**哈尔滨工业大学**

**编译系统 2022春**

**实验三**

|  |  |
| --- | --- |
| 学院： | 计算学部 |
| 姓名： | 冯开来 |
| 学号： | 1190201215 |
| 指导教师： | 陈鄞 |

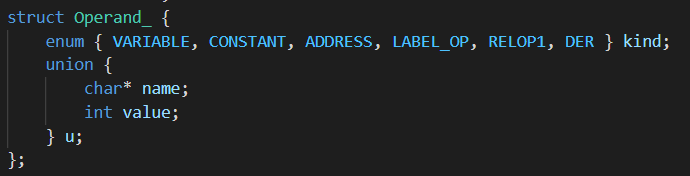
1. 实验目的
   1. 巩固对中间代码生成的基本功能和原理的认识
   2. 能够基于语法指导翻译的知识进行中间代码生成
   3. 掌握类高级语言中基本语句所对应的语义动作
2. 实验内容
   1. 实验过程
3. **实现功能**

根据之前实验的词法分析和语法分析我们已经构造除了语法树和符号表，在这个基础上，我们完成了intercodegenerate.c的设计，即中间代码生成。整体的流程就是将源程序生成的中间代码内容存到InterCodes这个链表中，所以我们需要做的就是对一些语法单元编写translate函数，通过语法树根节点Program的翻译调用到子树，进行递归翻译实现所有中间代码的生成。

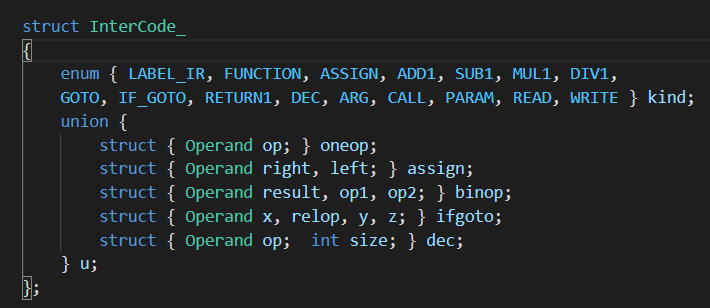
本次实验实现了以上所说的功能，对于实验指导书上的样例只完成了必做样例的中间代码生成。

1. **数据结构**

本次实验主要用到的数据结构是两个自己定义的结构体。一个是操作数结构体，因为我们需要打印中间代码生成的每个操作数是什么类型和数值。这样我们在生成中间代码的过程中能明确操作数是什么，其中CONSTANT和RELOP1需要拥有value值，其他类型需要拥有name值。具体构造如下图所示：

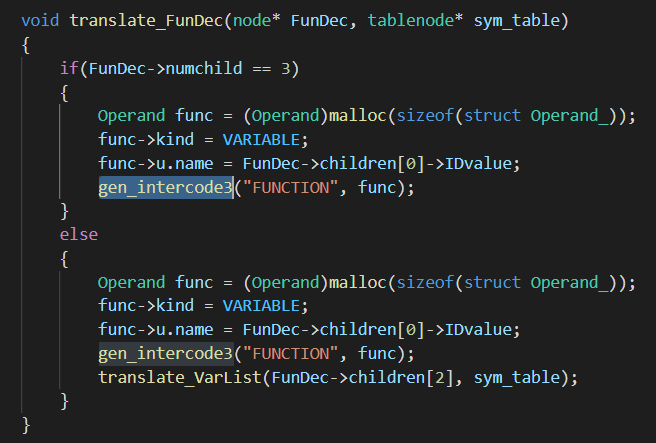


还有一个结构体是就是用来保存生成结果的InterCode。为了写入和读出方便，以及保存所有中间生成的代码，我定义了双向链表InterCodes来保存前后节点的信息。具体构造如下图：



