

Linux for Developers
Jumpstart Your Linux Programming Skills

Linux 程序设计基础

[美] 威廉·罗思韦尔 (William Rothwell) 著
陈光欣 译

■ 从零入手，快速掌握Linux基本技能



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Linux 程序设计基础

[美] 威廉·罗思韦尔 (William Rothwell) 著
陈光欣 译



Linux for Developers
Jumpstart Your Linux Programming Skills

人民邮电出版社
北京

图书在版编目(CIP)数据

Linux程序设计基础 / (美) 威廉·罗思韦尔
(William Rothwell) 著; 陈光欣 译. — 北京: 人
民邮电出版社, 2019. 8
(图灵程序设计丛书)
ISBN 978-7-115-51544-5

I. ①L… II. ①威… ②陈… III. ①Linux操作系统
—程序设计 IV. ①TP316.85

中国版本图书馆CIP数据核字(2019)第124045号

内 容 提 要

本书是Linux入门书,旨在介绍在Linux操作系统上开发软件所需具备的知识。本书共分四部分,主要内容包括:开源软件简介;Linux操作系统基础知识介绍,包括文件系统、Linux基本命令、文本编辑器、系统管理等内容;Linux编程语言概述,内容涵盖BASH shell脚本、Perl脚本、Python脚本,以及C、C++和Java;流行的软件版本控制工具Git。

本书适合所有对Linux编程感兴趣的读者阅读。

-
- ◆ 著 [美] 威廉·罗思韦尔
译 陈光欣
责任编辑 岳新欣
责任印制 周昇亮
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 11
字数: 260千字 2019年8月第1版
印数: 1-3 500册 2019年8月北京第1次印刷
著作权合同登记号 图字: 01-2018-8361号
-

定价: 59.00元

读者服务热线: (010)51095183转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147号

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

“最好的朋友，就是当全世界都远离你时，唯一走进你生活的那个人。”

——香农·奥尔德

谢谢你，萨拉，我的爱人和妻子，谢谢你走进了我的生活。

“强者将人扛在肩头，而非踩在脚下。”

——迈克尔·沃森

爸爸妈妈，谢谢你们，谢谢你们的支持。

“七次跌倒，八次站起来。”

——日本谚语

谢谢你，茉莉娅，谢谢你的理解。

致 谢

感谢所有帮助我完成本书的朋友。就像任何知书达理的作者应该说的，出版这样一本书要付出很多心血，本书是很多人一起努力和奉献的结果。

凯斯·莱特和马修·埃尔姆克，感谢你们的技术审阅。你们的反馈使得本书比我的原稿好得多，这是毫无疑问的。

克里斯·赞恩，我肯定找不到比你更好的编辑了，你让我看起来貌似也能自己组织出连贯的句子——这不是一件简单的事！

德布拉·威廉姆斯·考利，感谢你看到了本书的价值，并在整个写作过程中一直施以援手。

前 言

我认为本书是一段旅程的开始。你在这段旅程中的起点可能与他人不同，本书的目的是帮你掌握在 Linux 操作系统上开发软件所需具备的知识。

某些读者已有在基于 Windows 的平台上开发软件的经验，这些读者可以将本书作为一本指南，了解一下 Linux 系统上的软件开发与自己使用的平台有什么不同。

可能你已经在使用 Linux 进行工作了，现在想开始编写代码。同样，本书也会为你提供一个非常棒的起点。

本书分为四个部分。

第一部分，开源软件。这部分只有一章：第 1 章，开源软件简介。这一章介绍开源软件，包括它相对于闭源软件的优点，以及软件许可的一些基础知识。

第二部分，Linux 基础。这部分介绍 Linux 操作系统，内容既涉及终端用户，也涉及系统管理员，目标是帮助软件开发人员掌握使用 Linux 系统的必备知识。

- 第 2 章，Linux 简介。在这一章中，你将学习 Linux 的基础知识，包括如何访问 Linux 系统、如何使用基于 Linux 的图形用户界面（GUI），以及一些基本的命令行操作。
- 第 3 章，文件系统。这一章重点说明 Linux 是如何组织文件的。你将学习文件系统的概念，以及如何使用和管理文件系统。
- 第 4 章，基本命令。在这一章中，你将学习多个 Linux 命令，它们对于任何开发人员来说都至关重要。
- 第 5 章，文本编辑器。作为开发人员，你需要知道如何编辑文件。这一章重点介绍 vi 编辑器，这是一个在 Unix 和 Linux 中都最常用的文本编辑器。此外还会介绍其他几个 Linux 文本编辑器。
- 第 6 章，系统管理。即使是开发人员，知道如何进行系统管理也是非常有用的。在这一章中，你将学会如何添加软件以及如何管理用户。

