

# 编译原理实验报告

词义分析

121220100 王梓轩

## 完成的功能点：

1. 基本功能：包括检查变量未定义、重定义，操作数的匹配，结构体使用等各种错误。

## 编译及运行方法：

依次输入：（XXX 为测试用的 C 文件，与 parser 在同一文件夹下）

```
make
./parser XXX
```

## 符号结构：

符号表中每个元素的结构及意义：

```
struct Item_
{
    char* name;          //该符号的名称
    int kind;            //该符号的类型
    union
    {
        Var_* v;         //普通变量
        Func_* f;        //函数变量
        FieldList_* fl;   //域中的变量
        Type_* type;      //类型变量
    }u;                 //联合体
    Item_* next;         //指向符号表中的下一个元素
};
```

## 亮点：

1. 对于树的遍历来说，从 Program 节点开始，树中的每个父节点调用子节点的对应函数。这样使得子节点可以继承父节点的属性，并可以在父节点的函数内传递子节点的兄弟调用函数的返回值，实现继承属性。
2. 针对每个节点均需完整考虑是否可能会在这个节点出现错误，可能会出现哪些错误，并处理所有可能出现的错误。
3. 代码中用了一些 assert 语句，使得调试时便于快速找到出错所在的节点。
4. 符号表使用哈希表，这种数据结构有利于查找和增加，其平均时间复杂度均为  $O(1)$ 。并且利用的是链地址的哈希，不会出现符号过多哈希表中无法存放的情况。
5. 符号表中用 Item\_ 结构来表示一个符号，涵盖符号的名称、类型、具体的变量情况等。Exp 节点返回专门的数据结构 ExpReturnType，用来判断返回的类型。