

学号: 20376158

姓名: 安达楷

班级: 212114

总结感想

一学期的编译实验终于结束了, 这学期在编译实验上的花的时间可以说相当的多了, 从9月份放出第一个实验开始, 感觉自己就一直在写编译实验。

刚开始还很迷茫, 不知道如何下手, 甚至连文法解读都看不懂。词法分析写的还算顺畅, 不算太难, 接着就马不停蹄接着写语法分析, **这时就陷入了第一个困难点: 语法树的建立**。我这时还困惑于什么是语法树, 担心自己语法树建的结构不对, 导致后面的实验非常的困难, 于是漫无目的地构思了很久, 最后通过咨询助教才得到了解答。

国庆节前后我就开始了错误处理, 参考了其他代码, 我发现错误处理和中端代码可以一起做, 但是我在做错误处理的时候还不知道怎么做中端代码生成, 虽然遍历了语法树, 但仍然需要重构许多内容。在做中端代码生成时就遇到了我的**第二个困难点: 不知道后端代码怎么生成**。由于我选择的是四元式, 导致代码是需要自己设计的, 设计了类似于mips的代码, 但是其实在写中端的时候完全不知道后端要怎么做, 我就先写了一版简单的, 打算在写后端的时候再完善, 导致后面在写后端的时候反复地来到中端重构代码。

从十月中旬到十一月中旬, 我一直在写中端代码和后端代码, 由于我的四元式设计得不够好, 所以导致这两者其实耦合地比较严重, 中端更多的是为了后端生成服务的, 而不是与后端独立的两个部分。在写后端的时候, **遇到了第三个困难点: 不知道mips的机制**。虽然会简单地写mips, 但是其实对于mips内部是如何内存管理的、如何分配寄存器这些问题, 一开始还是非常的懵, 后面也是学习咨询了很久才理解了。但是在写的时候还是遇到了很多的困难, 函数的调用、数组的处理、寄存器的分配, 这三块内容我感觉是写后端代码的几个难点, 尤其是寄存器的分配, debug非常难de, 经常需要用mars一行行调试mips代码来de, 一de就是一两个小时。好在, 最后还是写完了, 实现功能的那一刻真的非常有成就感!

之后十一月中旬到十二月中旬, 就开始了我的优化。然而一开始就面临着要重构中端代码。因为我在写中端的时候对基本块的概念不是很清晰, 虽然划了, 但没完全划, 不符合规则, 导致我花了挺多时间重构基本块的。重构后就顺其自然地建立了数据流图, 之后就进行了数据流分析, 活跃变量分析、到达定义分析(没用上), 然后做了死代码删除、常量传播(虽然失败了)。死代码删除时, 也碰到了一个bug好几天没解决, 最后多亏了助教哥哥的帮助。后端优化方面, 主要就是做了寄存器分配, 图着色寄存器分配应该做了我有两周的时间, 非常的痛苦。一开始不知道怎么做图着色, 参考虎书的内容, 他是用了虚拟寄存器, 但是我之前没用又不想改架构了, 就没用他的方法。不过核心思想还是差不多, 先对局部变量和参数建立了冲突图, 然后用全局寄存器对冲突图的节点进行染色, 对其他的变量分配临时寄存器。图着色虽然做了, 但是发现testfile8还是没过! 然后又把临时变量用了引用计数, 在替换时使用OPT策略进行最优替换。还是没过! 最后发现是建立冲突图时变量太多导致运行时太慢的TLE! 改了之后就有了不错的速度。然而, 最后测试了下, 真正最起作用的还是引用计数!!! 可能是因为我的图着色没做“合并”的操作? 之后没啥时间了, 做了点简单的优化: 乘除模优化和窥孔优化。除法优化的效果很不错, 窥孔优化的效果更是惊人, 真正理解了什么叫细节决定成败, 少量的优化积少成多, 也能取得非常不错的优化效果。

上面对整个编译实验的过程做了一个总结。

先提点小意见吧, 一方面就是希望前端可以使用自动化的工具, 而不是只能用递归下降进行编写, 现在学界和业界编译器的重点都是在后端, 希望可以减轻前端和中端编写的压力, 能够把时间腾出来多做些后端相关的优化。另一方面, 希望能在指导书中多给一些设计的指导, 比如涉及到mips的函数调用、数组、寄存器分配等知识, 这块对于基础较差的同学很难理解(比如我), 我就是在看了很久其他资料后

才懂的，并且我身边还有一些也没搞懂这个问题的同学来询问我，希望课程组可以把指导书的讲解更全面细致一些。

从零开始写一个编译器还是很有成就感的，虽然整个代码有点shi，有许多地方都是由于没有提前设计好导致的，不过这也让我积累了很多的经验，一定是要设计先行，多想再写。同时，之前从没有个人接触过上千行的项目（web开发项目除外），个人感觉难度梯度有点大，之前两个学期的CO和OS都没有这么大的工程量和代码量，突然之间上手很难不写出屎山代码。这次的编译实验收获很大，是对理论的一次很好的实践。（好，得去复习理论了

最后，感谢一下老师助教和我身边的同学们，他们都在这个过程中给我提供了很多的帮助。

language	files	code	comment	blank	total
C++	84	7,700	1,111	1,161	9,972