第三部分，Linux 编程语言。这部分概述 Linux 系统上可用的编程语言。有多种语言供你选择！这部分的目标不是教会你每种语言的所有知识，而是通过对它们的介绍，帮助你确定哪种语

言最适合你。

- 第 7 章，Linux 编程语言概述。这一章对编程语言进行总体介绍，重点在于说明脚本语言和结构化语言（即编译型语言）之间的区别。
- 第 8 章，BASH shell 脚本。这一章介绍 BASH shell 脚本语言。你将学习如何创建能与用户交互的代码，以及 BASH shell 编程语言的其他功能。
- 第 9 章，Perl 脚本。这一章重点介绍如何使用 Perl 脚本语言编程，还包括流控制和变量使用等内容。
- 第 10 章，Python 脚本。你将学习 Python 脚本的基础知识，包括多种 Python 变量类型，以及如何重用代码和进行流控制。
- 第 11 章，C、C++ 和 Java。你将学习在基于 Linux 的系统上创建 C、C++ 和 Java 代码的基本技术。

第四部分，使用 Git。这部分介绍一种非常流行的软件版本控制工具：Git。使用版本控制工具可以节省大量时间、金钱和精力，在多个开发团队协同工作时尤为如此。

- 第 12 章，Git 基础。这一章介绍 Git 的概念，包括版本控制的概念，以及 Git 的安装和功能。
- 第 13 章，使用 Git 管理文件。这一章介绍如何使用 Git 的一些功能，比如暂存、提交和分支。
- 第 14 章，管理文件差异。这一章重点介绍如何处理不同版本的文件。你将学习如何处理文件差异以及如何合并文件。
- 第 15 章，Git 高级特性。你将学习如何管理 Git 仓库以及如何进行补丁操作。

祝你一路顺风！

在 informit.com 上注册你购买的 *Linux for Developers*，就可以方便地获取已有的可下载资料、更新和勘误。^①要进行注册，请先在 informit.com/register 页面上登录或创建新账户，然后输入本书的 ISBN 号 9780134657288 并点击 Submit 按钮。注册完成后，你可以在“Registered Products”中找到所有可用的优惠内容。

^① 本书中文版勘误请到 <http://www.ituring.com.cn/book/2468> 查看和提交。——编者注

目 录

第一部分 开源软件

第 1 章 开源软件简介	2
1.1 定义源代码	2
1.1.1 闭源软件	3
1.1.2 开源软件	4
1.1.3 “自由”软件	4
1.2 选择开源许可证	6
1.2.1 选项	6
1.2.2 关键名词	6
1.2.3 示例	7
1.2.4 有用链接	8
1.3 小结	8

第二部分 Linux 基础

第 2 章 Linux 简介	10
2.1 访问 Linux 系统	10
2.1.1 选择正确的发行版	11
2.1.1 登录	12
2.2 使用 GUI	14
2.3 基本命令行操作	16
2.3.1 命令行结构	17
2.3.2 获取帮助	17
2.4 小结	22
第 3 章 文件系统	23
3.1 理解文件系统	23
3.1.1 了解最常用的目录	24

