Ruby 原理剖析

Ruby Under a Microscope

[美] Patrick Shaughnessy 著
张汉东 译
秦凡鹏 审校

華中科技大學出版社

Ruby原理剖析

Ruby Under a Microscope

[美] Patrick Shaughnessy 著
张汉东 译
秦凡鹏 审校

華中科技大學出版社

内容简介

本书解开 Ruby 编程语言的魔法面纱。全书图文并茂、深入浅出地剖析了 Ruby 编程语言的核心工作原理。作者本着科学实证的精神，设计了一系列实验，帮助读者轻松了解这门编程语言的工作奥秘，包括 Ruby 如何用虚拟机执行代码，Ruby 的垃圾回收算法，以及类和模块在 Ruby 内部的关系等。

湖北省版权局著作权合同登记 图字:17－2016－390 号

图书在版编目(CIP)数据

Ruby 原理剖析/(美)帕特里克·肖内西著；张汉东译；秦凡鹏审校. —武汉：华中科技大学出版社，2016.10

ISBN 978-7-5680-2262-0

Ⅰ. ①R… Ⅱ. ①帕… ②张… ③秦… Ⅲ. ①网页制作工具-程序设计 Ⅳ. ①TP393.092.2

中国版本图书馆 CIP 数据核字(2016)第 243475 号

Ruby 原理剖析

Ruby Yuanli Pouxi

[美]Patrick Shaughnessy 著

张汉东 译 秦凡鹏 审校

策划编辑：徐定翔　　责任校对：张　琳

责任编辑：徐定翔　　责任监印：周治超

出版发行：华中科技大学出版社(中国·武汉)　　电话：(027)81321913

武汉市东湖新技术开发区华工科技园　　邮编：430223

录　　排：华中科技大学惠友文印中心　　印　　刷：湖北新华印务有限公司

开　　本：787mm×960mm　1/16

印　　张：23.75　　字　　数：490 千字

版　　次：2016 年 10 月第 1 版第 1 次印刷　　定　　价：78.80 元

献给我的妻子 Cristina、女儿 Ana、儿子 Liam！

感谢你们一直以来对我的支持！

业界评论

ADVANCE PRAISE FOR RUBY UNDER A MICROSCOPE

“很多人研究过 Ruby 的源码，但很少有人像 Patrick 这样把研究成果写成一本书。我特别喜欢书里的图表，加上 Patrick 恰到好处的解说，晦涩难懂的内容变得易于理解。本书是编程极客和喜欢深入研究软件工具的 Ruby 爱好者的福音。”

——PETER COOPER (@PETERC)，《RUBY INSIDE》《RUBY WEEKLY》的编辑

“这本书填补了 Ruby 领域的空白——内容太棒了！”

——XAVIER NORIA (@FXN)，RUBY HERO、RUBY ON RAILS 核心团队成员

“Patrick Shaughnessy 做了一件很棒的事，写了这本关于 Ruby 内部原理的书。你一定要看，因为其他书里找不到这样的内容。”

——SANTIAGO PASTORINO (@SPASTORINO)，WYEWORKS 联合创始人、RUBY ON RAILS 核心团队成员

“这本书让我爱不释手，它让我对 Ruby 和 CS 有了更深的理解。书中的图表真的非常棒，我写代码时会浮现在我的脑海里。它是我最喜欢的三本 Ruby 书籍之一。”

——VLAD IVANOVIC (@VLADIIM)，HOLLER SYDNEY 的数字媒体策略师

“虽然我不经常研究 Ruby 的内部原理，但是这本书绝对值得一读。”

——DAVID DERYL DOWNEY (@DAVIDDWDOWNEY)，CYBERSPACE TECHNOLOGIES GROUP 创始人

关于作者
ABOUT THE AUTHOR

Patrick Shaughnessy 是一名 Ruby 开发者，目前在麦肯锡管理咨询公司（McKinsey & Co.）工作。Patrick 原本在麻省理工学院（MIT）学习，准备成为一名物理学家，后来却阴差阳错地干了 20 多年软件开发工作。他用过 C、Java、PHP、Ruby 等编程语言。编写这本书使他有机会把接受的科学训练用于研究 Ruby。他能说一口流利的西班牙语，经常陪妻子回西班牙北部的娘家。目前，他与妻子和两个孩子居住在波士顿郊外。

