

## 编译原理 实验 3 实验报告

201220062 黄子睿

### 一、实验功能简介：

- a) 实验实现了 c—语言要求的所有中间代码生成，并通过了 OJ 平台上的所有测试用例。
- b) 选作部分完成要求如下：
  - i. 要求 3.1:
    - 1. 可以出现结构体类型的变量，但是不存在结构体赋值，同时结构体作为参数为地址传递；
    - 2. 结构体类型的变量可以作为函数的值，但函数返回值不作为结构体
  - ii. 要求 3.2:
    - 1. 一维数组类型的变量可以作为函数参数(但函数不会返回一维数组类型的值)。
    - 2. 可以出现高维数组类型的变量(但高维数组类型的变量不会作为函数的参数或返回类值)
- c) 下面简要叙述一下语义分析的代码实现方式：
  - i. 准备工作：首先对实验 2 的内容进行修改，修改了上个实验中的部分 Bug，同时考虑到每阶段实验不存在重复变量名，因此为了向实验三提供符号表，注释掉实验二中 traverseCompSt 结尾部分删除本函数定义变量的语句，同时将函数的参数实现加入表中。为了实现对 read 与 write 函数的调用，在 initTable 中事先加入代表 read 与 write 的 item
  - ii. 中间代码生成：(实验三主题代码在 inter.h 与 inter.c 文件中)
    - 1. 由于每次实验没有全体变量，因此实验三功能通过归纳式 Def -> Specifier FunDec CompSt 进入 translateFuncDec，translateCompSt 函数实现对函数的中间代码生成。
    - 2. 生成的中间代码由双向链表构成，记录在结构体 InterCodeList 中，每条三地址指令由 InterCode 结构体表示，其中用成员变量 kind 标注该条指令的类型，并用一个联合 u 记录指令用到的 pOperand。pOperand 指向结构体 Operand，记录三地址变量每个部分的临时变量名或常量值。
    - 3. 沿着语法树遍历，依次生成中间代码。

### 二、编译方式介绍：

- a) 提交的文件结构为：
  - i. Code
  - ii. Report.pdf
- b) 代码的运行在 Code 文件夹下，通过指令 make 生成./parser；通过指令./parser <source file> <destination file>将<source file>中的代码翻译到<destination file>中。