Lab 2 实验报告：

一．主要文件说明：

在 lab1 的基础上，主要增加了 Symboltable.h/.c，Semantic.h/.c 四个文件，对 SyntaxTree.h/c，lexical.l，main.c 五个文件进行了修改。

1. Symboltable.h/.c：针对变量，类型，自定义结构体，函数等给出了封装好的结构体进行定义，并定义了相应的哈希表，用于存储变量表、函数表、结构体表等。
2. Semantic.h/.c：主要设计 dfs 函数，针对语法树上的每个节点，以及这个节点对应的产生式，去设计他的子节点遍历顺序，并计算继承属性、综合属性。
3. SyntaxTree.h/c：对于语法树上的每个节点的结构进行了修改，在节点上加入了这个节点的继承、综合属性，这个节点对应的产生式。
4. Lexical.l：对每个产生式进行标号，在创建语法树的节点时，在节点中记录对应的产生式的标号。
5. Main.c：在词法/语法分析不报错后，得到语法分析树，并从语法分析树的根节点开始对语法树进行 dfs，在 dfs 的过程中进行语义分析。

二．编译测试条件：

未对 makefile 文件进行修改，编译指令如下：

```
cd ./Code
make clean
make
./parser ../Test/test1.cmm
```

三．核心功能实现：

1. 符号表的存储：

（1）针对变量：对于一个变量，其类型主要设计 3 个属性：变量基本类型，变量的维度，变量的结构体编号（如果基本类型是结构体的话）。对于变量的存储，只需要在变量类型的基础上加入变量名，使用变量名去唯一标识一个变量。我们将变量名看作一个不超过 40 长度的字符串，通过字符串哈希的方式制作哈希表，在哈希表中存储变量。对于变量表的操作主要为：初始化、插入一个变量、根据变量名去寻找一个变量。

（2）针对结构体：对于一个自定义的结构体，其主要有如下 3 个属性：1.结构体的名称（可以为空），2.结构体的 ID（我们给每一个结构体分配一个 ID，用于唯一确定这个结构体，由于结构体名称可以为空，因此对于空名称的结构体，无法仅用名称去唯一标识），3.结构体域中的变量，我们使用一个变量类型的链表进行存储。同样的，我们将结构体名看作一个不超过 40 长度的字符串，通过字符串哈希的方式制作哈希表，在哈希表中存储变量，通过哈希表实现高效的名称到结构体的映射。由于定义结构体时，空类型的结构体，不会再后续重新用于定义变量，因此我们可以在哈希表中不记录这些结构体。但我们仍然要给这些结构体分配 ID，存放在一个新的表中，用普通的数组的数字下标对应结构体的 ID。
（3）针对函数：对于一个函数，其主要有如下 3 个属性：1.函数的名称，2.函数的返回类型，3.函数的参数列表。类似的，我们用一个变量类型的链表存储函数的参数列表。

2. 语法符号的属性：

对于每个语法符号，我们在其语法分析树的对应节点上增加其综合属性与继承属性。我们在设计属性时，设计为 L 属性，以确保可以通过 dfs 遍历求出属性的值。属性的定义如下：

```
typedef struct attribute
{
    ps_link* ph;
    style_link *sh;
    int sign;
    char name[40];
    int in_statement;//Used to determine if a variable is within a function declaration
}attribute;
```

（1）其中 ph 是变量类型的链表（包括变量基本类型，变量的维度，变量的结构体编号）。其主要用法如下：

1）作为继承属性时：定义变量时，作为 VarDec 的变量类型。定义函数时，在函数体中传递函数的返回值。

2）作为综合属性时，记录 Specifiers 最终产生的类型；EXP 的类型。

（2）sh 为变量的链表（在变量类型的基础上加入变量名）。其主要用法如下：

1）作为继承属性时，主要用于在结构体内部的变量定义语句中，记录结构体内已经定义了哪些变量，主要用于避免结构体域内的域名重复。

2）作为综合属性时，记录函数的 FunDec 中定义的变量、结构体中定义的变量、Varlist 中定义的变量。

（3）Sign 用于标记。

1）作为继承属性时，在 Deflist 中，标记是否为结构体内部的变量定义。

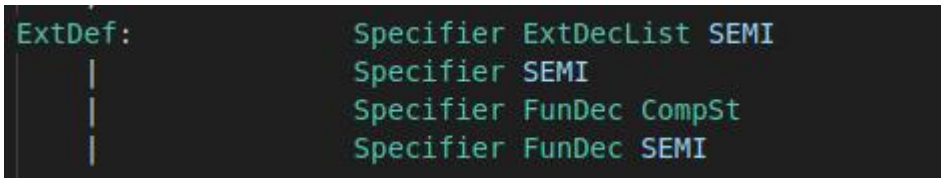
2）用于综合属性时，在 EXP 中标记是否为右值表达式。

（4）Name：在产生式的某个部分可以推导出 ID 时，使用 name 去记录 ID。

（5）in_statement：同 Sign，用于标记，主要在 FunDec 中标记是声明还是定义。

3. 对扩展功能 3.1（函数声明）的支持：

（1）修改产生式：新加入针对函数声明的产生式



（2）修改符号表中的函数表：将函数表拆分为声明表和定义表，一个函数在定义时同时加入声明表、定义表，在声明时加入且仅加入声明表。

（3）在 dfs 后对声明表进行检查，检查有哪些函数声明了但没有定义。