编译技术实验总结感想

这个学期有幸学习6系第三门计算机系统相关的核心专业课，经过了一个学期的理论 + 实验的磨练，我感触颇深，下面来简单聊一聊这一个学期学习编译的心路历程吧。

在第一次实验作业放出来的时候，我看到时是非常慌张的。在上个学期，OO课程第一单元表达式求导作业就给出了一长串的产生式，当时我从0开始学习表达式的递归下降解析，没找到什么很好的学习资料，硬着头皮在子程序接口十分混乱的情况下写了一个看起来有点像递归下降的分析过程，才惊险的通过了第一单元的作业训练。而编译的文法比表达式求导中的文法复杂得多，一比较整个一小巫见大巫。当时的我，**十分担心自己不能亲手把这个编译器实现出来**（讲真，从大一程设开始我就患上了"码力不足恐惧症"）。在实现语言的选择上，因为有码力不足恐惧症，且怕出现玄学bug，所以我选择了比较熟悉的 Java 语言进行开发。

不知不觉就是一个月，在国庆期间，因为害怕写不出来这个庞然大物的语法分析，我连续编码 + debug 12小时，终于把这个大玩意儿给搞定了。而这个时候，正好OO助教工作中我在负责作业标程编写，这时我就想既然我已经把这么一个大家伙搞定了，那么表达式这种小玩意儿应该手到擒来吧？于是我直接复制过来了编译器中的核心代码，改了改词法分析部分的正则，然后做语法分析，果然比较顺利地写出来了。在这个时候，我比较深刻的体会到了，在给了文法的情况下，词法分析和语法分析是一般性的解析方法。以后不管让我分析什么字符串，只要给了文法，似乎都可以迎刃而解。

错误处理是做的比较挣扎的一部分。当时吴佬向我们安利抽象语法树架构，这样可以比较OO地去做后面的部分。我最开始的语法分析是扔到一个 `Parser` 里面写完的，总共有一千多行，作业让我输出什么我就输出了什么，并没有留下什么比较有用的信息。因为下学期得负责助教工作，所以我比较希望能把架构设计得好一点，于是就开始了为期4天的大力coding。在那几天里，我差不多每天都要写 + 调通800行代码，这个小编译器的代码从一千多行直逼四千行。在重构出来AST，接着写错误处理部分的时候，我被谜语的错误处理需求给恶心到了，有好几种类型的错误的语言描述十分晦涩难懂，好多地方说不清楚算不算错误，全靠助教在讨论区和微信群补充答疑，不然我真的是对着空气在debug。由于这次作业没有公开的测试用例库，所以debug起来也挺痛苦的，4天的时间里面大概有1天是在debug。这次作业给我的感觉就很奇怪：我大概可以理解课程组想要我们通过建立符号表来处理一些语法甚至语义错误，找一找感觉，但是因为我们没有生成代码，所以（至少对我来说）我建立的符号表并不能圣人后面的代码生成作业，这导致我在代码生成作业的时候花了一些时间去调整符号表部分的架构，并且丰富了符号表的表项，这时候大概才可以做代码生成。后面由于一些同学的反馈，课程组把错误处理部分的作业延期了。虽然我当时早就写完了错误处理作业，但是我在写代码生成的时候认为确实最好不要把错误处理和代码生成作业弄得时间上毫无交集，不然期末考试的时候下载下来的错误处理代码和后面的代码生成差距可能会比较大，恐怕也会影响大家的发挥。

