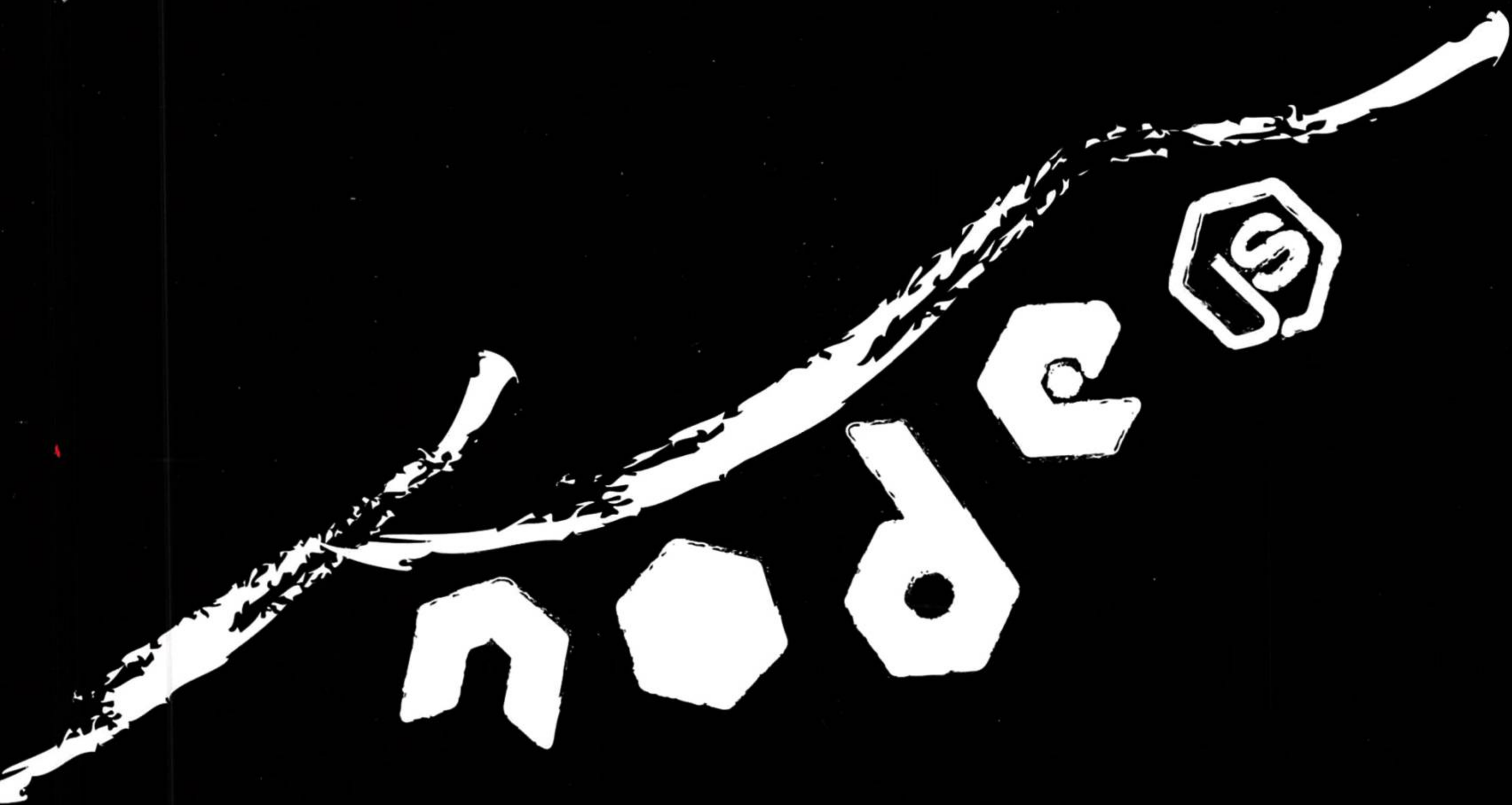


Broadview[®]
www.broadview.com.cn



Node.js

来一打 C++ 扩展

死月 著

 中国工信出版集团

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



Node.js

来一打 C++ 扩展

死月 著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

Node.js 作为近几年新兴的一种编程运行时，托 V8 引擎的福，在作为后端服务时有比较高的运行效率，在很多场景下对于我们的日常开发足够用了。不过，它还为开发者开了一个使用 C++ 开发 Node.js 原生扩展的口子，让开发者进行项目开发时有了更多的选择。

