3) Shell 是一种程序设计语言。使用 Shell 语言编写的程序文件称为 Shell 脚本 (Shell Script)。Shell 作为一种程序设计语言，有自己完整的语法规则，支持分支结构、循环结构和函数定义等。Shell 脚本中还可以直接调用常用的 Linux 命令，通过编写 Shell 脚本，可以实现更为复杂的管理功能。

(二) Linux 命令行界面

命令行界面是操作 Linux 最常用的人机交互界面。用户既可以通过终端仿真器 (terminal emulator) 进入命令行界面，也可以将计算机系统配置成启动后默认进入命令行界面，还可以直接使用远程登

录的方式进入命令行界面。不同类型 Linux 发行版的命令行界面会略有差别。通过不同方式进入命令行界面后，其界面样式也存在细微差异。

大多数 Linux 发行版中都配置了终端仿真器。这是一种 GUI 环境下的终端窗口（terminal window）应用程序，方便用户使用命令行方式与 Linux 内核进行交互。用户启动终端仿真器时，系统将自动启动一个默认的 Shell 解释程序（通常是 Bash），以解释用户在终端窗口中输入的命令。用户可以看到 Shell 的提示符，在提示符后输入一串字符，Shell 解释程序将对这一串字符进行解释。我们可以使用以下多种方式打开终端仿真器：

(1) 单击桌面左上角的“活动”按钮，在进入的界面中浏览/搜索“终端”（或“gnome-terminal”）。

(2) 在文件系统中查找 `gnome-terminal` 可执行文件，路径通常是 `/usr/bin/gnome-terminal`。

(3) 用户也可以设置键盘快捷键。其中命令设置为 `/usr/bin/gnome-terminal`，快捷键建议设置为 `Ctrl+Alt+T`，因为在 Ubuntu 等操作系统中默认预置该快捷键。

【实例 2-1】 查看默认 Shell 解释程序和所有有效的 Shell 解释程序。

在命令行界面中输入 `echo $SHELL`，可以查看当前使用的 Shell 解释程序。绝大多数 Linux 发行版默认 Shell 解释程序都是 Bash。文件 `/etc/shells` 中保存了当前系统中有效的 Shell 程序列表，读者可以通过 `cat` 命令查看该文件内容。执行如下命令：

```
[zp@localhost ~]$ echo $SHELL
[zp@localhost ~]$ cat /etc/shells
```

执行效果如图 2-1 所示。

```
[zp@localhost ~]$ echo $SHELL
/bin/bash
[zp@localhost ~]$ cat /etc/shells
/bin/sh
/bin/bash
```

图 2-1 当前系统的 Shell 程序列表

（三）Linux 命令提示符

打开 Linux 命令行界面后，通常会在界面的最后一行显示标准的 Linux 命令提示符。完整的 Linux 命令提示符包括当前用户名、登录的主机名、当前所在的工作目录和提示符。Linux 命令提示符的基本格式如下：

[当前用户名@ 主机名 当前目录] 提示符

【实例 2-2】 查看 Linux 操作系统命令提示符。

图 2-2 展示了 CentOS Stream 操作系统命令提示符的典型样式及其变化情况。

```
[zp@localhost ~]$ cd /home/
[zp@localhost home]$
```

图 2-2 Linux 操作系统命令提示符（普通用户）

图 2-2 的第 1 行显示的命令提示符格式为 `[zp@localhost ~]$`。其内容表明：当前用户名为 `zp`，主机名是 `localhost`，“`~`”代表当前登录用户的主目录。需要注意的是，不同用户的主目录通常并不相同。本实例中，“`~`”表示当前工作目录是用户 `zp` 的主目录 `SP/home/zp`。常见的提示符有两个：`$` 和 `#`。提示符 `$` 表示当前登录用户为普通用户；`#` 表示当前用户为 `root` 用户。以 `root` 用户身份登录系统时，完整的命令提示符样式如图 2-3 所示。`root` 用户具有最高的权限，但使用 `root` 用户

