结束语 | 用程序语言，推动这个世界的演化

2019-11-22 宫文学 来自北京

《编译原理之美》

据说，第二次世界大战期间，图灵和同事破译的情报，在盟军诺曼底登陆等重大军事行动中发挥了重要作用。历史学家认为，他让二战提早了 2 年结束，至少拯救了 2000 万人的生命。也据说，苹果公司的 Logo 就是用来纪念图灵的。

图灵的故事我不再赘述，你上网随便搜个关键词都能找到。不过，通过这个故事，我们能得到两点启示：

> 对信息的处理能力至关重要，从此信息技术成为了科技进步的主角，一直到现在。
>
> 科技永远关乎人性，科技是客观的，而推动科技发展的人，是有温度、有故事的。

所以，在《编译原理之美》这个课程结束的今天，除了想跟你好好地说声再见之外，我更多地是想分享作为一个程序员，我们的挣扎、骄傲，以及跟这个社会的关系，跟时代洪流的关系。**我有一些感受分享一下。**

学习技术的过程，是跟大师对话的过程，是融入科技发展这条历史河流的过程，是一个有温度的心路历程。

有同学在留言区说，这门课，串联了计算机领域的很多基础课程。的确如他所说，当然，我也认为编译原理这门课，串联着整个计算机发展的历史，以及做出重要贡献的一代代大师。

什么是大师? 这么说吧。比如你针对某方面的问题琢磨了很多年，有所心得。刚想进一步梳理头绪，就发现有人在多年前，已经针对这方面的问题发表了一个理论，并且论述得很完整，很严密。这个人，就可以叫做大师。

我的一个朋友，某上市公司的副总，原来是在大学教物理的，闲暇时间还会琢磨物理学的理论。有时候，他在琢磨一个点的时候，觉得很有心得，刚想整理出来，再一查文献，发现某个人已经在这个方向发表了成果。他形象地比喻说，刚想写《红楼梦》呢，发现一个叫曹雪芹的已经写了。

计算机领域也有很多大师。我们在学编译原理的时候，其实一直在跟各位大师邂逅。

比如，当讨论有限自动机的时候，你知道那是一个最简单的图灵机（Turling Machine）。你再去阅读这方面的资料，会发现图灵那时在思考什么是计算，这种根本性的问题。

当我们探讨到程序运行环境、汇编语言、机器语言的时候，你会感觉似乎跟**冯•诺依曼（John Von Neumann）**走近了。你会感受到第一代程序员，用机器码写程序是什么感觉。

第一代程序员的人数只有个位数，他们甚至当时都没有考虑到，还可以用别的方式写程序。所以，当冯·诺依曼的一个学生发明汇编的写法时，这位老师甚至觉得那不叫写程序。

而只有你自己动手写了汇编代码，你才能体会到，第二代程序员是怎样写程序的，其中包括**比尔•盖茨（Bill Gates）。**显然，比尔·盖茨认为普通程序员应该用更简单的语言，于是他写了一个 Basic 语言的解释器。其他熟练使用汇编语言的程序员，还包括为阿波罗登月计划，编写程序的传奇女程序员，**玛格丽特•希菲尔德•汉密尔顿（Margaret Heafield Hamilton）。以及中国的雷军等等。题外话，**我看过一个图表，早期程序员中，女性的比例很高，希望未来更多的女性回归这个行业。

接下来，你会遇到 C 语言的发明人**丹尼斯•里奇（Dennis Ritchie），** 他的工作是基于**肯. 汤普森（Ken Thompson）**的 B 语言。这俩人还是 Unix 操作系统的发明者。目前，肯. 汤普森仍在 Go 语言项目组中工作。

我们使用的 Java、JavaScript、Go 语言等的语法风格，都是一路受到 C 语言的影响。我们做编译器的时候，要考虑调用约定、二进制接口，也能从这里找到源头。

在前端部分，我们讨论过面向对象的语义特征，和类型系统。而面向对象的编程思想，在 60 年代就被提出了，经由 80 年代的 C++ 和 90 年代的 Java 才开始盛行。

我们同样简单实现过一等公民的函数和高阶函数，它们是函数式编程的特征。最近几年函数式编程的思想开始热起来，但它的起源更早，可以追溯到 30 年代**阿隆佐•邱奇（Alonzo Church）**提出的 Lambda 演算理论中。

