2019年11月25日 星期一

# 编译原理实验三　实验报告

171860689 蔡跃龙 857570220@qq.com

171860673 林昊 1184264181@qq.com

---

## 实验内容

本次实验任务是在前两次实验的基础上编写一个C语言程序，将C—源代码翻译为线性结构中间代码并用虚拟机小程序来测试中间代码的运行结果。同时要求完成选做3.2——完成一维数组类型传参和高维数组类型变量。另外需要完成初步的代码优化。

---

## 程序实现功能以及过程

- 数据结构与文件接口

数据结构方面，设计了用于区分不同操作符的Operand结构体：它包含kind和u两个域，分别存储指明操作符的类型和对应类型的信息；用于区分不同中间代码的InterCode结构体：kind、u、prev、next域，分别用于存储中间代码的类型、中间代码结构信息和前向、后向指针。整个程序的中间代码线性序列以双向链表组织。

文件接口方面，将从命令行传入的第2、3个参数分别作为输入的测试文件路径和输出的中间代码文件路径。注意到文件路径传入main函数的argv参数后需要对文件进行安全性检查，才能传入PrintCode函数将已经组织好的，连接所有中间代码的双向链表按照格式要求输出到.ir文件当中。

- 中间代码生成

中间代码生成是此次实验的主要部分，这一部分基本是按照书上的伪代码完成，为了实现变量的对应，在符号表里添加了Operand属性，使得每个变量只对应一个no，例如在第一次使用变量i时，就生成一个v_k（k是当前的计数），这样变量就只对应一个v，也不会重复，后面使用这个变量时直接查找符号表，返回对应的v即可。

- 选做要求

另外需要考虑完成选做3.2：对于数组来说，分两种情况，一种是数组作为参数，实际上是地址的传递，需要用到&。另一种比较复杂，是数组变量，即Exp【Exp】d的使用，通过translate_Exp得到【】中的值，然后计算偏移量，当Exp还是数组时，使用循环语句对偏移量进行累加。因此第二种情况在处理数组时，要先将数组每一维的size记录下来，便于对偏移量的计算。

## 程序编译与运行

命令行进入Lab/Code目录，依次键入make clean、make命令并回车即可生成parser处理程序。同时，需要对parser的第三个参数进行指定，才可能进行中间代码生成并生成中间代码序列的.ir输出文件。另外，为了选择.ir文件进行测试，需要切换到irsim.pyc所在目录下，键入python irsim.pyc打开程序才可选择测试文件。

## 程序设计创新

- 代码优化

代码优化部分，主要完成了删除冗余GOTO指令和冗余LABEL的工作。对前者，我们采用遍历中间代码链表的方式，①每发现一个IFGOTO中间代码，就对其和紧跟着的GOTO和LABEL语句进行判断，由判断结果可以选择是否删除第二条GOTO语句及修改第一条IFGOTO语句的条件和跳转目标；②每发现一个GOTO语句，若下一句即为跳转目标，则将该语句删除。对后者，代码中用bitmap数据结构记录存在的GOTO目标和LABEL标号，在两次对中间代码双向链表的遍历中先后记录信息、删除对应位置LABEL存在但GOTO目标不存在的LABLE语句。

- 程序测试

曾编写脚本对27个样例进行程序鲁棒性的批处理检测，提升程序可靠性。