实验三 中间代码生成

161220033 刁含顺 161220055 金亦凡

一、 实现功能

1、在词法分析、语法分析和语义分析程序的基础上，将 C--源代码翻译为中间代码。打印到终端并且写入一个测试文件 out.ir，同时可以在虚拟机小程序输出结果

2、本次实验根据通过遍历实验一中实现的语法树中的每一个结点产生出中间代码，然后将中间代码按照输出格式打印出来。我们为每个主要的语法单元“X”都设计相应的翻译函数“translate_X”，为语法树中出现的特定结构，生成相应的中间代码。根据附录中的内容，除了实验指导中提示的函数外，还实现了所有必须的翻译函数

3、本次实验在实现必做的基础上完成了选做3.1

选做实现了单层的结构体

二、数据结构

（1）如下所示定义了比较运算符类型、操作数类型以及词法类型

```
enum R_KIND{
        G, GE, L, LE, E, NE,
};

enum O_KIND{
        VARIABLE, CONSTANT, TEMP, O_LABEL,
};

enum I_KIND{
        LABEL, FUNC, ASSIGN, ADD,
        SUB, MUL, DIV, CITE,
        GETPOINTER, ASSIGNPOINTER, GOTO, IFGOTO,
        RETURN, DEC, ARG, CALL,
        PARAM, READ, WRITE,
};
```

（2）定义了操作数的struct结构，分别存放操作数类型、变量以及标签

```
struct Operand_ {
    enum O_KIND kind;
    union {
        int var_no;
        int value;
        // char name[20];
        struct {
            int temp_no;
            char name[5];
            Operand prev;
            Operand next;
        } temp;
        struct {
            int label_no;
            char name[5];
        } label;
    } u;
};
```

（3）定义用于生成线形中间代码的结构体，这是一个双向链表，定义kind以及不同种类的操作数形式

```
struct InterCode_ {
    enum I_KIND kind;
    union {
        struct { char name[20]; } func;
        struct { Operand op; } single;
        struct { Operand result, op; } assign;
        struct { Operand result, op1, op2; } binop;
        struct { Operand re1, re2, label; enum R_KIND kind; } ifgoto;
        struct { Operand dec; int size; } dec;
        struct { Operand ret; char name[20]; } call;
    } u;
    InterCode prev;
    InterCode next;
};
```

三、运行方式

(1)修改Makefile中的

test:

./parser ../Test/test1.cmm

部分，目前默认测试文件为**test1.cmm**，可以将待测试代码放入这个文件中

(2)make clean

(3)make

(4)make test

即可在终端打印出中间代码

```
FUNCTION main :
READ t1
n := t1
t2 := #0
IF n > t2 GOTO label1
GOTO label2
LABEL label1 :
t4 := #1
WRITE t4
GOTO label3
LABEL label2 :
t5 := #0
IF n < t5 GOTO label4
GOTO label5
LABEL label4 :
t8 := #1
t7 := #0 - t8
WRITE t7
GOTO label6
LABEL label5 :
t10 := #0
WRITE t10
LABEL label6 :
LABEL label3 :
t11 := #0
RETURN t11
dhs@debian:~/compilers/lab3/Code$
```

同时生成了文件**out.ir**用于存放中间代码

根据实验要求，可以使用指令**python irsim.pyc**执行虚拟小程序，中间代码在虚拟小程序中运行正确