

79 | 程序员练级攻略：Java底层知识

2018-7-3 陈皓

前两篇文章分享的是系统底层方面的内容，今天我们进入高手成长篇的第二部分——Java底层知识。

Java 字节码相关

首先，Java最黑科技的玩法就是字节码编程，也就是动态修改或是动态生成Java字节码。Java的字节码相当于汇编，其中的一些细节你可以从下面的这几个教程中学习。

[Java Zone: Introduction to Java Bytecode](#)，这篇文章图文并茂地向你讲述了Java字节码的一些细节，是一篇很不错的入门文章。

[IBM DeveloperWorks: Java bytecode](#)，虽然这篇文章很老了，但是这篇文章是一篇非常好的讲Java 字节码的文章。

[Java Bytecode and JVMTI Examples](#)，这是一些使用 [JVM Tool Interface](#) 操作字节码的比较实用的例子。包括方法调用统计、静态字节码修改、Heap Tagging和Heap Walking。

当然，一般来说，我们不使用JVMTI操作字节码，而是用一些更好用的库。这里有三个库可以帮你比较容易地做这个事。

[asmtools](#) - 用于生产环境的Java .class文件开发工具。

[Byte Buddy](#) - 代码生成库：运行时创建Class文件而不需要编译器帮助。

[Jitescript](#) - 和 [BiteScript](#) 类似的字节码生成库。

就我而言，我更喜欢Byte Buddy，它在2015年还获了Oracle的 [“Duke’s Choice”](#) 大奖，其中说Byte Buddy极大地发展了Java的技术。

使用字节码编程可以玩出很多高级玩法，最高级的还是在Java程序运行时进行字节码修改和代码注入。听起来是不是一些很黑客，也很黑科技的事？是的，这个方式使用Java这门静态语言在运行时可以进行各种动态的代码修改，而且可以进行无侵入的编程。

比如，我们不需要在代码中埋点做统计或监控，可以使用这种技术把我们的监控代码直接以字节码的方式注入到别人的代码中，从而实现对实际程序运行情况进行统计和监控。如果你

看过我的《编程范式游记》，你就知道这种技术的威力了，其可以很魔法地把业务逻辑和代码控制分离开来。

要做到这个事，你还需要学习一个叫Java Agent的技术。Java Agent使用的是 [“Java Instrumentation API”](#)，其主要方法是实现一个叫 `premain()` 的方法（嗯，一个比 `main()` 函数还要超前执行的 `main` 函数），然后把你的代码编译成一个jar文件。

在JVM启动时，使用这样的命令行来引入你的jar文件：`java -javaagent:yourAwesomeAgent.jar -jar App.jar`。更为详细的文章你可以参看：[“Java Code Geeks: Java Agents”](#)，你还可以看一下这个示例项目：[jvm-monitoring-agent](#) 或是 [EntryPointKR/Agent.java](#)。如果想用ByteBuddy来玩，你可以看看这篇文章 [“通过使用Byte Buddy，便捷地创建Java Agent”](#)。如果你想学习如何用Java Agent做监控，你可以看一下这个项目 [Stage Monitor](#)。

JVM 相关

接下来讲讲Java底层知识中另一个非常重要的内容——JVM。

说起JVM，你有必要读一下JVM的规格说明书，我在这里放一个Java 8的，[The Java Virtual Machine Specification Java SE 8 Edition](#)。对于规格说明书的阅读，我认为是系统了解JVM规范的最佳文档，这个文档可以让你对于搞不清楚或是诡异的问题恍然大悟。关于中文翻译，有人在GitHub上开了个Repo - [“java-virtual-machine-specification”](#)。

另外，也推荐一下 [JVM Anatomy Park](#) JVM解剖公园，这是一个系列的文章，每篇文章都不长，但是都很精彩，带你一点一点地把JVM中的一些技术解开。