身份操作存在较大的风险，一旦操作不当容易导致极具破坏性的结果，因此不建议用户频繁以 root 用户身份登录。不同发行版的 Linux 命令提示符并不完全相同。Linux 命令提示符每一部分的样式都可以根据需要进行修改，有兴趣的读者可以自行查阅资料尝试修改。

```
[root@lab ~]# cd..  
[root@lab /]#
```

图 2-3 Linux 操作系统命令提示符 (root 用户)

用户在命令提示符 \$ 或者 # 之后输入 Linux 命令，然后按 “Enter” 键执行该命令。例如，在图 2-2 的第 1 行中，我们输入了命令 “cd/home/”，命令 cd 用于改变 Shell 工作目录，输入的命令表示将工作目录切换成 “home”。用户进行目录切换等操作后，当前工作目录会发生变化，原来显示 “~” 的位置，其内容也随之变成 “home”。与此类似，在图 2-3 中的第 1 行，我们输入了命令 “cd..”，其中 “..” 代表当前目录的父目录，本实例中该父目录为根目录 “/”。

二、Linux 命令行基本操作

(一) Linux 命令语法格式

Linux 命令语法格式如下：

命令 [选项] [参数]

用户在 Linux 命令提示符之后输入命令、选项和参数，按 “Enter” 键即可执行 Linux 命令。Linux 命令行严格区分字母大小写，命令、选项和参数都是如此。本书后续章节在不产生歧义的情况下，通常也会将选项和参数统称为参数。

命令的语法格式说明中，通常使用 “[]” 来标记可选项，选项和参数都是可选项。选项常用于调整命令功能，通过添加不同的选项，可以改变命令执行动作的类型。选项有短选项和长选项两种，短选项之前通常使用连字符 “-”，长选项之前通常使用连字符 “--”。短选项更为简洁，本书后续章节一般采用短选项。Linux 命令中，参数通常是命令的操作对象，多数命令都可使用参数。一般而言，文件、目录、用户和进程等都可以作为参数被命令操作。【实例 2-1】和【实例 2-2】中涉及的几条命令都只包含参数，而没有包含选项。这些参数中，既有变量名，也有文件和文件的路径。

大多数命令都提供了数量众多的选项。用户可以使用帮助功能获取更为详细的命令帮助信息，因此不用记住所有命令的所有选项。各条命令的常用选项或者选项组合并不多。许多命令执行时，甚至可以不使用选项或参数，但此时命令只能执行最基本的功能。本书给出的实例中涉及的选项大多是常用的选项。读者可以根据实例熟悉 Linux 命令常见的选项功能。

【实例 2-3】 执行不包含选项和参数的命令。

本实例中，我们以 ls 命令为例，解释 Linux 命令的基本格式。ls 命令用于列出目录中的内容。执行如下命令。

```
[zp@localhost ~]$ ls
```

执行效果如图 2-4 所示。本实例中，ls 命令后面不包括任何选项或参数。此时列出的是当前目录（本实例中为用户 zp 的主目录）下的内容。

```
[zp@localhost ~]$ ls  
公共 模板 视频 图片 文档 下载 音乐 桌面 anaconda3
```

图 2-4 执行不包含选项和参数的命令

【实例 2-4】 执行包含选项的命令。

执行如下命令：

```
[zp@localhost ~]$ ls -l
```

与【实例 2-3】相比，本实例增加了一个“-l”选项，此时返回的结果信息更加详细。执行效果如图 2-5 所示。限于篇幅，这里只给出部分截图。

```
[zp@localhost ~]$ ls -l
总用量 674676
drwxr-xr-x. 2 zp zp      6  3月 26 18:06 公共
drwxr-xr-x. 2 zp zp      6  3月 26 18:06 模板
drwxr-xr-x. 2 zp zp      6  3月 26 18:06 视频
drwxr-xr-x. 2 zp zp      6  3月 26 18:06 图片
drwxr-xr-x. 2 zp zp      6  3月 26 18:06 文档
```