3.1.2 命名须知	24
3.2 浏览文件系统	25
3.3 管理文件系统	27
3.3.1 管理目录	29
3.3.2 管理文件	30
3.4 小结	38
第 4 章 基本命令	39
4.1 命令行工具	39
4.1.1 查看文件	40
4.1.2 搜索文件	43
4.1.3 比较文件	45
4.1.4 shell 特性	46
4.1.5 权限	48
4.2 开发人员工具	50
4.2.1 文件压缩命令	50
4.2.2 grep 命令	53
4.3 小结	56
第 5 章 文本编辑器	57
5.1 vi 编辑器	57
5.1.1 为什么要学习 vi	58
5.1.2 什么是 vim	58
5.1.3 基本 vi 命令	59
5.2 其他编辑器	67
5.2.1 Emacs	67
5.2.2 gedit 和 kwrite	69
5.2.3 nano 和 joe	69
5.2.4 lime 和 bluefish	70
5.3 小结	70

第 6 章 系统管理	71	8.3.1 while 循环	99
6.1 基本任务	71	8.3.2 for 循环	100
6.1.1 获得 root 账户的访问权限	71	8.3.3 case 语句	100
6.1.2 显示磁盘使用状态	74	8.4 用户交互	101
6.2 管理软件	75	8.5 附加信息	101
6.2.1 查找与列出软件	76	8.6 小结	102
6.2.2 安装软件	77	第 9 章 Perl 脚本	103
6.3 用户账户	78	9.1 Perl 脚本基础	103
6.3.1 添加用户账户	78	9.1.1 运行 Perl 代码	103
6.3.2 修改用户账户	80	9.1.2 其他 Perl 文档	105
6.3.3 删除用户账户	80	9.1.3 变量和值	105
6.3.4 理解组	80	9.2 流控制	108
6.3.5 管理组	81	9.3 条件	109
6.4 小结	82	9.4 其他特性	111
		9.5 小结	113
		第 10 章 Python 脚本	114
		10.1 Python 脚本基础	114
		10.1.1 运行 Python 代码	115
		10.1.2 附加文档	116
		10.1.3 变量和值	116
		10.2 流控制	118
		10.3 条件	119
		10.4 其他特性	120
		10.5 小结	121
		第 11 章 C、C++和 Java	122
		11.1 理解系统库	122
		11.1.1 管理共享库文件	123
		11.1.2 查看共享库文件	124
		11.2 建立软件包	125
		11.2.1 建立 RPM 包	125
		11.2.2 建立 Debian 包	127
		11.3 Java 安装基础	128
		11.4 小结	128
第三部分 Linux 编程语言			
第 7 章 Linux 编程语言概述	84		
7.1 脚本语言	84		
7.1.1 BASH shell 脚本	84		
7.1.2 Perl 脚本	85		
7.1.3 Python 脚本	87		
7.1.4 其他脚本语言	88		
7.2 编译型语言	93		
7.2.1 C 程序基础	93		
7.2.2 C++程序基础	93		
7.2.3 Java 程序基础	94		
7.3 IDE	94		
7.4 小结	94		
第 8 章 BASH shell 脚本	95		
8.1 BASH 脚本基础	95		
8.2 条件表达式	96		
8.2.1 整数比较	98		
8.2.2 文件比较	99		
8.3 流控制语句	99		