学习Java底层原理还有Java的内存模型，官方文章是 [JSR 133](#)。还有马里兰大学的威廉·皮尤（William Pugh）教授收集的和Java内存模型相关的文献 - [The Java Memory Model](#)，你可以前往浏览。

对于内存方面，道格·利（Doug Lea）有两篇文章也是很有价值的。

[The JSR-133 Cookbook for Compiler Writers](#)，解释了怎样实现Java内存模型，特别是在考虑到多处理器（或多核）系统的情况下，多线程和读写屏障的实现。

[Using JDK 9 Memory Order Modes](#)，讲了怎样通过VarHandle来使用plain、opaque、release/acquire和volatile四种共享内存的访问模式，并剖析了底层的原理。

垃圾回收机制也是需要好好学习的，在这里推荐一本书《[The Garbage Collection Handbook](#)》，在豆瓣上的得分居然是9.9（当然，评价人数不多）。这本书非常全面地介绍了垃圾收集的原理、设计和算法。但是这本书也是相当难啃的。中文翻译《[垃圾回收算法手册](#)》翻译得很一般，有人说翻译得很烂。所以，如果可能，还是读英文版的。如果你对从事垃圾回收相关的工作有兴趣，那么你需要好好看一下这本书。

当然，更多的人可能只需要知道怎么调优垃圾回收，那么推荐读读[Garbage Collection Tuning Guide](#)，它是Hotspot Java虚拟机的垃圾回收调优指南，对你很有帮助。

[Quick Tips for Fast Code on the JVM](#)也是一篇很不错的文章，里面有写出更快的Java代码的几个小提示，值得一读。

小结

好了，总结一下今天学到的内容。Java最黑科技的玩法就是字节码编程，也就是动态修改或是动态生成Java字节码。Java的字节码相当于汇编，学习其中的细节很有意思，为此我精心挑选了3篇文章，供你学习。我们一般不使用JVMTI操作字节码，而是用一些更好用的库，如asmtools、Byte Buddy和BiteScript等。使用字节码编程可以玩出很多高级玩法，其中最高级的玩法是在Java程序运行时进行字节码修改和代码注入。同时，我介绍了Java Agent技术，帮助你更好地实现这种高级玩法。

JVM也是学习Java过程中非常重要的一部分内容。我推荐阅读一下JVM的规格说明书，我认为，它是系统了解JVM规范的最佳文档，可以让你对于搞不清楚或是诡异的问题恍然大悟。同时推荐了[JVM Anatomy Park](#)系列文章，也非常值得一读。

随后介绍的是Java的内存模型和垃圾回收机制，尤其给出了如何调优垃圾回收方面的资料。这些内容都很底层，但也都很重要。对于想成为高手的你来说，还是有必要花时间来啃一啃的。

下篇文章是数据库方面的内容，我们将探讨各种类型的数据库，非常有趣。敬请期待。

下面是《程序员练级攻略》系列文章的目录。

[开篇词](#)

入门篇

[零基础启蒙](#)

[正式入门](#)

修养篇

[程序员修养](#)

专业基础篇

[编程语言](#)

[理论学科](#)

[系统知识](#)

软件设计篇

[软件设计](#)

高手成长篇

[Linux系统、内存和网络（系统底层知识）](#)

[异步I/O模型和Lock-Free编程（系统底层知识）](#)

[Java底层知识](#)

[数据库](#)

[分布式架构入门（分布式架构）](#)

[分布式架构经典图书和论文（分布式架构）](#)

[分布式架构工程设计\(分布式架构\)](#)

[微服务](#)

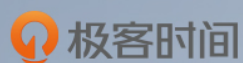
[容器化和自动化运维](#)

[机器学习和人工智能](#)

[前端基础和底层原理（前端方向）](#)

[前端性能优化和框架（前端方向）](#)

[UI/UX设计（前端方向）](#)