图 2-5 执行包含选项的命令

【实例 2-5】 执行包含参数的命令。

执行如下命令：

```
[zp@localhost ~]$ ls /dev/
```

本实例中，目录“/dev/”被用作 ls 命令的参数。本实例的“/dev/”也可以写成“/dev”。事实上，最后面的那个“/”是使用自动补全功能时自动生成的。关于自动补全，后面会详细介绍。以传入目录名称作为参数，ls 将显示指定目录下的内容。执行效果如图 2-6 所示。

```
[zp@localhost ~]$ ls /dev/
autofs      mapper      tty13      tty44      usbmon2
block       mcelog     tty14      tty45      vcs
bsg         mem         tty15      tty46      vcs1
bus         midi        tty16      tty47      vcs2
cdrom       mqueue     tty17      tty48      vcs3
```

图 2-6 执行包含参数的命令

【实例 2-6】 执行同时包含参数和选项的命令。

执行如下命令：

```
[zp@localhost ~]$ ls /dev/-l
```

执行效果如图 2-7 所示。本实例同时增加了“-l”选项和“/dev/”参数，两者的位置可以交换。交换后的效果如下所示。

```
[zp@localhost ~]$ ls -l /dev/
```

需要注意的是，并不是所有的选项和参数都可以交换位置。如果某个参数是某个选项对应的参数，那么该参数通常只能放在该选项之后。

```
[zp@localhost ~]$ ls /dev/ -l
总用量 0
crw-r--r--. 1 root root    10, 235 6月 4 22:24 autofs
drwxr-xr-x. 2 root root    160 6月 4 22:25 block
drwxr-xr-x. 2 root root     80 6月 4 22:24 bsg
drwxr-xr-x. 3 root root     60 6月 4 22:24 bus
lrwxrwxrwx. 1 root root      3 6月 4 22:24 cdrom -> sr0
```

图 2-7 执行同时包含参数和选项的命令

【实例 2-7】 积累更多的 Linux 命令。

本实例分别执行了 `pwd`、`hostname`、`uname` 三条新命令，针对 `uname` 命令还演示了携带不同选项时的执行效果，如图 2-8 所示。`pwd` 命令用于输出当前的工作目录名称。`hostname` 命令用于显示或者设置系统主机名。`uname` 命令用于输出系统信息。

```
[zp@localhost ~]$ pwd
/home/zp
[zp@localhost ~]$ hostname
localhost.localdomain
[zp@localhost ~]$ uname
Linux
[zp@localhost ~]$ uname -r
5.14.0-96.el9.x86_64
[zp@localhost ~]$ uname -n
localhost.localdomain
```

图 2-8 使用命令行界面执行 Linux 命令

(二) 命令自动补全

Linux 操作系统的 Bash 相当智能化，支持命令和文件名自动补全功能。用户可以通过“Tab”键使用自动补全功能，将输入的部分命令或者文件名快速补充完整。例如，输入命令或者文件名的时候，通常只需要输入该命令或者文件名的前几个字符，然后按“Tab”键，Shell 就可以自动将其补全。当匹配结果只有一个时，按“Tab”键可以自动补全命令。对于存在多个可能匹配结果的情况，连续按两次“Tab”键可查看以指定字符开头的所有相关匹配结果。

【实例 2-8】 自动补全功能。

以按一次“Tab”键自动补全命令或者文件名为例，首先，在命令行界面中输入“`ls/e`”，然后按“Tab”键，系统会自动将该命令补全成“`ls/etc/`”，接下来按“Enter”键，可以执行该命令。

```
[zp@localhost ~]$ ls /e 【Tab】
```

执行效果如图 2-9 所示。

```
[zp@localhost ~]$ ls /etc/
accountsservice      machine-info
adjtime              magic
aliases              mailcap
alsa                 makedumpfile.conf.sample
```