本书以 Chrome V8 的知识作为基础，配合 GYP 的一些内容，将教会大家如何使用 Node.js 提供的一些 API 来编写其 C++ 的原生扩展。此外，在后续的进阶章节中，还会介绍原生抽象 NAN 以及与异步相关的 libuv 知识，最后辅以几个实例来加深理解。不过，在学习本书内容之前，希望读者已经具备了初步的 Node.js 以及 C++ 基础。

阅读本书，相当于同时学习 Chrome V8 开发、libuv 开发以及 Node.js 的原生 C++ 扩展开发知识，非常值得！

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

Node.js: 来一打 C++ 扩展 / 死月著 . — 北京: 电子工业出版社, 2018.6

ISBN 978-7-121-33642-3

I . ① N… II . ①死… III . ① C++ 语言 - 程序设计 IV . ① TP312.8

中国版本图书馆 CIP 数据核字 (2018) 第 022884 号

策划编辑: 刘 皎

责任编辑: 李云静

印 刷: 三河市华成印务有限公司

装 订: 三河市华成印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×980 1/16 印张: 30 字数: 655 千字

版 次: 2018 年 6 月第 1 版

印 次: 2018 年 6 月第 1 次印刷

定 价: 109.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）51260888-819，faq@phei.com.cn。

好评袭来

This book contains absolutely everything you need to know about how all the pieces of Node.js' C++ code work and interact, explaining the necessary concepts without needing prior knowledge about the internals of V8, libuv or other pieces of Node.js. It shows well how Node.js' own built-in modules are constructed using the APIs provided by V8, so that they are usable from JavaScript, and how you can create the same kind of modules from scratch.

After having read this book, you will be able to write a production-quality, future-proof C++ extension for Node.js if you need to do that, or maybe even make changes Node.js itself if you're interested in that!

本书包含了所有你需要了解的有关 Node.js C++ 代码是如何运行和交互的知识，解释了一些你不需要知道 V8 的内部机制就能理解的必要概念，并介绍了 libuv 以及其他一些内容的方方面面。本书还展示了 Node.js 的内置模块是如何使用 V8 的 API 进行构建并可在 JavaScript 层面使用的——并且你也能用这种方法从头开始创建相同类型的模块。

读完本书，你将学到如何写出产品级质量的、面向未来的 Node.js C++ 扩展。感兴趣的话，你甚至可以对 Node.js 自身进行修改！

——安娜·亨宁森 (Anna Henningsen, addaleax)

Node.js 技术指导委员会成员 (Node.js TSC)

Node.js 不是第一个将 JavaScript 带入服务端领域的技术，然而它却成为史上最热门、最有影响力的工具之一。究其原因，其一，在于 Node.js 适逢后端高并发潮流，巧妙结合 Reactor 模型和 JavaScript 所擅长的回调风格，大大降低了开发高并发服务器应用的成本；

其二，在于恰逢浏览器大战，前端技术突飞猛进，急需一套适合 JavaScript 和前端工程师的生态和工具链，Node.js 刚好成为前端 JavaScript 最易上手掌握的命令行环境。在 Node.js 发展得如此火热之际，Node.js 的开发体验在不断提升，上手门槛也在不断降低。

然而，如果大家真正想突破自己并成为个中高手，无论是后端程序员希望在服务端及架构方面有所建树，还是前端程序员想跨越边界，都应该去了解 Node.js 的底层机制，去学习写一些 Node.js 的扩展。从 Node.js 的内在机制，我们可以学到更多有关计算机体系的知识，如内存管理、多线程编程等，真正向一个架构师、一个大牛迈进。

