编译原理实验报告

实验名称：实验一 词法分析与语法分析

指导教师：许畅

实验时间：2023.4.27

任务号：14

报告人：

分配	姓名	学号
组长	张铭俊	201220065
组员	吴浩然	201220064

一、程序功能

必做内容

- 你的程序需要将符合假设的C--源代码翻译为中间代码，输出至文件中。

选做内容

- 修改前面对C--源代码的假设2和3，使源代码中：
  a) 可以出现结构体类型的变量（但不会有结构体变量之间直接赋值）。
  b) 结构体类型的变量可以作为函数的参数（但函数不会返回结构体类型的值）。

二、程序运行

1. 程序划分

程序在实验三中新增加了以下几个文件

- inter.h：存放中间代码部分相关的数据结构定义以及函数声明，包括操作数operand，中间代码interCode，参数列表arg等等数据机构以及translateexp等等的函数声明。
- inter.c: 中间代码部分相关的实现部分，包括各个函数的实际实现等等。

2. 程序实现

1. 线性链表

   此次中间代码部分选择采用双向链表的方式来实现，每个链表节点连接一个中间代码语句，操作数及中间代码结构与PDF所述一致。

```
// 中间代码列表结点
typedef struct _interCodes
{
    pInterCode code;
    pInterCodes prev, next;
} InterCodes;

// 中间代码列表
```

```
typedef struct _interCodeList
{
    pInterCodes head, tail;
    Type arrType;
    int norVarCnt;
    int tmpVarCnt;
    int labelCnt;
} InterCodeList;
```

2. 输出方式

   通过在语法分析语义分析完成后，通过遍历生成的语法分析树，从上到下的进行遍历进而输出中间代码。

```
if (iftrue)//不存在语法词法错误
    {
        if (firstNode != NULL)
        {
            dfs(firstNode, 0);
            interCodeList = newInterCodeList();
            genInterCodes(firstNode);
            printInterCode(fw, interCodeList);
        }
    }
```

3. 对于不应完成选做部分

   为了使得在遇到在不该完成的选做3-2用例时，能够不生成.ir文件并防止段错误，在遇到高维数组的情况，输出错误信息至控制台并直接exit(0)退出。

```
// VarDec LB INT RB
        if(getArray(child)->type->u.array.elem->kind != ARRAY)
            translateVarDec(child, place);
        else
        {
            printf("Cannot translate: Code contains variables of multi-
dimensional array type or parameters of array type.\n");
            exit(0);
        }
```

4. 程序打包结构

   -- Code

   -- makefile

   -- inter.h

   -- inter.c

   -- ……

   -- report.pdf

5. 编译方式

   编译的三条指令如下

```
bison -d syntax.y
flex lexical.l
gcc syntax.tab.c main.c tree.c inter.c -lfl -ly -o parser
```

在默认test.cmm在Code同目录下时，执行如下指令输出中间代码到out.ir中

```
./parser test.cmm out.ir
```