图 2-9 自动补全命令

用户还可以进行更复杂的尝试。执行如下命令：首先，在命令行界面中输入“`cat/e`”，接着按“Tab”键，系统会自动将该命令补全成“`cat/etc/`”。但是这次，我们不再直接按“Enter”键，而是继续在后面输入“`ad`”，接着按“Tab”键，系统会自动将该命令补全成“`cat/etc/adjtime`”，此时按“Enter”键显示的将是文件“`/etc/adjtime`”的内容。

```
[zp@localhost ~]$ cat /e 【Tab】 ad 【Tab】
```

执行效果如图 2-10 所示。

```
[zp@localhost ~]$ cat /etc/adjtime
0.0 0 0.0
UTC
```

图 2-10 自动补全命令和参数

【实例 2-9】 自动列出候选项。

以按两次“Tab”键自动列出候选项为例，依次执行如下命令：

```
[zp@localhost ~]$ ls/b 【Tab】 【Tab】
```

```
[zp@localhost ~]$ ls /bo 【Tab】
```

首先，在命令行界面中输入“ls/b”，然后按“Tab”键，系统并没有自动补全命令，而是发出提示音。此时再按“Tab”键，系统将反馈“bin/”和“boot/”两个候选项供选择。我们继续输入一个字符“o”，然后继续按“Tab”键，系统自动将该命令补全成“ls/boot/”。最后按“Enter”键，可以执行该命令。执行效果如图 2-11 所示。

```
[zp@localhost ~]$ ls /b
bin/ boot/
[zp@localhost ~]$ ls /boot/
config-5.14.0-71.el9.x86_64
config-5.14.0-96.el9.x86_64
efi
```

图 2-11 自动列出候选项

（三）强制中断命令执行

部分命令执行时间较长，例如，许多与网络相关的命令由于网络状况不佳，可能会出现长时间等待被执行的情况。如果想提前终止该命令，此时可以使用“Ctrl+C”组合键强制中断命令执行。

【实例 2-10】 强制中断命令执行。

执行如下命令：

```
[zp@localhost ~]$ ping 127.0.0.1
```

ping 命令是一种比较基础的网络诊断命令。它通过向特定的目的主机发送因特网报文控制协议（internet control message protocol, ICMP）Echo 请求报文，测试目的主机是否可达以了解其有关状态。“127.0.0.1”是本机的环回地址，读者也可以将该地址换成其他有效的域名地址。该命令在默认情况下，将继续执行。用户可以随时通过“Ctrl+C”组合键结束命令的执行过程，执行效果如图 2-12 所示。

```
[zp@localhost ~]$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) 比特的数据:
64 比特, 来自 127.0.0.1: icmp_seq=1 ttl=64 时间=0.050 毫秒
64 比特, 来自 127.0.0.1: icmp_seq=2 ttl=64 时间=0.044 毫秒
64 比特, 来自 127.0.0.1: icmp_seq=3 ttl=64 时间=0.059 毫秒
^C
---127.0.0.1 ping 统计 ---
已发送 3 个包, 已接收 3 个包, 0% packet loss, time 2073ms
rtt min/avg/max/mdev = 0.044/0.051/0.059/0.006 ms
```

图 2-12 强制中断命令执行

（四）使用 root 权限

Linux 操作系统部分命令的执行需要 root 权限。如果确实需要 root 权限执行某些操作，此时可以使用 su 或 sudo 命令。建议读者使用 sudo 命令，临时获取 root 权限，以便执行一条需要 root 权限的命令。sudo 命令的常用格式如下：

```
sudo cmd_name [其他可选的参数或选项]
```

【实例 2-11】 使用 root 权限执行命令。

本实例中，我们试图使用 cat 命令查看 /etc/shadow 文件内容。cat 命令可以将指定文件内容写到标准输出，也就是可以查看文件内容。文件 /etc/shadow 保存了与密码相关的信息，属于对安全性要求较高的内容，访问者需要有 root 权限才能执行“查看”命令。本实例涉及两条命令：前一条命令直接使用 zp 用户身份查看文件内容，结果提示权限不够；后一条命令前增加了 sudo 命令，用于临时获取 root 权限，结果命令得以成功执行。

