

前言

我很喜欢JS这门语言，感觉它和C语言一样，在C语言里，很多东西都需要自己实现，让我们可以发挥无限的创造力和想象力。在JS中，虽然很多东西在V8里已经提供，但是用JS，依然可以创造很多好玩的东西，还有好玩的写法。另外，JS应该我见过唯一的一门没有实现网络和文件功能的语言，网络和文件，是一个很重要的能力，对于程序员来说，也是很核心很基础的知识。很幸运，Node.js被创造出来了，Node.js在JS的基础上，使用V8和Libuv提供的能力，极大地拓展、丰富了JS的能力，尤其是网络和文件，这样我就不仅可以使用JS，还可以使用网络、文件等功能，这是我逐渐转向Node.js方向的原因之一，也是我开始研究Node.js源码的原因之一。虽然Node.js满足了我喜好和技术上的需求，不过一开始的时候，我并没有全身心地投入代码的研究，只是偶尔会看一下某些模块的实现，真正的开始，是为了做《Node.js是如何利用Libuv实现事件循环和异步》的分享，从那时候起，大部分业余时间和精力都投入源码的研究。我首先从Libuv开始研究，因为Libuv是Node.js的核心之一。由于曾经研究过一些Linux的源码，也一直在学习操作系统的一些原理和实现，所以在阅读Libuv的时候，算是没有遇到太大的困难，C语言函数的使用和原理，基本都可以看明白，重点在于需要把各个逻辑捋清楚。我使用的方法就是注释和画图，我个人比较喜欢写注释。虽然说代码是最好的注释，但是我还是愿意花时间用注释去把代码的背景和意义阐述一下，而且注释会让大部分人更快地能读懂代码的含义。读Libuv的时候，也穿插地读了一些JS和C++层的代码。我阅读Node.js源码的方式是，选择一个模块，垂直地从JS层分析到C++层，然后到Libuv层。读完Libuv，接下来读的是JS层的代码，JS虽然容易看懂，但是JS层的代码非常多，而且我感觉逻辑上也非常绕，所以至今，我还有很多没有细读，这个作为后续的计划。Node.js中，C++算是胶水层，很多时候，不会C++，其实也不影响Node.js源码的阅读，因为C++很多时候，只是一种透传的功能，它把JS层的请求，通过V8，传给Libuv，然后再反过来，所以C++层我是放到最后才细读。C++层我觉得是最难的，这时候，我又不得不开始读V8的源

码了，理解V8非常难，我选取的几乎是最早的版本0.1.5，然后结合8.x版本。通过早期版本，先学习V8的大概原理和一些早期实现上的细节。因为后续的版本虽然变化很大，但是更多只是功能的增强和优化，有很多核心的概念还是没有变化的，这是我选取早期版本的原因，避免一开始就陷入无穷无尽的代码中，迷失了方向，失去了动力。但是哪怕是早期的版本，有很多内容依然非常复杂，结合新版本是因为有些功能在早期版本里没有实现，这时候要明白它的原理，就只能看新版的代码，有了早期版本的经验，阅读新版的代码也有一定的好处，多少也知道了一些阅读技巧。Node.js的大部分代码都在C++和JS层，所以目前仍然是在不断地阅读这两层的代码。还是按照模块垂直分析。阅读Node.js代码，让我更了解Node.js的原理，也更了解JS。不过代码量非常大，需要源源不断的时间和精力投入。但是做技术，知其然知其所以然的感觉是非常美妙的，你靠着一门技术谋生，却对它知之甚少，这种感觉并不好。阅读源码，虽然不会为你带来直接的、迅速的收益，但是有几个好处是必然的。第一是它会决定你的高度，第二你写代码的时候，你看到的不再是一些冰冷、无生命的字符。这可能有点夸张，但是你了解了技术的原理，你在使用技术的时候，的确会有不同的体验，你的思维也会有了更多的变化。第三是提高了你的学习能力，当你对底层原理有了更多的了解和理解，你在学习其它技术的时候，就会更快地学会，比如你了解了epoll的原理，那你看Nginx、Redis、Libuv等源码的时候，关于事件驱动的逻辑，基本上很快就能看懂。很高兴有这些经历，也投入了很多时间和精力，希望以后对Node.js有更多的理解和了解，也希望在Node.js方向有更多的实践。

本书的目的

阅读Node.js源码的初衷是让自己深入理解Node.js的原理，但是我发现有很多同学对Node.js原理也非常感兴趣，因为业余时间里也一直在写一些关于Node.js源码分析的文章（基于Node.js V10和V14），所以就打算把这些内容整理成一本有体系的书，让感兴趣的同学能系统地去了解和理解Node.js的原理。不过我更希望的是，读者从书中不仅学到Node.js的知识，而且也学到如何阅读Node.js源码，可以自己独立完成源码的研究。也希望更多同学分享自己的心得。本书不是Node.js的全部，但是尽量去讲得更多，源码非常多，错综复杂，

