

# lab2 - 语义分析

---

## 1. 需求

---

建立**类型系统**, 实现**嵌套**作用域, 通过**符号表**收集上下文无关文法无法处理的语义信息, 解析符号引用进行**语义错误检查**, 并为后续**代码生成**收集必要信息.

## 2. 设计

---

### 2.1 类型系统

- 借鉴手册中的Type类型, 作为后续Symbol结构体中的字段的类型.

### 2.2 符号表

基于结构体的继承实现多态(以xv6 shell中的cmd为例: pipecmd, redircmd...)

- 实现varSymbol, funcSymbol, structSymbol, typeSymbol, 均继承自Symbol基类, 各子类中有各自**特定需要的符号信息**(包括上述类型系统的成员)
- 基于hashmap实现符号表, 符号表中存储了Symbol

### 2.3 嵌套作用域

- 实现globalScope, funcScope, localScope, structScope, 继承自Scope基类, Scope中包含了**本作用域的符号表**
- 通过栈来模拟遍历语法分析树的过程中, **作用域的更迭**

### 2.4 语义错误检查

- 遍历语法树的过程中, 遇到全局变量, 结构体, 函数的**定义**就主要做**符号表填表**的操作, 以及少数填表前的查表

- 遇到Exp, 就主要根据手册中的错误类型, 进行分类讨论, **查表并对Exp进行解析**, 遇到错误的情况就输出错误信息
- 做一定的**错误恢复**(当作没有语义错误照常处理), 防止后续代码**再次引用**时出现**False Positive**

## 3. 编译运行

---

- include目录下包含了头文件, src目录下包含源文件, make即可