

2022编译总结感想

黄雨石

20376156

2022编译总结感想

感想

总结

对课程组的建议

感想

这个学期真的是很难呀，各种大作业语言考试，语言补习叠加，先简单发表一下感想吧。刚开学时，初次与编译相识，文法解读、词法分析、语法分析包括期中考都过的十分顺利，毕竟唯一有点难度的递归下降在上个学期第一次 oo 作业中已经学过了，只是代码量的拔高而已，于是之后在编译上面摆了几周，去学英语了。结果过了几周，不看不知道，一看吓一跳，直接快进到第一次代码生成，并且还是直接就 mips 了，我还以为还有生成中间代码的作业过度一波，是我太天真了，于是我需要先同时完成错误处理、符号表建立、中间代码生成，在看过指导书后错误处理、符号表建立都很快一步到位了，为了加快步伐于是我选择在完成前面两项工作的同时生成中间代码，这部分只是费时间，但是每一分钟都不是虚度的，可是到了 mips 生成阶段，我的噩梦来了，细节是在是太多了，函数压栈，寄存器保护回写等等。。。并且写了后面就忘了前面，每次写完都会怀疑自己前面有没有写错，当然由于自己本身代码能力还算ok，于是也艰难地独立写完了，但是许多地方已经没有遵循 oo 课所教授的一些代码规范。虽然完成了，但是对着大几百行的 testfile 以及标准输入输出在 mars 中单步调试的日子非常痛苦，整整持续了完整的两天，这两天我几乎什么都没有做，就对着一堆文件 deBug，戏剧性的是我过完所有辅助测试才发现代码生成只要求翻译完部分文法，于是又有些幸运地提前完成了代码生成二并且 Bug 也 de 完了，但是这又是另一个炸药，因为我直到优化发布我才开始做一系列优化，此时对我自己的代码都有一些不熟悉了。于是在优化的过程中我首先重新阅读了一遍自己的代码，然后将代码改得基本符合 oo 的规范，但是优化也是一件十分痛苦的事情，每做一个优化过了竞速就会发现自己辅助库或者代码生成二有点挂了，于是一 de 就是两三个小时，最后发现只是一两个细节问题或者是编码疏忽时，我内心是崩溃的，但是最难受的是发现好多优化都需要改之前的一些架构，于是 Bug 愈发多了起来。每当我改完一个 Bug 就会发现连带出了另一个 Bug，都是因为忘了之前的设计，在改之前的设计时没有回忆清楚的原因。

虽然很难，但是每当完成了一个优化，看着自己的排名飙升并且正确性得到保证就会非常的自豪，并且偶尔发呆时看着自己7000行左右的编译器就非常有成就感，可以说编译实验是一个痛并快乐着的过程

总结

发表完感想，舒畅了不少，接下来总结一下我在编译实验中学到的东西：

- **设计优先**：我在编写数据流分析的代码时，认为教材中已经有指导，理论课也学过算法，没有仔细思考设计就打算开始写代码。结果动键盘前却发现自己的许多实现在当前环境下不适用(需要推翻之前的设计进行较大的修改)，只好老老实实进行设计。因此，对于较为复杂的算法和模块的实现，一定要搞清概念后，先再脑海中过一遍实现的细节，再给出一个合理的设计，最后再给予实现。若在实现的过程中一发现不妥当的地方，要及时修正设计而不是硬写，这样能及时止损。
- **多做模块化测试**：我其实在设计这个编译器之前是明白这个道理的，并且由于强迫症，我都会在编写完每一个模块后进行一波测试，但是我身边有些同学总喜欢全部写完再测试，看着他们 Bug 一 de 就是一个星期，我就会很庆幸自己早早就做了测试

- 当性能存在问题时，首先看自己的实现的是否有大的问题，而不是去做细枝末节上的优化：例如刚开始我做的优化和舍友做的优化其实是一致的，但是第三个点始终是在70名之间徘徊就是上不去，于是我开始做一些窥孔的细节优化，但是直到最后才发现原来是我的临时寄存器分配的 `CLOCK` 写错了，要是一开始我就去找大的优化是不是在实现上有问题那么也不会浪费那么多的时间，最后改完后第三个点就正常了

对课程组的建议

- 希望课程组多给一些可以参考的编译器资料，毕竟理论课和实验课的差距还是很大的。同时对于课程组给的参考的 `p10` 编译器，我认为它其实只是设计理念上大体符合我们的任务，但是与课程过为脱节，两者是在没有什么关系，所以没有什么参考价值，其实可以将一些简化的 `LLVM` 的源码分享给大家查看。当然也建议参考软件学院之前编译的平台，感觉他们的指导非常完善
- 评测机比较慢，希望可以让他变快一点，我甚至怀疑每一个点是不是都重新把我的 `src` 又编译了一遍
- 对于竞速的一些指标，感觉不是太满意，为什么乘除指令占的权重那么高呢，建议模拟一个较为真实的 `CPU` 和内存访存的模式，以在此模式下完成程序运行所需要的时钟周期数为指标。同时也感觉可以将代码辅助样例库以及代码生成二都纳入竞速中，毕竟只有大量的测试点都快才能证明编译器的性能，就好比比我比某人竞速有一个点慢很多，但是我代码生成二、辅助库所有点都比他快，这并不能说明他的编译器性能比我的更优
- 希望课程组也像 `oo` 那样发布一些代码规范检测工具，不少同学只是以完成作业，提高竞速排名为目标，代码基本没有办法看，我甚至还了解到有同学将所有代码写到一个文件夹里面的。当然代码规范、耦合性、架构好坏等等我认为也都可以纳入评分标准中来
- 希望错误处理作业跟生成中间代码放在一起，输出形式上可以仍然只检查错误代码。这样可以保证语义错误的检查与后续流程可以使用同样的符号表，减少因为错误处理阶段因为考虑不全、仓促实现而造成的不必要的麻烦。

最后希望编译实验课程组越办越好，老师与助教都辛苦了！