执行如下命令。

```
[zp@localhost ~]$ cat /etc/shadow
[zp@localhost ~]$ sudo cat /etc/shadow
```

执行效果如图 2-13 所示。

```
[zp@localhost ~]$ cat /etc/shadow
cat: /etc/shadow: 权限不够
[zp@localhost ~]$ sudo cat /etc/shadow
[sudo] zp 的密码:
root:$6$NU.LDVJsPHTK2r.z$ndokwPMFBj7GRjut1QJFrUaol8QegRIkz95tpRxEEc
```

图 2-13 使用 root 权限执行命令

【实例 2-12】 切换到 root 用户身份。

在命令行里执行 su 命令可以切换到 root 用户身份。执行 su 命令后会提示输入密码。因为 root 用户拥有最高的系统控制权，稍有不慎就可能完全破坏 Linux 操作系统，所以在实际使用中，不建议直接使用 root 用户身份。如果已经切换成功，建议使用 exit 命令退出。部分操作系统（例如 Ubuntu）为了安全起见，甚至默认不启用 root 用户身份，此时执行本实例命令将不会成功。执行效果如图 2-14 所示。

```
[zp@localhost ~]$ su
密码:
[root@localhost zp]# exit
exit
```

图 2-14 切换到 root 用户身份

（五）Linux 命令行帮助信息

本小节介绍三种查看命令帮助信息的方法。由于这三种方法的信息来源不同，显示的结果并不完全一致，但任何一种基本上都能满足需要。编者安装时选择的是中文版的操作系统，最后方法显示的提示信息通常也是中文的，因此该方法适合英语基础一般的用户。

1. 使用 man 命令获取帮助信息

man 命令用于查看 Linux 操作系统的手册，是 Linux 中使用最为广泛的帮助方法。一般情况下，手册资源主要位于 /usr/share/man 目录下。man 命令的基本格式如下：

man 命令名称

在命令提示符后输入“man 命令名称”可以显示该命令的帮助信息。man 命令格式化并显示在线的手册页，其内容包括命令语法、各选项的含义以及相关命令等。

【实例 2-13】 使用 man 命令获取帮助信息。

在命令行中输入“man sudo”可以获取 sudo 的帮助信息。执行如下命令：

```
[zp@localhost ~]$ man sudo
```

执行效果如图 2-15 所示。在该界面中，使用键盘上、下方向键可以滚动屏幕，以查看更多内容。输入“q”，可以退出该帮助信息界面，返回到命令行界面。

```
SUDO(8)  BSD System Manager's Manual  SUDO(8)
NAME
  sudo, sudoedit-execute a command as another user

SYNOPSIS
  sudo -h | -k | -k | -v
  sudo -v [-ABknS] [-g group] [-h host] [-p prompt] [-u user]
  sudo -l [-ABknS] [-g group] [-h host] [-p prompt] [-u user]
  sudo [-ABbEHnPS] [-C num] [-D directory] [-g group]
  sudo [-h host] [-p prompt] [-R directory] [-r role]
Manual page sudo(8) line 1 (press h for help or q to quit)
```

图 2-15 使用 man 命令获取帮助信息

2. 使用 info 命令获取帮助信息

info 文档是 Linux 操作系统提供的另一种格式的文档。info 命令的基本格式如下：

info 命令名称

【实例 2-14】使用 info 命令获取帮助信息。

在命令行提示符后输入“info uname”可以获取 uname 的帮助信息。执行如下命令：

```
[zp@ localhost ~]$ info uname
```

执行效果如图 2-16 所示。在界面中输入“q”（没有回显）后，可以退回到命令行界面。

```
Next: hostname invocation, Prev: nproc invocation, Up: System co\
ntext
21.4 'uname': Print system information
=====
'uname' prints information about the machine and operating system \
it is
run on. If no options are given, 'uname' acts as if the '-s' opti\
on
were given. Synopsis:
---Info: (coreutils)uname invocation, 97 lines --Top-----
welcome to Info version 6.7. Type H for help, h for tutorial.
```

