编程语言漫谈

2018-01-05 朱赟

编程语言漫谈
朱赟
- 00:05 / 09:27

编程语言是一个已经被谈到耳朵发烫的话题，很多工程师都聊过。似乎无论怎么写，要么落入老生常谈的俗套，要么就是一堆理论上正确，但是对学习和理解编程语言并无多大益处的内容。

我跟池老师说：这篇可说的内容太多，反而不知道从何说起。池老师建议我：“任思维流动，想到哪写到哪。比如你可以点评一下自己最擅长的语言特性，比如新人学编程如何做到触类旁通，最后一通百通。”回想一下，这两个思路我似乎都用过了，以前写过点评 Ruby 和 Java 的文章，也写过自己的编程之路，今天说点平时不说的内容，可能不讨喜，但至少可以提供一些思考。

我在莱斯大学读硕士期间，课题就是程序语言和编译器设计。翻看当时用到的教科书和论文，需要我学习和掌握的内容大概都是这样的：

1. 一些关于图灵机和状态机的计算理论
2. 类型和类型系统，类型系统的证明和推断
3. 函数中的递归、迭代及其实现原理
4. 关于 Lambda 的演算和模型
5. 命令式程序语言、函数式程序语言、逻辑式程序语言、以及面向对象程序语言的本质区别
6. 语法器、词法器、编译器、解释器的原理及实现

所以，很多人刚开始接触编程语言，第一反应是怎么用、好不好用。而我看到一门语言，第一反应是这门语言是怎么实现的，类型系统是什么，计算能力的边界在哪等等。最早接触的几种语言，比如 C、OCaml、Schema、Python、Java 等，都被我拿着“手术刀”解剖过。那个时候，我想到最多的成语是庖丁解牛和洞若观火。

我的硕士毕业设计是用 OCaml 去实现一个机器人仿真和控制的语言，有点像 Matlab 里的机械模拟库。研究的目的，说白了就是不断用 0/1 这样的离散数学和语言，去逼近物理世界中很常见的连续函数（如导数和积分），而挑战的就是有限的计算资源和时间。

后来到了博士阶段，课题变成了生物信息学，那个时候常常要做的是生物学中的大数据处理和建模。由于业界很多数据科学相关的库都是 Python 实现的，于是我使用到了大量的 Python 编程。

不过在处理海量数据的时候，Python 的性能就成为一个很大的瓶颈。经常遇到的情况是，一个脚本或者一个函数库，其复杂度在一个临界点后是指数增长的，稍微大一点的数据量，动不动就要跑几天，或者干脆跑不出来。

于是我又开始学习并使用一种叫做 Cython 的语言。这种语言在语法方面算是 Python 和 C 的混合体，其编译器可以将 Cython 代码转化为 C 并编译成性能很好的可执行代码。那时候除了建模需要的数据清洗和数据模型的训练，另一个挑战就是写出高性能的代码。

博士毕业后，我没有选择留在学术界，而是进入了互联网公司工作。

Square 和 Airbnb 都是以 Ruby 和 Java 为主要编程语言的公司。这个时候，语言上面临的挑战已经不再是其计算能力或者性能，而是如何在工程中用适当的语言搭建出一个方便协作、性能过得去、可读性好、模块化好、可重用、易扩展的代码库。

很多时候工程师们争论的问题，不再是对和错、是与否的问题，而是每个人的观点应用到相关的场景中的时候带来的优劣比较。也就是说，是不是把合适的技术用到了正确的场景中。平衡是我们在这个阶段要着重考虑的，这种平衡有时候是时间复杂度，有时候是空间复杂度。

最近几年我主要使用 Ruby 和 Java 编程，这两门语言的优缺点就不在这里说了，网上有很多类似的观点。关于 Java，后面我会写一篇“Java开发的中常见问题”，下面我会来简单聊聊自己对软件工程和编程语言的一些看法，有的是点评别人的观点，有的是我自己的观点，分享给你，希望可以给你带来一些启发。

1 初学者不要纠结“先学哪种语言”，这种时间花的很不值得，还不如随便挑一个语言，跳进去游几圈试试。对于工程师来说，学习第一门编程语言只是万里长征的第一步，只要你还在这个领域，就不可能只学一种语言，只会一种语言的工程师根本就不能称之为工程师。

2 如果你不能用一种编程语言的基本特性写出好代码，那换成另外一种语言也无济于事，你会写出同样差的代码。比如，你的 Java 代码写得很糟糕，那么换成 Go、Ruby，你的代码也会一样糟糕，甚至更差。

所以，基本掌握了一门语言的功能和语法特性之后，要去做实践和练习，能写生产代码了，再回过头来去看编程语言的本质，了解这门编程语言的设计原理，能力边界和高级功能，这样有助于你更快更好地掌握其他编程语言。

3 很多人觉得不要用脚本语言入门，我觉得不一定，尤其现在就着人工智能浪潮搞机器学习的人，用 Python 入门就很好。另外，脚本语言在面试中绝对占优势。平时工作中我对 Ruby、Python、C++、和 Java 的熟练程度差不多，但是面试中使用 Ruby 或者 Python 答题，写代码的时间估计是那两者的一半。

