01 开篇词-多线程为什么是你必需要掌握的知识

更新时间：2019-09-09 19:19:41

> 当你做成功一件事，千万不要等待着享受荣誉，应该再做那些需要的事。
>
> —— 巴斯德

你好，很荣幸你打开本专栏，看到我的这些文字。相信每一位开发者，都应该对多线程开发有所了解。作为程序员，如果不知道多线程，都不好意思和同行打招呼。但是对多线程有深入了解的开发人员却并不多。有着丰富经验的更是凤毛麟角。多线程开发其实在软件开发领域有着很重要的地位，绝大多数框架或者软件工具都使用了多线程。比如耳熟能详的Tomcat、Kafka、Akka等等。JVM的垃圾回收也是由单独的线程执行。正是有着如此多的优秀框架，才使得我们编写绝大多数业务代码时无需考虑使用多线程。

但这是否可以认为多线程开发对于普通开发者并不重要？其实恰恰相反。多线程开发在当今软件领域变得越来越重要，是每个开发人员不但要了解，而且要彻底掌握的开发知识。多线程开发重要性从不同角度来看，有如下原因：

提升代码性能

现今是大数据的时代。随着数据分析的需要、AI学习的需要、存储设备的廉价，越来越多的数据被采集下来，通过程序进行处理。面对海量的数据，如何榨取CPU的运算能力，提升运算效率，开发人员需要重点考虑。而CPU的发展从提升主频转为多核，使得多线程开发有了更大的用武之地。

另外在微服务大行其道的时代，恰当使用多线程，也能令你的程序性能大大提升。把没有依赖的API调用以多线程的方式发送出去，并行处理拿到结果后再做进一步计算。执行比串行提高了几倍，而且可以充分发挥出微服务分布式的计算优势。

更优秀的软件设计和架构

相信做过Java开发的攻城狮都熟知面向对象。面向对象的出现，使得我们设计软件更加贴近于真实世界，代码封装得更为合理。没错，其实软件世界即现实世界。设计和开发无形的软件，都是参考现实世界中有形的物体。现实世界可以认为是"多线程"的世界。每一个人是一个线程，每一台运转的机器是一个线程。掌握了多线程开发，能让你设计出更加贴近真实世界的软件，而不是凭空做出设计。其实优秀的软件设计都是如此。比如Java中NIO的设计，和快递投放极为相似。

更好的工作机会
这个原因就比较现实了。目前绝大多数技术面试都会问到多线程的相关知识，尤其是互联网大厂。通过面试多线程知识，除了可以看出你的技术深度，更重要可以看出你的学习能力。你可以没用过多线程，但是如果在短时间内能够把多线程深入掌握，说明候选人的学习能力、领悟能力都很高。

专栏特色

关于多线程的学习其实还是有一定难度，一是多线程推翻了单线程开发的一些认知，二是理解多线程中的问题，需要学习部分JVM底层知识。这也是大多数开发人员对多线程仅停留在简单了解层面的原因。本专栏目标是消除掉多线程学习上的拦路虎，让每个人都能轻轻松松学习，并且最终能深入理解底层原理。为了达到这个目标，
专栏编写会有如下特点：

多以每个人都熟知的现实世界示例做类比，彻底消除掉理解上的鸿沟。

配以生动有趣的插图，让知识的学习变得更为轻松。在枯燥的学习中也能开心一笑。相信有小慕陪伴的学习也会充满了乐趣：

由浅入深，从问题提出，到代码示例。从解决方案到底层原理。彻底搞懂多线程开发的知识点。

结合实践进行讲解。不只是停留在“懂”，而是要再进一步到“用”。技术学习的价值要体现在使用上。

适合人群

多线程是 Java 的高级特性，所以本专栏并不适合没有任何开发基础的人员。不过只要你有一到两年的开发经验，阅读本专栏不会有任何障碍。本专栏将会深入源代码层面讲解多线程相关知识，所以也适合有着丰富开发经验，但想对多线程有更为深入了解的高级开发者。

本专栏适合的群体如下：

1、初入职场，有简单的Java开发基础，想要从基础开始学习Java多线程开发。

2、有3-5年甚至更多开发经验，想要深入了解Java多线程开发。

3、学习过多线程，想要了解在实际项目中如何应用。

4、准备跳槽面试，想要全方位，较为深入的学习多线程。

专栏设置

为了达到由浅入深，从“懂”到“用”的目标。专栏设置如下四大模块，每个模块又由多个章节所组成。

Java多线程开发基础

本模块为多线程开发入门。主要讲解了多线程的基本概念以及在Java中如何进行基本的多线程开发。深入讲解了Thread类的API以及多线程的启动过程。

Java并发问题及解决方法

多线程并发会遇到单线程程序中看似古怪的各种问题。多线程开发的难点也在于此。本模块会深入分析这些问题产生的原因，并给出解决方法。在源代码层面深入分析atomic、volatile、synchronized等多线程关键的同步技术。

Java并发工具

本模块会对Java提供的主要并发工具进行讲解。包括线程池、Lock、CountDownLatch、CyclicBarrier等。此外还有并发容器的深入分析，如ConcurrentHashMap等。这些都是实际开发中常用的并发技术，也是多线程面试的重点知识点。

多线程程序设计模式

学习了以上这么多的多线程知识。最后我们来看看实际开发工作中如何设计多线程程序。通过合理的多线程程序设计，才能真正让多线程体现出它的优势。

本模块涵盖了Future模式、Master/Slave模式等多线程程序设计的最佳实践。通过本章学习，结合前面所学的基本知识，你已经可以大胆实践多线程开发了！

专栏学习建议

在专栏的学习上，从基础概念开始，不要图快直奔使用。概念的理解和学习自然枯燥，但却是根基所在。只有深入理解理论知识，才能做到举一反三，遇到问题从容应对。

建议在学习的过程中，做好学习笔记，把学到的知识以自己的语言记录下来。这也是我学习的一个经验，做记录的过程也是一种输出，会让你的知识掌握更清晰、牢固。另外有问题一定要多交流，无论是和身边的朋友，还是和我。学习就是不断提问和解答的过程。没有问题说明你没有思考，没有思考的学习效果也不会很好。

最后我很期待和你一起开启Java多线程的学习。相信一起学习的每个人在专栏结束之时，都可以自信地说我彻底掌握了Java多线程开发！

}

02 绝对不仅仅是为了面试—我们为什么需要学习多线程