本书在这些方面提供了一个非常系统的指南。死月通过精彩的内容告诉大家：底层的知识并不枯燥，用 C++ 写一个扩展很有意思也很简单。作为 Node.js 工程师 / 爱好者的你，值得拥有本书。

——曹力 (ShiningRay)

酷链科技 CEO，暴走漫画前 CTO，糗事百科前联合创始人，
高级 Node.js 技术专家，《JavaScript 高级程序设计》译者

Native module is one of the most underappreciated features of Node.js. But even in the age of asm.js and WebAssembly, it is an irreplaceable part of the Node.js ecosystem due to its versatility and performance. XadillaX's book provides a refreshing introduction (or reintroduction), and is a must-read for all low-level Node.js engineers.

原生模块是 Node.js 中最被低估的功能之一。因为自身的性能和多样性，使其即使是在 asm.js 和 WebAssembly 时代，仍旧能作为 Node.js 生态系统中不可替代的部分存在。死月的书对其进行了一个令人耳目一新的介绍，它是所有底层 (Low-Level) Node.js 工程师的必读之物。

——顾天骋 (Timothy Gu)

pug、ejs 前 Maintainer, Node.js Core Collaborator 之一

本书全面讲解了 V8、libuv 的原理并且手把手教你编写一打 Node.js 的 C++ 扩展，它是目前市面上相关领域非常稀缺的技术书籍。如果你想更深入地了解 Node.js 的实现原理，除了熟读内置 API 文档之外，阅读本书也会是一个很好的选择。

——雷宗民 (老雷)

《Node.js 实战》作者之一

这是一本角度“刁钻”的 Node.js 相关书籍，其与市面上大多数 Node.js 书籍的定位不同。本书借为 Node.js 开发 C++ 扩展做基石，顺带介绍了 Chrome V8 和 libuv 的内容，填补了市场上这一类书籍的空白，值得一读。

——李启雷博士
趣链科技 CTO

无论是基础部分的 V8 练习，还是使用 Node.js 经典的 Addon 开发、用 NAN 来改写，或是 libuv 里的 WatchDog 案例、EFSW 的封装，死月一直把实战贯穿在整本书之中。甚至第 8 章里他还特意剖析了两个 C++ 模块，把之前讲解的基础知识部分综合起来，以便让读者可以边学边练。

在如今追求大而全的时代，这本《Node.js：来一打 C++ 扩展》单纯地讲 Node.js 的某一个方面，而且讲得特别棒，真的很难得。

——刘琥（响马）
西祠胡同创始人，fibjs 作者

当你掌握了 Node.js 的上层使用，下一步进阶的方向就是研究 Node.js 的底层原理。本书为学习 Node.js 的实现机制打开了一扇门。书中介绍的上下文（Context）、句柄（Handle）、句柄作用域（Handle Scope）等概念直接来自源码，对于阅读 Node.js 及 V8 的源码具有极高的参考价值。

——潘旻琦（pmq20）
Node.js 技术专家，Node.js Collaborator 之一，RubyConf 讲师之一

国内 Node.js 偏向于原理的书目前只有朴灵的《深入浅出 Node.js》一本。至今 4 年过去了，Node.js 的版本已经从 v0.10 发展到 v9，中间几乎没有这样系统、有深度的书籍。

很高兴死月的新书弥补了这一遗憾。本书以 C++ 为主线，涵盖 Node.js 最核心的 libuv 和 V8，对理解 Node.js 原理有极大的好处。当然最大的好处在于，使用 C++ 编写 Node.js Addon 可以让 Node.js 有更广阔的应用空间。我们都知道 Node.js 擅长的是 I/O 密集型任务，对于 CPU 密集型运算这是极好的弥补。

特别推荐大家阅读此书，在 Node.js 应用极其广泛的今天，使用 C++ 编写 Node.js Addon 是更出彩的部分，你值得拥有。

——桑世龙（i5ting、狼叔）
StuQ 明星讲师，Node.js 技术布道者，《更了不起的 Node.js》作者

死月对 Node.js 底层机制有非常深入的了解。阅读本书，除了学习 C++ 扩展开发外，还会跟随死月了解 V8、libuv，相信大家对于 Node.js 的理解会更上一层楼。