图 2-16 使用 info 命令获取帮助信息

3. 使用 --help 选项获取帮助信息

使用 --help 选项可以显示命令的使用方法以及命令选项的含义。只要在需要显示帮助信息的命令后面输入 --help 选项，就可以看到所查命令的帮助信息了。使用 --help 选项的基本格式如下：

命令名称 --help

与前两种方法不同，使用 --help 选项获取到的帮助信息直接在用户输入的命令的下一行开始显示，并且鼠标光标将停留在新的命令提示符之后。在该界面中，使用鼠标中键（滚轮键）可以上下滚动屏幕，查看更多内容。在中文环境中，使用 --help 选项通常可以得到中文内容的帮助信息，但其内容相对也更为简单。

【实例 2-15】使用 -help 选项获取帮助信息。

执行如下命令：

```
[zp@ localhost ~]$ ls --help
```

执行效果如图 2-17 所示。

```
[zp@localhost ~]$ ls --help
用法: ls [选项]... [文件]...
列出给定文件 (默认为当前目录) 的信息。
如果不指定 -cftuvSUX 中任意一个或 --sort 选项, 则根据字母大小排序。
必选参数对长短选项同时适用。
-a, --all                不隐藏任何以 . 开始的项目
-A, --almost-all        列出除 . 及 .. 以外的任何项目
--author                 与 -l 同时使用时, 列出每个文件的作者
-b, --escape             以 C 风格的转义序列表示不可打印的字符
--block-size=大小        与 -l 同时使用时, 将文件大小以此处给定的大小为
```

图 2-17 使用--help 选项获取帮助信息

(六) 历史命令记录

Bash 还具备完善的历史命令记录功能。在用户操作 Linux 操作系统的时候, 每一个操作的命令都会被记录到历史命令记录中, 因此用户可以通过历史命令记录查看和使用以前操作的命令。为了保持知识结构的完整性, 本小节将介绍使用历史命令记录的三类方法, 并给出详细的实例。

注意, 初学者实在记不住历史命令记录的相关使用方法也没关系, 平时用得最多的与历史命令记录相关的快捷键其实还是 ↑ (上方向键) 和 ↓ (下方向键)。

1. 使用快捷键方法

Shell 环境提供了许多快捷键, 用以搜索历史命令的快捷键如表 2-1 所示。

表 2-1 用于搜索历史命令的快捷键

快捷键	描述
↑ (上方向键)	查看上一条命令
↓ (下方向键)	查看下一条命令
Ctrl+P	查看历史命令记录中的上一条命令
Ctrl+N	查看历史命令记录中的下一条命令
Ctrl+R	进入反向搜索模式
Alt+>	移动到历史命令记录末尾

【实例 2-16】使用快捷键搜索历史命令。

首先, 执行 pwd 和 whoami 两条命令。执行效果如图 2-18 所示。

```
[zp@localhost ~]$ pwd
/home/zp
[zp@localhost ~]$ whoami
zp
```

图 2-18 执行 pwd 和 whoami 两条命令

接下来, 通过使用键盘的 ↑ 和 ↓, 用户不仅可以快速地查找出这两条命令, 还可以查找出之前使用的命令。

注意, 对于表 2-1 中的快捷键, 学生做到了解即可; 即便没有掌握, 也不影响对本书后续内容的学习。

2. 使用 history 命令方法

使用 history 命令可以列出所有使用过的命令并为其编号。这些信息被存储在用户主目录的 .bash_history 文件中, 这个文件默认情况下可以存储 1000 条历史命令记录。Bash 启动的时候会读取 ~/.bash_history 文件, 并将其载入内存中。Bash 退出时也会把内存中的历史命令记录回写到 ~/.

bash_history 文件中。用户可以直接使用 cat 命令查看该文件内容，也可以使用 history 命令查看历史命令记录，此时每一条命令前面都会标示一个序列号。

history 命令语法格式如下：

history [选项]

