记一次编译课设与困难解决

16231256 李天宇

在编译课程设计这门课中，我选择了难度3的任务。课程设计的目标是，根据给定的文法（扩充C0文法），实现一个针对该文法的编译器。编译器的目标代码是mips汇编代码。需要实现命令行窗口路径输入，尽量多报错，优化等目标。这里将着重讲解我在本次编译课设中遇到的困难和我的解决方法。

首先遇到的第一个困难就是，这是我头一次完成这么大的一个任务，这是一种心理上的压力——我能不能完成呢？我会不会失败呢？实际上我们在动手写这个编译器前已经阅读过两个编译器，本质上已经为我们打下了一开始的基础，让我们了解了基本的工作量等。最终我克服了心理上的障碍，并且成功打出了我的第一行代码——include <stdio.h>（笑）。

第一个任务是完成语法分析程序，由于之前上课的时候已经学习过了相应的部分，所以我也想当然地按照既定的目标开始实现。语法分析部分的要求很简单，只需要输出读入的词（按规则）即可。在完成了之后，我直接写了自己的测试程序并且进行了测试，当然，结果并不如我料想的那样好。其中出现的问题主要是处理换行的逻辑上（行计数器好像有点问题）。最后我通过逐语句执行的方式发现是我在处理\n 和 \0的时候有点问题。经过其他微小调试之后，第一部分的内容就完成了（当然后续还是发现了其他bug的）。

第二个任务就是语法分析部分了。我利用递归下降分析法来书写我的程序。实际上写到void main部分才意识到，void不仅可以跟在main前面，也可以跟在函数名前面表示函数定义。这里我改写了文法以达到LL（1）文法，实现实际需求。语法分析的第一个bug出现了，我重复引用了很多东西，换句话说就是我循环引用了。我头一次书写这样的程序，之前不了解去掉循环引用的方法。所以我去网上查了解决办法，最终解决了这个循环引用的bug（通过头文件、C源文件分离的方式和禁止头文件循环的方式）。

这时候终于可以开始试运行语法分析了，可是语法分析的试运行并没有想象的顺利，我又开始了逐语句执行……最终发现在很多地方我没有读入下一个符号就调用下一个分析过程了（这里需要说一下，我的程序约定是，在检查下一语法成分之前为它读入好第一个符号）。在仔细检查debug之后，我终于完成了语法分析的部分。

最终，语法分析部分我实际书写时间大约是6小时，debug时间大约也是6小时，这告诉我们如果事先不想好随意写的话是会出很大的问题的。

下一任务就是语义分析、中间代码生成、目标代码生成了。这部分花费了我最多的时间，因为我需要设计好运行方式（运行栈）、符号表结构等。而且之前我的语法分析部分没有继承属性和综合属性，也没有返回值（全是void）。为了改这些，我就花了1小时左右。继承属性和综合属性通过参数方式实现，继承属性直接传入实参，综合属性传入地址以获取下层的返回，返回值则是用来判断下层是否正确的。运行栈的设计我全凭脑补，在纸上画出我的设想，反复推敲是否会导致意外情况……符号表中途改过一次设计，是为了实现局部变量和全局变量同名时候处理方式。我采用了两个表方式，优先查找局部表，局部表不含时再查找全局表。最后我通过大概十几个小时的代码完成了这部分的最初工作。当然结果也和我想象的一样（我已经习惯了），有很多bug，甚至不能通过最简单的测试程序（helloworld）。我开始逐句对比中间代码和源代码、中间代码和目标代码，化身人肉翻译机。在一行行地对比中，我认识到运行栈设计的不足等问题，逐个改bug，终于可以通过自己的测试程序。

随即我就兴奋地上论坛查找别人的测试程序来测试，悲剧的是，一个都过不了……漫长的debug之路依然等待着我。这部分debug的过程和debug自己测试程序的过程类似：首先阅读源代码理解数据流向，预测输出期望。然后阅读生成的中间代码和目标代码，化身mips处理器，对比期望数据流向判断bug位置。最后也是完成了小部分同学的测试代码debug。

报错程序和出错处理部分是我认为较为简单却需要细心的部分。首先你需要考虑到所有可能出错的部分，在这些部分加上报错函数。另外出错处理部分，我们需要尽量多报潜在错误，意味着我们应该尽量少跳读。对于每一个出错的位置，都需要我们细心地思考应当跳读到的语法成分。就这样，我也完成了报错程序和错误处理部分。

测试程序一和测试程序二的测试和上面类似，帮我找出了不少待优化的地方和不少致命闪退bug。

正是这时候，我的其他课程纷纷来到结课时期、同时我不小心把水撒到电脑上导致电脑送修，这一切导致了我的手忙脚乱，最终也就没有完成代码优化的任务。虽然我在前期曾经找老师、助教、同学多次讨论过优化的实现方式，但是自己的宏图伟志最终还是没能实现，也是我在课设中的遗憾了（当然宏图伟志究竟对不对就另说了哈哈）。

这也就是我这次编译课设中遇到的主要问题和我的解决方式。从上面的经历可以看出，人人的bug都是不同的，指望别的同学和你出一样的bug然后人家debug完成告诉你是不现实的。实际上还是需要自己静下心来逐语句、逐过程执行debug。完成这样一个大项目使我收获良多。