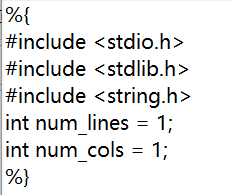
实验报告

一、实验目标

借助flex工具实现一个词法分析器，将SysY代码中的单词和符号进行分类，然后按照单词符号出现顺序依次输出：原始单词符号、种类、出现在源程序的位置（行数和列数）。其中单词的符号共分为K、I、C、O、D、T 共6类。还需要对注释进行处理，包括单行注释和多行注释。

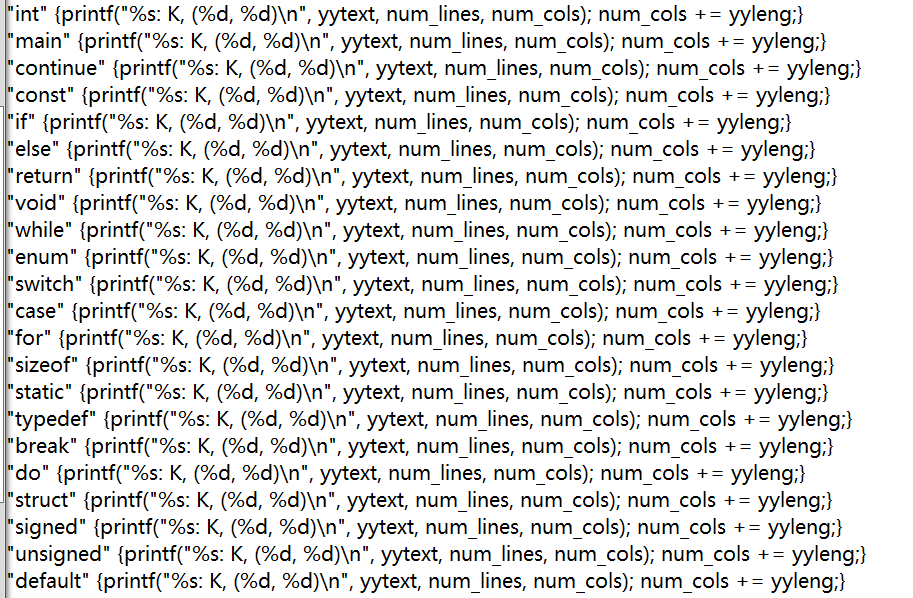
二、代码说明及思路

首先，在声明部分，除了需要的头文件之外，还声明了两个全局变量，其中num\_lines用于统计当前行数，num\_cols用于统计当前列数。



1. K类，关键字

关键字包括int、main、if、else等等，直接在识别规则处以 “int” 的形式要求完全匹配即可。动作中的输出按照要求的格式，种类为K（已匹配的部分存放在yytext中）；此外动作中还要记得将num\_cols加上已匹配内容的长度（yyleng），以统计当前所在列。（这一操作在以后的每次识别中都要进行，此后不再赘述）



2. I类，标识符

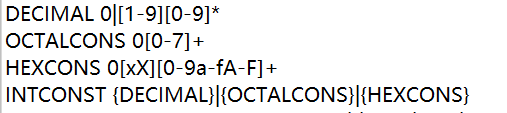
标识符以字母或下划线开头，后接任意个字母数字下划线。采用辅助定义IDENTIFIER，识别时输出为I类。





3. C类，常量

常量有三种表示方法：十进制，八进制，十六进制。十进制以非0的数开头，后接任意数字；八进制以0开头，后接0-7的任意数字；十六进制以0x或0X开头，后接任意个0-F。分别进行辅助定义，并汇总为INTCONST。





4. O，算符

算符用辅助定义OPERATOR表示，将可能的算符全部列出即可。





5. D，界符

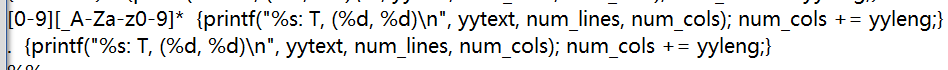
界符同上，辅助定义为BORDER





6. T，其他

标识符不能以数字开头，若以数字开头则应识别为T。还有在识别规则的的最后，用 . 匹配除换行符之外的任意字符，已匹配可能出现的其他情况。



7. 注释

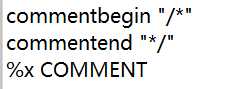
注释不进行输出，但要统计行数

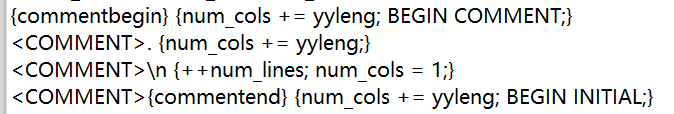
对于单行注释，先匹配//，中间为除换行符之外的任意字符，最后匹配换行符\n。识别到单行注释时行数要加一。



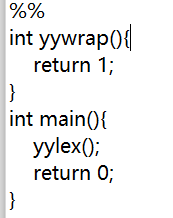


对于多行注释，为了统计行数，采用BEGIN条件激活规则，识别到/\*后即进入COMMENT部分，然后匹配到换行符就增加行数，其他字符只增加列数，直至匹配到\*/为止，结束该部分。



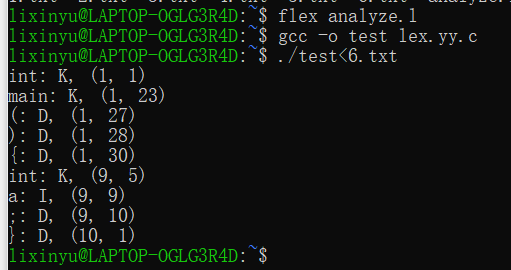


最后的用户子程序部分：



三、实验过程

.l源程序写好后，用flex运行生成lex.yy.c文件，再用gcc编译运行即可。图中所示为输入样例6的输出结果，经检验符合要求。



用同样方法测试了其他样例，输出均符合要求。受篇幅限制，在此只展示样例1的结果。