译者序

作为一名资深的 Rubyist，我本应该主动去了解 Ruby 语言底层的实现细节，但我却没有，因为我被 Ruby 底层的复杂性吓住了。虽然自己学过 C 语言，也读过网上流传的《Ruby Hacking Guide》，但是一直没能对 Ruby 的底层实现形成系统性的认识。直到我读到了 Patrick 写的这本书，才发现这就是我要的书！读完一遍后，我马上产生了一个念头：我要把它翻译出来分享给更多的人！书里的知识是每位 Rubyist 都应该掌握的！庆幸的是，华中科技大学出版社引进了这本书的版权，我在得知此消息的第一时间争取到了翻译权，因为我实在太喜欢这本书了，我实在太想翻译这本书了！

本书图文并茂，系统地展示了 Ruby 核心的底层实现原理，全书共 12 章，以循序渐进的方式带领读者一步步探索 Ruby 的底层原理。前 3 章讲解了 Ruby 代码从分词到编译的过程，学完这些内容，不仅可以让你对 Ruby 如何执行代码有一个系统的认识，而且可以让你对编程语言的实现有更深的理解。因为排除细节不谈，从宏观上看，这个过程不是 Ruby 语言特有的，它对读者理解其他编程语言也会有一定帮助。中间 6 章，作者讲解了 Ruby 类、对象、控制流程的底层实现，以及最重要的块（block）和元编程。学完这些内容，你会发现 Ruby 原来如此简单！接下来的 2 章讲解 JRuby 和 Rubinius 的底层实现，分析了 JRuby、Rubinius 和 MRI Ruby 的异同。最后一章分析了 Ruby 的 GC 技术，并比较了 JRuby、Rubinius、MRI Ruby 的 GC 差异。说实话，这么详细的内容在其他书里真没见过！

当然，本书并没有涵盖 Ruby 的所有方面，比如书中未提及 GIL、线程、纤程等技术的底层实现。但是它通过分析 Ruby 核心底层的实现方式，为读者提供了自学 Ruby 底层实现原理的思路和勇气。就让本书开启你深入探索 Ruby 之旅吧！

最后，我要感谢徐定翔编辑在翻译过程中给予的指导和帮助；感谢审校者秦凡鹏，他认真的审校保证了本书的翻译质量；感谢我的妻子宋欣欣，感谢她对我的体谅和付出，让我有更多的时间翻译本书；也感谢朋友们对我的支持。俗话说，“有人的地方就有 Bug”，翻译也难免出现 Bug。为此，我建立了一个 Github 仓

库，发现 Bug 的读者可以给我提 Issues，大家一起进步！地址是：https://github.com/Ruby-Study/Discuss_Ruby_under_a_microscope。

张汉东

2015 年 12 月

张汉东，Rubyist、资深程序员、独立企业培训师/咨询师、创业者。2006 年年底接触 Ruby，被其魅力征服，从此开启编程生涯的快乐篇章。

推荐序
FOREWORD

哦，你好！虽然我向来喜欢含蓄，但我不得不说：你应该买这本书！

我的名字叫 Aaron Patterson，但是网络上的朋友都叫我 tenderlove。我效力于 Ruby 和 Ruby on Rails 的核心团队，同时也是这本书的技术顾问。这是不是意味着你就应该听我的呢？不是的。好吧，也许。

实际上，当 Patrick 要我做这本书的技术顾问时，我兴奋得大礼帽都歪了，单片眼镜也掉到咖啡里去了！我知道很多开发者都被 Ruby 的底层原理吓住了，不敢深入研究。常常有人问我该如何学习 Ruby 的底层原理，或者该从哪里入手。不幸的是，我没能给他们一个好答案，但现在我可以回答他们了。

Patrick 科研式的写作风格让 Ruby 的底层原理变得更加直观。实验与讲解的结合让 Ruby 的行为和性能更容易理解。如果你对 Ruby 代码产生疑惑，无论是性能表现、局部变量，还是垃圾回收，你都能在这本书里找到解答。

