编译原理系统学习路线（2024年最新）

分享一下编译原理相关学习攻略。

编译原理这门课算是计算机很重要的专业课之一，但是呢，由于面试几乎不考，考研也不考，并且呢，大家学习了感觉作用也不大。

所以在不少学校，这门课就是选修课，大多数学校上成了理论课程这种，我们学校当时也是这样的，基本上就是学了一堆像有限状态机、上下文无关文法、递归下降分析、LL、LR这样的语法分析方法。

到现在像 LL、LR这样的分析算法具体步骤也早已忘了，不过我还是觉得如果还在大一、大二阶段，时间充足，编译原理还是非常值得学习的一门课。

当然，如果现在需要用到编译原理的知识，大多数场景都直接使用 yacc(自己定义文法，会帮你生成语法分析器)，根本不会自己手动去实现语法分析。

#

学编译原理有什么好处？

有助于帮助理解高级语言是如何转换为汇编代码、二进制代码的

在一些场景可以实现DSL（领域特定语言），类似 markdown、SQL、Latex 这种都算是 DSL，实际工作中类似美团、阿里这种大公司尤其是优惠券那一套体系，往往都需要一套DSL来表达各种优惠组合规则

能够更深入理解语言特性，各种语法规则

#

学习编译原理需要哪些基础

编译原理属于计算机高年级课程，至少熟练掌握一门编程语言，比如 C/C++、Java、Go、JS等，尤其是擅长某门编程语言最好，因为如果要实践可能使用 C++/Java 实现 TinyC、TinyJava 解释器( C语言和 Java语言的子集)

另外，最好了解一些基本的数据结构和编程概念，比如树、栈、队列、正则表达式、递归等，这些会有助于你学习编译器前端技术。

有一些计算机组成原理、汇编方面的知识，会有助于你学习编译器的后端技术，代码生成等。

#

理论课程

如果单纯想学习编译原理理论，应对学校考试，那么可以看下哈工大的编译原理网课：

https://www.icourse163.org/course/HIT-1002123007#/info

这门课也是当时我看过的，特别适合学校要考试的同学，理论讲解很清晰。

#

结合理论，做一些编译器实践

结合一点实践，中科大华老师的编译原理课程，会

https://mooc.study.163.com/course/1000002001

这门课会有几个 Lab：http://staff.ustc.edu.cn/~bjhua/courses/compiler/2014/

引用课程介绍:

在本课程的这些系列实验中，您的工作是为 MiniJava 编程语言从头开始设计和实现一个名为 Tiger 的编译器。MiniJava是Java编程语言的一个非平凡的子集，在Tiger书的附录中有描述。同时，您将使用 Java 作为实现编程语言（因此我们遵循编译器设计史上著名的先有鸡还是先有蛋的传统）。在本课程结束时，您将深入学习如何使用 Java 为现代 OO 语言（如 Java）实现编译器。 有六个实验室，并计划进行最终项目：

这门课有六个实验：

实验 1：词法分析和解析器，实验 1 将为 Tiger 设计和实现前端（词法分析器和解析器）: http://staff.ustc.edu.cn/~bjhua/courses/compiler/2014/labs/lab1/index.html

实验2：抽象语法树。本实验将要求设计和实现抽象语法树并构建一个Elaborator: http://staff.ustc.edu.cn/~bjhua/courses/compiler/2014/labs/lab2/index.html

实验3：代码生成器。在第三个实验中，将设计和实现多个代码生成器：C 代码生成器、Java 字节码生成器、Google Dalvik 生成器和 x86 代码生成器: http://staff.ustc.edu.cn/~bjhua/courses/compiler/2014/labs/lab3/index.html

实验4：垃圾收集器。在第 4 个实验室中，将构建一个垃圾收集器并将其链接到 Tiger: http://staff.ustc.edu.cn/~bjhua/courses/compiler/2014/labs/lab4/index.html

实验 5：优化。这个实验要求向 Tiger 添加优化，使其能够生成更高效的目标代码: http://staff.ustc.edu.cn/~bjhua/courses/compiler/2014/labs/lab5/index.html

实验 6：注册分配器。在第 6 个实验中，将为 Tiger 编写寄存器分配器，方便Tiger 编译器生成高效的生产质量本机代码。

#

斯坦佛 CS 143 课程

斯坦福大学的CS 143课程是编译原理领域非常有名的课程，课程目标是理解编译器的设计和实现。

将学习各个编译器的关键组成部分，包括词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成等。

另外，课程还涵盖了一些高级主题，比如并行编译和Just-In-Time（JIT）编译等。

同时课程也会设计和实现一个 Class-Object-Oriented-Language，简称 COOL 语言。

课程网站：

http://web.stanford.edu/class/cs143/

课程视频：

B站：https://www.bilibili.com/video/BV1NE411376V

课程教材：

中文名：编译原理技术和工具 英文名：Compilers: Principles,Techniques,and Tools

课程作业：5 个书面作业 + 5 个编程作业带你实现一个编译器

#

其它编译原理DIY资料

两周自制编程语言

初学者可以先阅读编译器 DIY 类的书籍因为这样可以先忽略复杂的理论，先获得直觉认识，甚至具备动手能力。

网易云课堂：自己动手用java写编译器

https://study.163.com/course/introduction.htm?courseId=1002830012

在线文档： 《自己动手写编译器》

这本书将介绍一个非常简单的编译器（ TinyC 编译器）的实现，用简单和易于实现的实例来描述基本的编译原理及过程

https://pandolia.net/tinyc/

非常推荐！

llvm:

https://llvm.org/docs/

antlr 项目

https://www.antlr.org/

编译三大书籍：龙书、虎书、鲸书

编译原理三大经典：龙书 虎书 鲸书

最后说一下，编译原理，适合时间充足学习计算机的同学，比如大一、大二，如果你在赶着准备校招、考研等，就先把这个放一放~（因为面试基本不考察