如果让我推荐一门脚本语言，那就是 Python，关于 Python的历史和语言特性，可以参考池老师之前写过的“人生苦短，我用 Python”一文。

https://mp.weixin.qq.com/s/sSl2PHiuQWmNuQMgoL4qcw

4 后端工程师要熟练掌握一门前端语言，前端工程师也要熟练掌握一门后端语言。倒不是为了提倡全栈或多个能力储备，而是两者的编程思维模式很不一样。知己知彼，在架构设计和解决具体问题时，才会有更精确的判断。

另外，现在大前端的概念也比较流行，也就是大前端工程师能够同时掌握 Web 编程语言、iOS 和 Android 编程语言，原生技术（iOS 和 Android）和 Web 的配合会越来越紧密。

5 SQL 是一门非常非常重要并且应该熟练掌握的语言（虽然它不能被称为程序语言）。我在这里用了两个非常，因为很多工程师有些过于轻视 SQL 了，并为此付出了惨重的代价。

如果你平时的编程工作涉及到业务功能，而不是纯粹的技术架构，一定会使用到数据库。SQL 就是数据的语言，通过它，你可以和数据建立连接和沟通。

如果你的数据访问模式写得很差，轻则代码性能一塌糊涂，重则引发 Bug，而涉及数据的问题，Bug 等级都比较高，后果可能很严重。

关于SQL，可以参考我之前的专栏文章“每个工程师都应该了解的：数据库知识”。

6 无论使用什么语言，工程师都应该能够基于这种语言搭建测试框架，写好测试代码和写业务代码一样重要，甚至更重要。工作后你会发现，可能有时候我们只花五分钟写了一个程序，而为其写一个差不多能够覆盖所有功能路径的测试用例集却花了一个小时。

关于测试，可以参考我之前的专栏文章“每个工程师都应该了解的：A/B 测试”。

7 最后的，也是最重要的是：在任何时候都要用并发的、分布式的思维去看待你的程序。因为竞争条件或者并发中的不确定因素（比如调用顺序）导致的 Bug，仅仅理解语言的基本特性，根本不能解释。

每种语言都有自己的并发编程模式（比如 Go 的 Goroutine，Java 的 ForkJoinPool，Swift 的 Swift Grand Central Dispatch 等）。学习每一种语言，都应该去深入了解它的并发模型，在这个多核的时代，不懂并发的程序员不可能是个好工程师。

今天我以漫谈的形式和你聊了聊上学和工作过程中学习和使用编程语言的经历，然后我讨论了一些对软件工程和编程语言的看法，算是经验之谈，希望对想要学习或正在使用编程语言的你有所帮助。

编程语言，你看它是山，它就是山；看它是水，它就是水。你可以把它当做一门简单的编程语言，有语法，有特性，也有优缺点，但这样的语言也可以复杂到去实现和解释各种计算模型和理论。一门编程语言到底能做什么，完全和工程师怎么去用，在什么场景中用息息相关。

最后留个讨论题，什么是真正的全栈工程师，他们需要掌握更多的编程语言吗？期待你的回复，我们下期再见。

Yezhiwei

2018-01-06

想看看安姐的代码

刘剑

2018-01-05

朱老师太牛了，你有做全栈的天赋。
我觉得朱老师应该问你是愿意将来成为一个全栈工程师？还是一个技术专家？

这两种的思维是完全不一样的。简单的说全栈是广度，技术专家是深度。但我想说的全栈工程师不能仅仅理解为前端+后端，全栈的核心是：世界充满了未解之谜，通过探索去解开自己感兴趣的谜底
全栈工程师学的东西越多，学习能力也变得越强，学习能力越强进入新领域就越快。这是我对全栈工程师的理解，如有不对之处还请朱老师指正。

作者回复

2018-01-06

理解的挺好。其实人们真正进入了高手模式，就能触类旁通，举一反三。这时候，你几乎可以做任何工作。

全栈只是工作的一种。

刘宗尧

2018-01-05

怎么理解语言的计算边界？

WTF

2018-01-05

除了并发和分布式思维，还得有量级的意识。

影浅

2018-04-20

只想专注做技术不想做管理怎么办？

齐帜

2018-04-19

安姐这篇文章里说的代码测试框架，跟"A/B测试"那篇的测试不是一回事吧，那篇文章说的是用户调研。

yeyuliunian

2018-01-16

关于编程语言设计有啥资料推荐

TT

2018-01-12

发现安姐所提到的每一点都是自身所缺乏的...现在刚从C语言转到Java开发，只能说业务功能开发好了，但数据读取，一些容错考虑都浅得很

作者回复

2018-01-13

多写多练，多读专栏

susu

2018-01-05

想了解一下，公司招聘的时候是否会更看重候选人使用特定语言的年限，还是相关领域的经验或者学习能力？

作者回复

2018-01-06

可以读读硅谷面试那篇，你提到的能力都会有考核。国内面试流程没有那么复杂，重点会考察使用语言的能力而不是年限。学习能力也需要，但更多是工作中体现出来的