编译原理 lab3 实验报告　201220180 李全昊

一. 实验的功能及实验内容

1. 实现的功能：任务是在词法分析、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

2. 实验内容：

首先在 lab2 的基础上我在符号表中加入了 read 和 write 函数；

并且我将 exit_domain 函数直接 return，不删除定义域中的定义变量（因为这次实验不会有任何重名）这样在后续在语法树的基础上进行生成中间代码的时候可以找到对应的 symbol；

整体数据结构的设计：采用双向链表的形式存储所有的中间代码，每一个中间代码是一个 Intercodes 指针，被分类为 4 种形式（1-4 个操作数）。

在打印中间代码到文件的时候，我定义了一个 void printIntercode(FILE *file)函数，负责向 file 中写入我生成的双向链表中的中间代码，它会根据中间代码的类型相应的打印中间代码的形式（比如如果是 Arg 就会打印 Arg）；

在这个打印函数中，还会调用 void printOP(Operand op, FILE *file)函数来打印中间代码中的操作数；操作数也有几种类型：常数，temp 变量，variable，function，以及 label 变量；这几种 operand 有不同的打印方法，其中值得一提的是上面的每一种 operand 都有自己的 type，也就是 Address_and Address_star 以及 Val，分别对应&,*以及值传递，这样方便打印函数知道应该怎么打印这个 operand

整体的生成中间代码的过程采用递归函数调用来实现，大部分参考着 pdf 中给出的 translate 函数

最后需要说明的一个点是在实现 translate_Exp 函数的时候，我给他设置了一个 isleft 函数，代表当前这个 exp 是在等号的左边还是右边。这一点是我在实现 exp 翻译的时候的非常重要的一个参考因素。translate_Exp 这个函数返回的是一个 operand，这个 operand 的 type 对应的是当前这个操作数中保存的是地址还是一个值。然后在递归调用 translate_Exp 的时候，我会根据当前的 isleft 属性来确定写入的中间代码是 Address_and，Address_star 还是 Val。

二. 实验总结

这次试验经过实验二的摸爬滚打还算完成的比较顺利，收获的也不少，对生成中间代码有了不少的了解