第四部分 使用 Git

第 12 章 Git 基础.....130

12.1 版本控制的概念.....130

12.1.1 第一代版本控制软件.....130

12.1.2 第二代版本控制软件.....131

12.1.3 第三代版本控制软件.....132

12.2 安装 Git.....135

12.3 Git 概念与特性.....136

12.3.1 Git 暂存机制.....136

12.3.2 选择 Git 仓库主机.....137

12.3.3 配置 Git.....138

12.4 小结.....140

第 13 章 使用 Git 管理文件.....141

13.1 基本配置.....141

13.2 git status.....142

13.2.1 处理多位置情形.....144

13.2.2 让 Git 忽略文件.....146

13.3 删除文件.....147

13.4 处理分支.....148

13.5 推送分支.....150

13.6 小结.....150

第 14 章 管理文件差异.....151

14.1 执行 diff 命令.....151

14.1.1 处理空白字符.....153

14.1.2 比较分支.....154

14.2 合并文件.....155

14.3 小结.....160

第 15 章 Git 高级特性.....161

15.1 管理文件仓库.....161

15.1.1 从远程服务器获取内容.....162

15.1.2 经由 SSH 连接.....164

15.2 补丁操作.....165

15.3 小结.....165

第一部分

开源软件

在开发软件时，你需要回答的一个重要问题是：“软件将以何种许可方式发布？”要找到该问题的答案，可能需要经历一个艰苦的过程。

你必须确定要对代码提供何种保护，以及允许他人以何种方式使用你创建的软件。本部分只有一章，重点在于帮助你确定软件的许可方式。在这一章中，你将学习以下内容：

- 闭源软件与开源软件的区别；
- 开源保护的概念；
- 主要开源许可证之间的差别。



如果你编写了一个非常棒的程序，想将其公之于众，这时就需要做一个非常重要的决定：软件使用哪种许可证？

这个决定将产生一些非常重要的影响，比如：

- 用户将如何使用你的软件；
- 代码对他人是可见的还是“隐藏”的；
- 其他开发人员能否使用你的代码来创建自己的程序；
- 其他人能否出售或转售你的程序。

免责声明

许可证的问题非常复杂，并且对软件的使用方式有重要影响。本书中的讨论旨在让你对各种不同的许可证有一个基本的了解，并不能作为法律上的建议。本书作者不想提供任何法律建议。你要做出任何关于软件许可证的决定，请一定考虑寻求正规的法律建议。

1.1 定义源代码

你需要回答的第一个问题最有可能是：“这个软件是开源还是闭源的呢？”要回答这个问题，你首先需要知道什么是源代码。

软件是由一组指令组成的，这些指令由编程语言写成。现在有多种编程语言，包括 C、C++、Java、Perl、Python，等等。这种指令集合就称为源代码。图 1-1 就是一个由 C 语言写成的源代码示例。

```

/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello World");
}

```

图 1-1 用 C 写成的源代码

通常不能直接使用这种源代码来运行程序。多数语言需要一个编译过程，将源代码转换为操作系统能够理解的指令。编译的结果对人来说就是一堆垃圾数据，但对操作系统来说是有意义的。图 1-2 给出了将源代码转换为编译代码的示例。

```

/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello World");
}

```

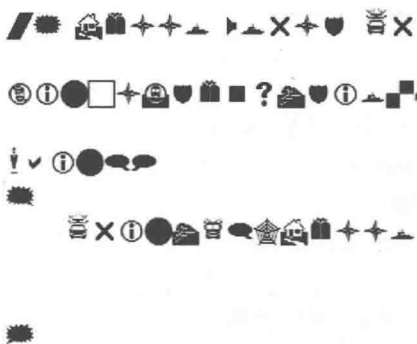


图 1-2 源代码转换为编译代码

如果你选择的软件许可方式为闭源，那么就可以只向客户提供编译后的代码。开源许可的软件还需要提供源代码。

1.1.1 闭源软件

闭源软件又称为专有软件，它的目的是将源代码作为一种秘密严密保护起来。它的思想是，如果他人看到了源代码，那么就可能会复制并非法使用，竞争对手可能会造成开发软件的组织遭受经济损失。可以想象，复制（抄袭）别人软件的成本要比自己开发软件低得多。

闭源软件这个名词还经常用来代替商业软件，但并不准确。商业软件必须购买才能使用。闭源软件和开源软件都可以商业化。具体的许可方式决定了软件是商业软件还是“自由”软件。^①

以下是闭源软件的例子：

□ Microsoft Windows

^① 这里给“自由”加引号是有原因的。很快你就会看到，“自由”这个词在软件使用方面必须有一个明确的定义。

- Adobe Photoshop
- Apple macOS

1.1.2 开源软件

当软件的源代码和编译代码都可以获取时，就可以认为它是开源软件。^① 版权所有者的软件许可会授予用户特定的查看、修改和分发软件的权限。现在开源许可证的种类非常多，可以让你选择授予何种权限。

尽管有些开源软件从经济意义上说是免费的，但这并不是它们被视为开源软件的必要条件。开源指的是能获取源代码，与如何使用软件以及软件是否需要费用都没有关系。

常见的开源软件如下：

- Linux^②
- Apache HTTP Server
- Firefox
- Git
- Openoffice.org

1.1.3 “自由”软件

对于软件来说，“自由”不见得是一个所有人意见一致的概念。有些人认为自由软件就是没有成本的软件，换言之，获取和使用这种软件没有任何花费。

但是，在软件使用方面，自由意味着什么呢？软件可以被用户任意使用，还是有某些限制？软件可以在世界上任何地方使用，还是有一些地理限制？你能自由地修改软件并分发修改后的免费版本，还是禁止这么做？如你所见，软件领域内的“自由”不是一个非常明确的概念。

