

总结

感谢

首先感谢各位读者朋友，这是我的第一本小册，在内容选题、文笔方面可能会有不少的问题。希望这是一个共创的小册，能够有更多的机会和你交流和不断完善这本小册。

为什么想写这本小册呢？

我 3 年前在公司的平台部门开始做服务质量监控平台，因为我们相信“代码未动、监控先行”的理念。最开始的切入点就是使用人肉埋点的方式来进行监控信息的上报，业务方的接入成本非常高，上线前需要花半天到一天的时间来进行埋点，苦不堪言。

后来我了解到 javaagent 这样一个技术能够自动注入埋点的代码，花了大概一两个月的时间苦学字节码、ASM 相关的知识，ASM 那本官方的英文手册读了不下 5 遍，深知其中的难点。另外 APM 这一套性能监控工具需要极高的可靠性和极低的性能损耗，倒逼着我对高性能 Java 方面有了更深的研究。

因为平时喜欢折腾破解一些东西，以前都是采用直接修改类文件的方式来处理的。后来掌握了 javaagent、ASM 这些工具以后，一发不可收拾，无痛破解了 censum、gceay、fastthread、heaphero、ELK、Mybatis idea 插件等软件，觉得还是蛮有成就感的。

后来带了更大的团队，很早期的时候就引入了 Kotlin 来进行后端开发。往往对语法糖背后的知识非常匮乏，基础较差的同学经常误解一些语法糖的意思，针对这个情况，我对大量语法糖背后的实现对他们做了详细的分析，让他们在使用的时候更加清楚背后发生了什么。

一路走来，觉得越底层的知识越是有价值，所以想写一本这样的小册，让更多的人能够对 JVM 底层的细节多一些的了解。

学习路径

这本小册作用只是带领大家入门，为了让大家能更全面的了解相关的知识，我准备了一些不错的学习资料供大家选择。

Java虚拟机规范(Java SE 8版)

这本书我看了很多遍，每次看都有新的收获，强烈推荐

自己动手写Java虚拟机

这本书主要讲的是用 Go 语言来实现 Java 虚拟机，需要你有一点点的 Go 语言基础，我对着这本书敲完了，然后用 Kotlin 重写了一遍，真正了解了 class 文件解析、字节码指令运行的详细细节，也推荐大家看一看

深入理解Java虚拟机:JVM高级特性与最佳实践

这本神书不用多说，很经典，笔试面试必备

JRockit权威指南 深入理解JVM

这本书也是了解 JVM 非常不多的书籍，里面提到了很多调优相关的东西

深入嵌入式 Java 虚拟机

这本书是 2003 年出版的，现在已经绝版了，这本书讲的是嵌入式 Java 虚拟机（KVM），KVM 麻雀虽小五脏俱全，代码非常精简，也比较好读，如果想入坑 JVM 源码的，建议从这本书入手

[ASM guide \(https://asm.ow2.io/asm4-guide.pdf\)](https://asm.ow2.io/asm4-guide.pdf)

ASM 官方的文档，154 页的英文文档，当时为了掌握它，看了好几遍才开始写代码，里面事无巨细的讲解了 ASM 的使用方法，推荐

Javassist (<http://www.javassist.org/>)

这也是一个老牌的字节码改写工具，性能比 ASM 稍差，但是用起来更加简单，如果只是实现简单的功能，可以考虑

slideshare

slideshare 有很多高质量的字节码相关的 PPT，有一些有对应的 YouTube 视频，干货较多

- [JVM Bytecode for Dummies](https://www.slideshare.net/CharlesNutter/javaone-2011-jvm-bytecode-for-dummies)
(<https://www.slideshare.net/CharlesNutter/javaone-2011-jvm-bytecode-for-dummies>)
- [Bytecode manipulation with Javassist and ASM](https://www.slideshare.net/ashleypuls/bytecode-manipulation)
(<https://www.slideshare.net/ashleypuls/bytecode-manipulation>)
- [Java Bytecode Fundamentals](https://www.slideshare.net/9ll/latjug-java-bytecode-fundamentals)
(<https://www.slideshare.net/9ll/latjug-java-bytecode-fundamentals>)

zipkin

分布式链路跟踪的开源实现，基于Google Dapper的论文设计而来，由Twitter 公司开发贡献。其主要功能是聚集来自各个异构系统的实时监控数据。可以作为你设计系统的参考

APM 开源

- [Glowroot](https://glowroot.org/) (<https://glowroot.org/>) 前期入门 APM 有大量阅读这个项目的代码，代码质量还挺高的，有问题联系作者也很及时的回复
- [pinpoint](https://github.com/naver/pinpoint) (<https://github.com/naver/pinpoint>): APM 整

套解决方案，比较庞大，性能经测试稍差，如果不想自己折腾可以直接拿这个来用

最后

不学习底层知识可能不会阻碍你称为一个称职的程序员，但也许会阻碍你成为一个优秀的程序员。学习底层的知识是痛苦的，但是需要相信道阻且长，行则将至。

感谢这么有耐心看到这里的读者，希望你能给我更多的意见。我还有几本小册正在酝酿中，下本小册见。

有任何问题，欢迎加微信与我交流



挖坑的张师傅

广东 广州