——孙信宇（芋头）

大搜车无线架构团队负责人，前端乱炖站长

C++ 扩展其实是从外在，用 C++ 的角度去观察 Node.js 内在的形式。因为 Node.js 整个系统自身几乎就是构建在 C/C++ 之上的，所以 C++ 扩展只是在 Node.js 内部被称为内置模块（built-in module），在第三方角度（user-land）则被称为 Addon，它们在本质上其实没有区别。死月凭借他在 C/C++ 方面的深厚积累，选择从 C++ 扩展作为突破口，带大家领略 Node.js 底层的风光。在本书中，你能看到真正发挥巨大价值的 V8、libuv 亦是精彩纷呈。

死月将 C++ 扩展写得这么透彻，我是服的。

——田永强（朴灵）

高级 Node.js 技术专家，《深入浅出 Node.js》作者

开发 C++ 扩展，可以扩充 Node.js 平台的本地 API，扩充 Node.js 应用的能力。本书详细介绍了包括 libuv、V8 在内的各种必要知识，是该领域比较难得的好书。对 C++ 开发者来说，本书既可以作为入门指引，又可以作为日常开发的有益参考。

——王文睿博士（Roger Wang）

node-webkit 和 NW.js 项目创始人和维护者，英特尔软件架构师

我至今仍然清晰记得，自己手写的第一个 Node.js C++ 扩展模块在 Node.js 0.6.9 跑通的那种愉悦感。随着应用升级到 Node.js 0.8，依赖的 C++ 扩展模块无法安装编译成功，最后发现是 V8 的 API 变化导致不兼容，从此我对 C++ 扩展模块产生了抗拒心理。如今看到《Node.js: 来一打 C++ 扩展》从实现原理到 V8 基础概念的一系列介绍，让我重新对 C++ 扩展模块产生了兴趣。参考本书中的实战例子，并在 NAN 的辅助下，编写一个跨 Node.js 版本的 C++ 扩展已经不是什么困难的事情了。通过最后一章，读者可以了解到 Node.js 官方的 N-API 计划，让 C++ 扩展不仅仅能跨版本复用，还能跨操作系统（平台）复用。

——袁锋（fengmk2）

Node.js 技术专家

序一

1995 年 Brendan Eich 花了 10 天时间开发出了一门脚本语言，用来弥补 Java Applet 的不足，随后 Marc Andreessen 给它起名为 Mocha。其最初的定位是，Java 用于大型专业级开发，而 Mocha 则是给测试脚本编写人员、业余爱好者、设计师使用的。

1995 年 5 月，Mocha 被集成到了 Netscape 浏览器中，其不久后改名为 LiveScript，当年年底网景公司和 Sun 公司达成协议并获得了 Java 商标的使用权，其正式更名为 JavaScript。

有人说 Sun 公司的介入限制了 Brendan Eich 的手脚。JavaScript 除了某些语法和 Java 类似以外，骨子里却是完全不一样的东西。

也有人说正式改名为 JavaScript 才使得这门语言成为浏览器执行的唯一语言。

时至今日 JavaScript 已经不仅仅局限于为网页做特效了，而真正发展成为一门全功能的编程语言：

- 2008 年 Chrome 发布、V8 发布；
- 2009 年 Node.js 发布；
- 2010 年 NPM 发布；
- 2014 年 12 月，多位核心 Node.js 开发者不满于 Joyent 对 Node.js 的管理制度，创建了 io.js；
- 2015 年初 Node.js 基金会成立；
- 2015 年 9 月 Node.js 4.0 发布，Node.js 和 io.js 正式合并。

Node.js 4.0 版引入了 ES6 的语言特性和“长期支持版本”的发布周期。

如今 Node.js 社区已经成为最活跃的编程社区之一，而从 NPM 的包数量来看，其已经超越了 Java 的 Maven、Ruby 的 gem、PHP 的 composer。