理查德·斯托曼给出了“自由”的一种定义方法，并通过自由软件基金会（FSF）公之于众：

“自由软件基金会（Free Software Foundation）这个名称中的‘free’指的不是免费，而是自由。首先，这种自由可以让你对程序进行复制并重新分发给周围的人，他们也能像你一样去使用；其次，这种自由下你可以对程序进行修改，可以控制程序，而不是让程序来控制你。因此，源代码必须要提供给你。”

① 实际上，一些开源项目只提供源代码，由使用代码的用户自行编译。此外，有些语言没有编译过程，所以用这些语言写成的程序只包括源代码。

② 从技术角度来讲，Linux 指的是 Linux 内核，也就是 Linux 操作系统的核心部分。Linux 操作系统上的多数软件也是开源的，但并不是说它们必须开源才能纳入 Linux 操作系统。

请注意，上面的定义强制规定了自由软件也必须是开源软件，但不是所有人都同意这一点，你在市场上会看到一些闭源的自由软件。

自由软件基金会定义了“四大自由”，由此给出了另一种定义“free”的方法。

- 自由之零：基于任何目的，按你的意愿运行软件的自由。
- 自由之一：学习软件工作原理并加以修改，使其按你的意愿进行工作的自由。可访问源代码是此项自由的先决条件。
- 自由之二：分发软件副本以帮助周围人的自由。
- 自由之三：将你修改过的软件版本分发给他人的自由，这样可以让整个社区有机会从你的修改中受益。可访问源代码是此项自由的先决条件。

这四大自由就是所谓 FOSS（Free and Open Source Software，自由开源软件）的核心思想。^① FOSS 试图解决这样一个问题：何种软件可以被视为“自由软件”？这个定义强调了一个事实，即不是所有自由软件都是开源的。相反，也不是所有开源软件都是通过这四大自由进行许可的。

开源软件是个复杂的世界，要理解它以及“自由”在其中的含义需要一段时间。图 1-3 给出了一个图形说明，包括了开源软件的各种组成部分。

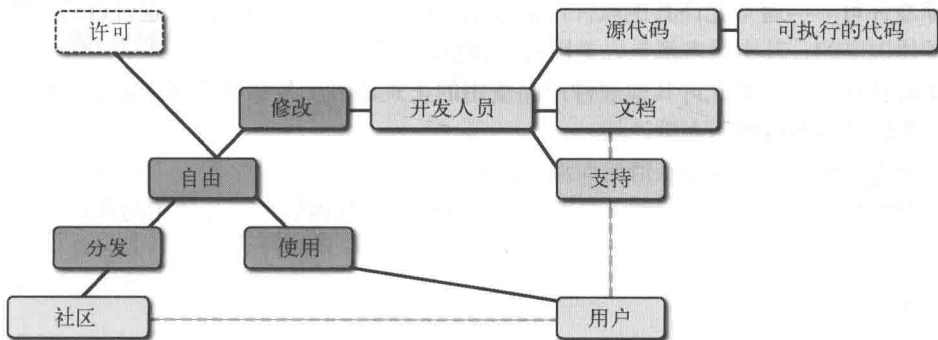


图 1-3 开源社区图解

图 1-3 突出体现了开源软件各个要素之间的复杂组合关系。开发人员编写源代码、创建文档并提供支持。然而，软件的用户也经常在这个过程中扮演重要角色。实际上，一些开源软件的支持和文档很少（甚至完全不是）来自于开发人员，而是依靠一个强大的用户群体（社区）来提供这些关键的环节。

请注意，图 1-3 还描述了修改、分发和使用软件的自由。这些自由是通过软件许可的方式实现的。

^① FOSS 常与 FLOSS（Free/Libre and Open Source Software）这个名词交替使用。