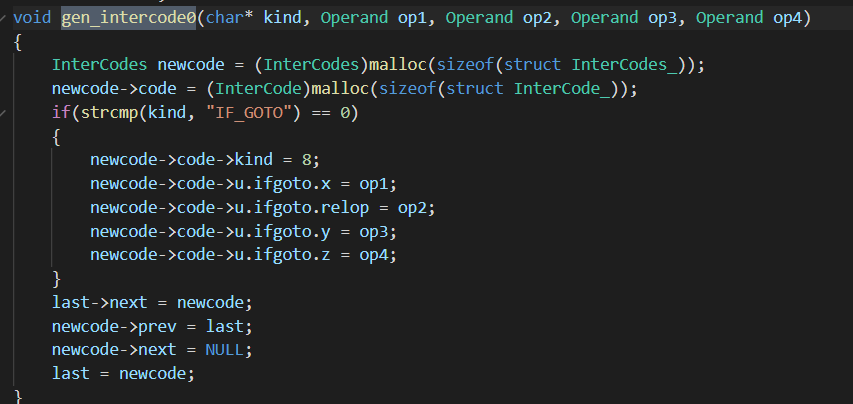
1. **翻译过程**

本来想面向测试用例编程翻译，但是有的翻译函数需要调用子节点的翻译函数，在编写每个翻译函数中，我发现一直有问题，所以最后还是保证了所有的语法单元都需要由对应的翻译函数。在定义操作的时候，我们需要定义一个操作数并赋予字段值，在生成一句中间代码是，我们需要调用对应的gen\_intercode()函数将新生成的中间代码加入到InterCodes这个链表中，比如在translate\_FunDec函数中就需要生成Function f代码，即调用gen\_intercode3(“FUNCTION”, func);函数部分如果所示：



1. **个性内容**

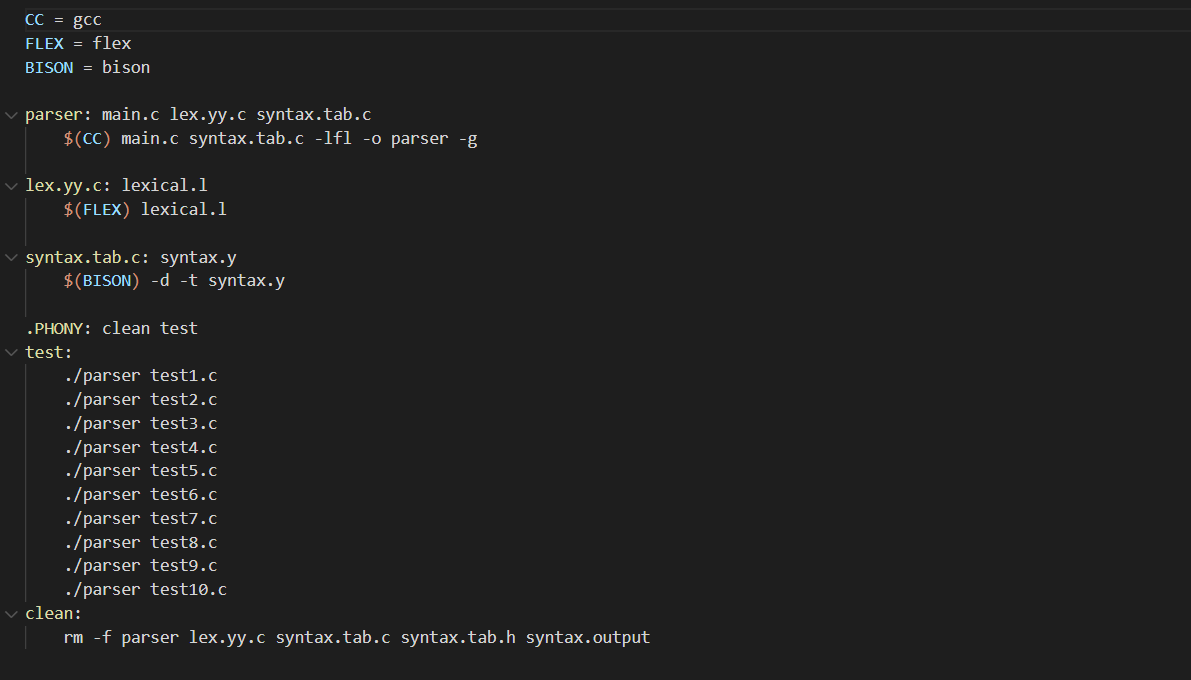
特别的，我在各个翻译函数生成对应中间代码之前，我编写了四个不同的中间代码生成函数，以此先根据操作数个数先统一进行函数调用，后续在通过对应的语法单元执行对应的生成函数。这些函数的功能在于能够根据传入的中间代码种类和操作数生成一个新的中间代码节点加入到InterCodes中。我们以四个参数的if\_goto()语句为例，如下图：



