

# Lab2实验报告

---

## 实现功能

语义分析，符号表增删查改和类型检查。

## 定义

### 类型Type

```
TypeDef struct Type {
    enum { eINT,
           eFLOAT,
           ARRAY,
           STRUCTURE,
           STRUCTVAR,
           FUNC,
           WRONGFUNC } kind;
    union {
        int basic;
        struct
        {
            Type* elem;
        } array;
        FieldList* structure;
        FieldList* structvar;
        struct
        {
            Type* ret;
            int argc;
            FieldList* args;
        } func;
    } u;
}Type;
```

**Type** 结构体表示变量或函数的类型。

**kind** 枚举类型表示具体的类型，如整数、浮点数、数组、结构体、函数等。

**u** 联合体根据 **kind** 的值存储不同类型的具体信息。

符号域 **FieldList\_**

```
typedef struct FieldList {
    char* name;
    Type* type;
    FieldList* tail;
}FieldList;
```

符号表中的一个条目 **TableList\_**

```
typedef struct TableList {
    char* name;
    Type* type;
    Operand* op;
    TableList* next;
}TableList;
```

符号表：哈希表，开放散列处理冲突

```
TableList* hashTable[HASHSIZE + 1];
```

表的初始化，添加和搜索操作

```
unsigned hash(char* name);
void initTable();
TableList* search(char* name);
void insert(TableList* item);
```

基本思路

遍历语法树，给语法树的每个节点作为函数，在上面进行相应的判断和操作

遇到定义添加符号表项，判断重复和错误定义

遇到调用检查符号表项，检查类型，判断错误调用

选做一

添加产生式，

```
ExtDef -> Specifier FunDec SEMI
```

并区分函数定义和函数声明

```
FUNC -> FUNCdef, FUNCsem
```

在遇到函数时检查是声明还是定义，对应处理

如何编译

使用群里发的Makefile