但是 Node.js 仍有很多不足之处，Node.js 的使用者绝大部分仅仅把 Node.js 作为前端开发的辅助工具。大家把 Node.js 作为后端主力开发平台使用时，遇到 CPU 密集的场景时又不得不借助 Java 或者 Go。虽然 V8 引擎一直致力于让 JavaScript 运行得更快，但是和 Java、C++ 相比，还有不小的性能差距。

虽然关于 JavaScript 的书已经汗牛充栋，但是有关 Node.js 原理的书却屈指可数。而目前真正能够深入介绍原理的，国内的图书中也只有朴灵的《深入浅出 Node.js》了，但如今四五年过去了依然没有等到该书的第 2 版，而死月的这本书却可以弥补这一方面的不足。

所有的编程语言底层都会回归 C/C++，Node.js 的底层依赖库 V8 使用 C++ 开发，libuv 则使用 C 语言。而使用 C++ 开发 Node.js 扩展将直接把擅长 CPU 的 C++ 和擅长 I/O 的 Node.js 结合在了一起，弥补了 JavaScript 在计算密集型应用方面的不足。

我从 2015 年开始研究 V8，认识死月的时间则更早。死月不仅仅精通 C++，他也是国内的 Node.js 布道师之一。从我认识他起，他就一直在使用 Node.js。如果你想深入了解 Node.js 的原理，或者想打开 Node.js 另一个世界的大门，这本《Node.js: 来一打 C++ 扩展》值得你精读。

——迷渡 (justjavac)，Flarum 中文社区创始人，国内知名前端技术专家
2018 年 3 月 22 日于天津

序二

我跟死月相识于 GitHub，那时我们经常向 Node.js 贡献一些代码，彼此也会在微信上讨论一些技术问题。当我听说死月在写一本关于 Node.js C++ 扩展相关的图书时，激动得几乎要从床上蹦起来。因为我深知一个对 Node.js 与 V8 引擎都如此了解之人，愿意将他所知所想分享出来，这将是给予社区的一份大礼。

从我个人的角度来看，这本书非常适合这类开发者：他们对于 Node.js 的使用已经了然于胸，但却苦于没有底层开发经验，对整个 V8 虚拟机也一知半解。这时，他们可以从第 3 章开始读起。本书用了很长的篇幅介绍 JavaScript 代码究竟在虚拟机里是怎么运行的，它们又都分别对应着哪一类数据结构等。因为作者深知，只有把这些基础理解透了，则无论是开发 C++ 扩展，还是写纯 JavaScript 代码，大家都能更得心应手。

本书像是在述说着 Node.js 在 C++ 扩展这一课题中曲折而又有趣的历史进程。首先从最原始的 V8 API 时代开始。对于每个原始时代，开发者最痛苦的莫过于解决各种版本的兼容问题。之后迎来的是 NAN 时代。它解决了原始时代的接口抽象问题，接口也更丰富多样，异步接口也封装在内。最后，是还在路上的 N-API。它与 NAN 一脉相承，拥有更官方的支持和更友好的接口。

另外，我们通常在写一个 C++ 扩展时，多数情况下会跟异步打交道，这其中包含着如何非阻塞地调用底层接口，如何将异步的结果返回到 JavaScript 的回调函数中，以及如何正确地在异步封装中释放你的资源。对这些内容特别感兴趣的读者，可以打开第 6 章一睹为快。

Node.js 已快走完它的第一个 10 年，尽管被人诟病于其回调地狱、虚假繁荣、超高并发场景下的不适应性，以及低端设备上的内存等问题，但这仍旧无法阻止它前进的步伐。然而对于我们 Node.js 工程师来说，除了掌握好这门语言之外，学习如何写 C++ 扩展、了解它如

欢迎访问：电子书学习和下载网站 (<https://www.shgis.com>)

文档名称：《Node.js 来一打C++扩展》死月 著.pdf

请登录 <https://shgis.com/post/3152.html> 下载完整文档。

手机端请扫码查看：