理解上可能有不对之处，欢迎交流。因为看过Linux早期内核（0.11和1.2.13）和早期V8（0.1.5）的一些实现，文章会引用其中的一些代码，目的在于让读者可以更了解一个知识点的大致实现原理，如果读者有兴趣，可以自行阅读相关代码。

本书结构

本书共分为二十二章，讲解的代码都是基于Linux系统的。

1. 主要介绍了Node.js的组成和整体的工作原理，另外分析了Node.js启动的过程，最后介绍了服务器架构的演变和Node.js的所选取的架构。
2. 主要介绍了Node.js中的基础数据结构和通用的逻辑，在后面的章节会用到。
3. 主要介绍了Libuv的事件循环，这是Node.js的核心所在,本章具体介绍了事件循环中每个阶段的实现。
4. 主要分析了Libuv中线程池的实现，Libuv线程池对Node.js来说是非常重要的，Node.js中很多模块都需要使用线程池，包括crypto、fs、dns等。如果没有线程池，Node.js的功能将会大打折扣。同时分析了Libuv中子线程和主线程的通信机制。同样适合其它子线程和主线程通信。
5. 主要分析了Libuv中流的实现，流在Node.js源码中很多地方都用到，可以说是非常核心的概念。
6. 主要分析了Node.js中C++层的一些重要模块和通用逻辑。
7. 主要分析了Node.js的信号处理机制，信号是进程间通信的另一种方式。
8. 主要分析了Node.js的dns模块的实现，包括cares的使用和原理。
9. 主要分析了Node.js中pipe模块（Unix域）的实现和使用，Unix域是实现进程间通信的方式，它解决了没有继承的进程无法通信的问题。而且支持传递文件描述符，极大地增强了Node.js的能力。
10. 主要分析了Node.js中定时器模块的实现。定时器是定时处理任务的利器。

11. 主要分析了Node.js setImmediate和nextTick的实现。
12. 主要介绍了Node.js中文件模块的实现，文件操作是我们经常会用到的功能。
13. 主要介绍了Node.js中进程模块的实现，多进程使得Node.js可以利用多核能力。
14. 主要介绍了Node.js中线程模块的实现，多进程和多线程有类似的功能但是也有一些差异。
15. 主要介绍了Node.js中cluster模块的使用和实现原理，cluster模块封装了多进程能力，使得Node.js是可以使用多进程的服务器架构，利用了多核的能力。
16. 主要分析了Node.js中UDP的实现和相关内容。
17. 主要分析了Node.js中TCP模块的实现，TCP是Node.js的核心模块，我们常用的HTTP，HTTPS都是基于net模块。
18. 主要介绍了HTTP模块的实现以及HTTP协议的一些原理。
19. 主要分析了Node.js中各种模块加载的原理，深入理解Node.js的require函数所做的事情。
20. 主要介绍了一些拓展Node.js的方法，使用Node.js，拓展Node.js。
21. 主要介绍了JS层Stream的实现，Stream模块的逻辑很绕，大概讲解了一下。
22. 主要介绍了Node.js中event模块的实现，event模块虽然简单，但是是Node.js的核心模块。

面对的读者

本书面向有一定Node.js使用经验并对Node.js原理感兴趣的同学，因为本书是Node.js源码的角度去分析Node.js的原理，其中部分是C、C++，所以需要读者

有一定的C、C++基础，另外，有一定的操作系统、计算机网络、V8基础会更好。

阅读建议

建议首先阅读前面几种基础和通用的内容，然后再阅读单个模块的实现，最后有兴趣的话，再阅读如何拓展Node.js章节。如果你已经比较熟悉Node.js，只是对某个模块或内容比较感兴趣，则可以直接阅读某个章节。刚开始阅读Node.js源码时，选取的是V10.x的版本，后来Node.js已经更新到了V14，所以书中的代码有的是V10有的是V14的。Libuv是V1.23。可以到我的github上获取。

源码阅读建议

Node.js的源码由JS、C++、C组成。1 Libuv是C语言编写。理解Libuv除了需要了解C语法外，更多的是对操作系统和网络的理解，有些经典的书籍可以参考，比如《Unix网络编程》1,2两册，《Linux系统编程手册》上下两册，《TCP/IP权威指南》等等。还有Linux的API文档以及网上优秀的文章都可以参考一下。

2 C++主要是利用V8提供的能力对JS进行拓展，也有一部分功能使用C++实现，总的来说C++的作用更多是胶水层，利用V8作为桥梁，连接Libuv和JS。不会C++，也不完全影响源码的阅读，但是会C++会更好。阅读C++层代码，除了语法外，还需要对V8的概念和使用有一定的了解和理解。

3 JS代码相信学习Node.js的同学都没什么问题。

其它资源

个人博客 csdn <https://blog.csdn.net/THEANARKH>

知乎<https://www.zhihu.com/people/theanarkh>

github <https://github.com/theanarkh>

阅读Node.js源码时，所用到的基础知识、所作积累和记录几乎都在上面的博客中。如果你有任何问题可以到<https://github.com/theanarkh/understand-nodejs>提issue或者联系我。