编译原理实验三报告

马珩峻　林立强

一、程序功能

实现了 c--语言的中间代码生成，选做了 3.1 结构体相关的中间代码生成，Operand 和 interCode 的结构体定义参考了讲义中给出的建议，不再细说，然后采用了链表式的 IR，

```
typedef struct _interCodes {
    pInterCode code;
    pInterCodes *prev, *next;
} InterCodes;
```

```
typedef struct _interCodeList {
    pInterCodes head;
    pInterCodes cur;
    char* lastArrayName;  // 针对结构体数组，因为需要数组名查表
    int tempVarNum;
    int labelNum;
} InterCodeList;
```

因为 3.1 虽然允许结构体的定义与函数参数传递，但是并未允许结构体的直接赋值，再根据示例可以看出 c--的结构体作为参数传递时是传址的，这与 c 语言的定义有区别，需要注意。

另外对中间代码的生成没有做太多优化，主要对直接使用的符号和立即数进行了一步去掉一步创建新临时变量的优化。

因为翻译规则在讲义中已经明确给出，主要工作是数据结构的设计，相关接口的具体实现，同实验二中间代码的生成也是进入 ExtDefList 后的事，另外更具我们的诸多假设，可以在这里选择不在语义分析时候生成 IR，而是语义分析后不删除符号表再次遍历语法树生成 IR，所以再次遍历语法树寻找 ExtDefList 节点然后开始处理

二、

因为不允许修改 makefile 但本次实验中./parse 相较而言需要多一个输出文件的参数，所以测试时是自己手动输入的，同样 make 编译，然后./parse 输入文件 输出文件 执行程序