

Lab1 Report

黄潇颖 2020201622

1.实验内容:

用flex工具生成一个 SYS 语言的 词法分析程序，对SYS语言的源程序进行扫描，识别出单词符号的类别，输出符号的相关信息。

输入:SYS 源程序

输出:把单词符号分为下面六类，然后按单词符号出现顺序依次输出各单词符号的 种类 和出现在源程序中的 位置 。

```
#define MYOPRATER "O" //算符
#define MYKEY "K" //关键字
#define MYDOMAIN "D" //界符
#define MYID "I" //标识符
#define MYCONST "C" //常量
#define MYOTHER "T" //其它
```

2.实验环境

操作系统:MacOS

分析工具:Flex

3.实现细节

• 3.1 位置的计算

使用全局变量 row 和 col 来计算位置，规则如下：

- row和col均从1开始计.
- 每匹配到一个换行符 \n : row+=1 , col 重置1.
- 每匹配出一个串， col 都要加上该串的长度 yyleng .

• 3.2 识别规则

- 为防止混乱，先对sysY语言中可能出现的符号的正则表达式进行 辅助定义 .
- WORDFUNC_* 函数是为六类不同的符号串所封装的功能函数。其中除了T类的实现较为特殊外，其它五类的功能均为打印类别、位置信息.
- T类函数的识别规则：

T类对编译器分析程序语言毫无用处，因此可以直接滤掉内容，但其在源文件中所占的位置仍需计算。

1) 空格、tab、换行符：思路均在 3.1 提过，实现于函数 WORDFUNC_space 和 WORDFUNC_lindfeed 中，不再赘述.

2) 注释：注释的识别需要用到flex提供的 Start condition 机制。先用 %x commentline comment 来为 // 和 /* */ 两种格式的注释声明与INITIAL无关（若不，则可能会匹配到 DIV ）的mini scanner来匹配注释。

// 的情况：

```
{COMMENTLINEBEGIN} BEGIN(commentline);  
<commentline>{COMMENTCONTENT} WORDFUNC_comment(yytext,yyleng,yyout);  
<commentline>{LINEFEED} {WORDFUNC_lindfeed();BEGIN(INITIAL);}
```

/* */ 的情况：

```
{COMMENTBEGIN} BEGIN(comment);  
<comment>{COMMENTCONTENT} WORDFUNC_comment(yytext,yyleng,yyout);  
<comment>{LINEFEED} WORDFUNC_lindfeed();  
<comment>{COMMENTEND} BEGIN(INITIAL);
```

4.测试

进入 lab1_code 文件夹输入命令：

make clean 删除已有输出文件.

make 进行新的测试.

输出结果在 out*.txt 文件中.