左耳朵耗子

全年独家专栏《左耳听风》

20000 名程序员的练级攻略

陈皓

资深技术专家
骨灰级程序员



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 16



javaee

1531930466

有同学认为这种介绍文章没用，一大堆引用。我觉得吧，这篇文章价值很大。如果只是要写一篇关于字节码或JVM的详细使用，那很多书籍或网站可能有了，反而不值得写。耗子叔这系列文章，在我看来很有大局观，自顶向下梳理了各种技术脉络。授人以渔其实更重要，好的老师是给你指出明路，让你少走弯路，而不是给你讲解几道题。不过这也许要工作几年后才能更深刻的体会到吧，这些总结的资源是一笔财富，至少不用走弯路，可以有选择性的去挑选适合你的认为有价值有兴趣的内容去学习。

作者回复 谢谢理解



怪盗キッド

1530598209

Hi，我利用ASM写了一个简单、快速且无侵入的Java方法监控工具MyPerf4J，通过JavaAgent方式对Java方法进行字节码注入，可以统计出方法的执行性能指标，包括RPS、Avg、TP50、TP90、TP99、TP999等，Github地址：
<https://github.com/ThinkpadNC5/MyPerf4J>

作者回复 🍷那些统计，你用到了蓄水池算法了吗？



吃桔子的攻城狮

1533289008

第一次评论。这个专栏看了这么久，第一次觉得有必要说几句，这种风格的专栏真的非常赞。看到有些同学说链接太多缺少耗子哥自己的东西，我想说这个系列随便一篇文章拿出来，如果纯自己写都能单独写成一个系列甚至一本书。这就像重复造轮子，明明已经有了优秀的文献资料，为什么要重新写一套？相反，能把这些优质资源做整合，串联，归纳，提供学习的路径和思路才是受益无穷的！

有同学说这些都是网上可以找到的，那不妨请想一下，如果只给你本系列某篇文章的题目，凭自己你真的可以找得到这些资料吗？不会陷入现在互相抄来抄去的劣质博客里迷惘困惑，百思不得其解吗？

支持这种风格，我认为订阅专栏的钱花的很超值！

作者回复 谢谢



superryanguo

1530625905

java有必要单独抽一篇来讲吗？而且都是引用



ruby

1530599521

皓哥，后面有大数据文章，怎么学spark.hadoop等吗？



鹤鸣

1530636339

C++程序员问个问题:怎样对一个已有的基于spring的项目优化性能?目前我这边首先要做的

事情是测试出性能瓶颈，但是目前为止我还在使用那种很土的办法，纯体力活的那种，我觉得这个路子不大对头。



Rolin

1530586457

Android 程序猿好好学！



墨梵

1530659627

耗叔有专门讲c++对象模型方面和内存分配的篇章嘛？



葛阳阳

1530578237

赞



庞雨青_Alice

1559392882

非常感谢左耳皓哥分享。

读精品的技术文章真是一件很爽快的事情。我个人是喜欢刨根究底的类型，之前在学习编程的过程中一直都没能找到多少成就感。现在看来一是没有找到最精品的文章，二是没有找到适合自己的方式。

这几天耐着性子慢慢读英文的文章，自己的英语能力也有所提高。

感谢皓哥🙏



godtrue

1546993394

恩，认同这种风格的文章，感觉很导师风格，指明路线自己去玩，每个人的收获取决于每个人的付出。



Tony

1531406635

耗子哥，我记得你以前写过一篇关于java进阶的比较全面的文章，怎么找不到了？



付彬

1530666549

好



LenX

1530621186

动态生成的代码 debug 起来挺费劲的，尤其是用 IDE。耗叔有什么工具或方法推荐吗？



怪盗キッド

1530619144

没有使用蓄水池算法，我是通过数组+Map的方式存储数据：

对于数组来说，下标代表方法的响应时间，下标对应的值代表该响应时间的数量；

对于Map来说，Key代表方法的响应时间，Value代表该响应时间的数量；（ Accurate模式 ）



ZYCHD(子玉)

1530584177

读耗子书的文章总给人带来新鲜的感觉。视野很开阔。前后穿插纵横千里！