【实例 2-17】查看历史命令记录。

使用不带参数的 history 命令可以查看所有近期命令记录。执行如下命令：

[zp@ localhost ~]\$ history

执行效果如图 2-19 所示。

```
[zp@localhost ~]$ history
 1 ifconfig
 2 ssh localhost
 3 python -v
 4 python --version
 5 ls /mnt/hgfs/
 6 ls
```

图 2-19 查看所有的历史命令记录

用户也可以指定查看最近的五条历史命令记录。执行如下命令：

[zp@ localhost ~]\$ history 5

【实例 2-18】清空所有历史命令记录：

使用“history-c”会立即清空所有历史命令记录。执行如下命令：

[zp@ localhost ~]\$ history-c #清空所有历史命令记录

[zp@ localhost ~]\$ history #查看所有历史命令记录，此时只有一条记录，即当前命令

执行效果如图 2-20 所示。

```
[zp@localhost ~]$ history-c
[zp@localhost ~]$ history
 1 history
```

图 2-20 清空所有历史命令记录

3. 使用历史命令方法

使用“! +编号”可以执行特定编号对应的历史命令，使用 fc 命令可以编辑历史命令。表 2-2 中给出了一些典型的历史命令用法，读者可以在命令行界面中输入实例进行尝试。需要说明的是，该表部分实例，特别是最后几行的实例对历史命令记录的内容存在要求。如果用户的命令列表中没有包含这些命令或者文本，并不一定能得到相应的结果。

表 2-2 历史命令使用实例

使用实例	功能描述
!!	重复执行上一条命令
! 3	执行历史命令记录中的第 3 条命令
! w	执行上一条 w 命令（或执行以 w 开头的历史命令）
fc	编辑并执行上一条历史命令
fc-2	编辑并执行倒数第 2 条历史命令
! -4	执行倒数第 4 条命令
! \$	使用前一条命令最后的参数

【实例 2-19】 使用历史命令记录。

首先，读者输入若干条命令，这些命令将作为历史命令被记录下来，后续可以直接操作这些命令。执行如下命令：

```
[zp@ localhost ~]$ whoami
[zp@ localhost ~]$ date
[zp@ localhost ~]$ time
[zp@ localhost ~]$ pwd
[zp@ localhost ~]$ uname
```

执行效果如图 2-21 所示。

```
[zp@localhost ~]$ whoami
zp
[zp@localhost ~]$ date
2022年 06月 10日 星期五 04:10:16 EDT
[zp@localhost ~]$ time
real    0m0.000s
user    0m0.000s
sys     0m0.000s
[zp@localhost ~]$ pwd
/home/zp
[zp@localhost ~]$ uname
Linux
```

图 2-21 执行若干条命令

然后，用户可以通过快捷方式直接调用前述历史命令。各条命令的含义请参考表 2-2 的实例进行理解。执行如下命令：

```
[zp@ localhost ~]$ !!
[zp@ localhost ~]$ !-2
[zp@ localhost ~]$ !w
[zp@ localhost ~]$ fc-3
```

执行效果如图 2-22 所示。其中，执行“fc-3”时，将弹出一个命令行编辑器（一般是 Vi/Vim，也可能是其他。例如，部分版本的 Ubuntu 操作系统中使用的是 GNU nano），显示命令“uname”。用户首先在键盘上按“I”键，然后可以在命令编辑界面将“uname”修改为“uname -r”。接下来，用户先按“Esc”键，然后按“Shift+:”组合键，此时在屏幕的左下角将出现“:”提示符，读者在该提示符后面输入“wq”，按“Enter”键确认即可，如图 2-23 所示。执行成功后将得到与【实例 2-7】中等价命令的执行效果。

```
[zp@localhost ~]$ !!
uname
Linux
[zp@localhost ~]$ !-2
pwd
/home/zp
[zp@localhost ~]$ !w
whoami
zp
[zp@localhost ~]$ fc -3
uname -r
5.14.0-96.el9.x86_64
```

图 2-22 调用历史命令记录