



下载APP



## 结束语 | 再聊Go并发编程的价值和精进之路

2020-11-27 晁岳攀

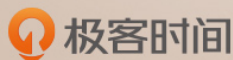
Go 并发编程实战课

[进入课程 >](#)**晁岳攀 (鸟窝)**

前微博技术专家、知名微服务框架 rpcx 的作者

你好, 我是晁岳攀 (鸟窝)。

并发原语, 初识时简单, 深交时复杂, 熟识后又觉简单, 这也是一种学习的乐趣。

**讲述: 安晓辉**

时长 10:00 大小 9.17M



你好, 我是鸟窝。很高兴和你一起度过了一个多月的时间, 到了和你说再见的时候了。

在过去的这些年里, 我一直在研究 Go 并发编程, 时间越久, 越觉得, 掌握 Go 并发原语是一件很有意思的事情。

很多刚开始学习并发原语的同学给我留言说: “使用 Go 写并发程序很容易啊, 为啥要学这么多并发原语呢?”

如果你也有这样的疑问, 我的答案就是在这节课的封面图中写的那句话: “**并发原语, 初识时简单, 深交时复杂, 熟识时又觉简单。**” 这是我的真实体会。



如果你处于刚开始接触并发原语的阶段，你可能会觉得：“这挺好理解的呀，我一看就会了。”但是随着学习的不断深入，你会看到各种复杂的用法，各种潜在的坑，这些东西打破了初印象，你会陷入到“千头万绪”的境地。只要你不畏困难，持续学习，最后你就可以轻松地使用这些并发原语了。如果说最初的“简单”是“初生牛犊不怕虎”的“简单”，那么“熟识”后的“简单”，就是“拨云见雾”的“简单”。这也是，我在这门课里想要带你达到的状态。

**总之，使用 Go 写并发程序很容易，使用 Go 写好并发程序很不容易。**

遗憾的是，很多人都没有意识并发编程的复杂性，甚至还没有意识到，并发编程错误带来的严重后果。所以，我想跟你分享关于并发编程 Bug 的两个小故事。

第一个故事，是我刚刚看到的澳大利亚交易所（ASX）的新系统上线后崩溃的故事。

11 月 16 日中午，ASX 发布声明说，当天将休市，会在次日的正常时间重新开放。官方给出的关闭原因是“局限于单个交易指令中交易多种证券（组合交易）的软件问题，导致了市场数据不准确。”

虽然我并没有看到这个 Bug 的细节，但是，从官方提供的关闭原因中，我们可以简单地推断出是“单个指令中交易多种证券的问题”，大概率是一个并发问题的 Bug。虽然经过一天的排查和修复，第二天这个交易所就恢复上线了。但是，耽误一天的时间，损失也是非常大的。

类似的软件 Bug，尤其是并发问题的 Bug，即使经过很长时间的测试，也不一定能被触发和发现。可是一旦出现，就可能是一个一级的 Bug。

如果看完这个故事，你还没有意识到并发编程的复杂性和并发问题的危害，我再给你讲一个故事。

1997 年 7 月，NASA 的 Mars Pathfinder（火星探路者）在降落火星表面后不久，就因并发软件中的一个缺陷受到了威胁。这是在飞行前的测试中发现的，但因为它只发生在某些没有预料到的重载条件下，所以被给予了较低的优先级。

但是，飞船开始采集气象数据的时候，它所使用的 vxWorks 操作系统就出现了问题，不断地重启。这是经典的优先级反转的并发 Bug。

幸好工程师上传了一小段 C 语言程序给飞船，在运行的时候，将优先级继承的互斥标志从 false 改成了 true，才成功地解决了这个 Bug。

这次人为的忽视，险些酿成惨剧。所以，学好并发编程，是我们的重要责任。

那么，该怎么在编写 Go 程序时，避免并发编程的 Bug 呢？在 [开篇词](#)里，我讲到了“两大主线”，现在学完了所有内容之后，你会发现，其实可以抽象成“三部曲”：

