**哈尔滨工业大学**

**编译系统 2022春**

**实验二**

|  |  |
| --- | --- |
| 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. 巩固对词法分析与语法分析的基本功能和原理的认识
   2. 能够应用自动机的知识进行词法与语法分析
   3. 理解并处理词法分析与语法分析中的异常和错误
2. 实验内容
   1. 实验环境

Ubuntu 12.04, kernel version 3.2.0-29；GCC version 4.6.3

GNU Flex version 2.5.35；GUN Bison version 2.5

* 1. 实验过程

1. **实现功能**

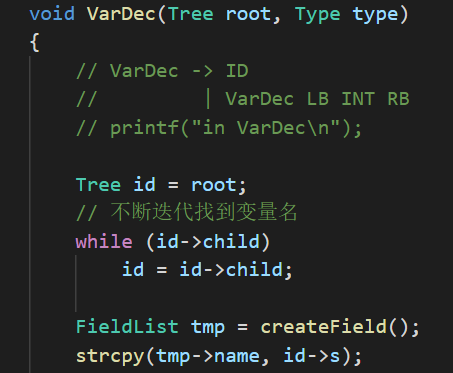
本次实验实现了必做的错误检测（未标黄项目）。以及部分标黄的选做内容。

1. **数据结构**

本次实验主要用到的数据结构包括变量类型Type，还有字符表中的一个个域FieldList。这一部分的书写参照实验指导书即可。

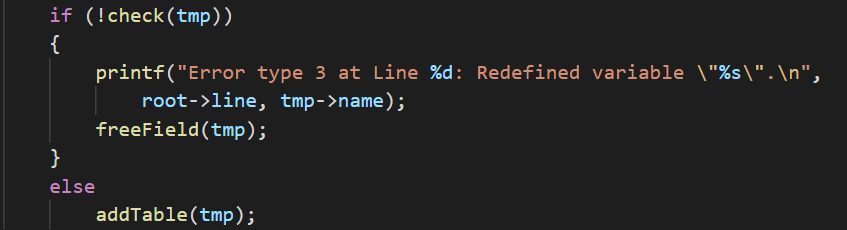
1. **语义分析**

语义分析是在前两次的实验基础上完成。大致流程就是在上一次成功建立语法树之后，再遍历一遍树，遍历树的过程中检查语义错误。我以几类错误检测为例，简单讲一下语义分析过程。

**3.1变量出现重复定义或变量定义过的的结构体名字重复**。如语法单元VarDec，给出的语法规则为VarDec→ID| VarDec LB INT RB。这是一个申明语句，可以申明一个普通的变量名或者一个数组。当语法树遍历到这个语法单元时，我们构造函数void VarDec(Tree root, Type type)，形参是此时的根节点和综合属性类型。然后我们通过儿子和兄弟的遍历这棵子树找到他的下一个叶节点。并且申请一个域用来存放下一个叶节点（也可能是子树）的信息。

如果是ID，那么我们就将ID的信息放入这个域tmp。如果产生式是VarDec LB INT RB，那么说明此时是一个数组类型，同理，我们创建一个数组变量信息放入这个域tmp。

最后，我们搜索我们之前创建的符号表是否包含tmp这个域，如果有则说明之前已经定义该变量名，程序输出Redefined variable ‘tmp->name’。如果符号表中没有该信息，我们则将该变量加入符号表。



**3.2赋值号两边的类型表达式不匹配**或者**赋值号左边只有一个右值的表达式**这类错误。对应于语法分析的EXP→Exp ASSIGNOP EXP这一部分。我们同样声明一个域，当匹配到root->child->brother->unit为ASSIGNOP时，我们进行错误检测，首先判断左边是否是INT或是FLOAT，如果是，则形如1=…这样的形式，这显然时左边是一个右值表达式的错误，输出**错误类型5**。如果左边id的basic和右边id的basic不一样说明两边表达式类型不匹配，说明不可进行赋值操作，输出**错误类型6**。

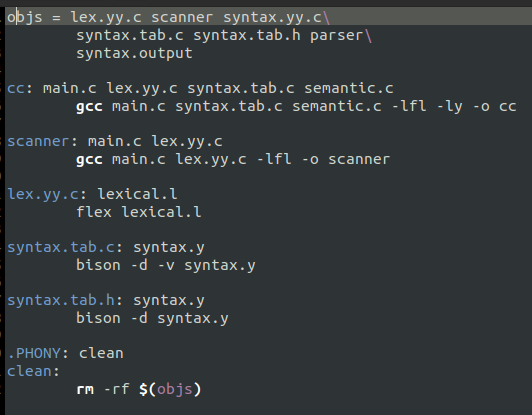
**3.3操作数类型不匹配或操作数类型与操作符不匹配。**在进行移入-规约操作时，即对应Exp→Exp PLUS Exp| Exp MINUS Exp时，程序对运算表达式两侧的表达式类型进行检验。若两侧表达式类型不相等则输出**错误类型7**，类似上面的赋值号两边不等。

**3.4数组访问操作符[.]中出现非整数**。每遇到数组访问语句的时候，对应于Exp→Exp LB Exp RB，程序对字节点第三个属性值root->child->brother->brother,若不为int，说明出现非整数，输出**错误类型12**。

**3.5对非数组型变量使用数组访问操作符**。同样对子节点的第三个属性值进行判断，若它的kind不为数组，则说明该节点对应的是非数组变量，输出**错误类型10**。

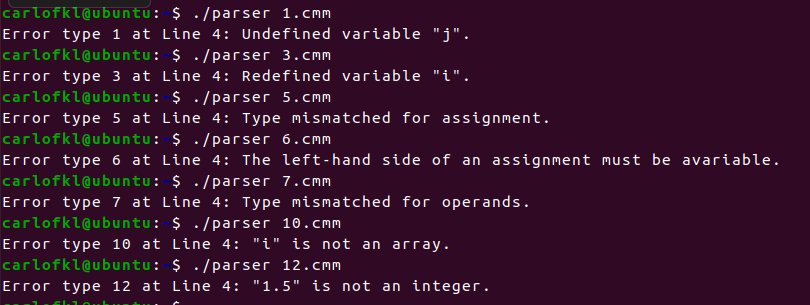
1. **编译过程**

本次实验采用Makefile进行编译。具体内容如下图所示：

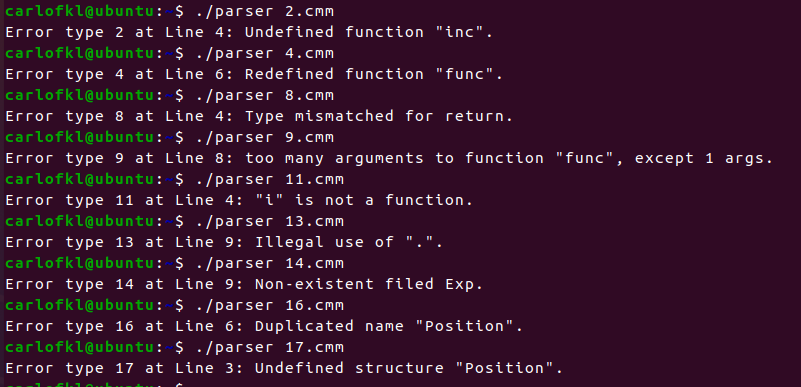


* 1. 实验结果

必做内容的几个结果：



选做内容的几个结果：



1. 实验总结
2. 学会了如何实现简单的语义分析任务。