

总结感想

18373599 崔建彬

总结感想

- 一、编译课设介绍：
- 二、初识编译课设
- 三、如临大敌的语法分析与错误处理
- 四、代码生成与优化
- 五、总结：

一、编译课设介绍：

本学期编译课设为计算机学院四大名捕之一，课程改革较早，难度较大，无论是对工程能力还是思维能力都有一定的要求。同时因为历届次次迭代的原因，课程的完成度极高，基本是从浅入深，层层递进地指导同学们独立自主从0完成一个编译器。没有说出现难度阶梯式跨越的现象。作为选了这门课的同学，经过本学期的学习，自然也是感触颇多。让我一一道来。

二、初识编译课设

从一开始，我对编译课设并没有一个明确的概念，只知道是写一个编译器，要优化竞速，很恐怖，很卷的一门课。也是最早进行难度改革的一门课，可以说是co他们的先驱了。最初两次作业难度较低，大体就是让我们理解本次课设的文法，加上词法分析只需要自动机便可以完成，让我有点轻敌。谁知道真正的难题还在后头。

三、如临大敌的语法分析与错误处理

没错，就如同小标题一样，第一个给我当头一棒的作业是语法分析。递归下降的语法分析，整个实验的复杂度远远高于之前的词法分析作业。但是因为前两次作业的轻敌，我并没有十分重视本次的作业。导致我最终光粗略完成就花了2天，再加上漫长的debug过程，前前后后花了小半周。好在课程组放出了部分测试样例，极大的降低了我们debug的过程，让我们不用从0制造样例，还是十分友善的。

其次就是错误处理。错误处理从理论上讲，难度应低于从0写一个语法分析。但谁知难度一点也不低，不仅要在语法分析中魔改自己的代码，使其从1200行徒增到2000行，还要进行错误的跳过，即使这一行报错，我们也要继续读下去，直到检查完文件所有的错。这个功能的实现就极大的增加了错误处理的难度。我前前后后又花了半周多，才勉强通过了课下的测试，但是即使到最后的目標代码生成时，还能看到我错误处理时的bug。

四、代码生成与优化

代码生成作业部分前前后后一共给了四周。有了前两次作业的惨痛经历和教训，我从一开始就着手去写代码生成。但是手里只有一个语法分析结果的我显得有点丈二和尚摸不着头脑，再仔细阅读完《2020官方文档》后，我对什么是中间代码，如何从中间代码生成目标代码有了个初步的概念。再初步完成第一版代码生成后，我开始着手去做更复杂的代码生成二。主要难题在于二维数组、函数调用返回的处理，不过有了第一次代码生成的经历，第二次作业也不是不可战胜的。

接着就到了我们的重头戏之一，代码优化。既然选择mips，就要使用上mips的特点：寄存器。如果只是内存的取放，那又和pcode有什么区别呢。于是我对整个代码加上了立即数优化、全局寄存器分配、临时寄存器分配。不过只做完这些确实还不是很全面，速度也没有到达前20%，不过与之前比较还是有了较为明显的进步。同时这些优化也耗费了我不少的精力。基本上每次优化完成都要从0去debug。虽然十分的累，但是最终还是受益匪浅。

五、总结：

总结一下本学期的课设经历，就是痛苦却快乐着。看着自己一次次满篇的绿色accept，之前的累一下次就忘光了，只剩下收获的喜悦与满足。最后看着自己从0完成一个几千行的工程项目，从0写出一个编译器还附带优化，还是颇具成就感的。整体课设感觉没有过度花费同学们的时间，却让同学们都收获了许多。可以说课程的设计还是十分合理的。在最后，也祝课程组越办越好，今后让更多的同学享受编程的乐趣。