

Lab3 实验报告

201840281 史成璐

实验三的任务是将c--源代码翻译为中间代码，将中间代码输出成线性结构。

我将关于中间代码生成的所有内容写在`generation.h`和`generation.c`中，在语义检查全部完成并通过之后生成中间代码。我没有处理结构体变量。

实现功能

中间代码表示

使用中间代码的线性表示，参考实验指导的提示和数据结构，使用`Operand_`和`InterCode_`结构体。用枚举类型`OPERAND_KIND`和`INTER_CODE_KIND`表示操作数和语句种类，便于分情况处理，枚举类型`Type`表示是否是取址或解引用。

使用双向链表来完成，结构体`InterCodes_`包含一个`InterCode_`结构和前后两个指针，插入和删除的效率 high。

还增加了结构体`Arg_`和`Param`，用于处理函数参数声明以及计算实参，类似链表的形式。

中间代码生成

参照实验指导中的翻译模式，基本上为每个主要的语法单元“X”都设计相应的翻译函数“`translate_X`”

数组的翻译采用递归，为了计算占用的空间，在语义分析`analysis.h`头文件的`Type_`结构体中新增变量`width`表示字节大小，递归时带入公式累计即可。识别最高维的部分，计算偏移量，在最高位对数组名取地址。将数组作为函数参数传递时，有对一个地址取地址的问题，于是在翻译时将地址参数存在一个临时数组中，后续先检查是否已经是地址。

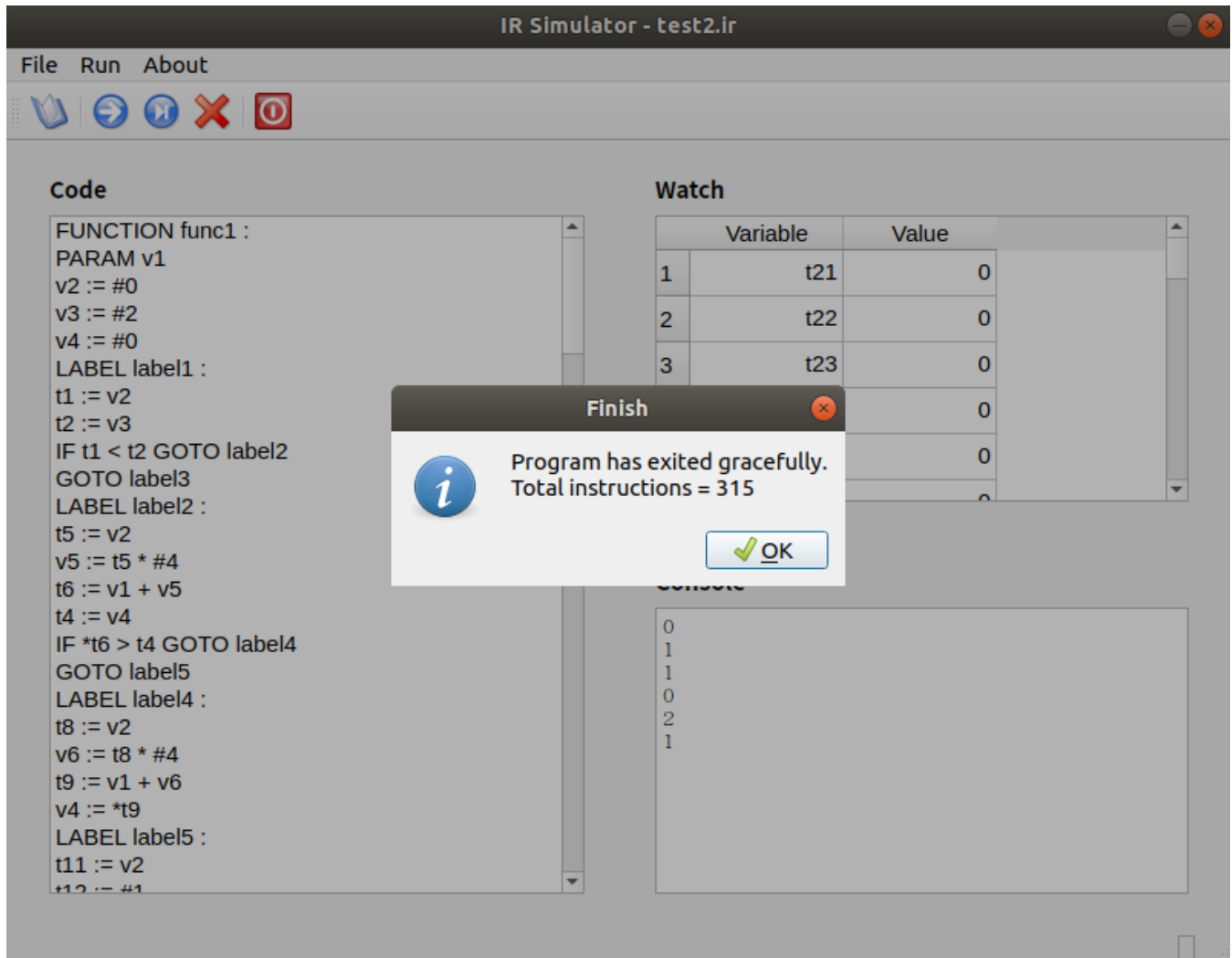
函数参数传递还要注意变量的顺序，与声明中相反。

`create_prefix()`函数根据临时变量种类生成变量名称，`write_code()`函数将中间代码按照格式写入文件中。

编译测试

可以直接使用Makefile，执行make命令即可生成parser程序。

正确实现了实验三PDF中的测试样例。



群内某测试文件，使用IR Simulator检验如图，仍有很大优化空间。

在函数的翻译中，留有一些问题，提示函数声明和定义参数不匹配，与语义分析有关，还未修复。