如果你想探索 Ruby 内部原理，或者想理解 Ruby 的工作方式，那就不用犹豫了，这本就是你要找的书。我很喜欢这本书，希望你也喜欢。

Aaron Patterson

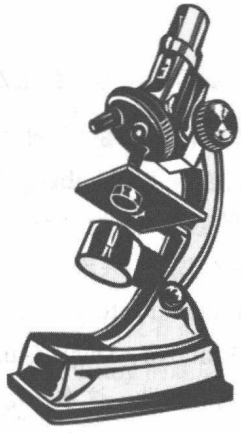
<3<3<3<3

致谢
ACKNOWLEDGMENTS

如果没有这么多人的支持，我可能永远不会完成这本书。

首先，感谢 Satty Bhens 和所有麦肯锡（管理咨询公司）的同事，让我可以在写书和工作之间灵活切换。感谢 Alex Rothenberg 和 Daniel Higginbotham 阅读了我的手稿，他们提出了宝贵的意见，并且在整个写作过程中给予了我帮助。感谢 Xavier Noria，他是最早对这本书产生兴趣的人，并提出了极好的建议，同时他也为实验 6-1 提供了灵感。Santiago Pastorino 也阅读了书稿。Jill Caporrimo、Prajakta Thakur、Yvannova Montalvo、Divya Ganesh 和 Yanwing Wong 是我的“审校特攻队”，如果没有他们的帮助，这本书是很难完成的。最后，如果没有不断给我鼓励和支持的 Peter Cooper，我可能永远都不会尝试写这本书。谢谢你，Peter。

感谢 No Starch 出版社所有帮助我出版这本书的人。我为本书感到自豪，它填补了 Ruby 内部原理类图书的空白。感谢我的文字编辑 Julianne Jigour，她让文字变得更加清晰和易懂。感谢 Riley Hoffman 和 Alison Law，你们精美地再现了数以百计的图表，和你们一起工作真是令人愉快。感谢 Charles Nutter 在技术上的帮助，以及对 JVM 垃圾回收的建议。特别感谢 Aaron Patterson，你的建议和技术审校让本书变得更加有趣和准确。最后，感谢 Bill Pollock 阅读和编辑本书的每一行文字，你的专业的知识和指导帮助我完成了写书的梦想。



复杂与简单之间的区别在于你的观察是否足够细致入微。

引言
INTRODUCTION

乍一看，学习 Ruby 似乎相当简单。世界各地的开发者都认为 Ruby 的语法简洁优雅。你可以用非常自然的方式表达算法，然后只需要在命令行输入 ruby，按下回车，Ruby 脚本就可以运行了。

然而，这只是表面现象。实际上，Ruby 借鉴了许多复杂语言（如 Lisp 和 Smalltalk）的精妙理念。此外，Ruby 是动态的，它使用元编程，即 Ruby 程序可以检查和修改自身。Ruby 外表“简单”，实际上是非常复杂的工具。

通过深度探索 Ruby——学习 Ruby 的内部工作原理——你会发现一些重要的计算机科学理论支撑着 Ruby 的特性。通过学习，你会对这门语言的底层行为有更深的理解。在这个过程中，你还将了解打造 Ruby 的团队是如何使用这门语言的。

本书将展示 Ruby 程序运行时的内部情形，让你学习 Ruby 如何理解和执行代码。书中丰富的图表将帮助你建立一套心智模型（mental model），以便更好地理解 Ruby 的行为（比如创建对象和调用块）。

本书适合哪些读者
Who This Book Is For

本书不是初学者的 Ruby 学习指南。我假设你已经知道如何写 Ruby 程序，并且每天都在使用它。优秀的 Ruby 入门教程已经很多了，不缺这一本。

虽然 Ruby 是用 C 这门偏底层且容易让人困惑的语言编写的，但是阅读本书不需要 C 语言编程基础。本书将从宏观层面解释 Ruby 的工作原理，不懂 C 语言编程的读者也能看懂。本书配有丰富的图表，可以帮助读者轻松理解 Ruby 的底层细节。

NOTE 我会提供一些 C 代码片段，并指明代码出处，以便有 C 语言基础的读者更好地理解 Ruby 的内部情形。如果读者对 C 代码的细节不感兴趣，那么可以忽略这些代码。

用 Ruby 测试 Ruby
Using Ruby to Test Itself

> 无论你有多聪明，无论你的理论有多完美，如果不符合实际，那么它就是错的。
>
> —— 理查德 • 费曼（Richard Feynman）

试想一下，整个世界的运作就像一个巨大的计算机程序。如果要解释自然现象或实验结果，像费曼这样的物理学家只要借助这个程序就可以了。（科学家的梦想成真了！）然而，宇宙并非如此简单。