由于前面独立完成了这个大家伙的递归下降分析，所以我的"码力不足恐惧症"的症状减轻了，于是毫不犹豫的选择生成mips，其实选mips的另外一个原因是想满足自己的求知欲，追求"用自己的编译器编译自己写的代码，然后扔到自己设计的处理器上运行"的星辰大海。奔向星辰大海的过程中，我遇到了不小的困难。困难一方面是理论知识的欠缺，当时我们班编译理论讲课的顺序做了调整，这导致代码生成1原定的ddl都快到了的时候，我们班还没有学到不同语句的代码生成办法，而我在那里胡乱写的编译器还并不能处理数组、函数这种语法成分。好在代码生成1延期了一周，救了一命。由于听说测试点中没有包含数组，所以我最开始也没写数组，把这个特别大的雷埋到了代码生成2。整个代码生成1我大概折腾了10天，有一个通宵是在做测试数据构造。之所以要做数据构造这个事情，一来是我没写数组，用不了测试用例库；二来是我认为自己不能老是靠课程组公开的测试数据和评测机去debug，都大三了，应该自食其力了。在那个晚上，我对每个语法成分都列了覆盖的情况表，然后对所有表格用加法原理和乘法原理人肉做组合，构造了一个覆盖到所提及的所有情况的数据，凭借它我自食其力通过了代码生成1的测试。经此一役，我算是亲身实践了规模比较大的一次人肉黑盒测试，也在构造数据的过程中学到了一些构造技巧。

代码生成2是最肝的一次作业了。由于之前没写数组，所以应该提前暴露出来的问题到了比较晚的时候才暴露出来。当我把控制流写完之后惊奇的发现自己目前的架构并没有办法处理"数组传参"这个东西时，我突然感觉编译课可能要寄了。在和多位大佬进行讨论之后，在离ddl还剩下6天的时候，我下定决心选择重构。不重构，只有一死，重构的话，或许还会有一线生机，另外，与其说犹豫到底要不要重构，不如亲身去try one try，犹豫的时间早就够我通过小规模的尝试来确定自己最终是继续重构下去还是放弃重构了。重构时我主要建立了对操作数的抽象，以及重新设计了自己的对数组/指针操作的指令。在那几天里，我感觉自己几乎一直在写编译，4天内在新主楼二楼走廊通了3个宵，总共睡了十几个小时，改了三千多行代码，提交了接近50发commit，其中有接近40发是在fix bug。我可能永远都忘不了，在某个通宵的夜晚，平均每30分钟就能de掉一个关于左值分析或者数组传参的bug的情形（这短短的四百多行代码里面居然藏了那么多bug）。逐渐地，我从连C库都WA一多半，逐渐到AK了C库，再到通过了作业的A和C级用例，接着又开始一个一个地啃B级用例，一点一点从每个testfile中提取让自己出错的pattern，一行一行地用 Mars 单步调试少则一两百行多则四五百行的MIPS汇编代码，最终我在离ddl还有两天左右的时候通过了所有的公测和民间数据。这个过程真的非常艰难，身心饱受着煎熬，但最终成功的那一刻到来时，一切又都释然了。最终我还是做到了，实现了自己独一无二的编译器。

在完成了实验的工作之后，我感觉自己再也不会犯"码力不足恐惧症"了，也不会怕有自己de不出来的bug了。整个编译课下来，最终我的编译器的代码量是6000-7000行的量级，这都是我自己一点一点敲出来的，其中中间代码里很多技巧是我独立设计出来的，测试数据构造我花大力气构造过，密密麻麻的汇编我调过好几百遍，这些给我带来的提升是真真切切的，它们已经与我融为一体了，时间是偷不走的。从这个角度来说，我非常感谢老师和助教们为我们设计了这样出色的编译实验，我们从中收获的东西比那些纸上谈兵的课程要多得多。

虽然编译实验让我们收获很大，但是可以观察到，不少同学们对其还是颇有微词，不满主要集中在两个方面：一个是竞速的内卷制赋分导致同学们有时会产生"做优化不是自己主动要做，而是被别人逼着做的"，这导致一些同学还没开始就对竞速有了很大的心理抵触。如果编译竞速能够给出一个合适的Level，比这个更高性能的编译器不在分数上做具体的区分，那么同学们可能从心理上就不那么反感了。第二点就是编译实验只提需求不给参考指导书，铺垫全都留给了理论课，而有些老师的理论课又几乎不对实验部分做提示和讲解，还有的班级理论课进度和实验课进度拓扑序混乱等问题，要不是有助教和大佬同学，好多同学很难去搞实验部分，个人认为课程组应该考虑如何让实验课和理论课更好地配合起来，不要对一些东西踢皮球，不应该一味地标榜课程的难度全国顶尖但却提供不了与之匹配的指导。

最后，还是感谢编译课程组的每位老师和助教在本学期内辛勤的付出，你们辛苦了！