邱奇用一种与图灵不同的方式，探讨了什么叫做计算，这个根本问题。他的思想于 50 年代体现在 Lisp 语言上。Lisp 的发明人是人工智能的先驱**约翰•麦卡锡（John McCarthy），**这门语言成了计算机语言一些重要基因的来源，JavaScript、Ruby、Clojure、Scala、Julia 等语言都从中汲取营养。我最近在研究云计算环境下的分布式数据库问题，发现可能还是要借鉴函数式编程的思想。

再有，编译原理中的属性语法和很多算法，不能不提**高德纳（Donald Ervin Knuth）**的贡献。他的著作应该成为你的必读。

当我们讨论 Java 的一些特征时，你可以试着体会 Java 语言之父**詹姆斯•高斯林（James Gosling）**当初设计字节码时在想什么。你还可以体会一下 **布兰登•艾奇（Brendan Eich）**用很短的时间发明 JavaScript 时，是汲取了前人的哪些思想，以及是如何做出那些重要的决定的，这些决定使得 JavaScript 在元编程能力、函数式编程等方面，直到现在都焕发出勃勃生机。

当你学会编译原理的一个个知识点的时候，就是一步步走近大师们的过程。他们的名字不再是教科书上抽象的符号，你已经能够逐渐欣赏他们的思想，感受到他们的感受，和他们隔着时空

交流。而当你凭着自己的经验，探索到了跟他们相同的方向上，你会更觉得有成就感，会觉得自己真正融入了科技演化的洪流中，算是开了窍了，算是其中的一份子了。

我想，真正在科技领域做出重大成绩的人，都会有这样一种，摸到了科技发展脉搏的感觉。据说，张小龙曾经说过，读懂了《失控》这本书的人，可以直接去他的团队上班。我猜，他对复杂系统科学情有独钟，产生了很多的心得。而任正非先生则对热力学中熵的理论感触很深，并把它深刻地融入到了华为的价值观和管理体系中。

除此之外，我们还要感谢 Antlr 工具的作者**特恩斯•帕尔（Terence Parr）**以及 LLVM 的核心作者 **克里斯•拉特纳（Chris Lattner）。**通过阅读他们的文章和代码，以及其他研究者的论文，你会感受到这个领域最前沿的脉搏。

而通过编译原理中的一些应用课程，我们还可以更好地理解 Spring 等工具的设计者的思维。并且思考，是否自己也有能力驾驭这样的项目，从而成为技术进步洪流中的博浪者。

我相信，如果你不想学习编译原理，可以轻松找到一百个理由。比如：

这个课程太难，我恐怕学不会；

这个课程跟我现在的工作关系不大；

我没有时间；

连谁谁谁都没有学，我就不凑这个热闹了；

…

但如果你想下定决心学会它的话，只要有一个理由就行了，那就是，你也可以成为技术进步洪流中的博浪者，而不是岸边的旁观者。这时，你的自我意识会觉醒：我来了，我要参与。在信息技术成为社会进步关键推动力的今天，这是作为一名程序员的傲骨。

更为重要的是，越来越多的中国程序员已经登上了舞台。越来越多高质量的开源项目，背后是一个个中国名字。我查阅自动化编程这个最前沿领域的文献时，发现文献上也不乏中国名字！

整个世界的目光也开始投向中国，因为他们越来越相信中国的创新能力。我们也确实有能力，因为我们已经有了云计算、人工智能和 5G 技术的积淀，我们正在芯片领域奋起直追，完全自主的操作系统已经开始萌芽。而在这些领域，编译技术都能大展身手。最重要的是，中国作为全球最大的市场之一，拥有最丰富的应用场景，也拥有越来越相信中国创新能力的消费者。

我相信，学习这门课的学员中，不管是大学生，还是已经很有工作经验的大侠，会有相当一批人，在下一个 10 年，使用编译技术做出一番成绩。

对我来说，我很高兴有机会专心致志地梳理编译原理相关的知识体系。而在梳理到每个知识点的时候，我都会迸发出很多灵感。这些灵感将会融入到我正在开发的一个软件和后续的工作中。

在这个过程中，我还有一个额外的收获，就是感觉自己的写作水平和普通话水平都提高了。**原因很简单：**因为每篇文稿都要改好几遍，录音有时也要录几遍。而把陡峭的学习曲线，变成一个让你缓缓爬坡的过程，也促使我必须竭尽全力！

我也觉得用仅仅 40 讲左右的课程，涵盖整个编译原理的知识体系，恐怕会显得不足。虽然涵盖了主要的知识点和脉络，但我在进入每个技术点的时候，发现要把这个点完全展开，可能都需要好几讲才行。不过没关系，我和极客时间还有进一步的计划，**你可以等待好消息！**