1. 全面地掌握 Go 并发编程的知识，不遗漏任何的知识点；
2. 熟悉每一个并发原语的功能和实现，在面对并发场景时，能够高效地选出最适合的并发原语；
3. 多看看别人踩的坑，避免自己再掉进相同的坑里。

在前面的课程中，我讲的所有内容，都是为了帮助你轻松地完成这三个目标。在课程的最后，我还想再给你多交代几句。

学完这门课，并不代表你已经掌握了 Go 并发编程的知识。Go 并发编程的知识广、内容深，现在你再回顾前面的知识，可能已经遗忘了一大半了。即使你现在记得很清楚，等过一段时间，再提到这些知识点，你也可能答不上来。

所以，学完这门课并不是一件一劳永逸的事情，你要在空闲的时候多复习下前面的内容。怎么复习呢？你可能也注意到了，每讲完一个并发原语，课程里都有一张知识导图，这些图既可以帮助你梳理知识主线，也可以帮助你高效地复习。

你可以对照着图中的核心要点，去回顾我们学习的重要内容，如果感觉有些地方比较陌生了，就及时回去复习下。另外，你也可以做一些摘录，并且写上你自己的收获和思考。**学习过不等于能输出**，你一定要记住这句话。

另外，这门课的核心是讲 Go 并发原语的知识，并没有涉及到 Go 并发模型和调度的事情。这不是说，我认为这部分内容不重要，而是很多大牛已经把这些内容写得很清楚、很明白了。如果你对这方面的知识还不太熟悉，可以搜索关键字“golang gpm

schedule”，你会看到很多资料。你读几篇，就明白了。如果要推荐的话，我建议你重点读一读欧长坤的 [🔗 《Go 语言原本》的 并发调度](#)，这一篇的逻辑非常顺畅，能看出非常多的经验。

当然，我还想再给你推荐一些补充资料，如果你还有余力，可以再扩展一下知识面。

首先是一本书，名字是“Concurrency in Go”。这是第一本全面介绍 Go 并发编程的图书。书中介绍了并发编程的背景知识、常见的原语和并发模式。我印象最深的，就是书里对 Channel 的介绍，比如 Channel 是粘合 goroutine 的胶水，而 select 是粘合 Channel 的胶水。这样形象的说法，可以帮助你快速地学到精髓。

除此之外，Go 官方博客列出的一些技术分享，比如 [🔗 Go Concurrency Patterns](#)、[🔗 Advanced Go Concurrency Patterns](#)，都是不错的阅读材料，我建议你好好读一读。

好了，关于结课后的学习方法，我就说到这里。在这节课的最后，我特别想再和你分享我自己的两个心得。

## 第一，开放的心态，可以拓展你的人生边界。

我始终认为，一个人衰老的标志，不是指他的容貌经历了太多岁月的刻画，而是他的内心封闭了，不再接收新的知识、新的事物。

在一些技术交流会上，我听到一些开发者说，Go 并发编程很简单，有什么可学的？遇到这种不是技术讨论的话题，我一般只会说：“你说得对。”

我当然认同我们应该把核心精力用在眼下有价值的事情上，在自己擅长的领域里深耕，但是我更相信，开放心会让你的人生与众不同。如果你碰见了新技术的发展，即使不需要深入地学习，也要尽量花时间去了解一下，也许这些新的东西，就是你人生的转折点。

我之前就是一直使用 Java、Scala，后来才开始了解 Go，但是，很显然，Go 给我的人生带来了不一样的东西。如果不是深入研究 Go，我就没有机会开设这么一门课了。

## 第二，无数人想要你的注意力，但只有你能决定你把它放在哪里。

我们总说这个时代是信息爆炸的时代，其实，信息爆炸就意味着千万的信息发送者想要占用你的注意力。你一定要保持谨慎，不要毫无感知地把你的时间扔给无价值、无意义的信息。

如果说上一条是让你延伸注意力的触角，那么这一条，就是让你收缩注意力的触角，但这两者并不矛盾，因为侧重点不同。“延伸”还是“收缩”，取决于你自己想要拥有的人生的样子，只有你能决定。我能做的，就是提醒你，要开放，也要谨慎。

