编译原理实验报告

实验内容：中间代码生成

组长：杨月洋　131220026

组员：林伯宏　131220029

任务编号：17　　联系方式：dxqzyyy@163.com

一、 实现的功能：

针对一份 C--源代码，在词法分析、语法分析、语义分析的基础上，将源代码翻译成中间代码，并以三地址代码形式打印在输出文件上。该中间代码文件可以通过虚拟机小程序运行，并获得预期的结果。除此之外，如何优化代码、提高效率也是实验的目标之一。

辅助工具：flex　bison　python-qt　虚拟机小程序

编程语言：C 语言　c99

执行方法：

A. 直接使用目录下的可执行程序 *parser*，输入命令：

./parser　测试文件相对路径　输出文件相对路径

B. 在 Code 目录下输入“make”，通过 Makefile 得到可执行文件 *parser*，再进行测试

二、 实现方法：

本次实验的内容基于实验一和实验二的完成成果，在保证程序没

有语法、语义错误且可以正确运行的前提下，将程序的源代码翻译成具体格式的线型三地址中间代码。主要使用到了实验一生成的语法树和实验二生成的符号表。

在之前的实验中我以“左子女右兄弟”的二叉树表示方法构建了语法树，并通过栈实现二叉树的先序遍历（等同于多叉树的先序遍历）。可知，先序遍历类似于自顶向下的分析方式，可在递归子程序中按序逐步审查每个文法中规定的特定结构。每当发现某个特定的结构 X，便可以执行相应的“翻译模板”translateX（……）。在翻译模板中再根据具体的产生式从而执行不同的翻译方式。

具体的翻译过程我选择“边扫描边生成”的方式，不通过返回值将翻译好的中间代码拼接返回，而是在过程调用中直接将代码插入到三地址代码的存储结构中。这要求翻译的顺序必须严格符合最终顺序，且由于没有临时变量保存临时结果，需保证前后的翻译过程互不影响（例如后续的翻译不能改变之前已翻译好的结果）。相应的翻译模板如实验讲义中所示，没有提供的模板大体思路如下：

变量的定义中仅需翻译数组与结构体的类型，这些变量需要申请相应的内存空间，可以从符号表中查询其类型大小，翻译成”*Dec v1 size”*，但应注意这里的 v1 既不是值也不是地址，而仅是指这个变量，其地址是 *&v1*，值由 **(&v1+offset)* 取出。

变量的初始化类似于“***Exp1 ->Exp2 ASSIGN Exp3***”的翻译。可以知道父节点 Exp1 的类型应与子节点 Exp2 和 Exp3 的类型相同（否则无法通过语义分析），通过递归过程 handleExp()得到 Exp 的类型。若

其类型为数组或结构体，可以报错（由于实验三中规定了没有结构体变量的直接赋值，而数组变量不可以直接赋值），故 Exp 的类型只能为基本数据类型。由于左值限定，子节点 Exp2 只能是 ***id***，***Exp[Exp]***，***id.[Exp]***,三中，按情况考虑。为 id 时直接取值翻译即可，剩下两种情况应该取其地址 addr，最终翻译成 “ **addr = xxx* “的形式。

三、 实验特色：

1.由于本次实验中需要大量的中间代码和操作数，并需设计成结构化的数据，我对每一个结构均设计了一个“池结构”，在运行开始时一次性在堆内申请空间，每次创建一条中间代码或者申请一个临时变量操作数仅需要从池中取出一个空间并初始化即可。如图：

```
InterCodeNode *getInterCodeNode() {
    InterCodeNode *temp;
    if(i_free == NULL) {
        i_free = (InterCodeNode *)malloc(sizeof(InterCodeNode) * i_NUM);
        for(temp = i_free; temp != i_free + i_NUM - 1; temp++) {
            temp->next = temp + 1;
        }
        temp->next = NULL;
    }
    temp = i_free;
    i_free = i_free->next;
    memset(temp,0,sizeof(InterCodeNode));
    return temp;
}
```

2.***函数的形参***与***局部变量***均可能存在于同一张符号表中，我在插入符号表的时候直接赋予一个变量标号 var_no，代表该变量对应 v1 或 v2。但应注意对于数组和结构体变量来说，形参中的 vNum 与局部变量 vNum 是不同的。形参中 vNum 本身代表该变量的地址，而局部变量如前文所说，&vNum 代表其地址。因此我在设计时在符号表中增加了标志位，便于区分形参和局部变量中的 数组或结构体。