幸运的是，要理解 Ruby 的工作原理，只需要阅读它的 C 源码，它就像一种描述 Ruby 行为的物理学理论。就像麦克斯韦方程解释电磁现象一样，Ruby 的源码可以解释传递参数或者在类中包含模块时发生了什么。

像科学家一样，也需要做实验来确保我们的假设是正确的。每学习一个主题，就会做一个实验，用 Ruby 来测试它自身！运行小的 Ruby 测试脚本，看输出结果或运行的快慢是否跟我们预想的一样。我们用实践检验 Ruby 的理论。这些实验都是用 Ruby 编写的，所以你也可以自己尝试。

哪种 Ruby
Which Implementation of Ruby

Ruby 是松本行弘（Yukihiro “Matz” Matsumoto）在 1993 年发明的，标准版的 Ruby 称为 Matz 的 Ruby 解释器（MRI）。本书的大部分内容将讨论 MRI 的工作原理，将学习 Matz 如何实现其自己的语言。

后来又出现了其他版本的 Ruby，像 RubyMotion、MacRuby 和 IronRuby，它们都运行在特定的平台之上。Topaz 和 JRuby 甚至不是用 C 语言构建的。还有一个版本叫 Rubinius，它是用 Ruby 来实现的。Matz 本人也在开发一个简化版本，叫 mruby，可以在其他应用程序里运行。

我将在第 10、11 和 12 章介绍 JRuby 和 Rubinius 的细节。你将了解它们是如何使用不同的理念和技术来实现相同的语言的。研究完这些 Ruby 的衍生版本后，你会对 MRI 的实现产生新的看法。

概述
Overview

第 1 章：分词与语法解析 本章学习 Ruby 是如何解析程序的。这是计算机科学最迷人的领域之一：计算机语言怎么会如此聪明可以理解给定的代码呢？这种智能到底是怎么实现的？

第 2 章：编译 本章解释 Ruby 如何用编译器把代码转换成不同的语言。

第 3 章：Ruby 如何执行代码 本章重点介绍 Ruby 用来执行程序的虚拟机。它的内部构造和工作机制。将深入虚拟机内部来搞清楚这些问题。

第 4 章：控制结构与方法调度 本章继续讲解 Ruby 虚拟机。先学习 Ruby 如何实现控制结构，比如 if...else 语句和 while...end 循环；然后探讨 Ruby 如何实现方法的调用。

第 5 章：对象与类 本章讨论 Ruby 的对象和类的实现。对象和类之间如何关联？Ruby 对象的内部是什么样的？

第 6 章：方法查找和常量查找 本章介绍 Ruby 模块及其与类的关系。学习 Ruby 如何查找代码中的方法和常量。

第 7 章：散列表：Ruby 内部的主力军 本章探讨 Ruby 散列表的实现。MRI 大部分内部数据使用了散列表，而不是仅用于保存散列对象。

第 8 章：Ruby 如何借鉴 Lsip 几十年前的理念 本章讲解 Ruby 最优雅、最有用的特性：块（block）。Ruby 的块是从 Lisp 借鉴来的。

第 9 章：元编程 这是 Ruby 开发最难的主题。学习 Ruby 内部如何实现元编程，将帮助你更有效地使用元编程。

第 10 章：JRuby：JVM 上的 Ruby 本章介绍 JRuby，一个用 Java 实现的 Ruby 版本。你将学习 JRuby 如何利用 Java 虚拟机（JVM）来更快地运行 Ruby 程序。

第 11 章：Rubinius：用 Ruby 实现 Ruby Rubinius 是 Ruby 最有趣且最

具创新性的版本。我们将查找并修改 Rubinius 中的 Ruby 代码，借此观察特定方法的工作原理。

第 12 章：MRI、JRuby 和 Rubinius 中的垃圾回收 垃圾回收（GC）是计算机科学中最神秘、最令人困惑的话题。你将了解 Rubinius、JRuby、MRI 各自的 GC 算法。

学习 Ruby 的内部实现细节后，你会对 Ruby 复杂功能和行为有更深入的理解。就像虎克在 17 世纪第一次用显微镜看到微生物和细胞一样，你也将看到 Ruby 内部各种有趣的结构和算法。我们将一起发现到底是什么赋予了 Ruby 生命！

目 录
CONTENTS