

# 程序员练级攻略（2018）：Java底层知识

2018-07-03 陈皓



程序员练级攻略（2018）：Java底层知识

朗读人：柴巍 07'25" | 3.40M

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 JVM TI Examples](#)，这是一些使用 [JVM Tool Interface](#) 操作字节码的比較实用的例子。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（2018）》系列文章的目录（持续更新中）。

- [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](#)
  - .....

# 左耳朵耗子

## 全年独家专栏《左耳听风》

拼团价 **¥199** / 3人成团  
原价: 299

陈皓  
资深技术专家  
骨灰级程序员



扫码拼团



版权归极客邦科技所有，未经许可不得转载

精选留言



葛阳阳

赞

2018-07-03

👍 0



React

前端前端

期待耗子叔对前端学习和发展的分析！

2018-07-03

👍 0



75819

我去，学完这个大篇章系列估计要35了，又要开始担心失业了。

2018-07-03

👍 0