**编译原理实验报告**

**------2020200982 闫世杰**

**实验目标:**

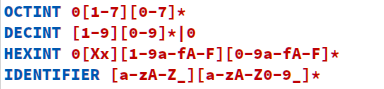
**设计并实现一个基于Sysy语言的编译器**

**实现过程:**

1. **词法分析: 基于flex实现词法分析**

**由于后续语法分析的需要,绝大多数内容都需要进行精准匹配,需要用到正则表达式的地方只有数字和标识符,因此该过程较为简单**

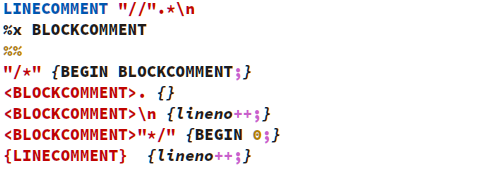
**数字及标识符正则式如下,其余均为精确匹配,此处省去:**



**需要注意的地方是对注释的匹配,个人是利用flex的状态转换功能实现**

**对于单行注释,直接使用正则匹配即可**

**对于块注释,由于可以跨越多行,因此使用状态转换来实现,匹配到/\*时进入注释态,任何匹配都不会进行,匹配到\*/时恢复到正常态,继续进行匹配**



**对于词法分析所匹配到的所有词都将作为一个终结符传递给语法分析器,进行后续的工作**

1. **语法分析: 基于bison(yacc)对匹配的词组进行语法分析**

**只需要写出Sysy的语法规则,并给出相关的语义动作.**

**具体的语法规则在文档中都已经给出了,只需要稍加改进,实现其中包含的正则匹配式,同时定义相关的优先级或者改写文法来消除冲突即可**

**正则匹配式的实现:**

1. **[T]:T 可有可无的情况:**

**把包含[T]的语句写成两句(含T\不含 T)**

1. **{T}:T可以0次或多次出现:**

**改写成 Ts->Ts T|empty 即可**

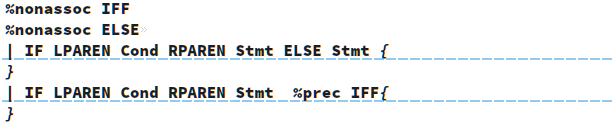
**冲突的消除:**

1. **四则运算优先级:**

**给出的语言定义文法已经解决了该问题**

1. **if\_else移进归约冲:**

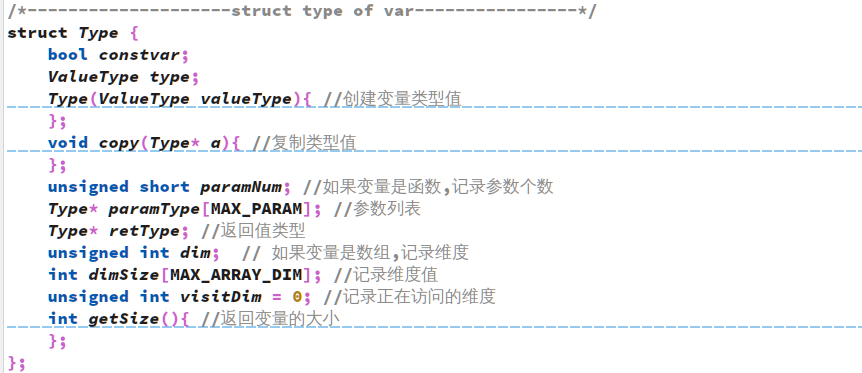
**利用yacc的优先级定义来解决**



**具体的语法规则实现在parser.y中,在此不多叙述**

**由于是先利用语法分析的过程生成语法树,然后分析整个语法树来完成语义分析并生成汇编代码,因此在整个语法分析中,任务不是特别多,只进行语法树节点的创建,语法树中父子关系的指向**

**记录变量的结构体**



**记录语法树节点的结构体**



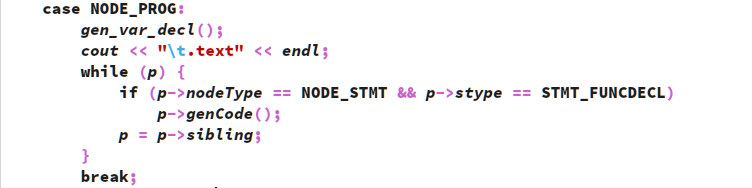
**语法分析之后会生成对应的语法树,语法树中记录着整个程序的信息,后续的语义分析会根据语法树,生成对应的汇编代码**

1. **语义分析: 遍历分析语法树,生成最终的汇编代码**

**从根节点开始,遍历整棵语法树,根据语法树的节点中记录的节点的类型来进行具体汇编代码的输出**

**根节点:即程序的开始节点**

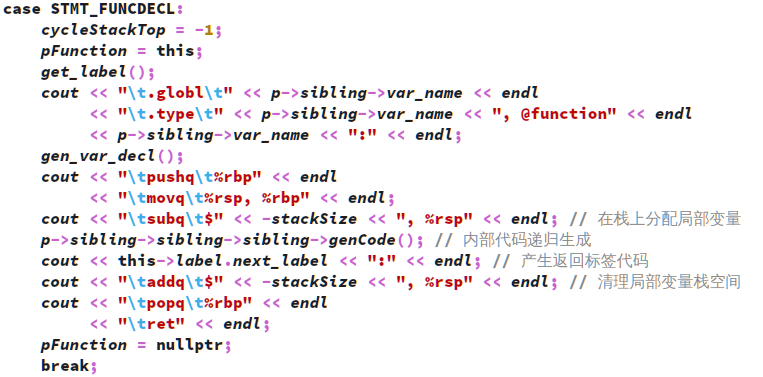
**输出整个程序所定义的全局变量,并递归进行后续函数及语句的代码生成**



**函数调用:传入参数并进行call操作**



**函数声明语句:分配内存,label,并递归进行函数内部汇编代码生成**



**由于分类较多,其余汇编代码的生成过程不再一一列举,具体内容在tree.cpp中**

**由于是先生成语法树,再进行语法树分析生成汇编代码,因此整个过程比较繁琐**

**由于对函数调用的理解不是特别深刻,在进行参数传递以及内存分配的时候存在存在着一些问题,因此函数调用功能可能不能正确完成**