总的来说，信息技术的进步史，也是一代代大师的人文故事史。而编译技术让我们有机会走近这些大师，与他们对话，并加入他们。**中国的程序员面临着历史的机遇，而抓住机遇的关键，是自我意识的觉醒，是敢于成为科技进步历史洪流中的博浪者的决心。**

希望与你共勉，一起进步！

最后，我为你准备了一份结课问卷，题目不多，两三分钟就可以完成。希望你能畅所欲言，把自己真实的学习感受和意见表达出来，我一定会认真看，期待你的反馈。当然，如果你对专栏内容还有什么问题，也欢迎你在留言区继续提问，我会持续回复你的留言，我们江湖再见！

宫文学

北京物演科技CEO

不知道在学习过程中，你有哪些体会和评价？
这里有一份专栏调查问卷，邀请你填写。

在12月3日前提交，
极客时间赠送给你专属优惠券。

我们一起继续成长！

去提交

精选留言 (56)

刘強
2019-11-23

唯有感谢。

作者回复: 我还想再力求完美。仅仅40讲的专栏，还没达到我的目标。
后续的计划在酝酿中。
我最近先把一些可以优化的地方优化一下，包括Lab。

9

沉淀的梦想
2019-11-22

期待老师和极客时间的进一步计划

作者回复: 上周我跟极客时间的朋友们讨论时，提到了几个一直能把课程跟得很紧，应该就包括你:)

9

P小二

2019-11-22

感谢老师

作者回复: 希望这门课能帮你“寻找自我”!

6

sugar

2020-04-06

苹果公司的logo与图灵无关（这个梗来自于图灵因同性恋被迫害后自杀是通过食用浸有氰化物的苹果，所以有人猜乔帮主是为了向他致敬），但是读过乔布斯传的朋友都知道，史蒂夫在自传中亲口否认了大家的猜想。真实的原因是帮主早年坚持素食主义时曾长期以苹果为主食，麦金塔电脑的macintosh一词就是苹果的一个品种。

作者回复: 收到。谢谢分享!

3

漂流

2019-12-19

感谢老师，虽然我今天才补完整个课程，并且其中几个章节可能需要 n 次回顾，但是这门课程已经超出我的预期了。

我习惯看完整的东西，这门可也是上周才开始看的，所以并没有一直追更新，这会失去每章互动的乐趣，但得到了连贯的整体思路，这就是选择的一得一失吧。在看之前其实我一直担心课程与龙虎书一样让人无法看下去(我看了大概三章，在语义分析附近就放弃了)，但这门课已经带我走完了整个编译过程，并且对每个子过程都有或简单或详细的描述和案例，这已经超出了我的预期。

我曾经自学本科的计算机大部分课程以达到学会编程的目的，目前也在以编程为工作并以此为乐，玩编程的人好奇心都非常重，每每碰到不懂的地方都想把它挖掘清楚，比如看源码实现从

框架一直追到 std 的各平台差异，从用别人的框架到自己实现并应用与公司的生产项目，而编译原理一直是心中的痛点之一，一直想搞明白这里面到底是怎么运作的。实际上，不懂编译原理的我在很多地方已经使用了编译前端的技术，比如状态机、字符串解析等等。

写到最后，我也不知道我想表达什么，但感谢必须再次提及，万分感谢。

作者回复: 谢谢你用心写这么多留言表达自己的感受!
我跟你也有相同的感受。我们学计算机，本能的就想把相关的技术点搞透，否则就觉得是在糊弄自己。能把过去拦住自己的知识点啃下来，会很有成就感。
我和极客时间会继续在编译原理这个领域耕耘，产出更多的好内容，让更多同学收益!

3

Geek_9c3134
2021-04-09

老师能讲讲 Chez Scheme编译器吗 为什么设计的好

作者回复: 在Lisp的世界里，对Chez Scheme还是很推崇的。由于Lisp/Scheme很强的元编程能力，它可以用很简短的方式别的语言用很长的代码才能实现的功能，这是用它写编译器的优势之一。在《编译原理实战课》中，你可以看到Julia的前端功能也是用Lisp来实现的。
另外，美国印第安纳大学的R. Kent Dybvig教授等人发明了一个叫做Nano Pass的技术，把编译过程分解成很多个小的Pass，每个小Pass只完成一项简单的事情，从而让编译器的模块化程度更高。这也是Chez Scheme中采用的技术。在https://github.com/nanopass/nanopass-framework-scheme页面，你能找到几篇论文。这几篇论文值得一读。
好消息是，Chez Scheme已经开源了，所以我们可以研究它的代码。
不过坏消息是，你要习惯看Lisp格式的代码才行。并且，虽然很多语言都受到了Lisp语言的启发，但Lisp本身的社区还是比较小，所以讨论Chez Scheme编译器的资料也就比较少。这使得我到现在还没有足够的动力去研究它。
不过，由于Chez Cheme完全实现了一个自己的后端，没有借助LLVM，所以在增加后端技术的见识方面，会有好处。当然，同样的还有方舟编译器的代码值得借鉴。

2

brian
2020-05-12

懵懵懂懂看完全部了，感谢老师，让我入门了计算机核心地位的编译原理技术，期待老师出新作!