虽然很舍不得，但还是要跟你说再见了。在课程的最后，我给你准备一份结课问卷，希望你花 1 分钟时间，点击下面的图片填写一下。如果你的建议被采纳，我将会给你赠送一个护腕垫或者价值 99 元的课程阅码。期待你的畅所欲言。



## 晁岳攀(鸟窝)

前微博技术专家、知名微服务框架 rpcx 的作者

感谢一起走过的这段时间，非常想听听你对我和这门课程的反馈与建议。在 12 月 11 日前提交问卷，将有机会获得



原创 | 手绘护腕垫

价值 **¥69**

或



极客时间课程阅码

价值 **¥99**

填写问卷 

提建议



# Go并发编程实战课

## 鸟窝带你攻克并发编程难题

晁岳攀 (鸟窝)

前微博技术专家

知名微服务框架 rpcx 的作者



新版升级：点击「 请朋友读」，20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。

上一篇 20 | 在分布式环境中，队列、栅栏和STM该如何实现？

### 精选留言 (26)


 写留言



昵称

2020-11-27

时间飞逝，感触颇深，受益匪浅，意犹未尽，期待作者再次回归！！！！

作者回复: 



2



郑鹏

2020-11-27

完结散花！！！！！！

展开 

作者回复: 

**keep**

2020-11-27

感谢老师，内容质量超预期~

展开 ∨

作者回复: 感谢，只要你感觉有收获我就满足了

**Fcou**

2020-12-04

完结撒花，一氧化二氢制造工厂代码例子真是激发思考，以后还是要多读源码，多写。期待老师更多实战分享

展开 ∨

作者回复: 坚持下来的同学都很优秀，加油！

**Vincent**

2020-12-03

看完之后，自己的头发掉了好几根；真的要，多读几遍，才能融入到自己的思维里  
感谢鸟窝，期待更多的干货

展开 ∨

**wangkaiyuan**

2020-12-03

感谢老师

展开 ∨

**网管**

2020-12-01

第一遍看完了，准备自己总结下每节的重点，做些笔记。





2020-11-30

展开 ∨

◀ ▶

💬 👍



2020-11-30

展开 ∨

A horizontal bar containing navigation icons: a left arrow, a right arrow, a speech bubble icon, and a thumbs up icon.



2020-11-30

展开 ∨

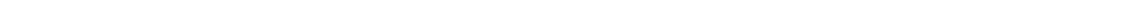
---






2020-11-30

展开 ∨





2020-11-29

展开 ∨

[!\[\]\(aff7c69c44a5e015f18c35867ef3f5c3\_img.jpg\)](#)
[!\[\]\(0008e7ee33c76c8da181b6f52bf57cf7\_img.jpg\)](#)



**Yimmy**

2020-11-28

"无数人想要你的注意力，但只有你能决定你把它放在哪里",很好的提醒

**Linuxer**

2020-11-28

感谢!我处于觉得有些部分容易，有些部分难，这种状态，还需要多刷几遍

**Yabo**

2020-11-28

还没看完，刚看了前几课，有些内容要反复看，受益匪浅，感谢老师的分享。

**张海鹏**

2020-11-28

这门课我收获很大，非常感谢，很值

展开 ▾

**大欧**

2020-11-28

学完打卡，收获很多，超值

展开 ▾

**青生先森**

2020-11-27

老师辛苦了，收益良多，感谢。

展开 ▾

作者回复: 谢谢, 💎💎💎💎💎

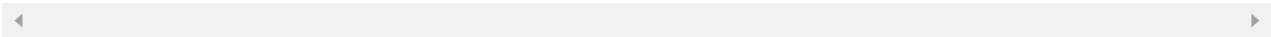
**橙子888**

2020-11-27

最后一堂课打卡，感谢老师！

展开

作者回复: 每次都看到你卡，感谢你和其他坚持下来的朋友们



伟伟

2020-11-27

受益匪浅的一门课，基本每节课都值得反复研读

展开

