

一．程序功能

完成了实验的基本功能，选做了结构体的实现；

Operand 的实现参考了指导书，定义了六种基本操作对象

```
//操作对象
typedef struct _operand {
    enum {
        OP_VARIABLE,
        OP_CONSTANT,
        OP_ADDRESS,
        OP_LABEL,
        OP_FUNCTION,
        OP_RELOP,
    } kind;

    union {
        int value;
        char* name;
    } u;
} Operand;
```

中间代码的结构体也参照了指导书，没有太多需要讲的

```
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;
```

关于中间代码的优化：主要实现了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

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．程序运行

Code 目录下：Make 编译后

利用两个脚本完成自动测试：

```
$ parse.sh
1  mkdir -p out2
2  file_name=$(basename $1)
3  ./Code/parser $1 ./out2/${file_name}.ir
4
```

```
$ auto-test.sh
1  for file in ./test2/*
2      do
3          echo Testing with $file
4          ./parse.sh $file
5      done
```