为了能够完好的连接所有的中间代码节点，我这里使用了geninterCodeList()生成一个中间代码链表的首节点，然后将它指向last，表明也同时指向尾节点。然后在所有生成中间代码函数中将新节点赋给last->next字段，last赋给newcode->prev字段，将newcode->next赋值为空，这样新节点作为尾节点继续延伸下去。

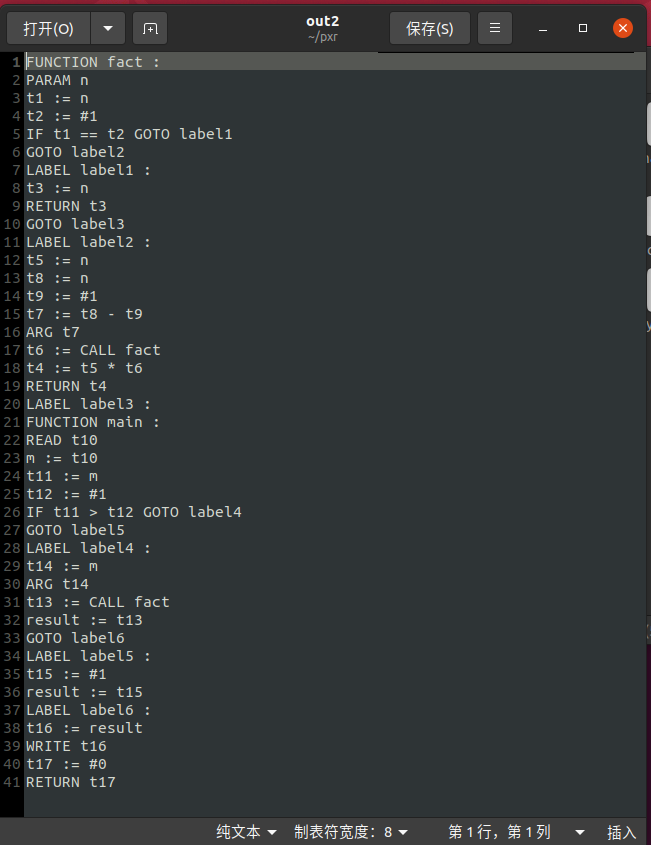
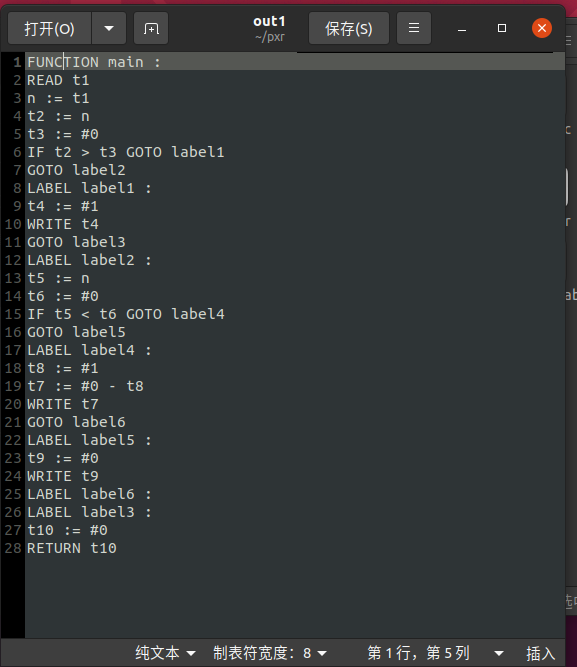
1. **编译过程**

本次实验采用Makefile进行编译。



* 1. 实验结果

在终端运行./parser test1.cmm out1和./parser test2.cmm out2



1. 实验总结
2. 学会了如何实现简单的中间代码生成。