

第 34 章 并发的总结对话

教授：那么，你现在头疼吗？

学生：（吃了两颗治头疼的药）有点儿。很难想象线程之间有这多种相互穿插的方式。

教授：的确如此。就这么几行代码，并发执行的时候，就变得难以理解，很让人意外啊。

学生：我也是。想到自己是计算机专业的学生，却不能理解这几行代码，有点尴尬。

教授：不用这么难过。你可以看看最早的并发算法的论文，有很多都是有问题的。这些作者通常都是专家教授呢。

学生：（吸气）教授也会……嗯……出错？

教授：是的。但是不要告诉别人——这是我们之间的秘密。

学生：并发程序难以理解，又很难正确实现，我们怎么才能写出正确的并发程序呢？

教授：这确实是个问题。我这里有一些简单的建议。首先，尽可能简单。避免复杂的线程交互，使用已被证实的线程交互方式。

学生：比如锁，或者生产者—消费者队列？

教授：对！你们也已经学过了这些常见的范式，应该能够找出解决方案。还有一点，只在需要的时候才并发，尽可能不用它。过早地优化是最糟糕的。

学生：我明白了——为什么在不需要时添加线程呢？

教授：是的。如果确实需要并行，那么应该采用一些简单的形式。使用 Map-Reduce 来写并行的数据分析代码，就是一个很好的例子，不需要我们考虑锁、条件变量和其他复杂的事情。

学生：Map-Reduce，听起来很不错呀——我得自己去了解一下。

教授：是这样的。我们学习到的知识仅仅是冰山一角，还需要大量的阅读学习，以及大量的编码练习。正如 Gladwell 在《异类》这本书中提到的，1 万小时的锤炼，才能成为专家。课程上的时间是远远不够的。

学生：听完感觉很振奋人心。我要去干活了！该写一些并发代码了……