作者回复: 假以时日，从懵懵懂懂逐步会变得门儿清，技术能力也就会脱胎换骨。
下一季马上会推出，会去实际考察好多个编译器的具体实现，让理论和实践互相印证。

2

milley
2020-03-28

感谢老师，第一遍了解一些概念，值得再刷二遍三遍

作者回复: 你能完整的刷完一遍，就很值得给你点赞！
如果多琢磨几遍，再动动手，你就要成为insider啦！

2

邙山
2020-02-07

科技是冰冷的，人是有温度的。 看了这文章能感受拳拳之心的跳动，共勉

作者回复: 共勉！

2

拉欧
2019-12-13

感谢老师，作为半路出家的半吊子程序员，这门课太超值了；话说我也看过《失控》，但对于能加入微信真是不敢奢求😁

作者回复: 谢谢肯定！
另外告诉你一个事情：我在厦门注册的一个公司名称，就叫"失控科技"。但后来发现这个名字太前卫了，包括在北京不允许注册这个名字，所以后面会改掉:-)

2

FengX
2019-11-27

由于不是科班出生，大学里没有学过编译原理，对于龙书又有点畏惧，因而一直没去学。
老师的课填补了我的知识空白。虽然头几节的课程内容，理解起来有难度，但我没放弃，多读

几遍，并结合同学们的提问和老师的解答，最后弄懂了。之后的课程内容也没落下，不懂就多读多思。
最后，非常感谢老师辛勤的付出！很期待老师的下一节课！

作者回复: 为你的努力点赞！

2

李二木
2021-08-08

”学习技术的过程，是跟大师对话的过程“，很有启发的一句话。

作者回复: 对。当你觉得你的某些解决问题的思路和大师越来越接近的时候，你就知道你真的是进步了。

1

崔伟协
2021-04-27

看完这篇文章，一二季看完了，已成为golang compiler的contributor,有接近10个pr被merge了，当然我以前就有基础的，科班出身，老师的课对我帮助很大

作者回复: Great，优秀！
多积累一些经验，后面可以参与国内的编译器和语言项目！

1

尔冬橙
2019-11-25

老师，能不能加餐讲一下java和kotlin的编译技术上的不同？

作者回复: 嗯。在准备加餐的话题。你这个作为候选！

1

至今未来
2019-11-24

还没学完 谢谢老师o(^o^)o的细心分享讲解✺◟(*❛◡❛*)◞✺

作者回复: 也谢谢你用心参与:)

1

风

2019-11-23

好感动。。
不学的理由千千万万，学习的理由只要一条就够了。
感谢老师，不仅仅分享了知识，还分享了自己的心路历程。
其实在这个信息化的时代里，知识本身并不宝贵，宝贵的是那些能让人奋不顾身地去追寻知识的力量，与其它课程不同，老师的课程里充满了这样的力量。

作者回复: 感谢读懂老师:)

1

写点啥呢

2019-11-22

感谢老师的一路指导，受益匪浅

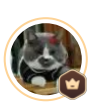
作者回复: 希望你们学过这门课以后，不仅收获知识，同时收获强大的自信，以及摆下一张安静的书桌的心态!

1

朋来先敬

2022-11-25 来自广东

还没看完，但是觉得这门课真的太好了，找了龙书和b站视频，发现都不如老师的课程。而且发现老师志向远大，一直致力推动中国编译技术发展，鼓励大家多为国家编译器技术努力!
👍👍👍



dll

2022-07-28

看完了，耗时快一个礼拜，很吃力，做了很多笔记，有些地方还看的迷迷糊糊的，今年或者到明年的上半年，计划把老师的编译原理实战，编写一门语言的课程和课后code，一块挨个刷一遍。然后再去有余力的时候看一看龙书，补充书本经典知识。谢谢老师领入门!

陌兮

2022-06-11

多谢老师。一路“看”下来，自己这方面是真正的小白。一遍过后，收获良多。打算找一本编译原理的书籍，再仔细研读下，基础才是最核心的技能。期待老师后续的课程。