# 编译原理实验二报告

周羽萱 211220074 1813156367@qq.com

## 一. 实现功能

所有的必做+第 3 个选做功能，具体如下：

(1) 在实验一词法分析和语法分析均正确后，自顶向下对程序进行语义分析，主要包含函数、结构体、数组定义使用检查。
(2) 维护符号表，完成符号信息的添加、类型检查等。
(3) 完成讲义中 17 种错误类型检测。
(4) 完成第 3 个选做功能：在“结构体间的类型等价机制采用按结构等价的方式”这个设定下完成语义分析。

亮点：
可以报出代码错误出现的模块，即使是同一种错误报错信息也有所不同，给出更多提示信息。
如对于样例：

```
int main()
{
float j;
10 + j;
10 || j;
10 - j;
}
```

报出提示错误

```
ubuntu@VM-4-8-ubuntu:~/compilers/lab2/code$ ./parser test.cmm
Error type 7 at Line 4: PLUS: operand type can't match.
Error type 7 at Line 5: OR: operand type can't match.
Error type 7 at Line 6: MINUS: operand type can't match.
```

（在加、或、减时发生了操作符类型不匹配的问题）

## 二. 程序编译

1. gcc -w main.c syntax.tab.c common.c semantic.c symbol_tab.c -lfl -ly -o parser
2. ./parser test.cmm

## 三. 代码实现

1. 符号表

关于符号表的代码我主要在 symbol_tab.h 和 symbol_tab.c 中完成，设计函数如下：

```c
unsigned int hash_pjw(char* name);//讲义上的哈希函数
void init_symbol_table();//初始化符号表
void insert(FieldList f);//插入一个符号
FieldList find(char* name);//查找一个符号
```

2. 数据结构

按照讲义和 PPT 上的提示，我设定的数据结构如下：

```c
struct FieldList_
{
  char* name; // 域的名字
  Type type; // 域的类型
  FieldList tail; // 下一个域
  int alive;//是否还存在
};

typedef struct {
    FieldList f;
    struct HashNode *next;
} HashNode;

typedef struct {
    HashNode* table[HASH_TABLE_SIZE+10];
} SymbolTable;

SymbolTable st;//全局符号表
```

```c
struct Function_{
  char* name;
  int line;
  Type ret;
  FieldList param;
};

struct Type_
{
  enum { BASIC=1, ARRAY=2, STRUCTURE=3 , FUNCTION=4 } kind;
  union
  {
    // 基本类型
    int basic;//用0表示int,用1表示float
    // 数组类型信息包括元素类型与数组大小构成
    struct {Type elem; int size; } array;
    // 结构体类型信息是一个链表
    FieldList structure;
    Function function;
  } u;
  enum {
    LEFT, // left value
    RIGHT, // right value
    BOTH // left | right value
  } assign;
} ;
```

基本和 PPT 上的内容一样，除了 structure 的类型做了一些小修改。

3. 语义分析：

a. 学习 ppt,对每个生成式分析，基于 lab1 的语法树展开。总的来说，即从语法树中获得信息,如果涉及到符号，就进行填表、查表等操作。如果出现讲义上的问题，就及时报错。

b. 语法分析部分我基本都采用递归完成。

## 四． 总结与感悟

1. 本次实验涉及到很多指针的操作，稍不留神就会 segmentation fault。我的教训：每个指针在用之前一定要判断是否为 NULL！
2. 有时程序中的一个错误会报多个不同类别的错误，这是由于我设计不当导致的，报错的顺序和时间点需要仔细思考。
3. 通过本次实验还发现了 lab1 的一个错误，说明完成实验时还是应该更细致一些，不能过了就万事大吉，如果可以，还是应该多设计一些样例。
4. 感谢助教哥哥给的 ppt 和讲义，给了我很多思路，帮助很大！

感谢助教哥哥批改！