编译原理词法分析程序实验报告

学院:计算机学院
班级:2015211306
姓名:魏 晓
学号:2015211301

实验目标和要求:

题目：词法分析程序的设计与实现。
实验内容：设计并实现C语言的词法分析程序，要求如下。
（1）可以识别出用C语言编写的源程序中的每个单词符号，并以记号的形式输出每个单词符号。
（2）可以识别并读取源程序中的注释。
（3）可以统计源程序汇总的语句行数、单词个数和字符个数，其中标点和空格不计算为单词，并输出统计结果
（4）检查源程序中存在的错误，并可以报告错误所在的行列位置。
（5）发现源程序中存在的错误后，进行适当的恢复，使词法分析可以继续进行，通过一次词法分析处理，可以检查并报告源程序中存在的所有错误。
实验要求：
方法：采用C/C++作为实现语言，手工编写词法分析程序。

实验个人实现概况:

编程语言:C++
实现环境:Mac OS 10.13.2
目标语言:C/C++

1.实现文件输入输出和命令行输入输出
2.**标识符采取Trie字典树实现,在标识符较多时有很好的效率从O(n)—>O(log n)**
3.在识别到符号能自动判断符号的种类,并自动分类存储
4.能够处理行中注释,多行注释,单行注释
5.采用面向对象编程,代码可重构
6.在进入自动机之前预先处理注释和空格,输出标准化代码,再进入自动机识别
7.对于指针,小数等简单结构无缝识别

系统说明

1.关键字(关键字可以任意添加)

“begin”,”end”,”if","then","else","while","write","read","do","call","const","char","until","procedure","repeat".....

2.运算符(运算符可以任意添加)

“+”,”-“,”*”,”/“,”=“,”%”,”^”,”&”…….

3.界符(界符可以任意添加)

“{“,”}”,”[“,”]”,”;”,”,”,”.”,"(",")",":".....

4.其他标记

如字符串，表示以字母开头的标识符；.空格、回车、换行符跳过

5.运行结果在屏幕上的显示

NUMBER　　$无符号整数

KEYWORD　$关键字

RELATIONOPTR　　$运算符

DELIMET　　$界符

ID　　$普通标识符　　//构建字典树,自动查重

6.测试功能的问题

可以修改de.txt文件,输入1就处理文件,输入其他就从键盘读入,输出到控制台

7.C关键字表

define	iostream	int	string	end
break	else	repeat	switch	begin
case	max	key	compare	then
char	cout	return	analyse	write
const	fpin	fgetc	fseek	read
main	for	fopen	void	call
default	in	fclose	while	until

Do	if	static	null	procedure

函数功能说明(重要)

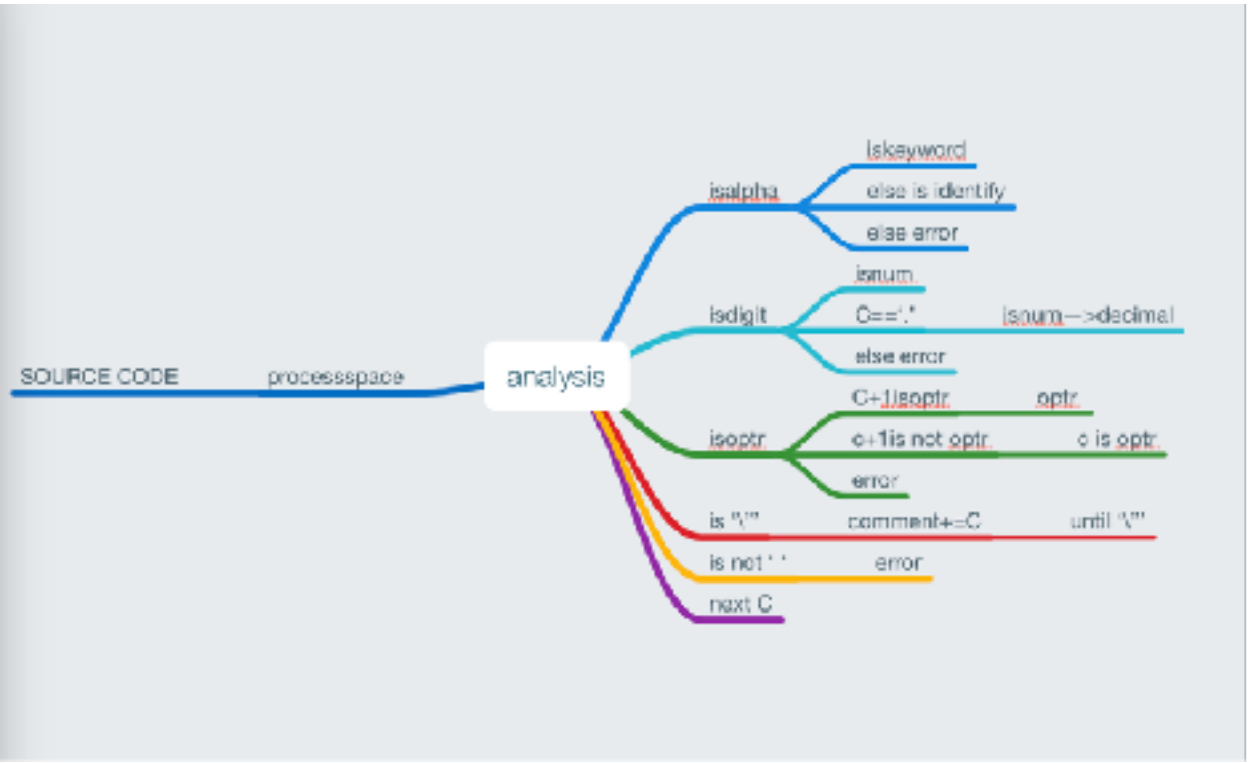
```
 string Lexical::processspace(std::string str)
```

//预处理函数,消去空格,输出注释

```
 void Lexical::run()
```

//运行

```
 void Lexical::analysis() {//自动机
     if((bool)isalpha(C)){
     else if((bool)isdigit(C))
     else if("ERR"!=isOptr(string(1,C)))
     else if(C == '"' || C== '\'')
     else if(C!=' ')
 }
```

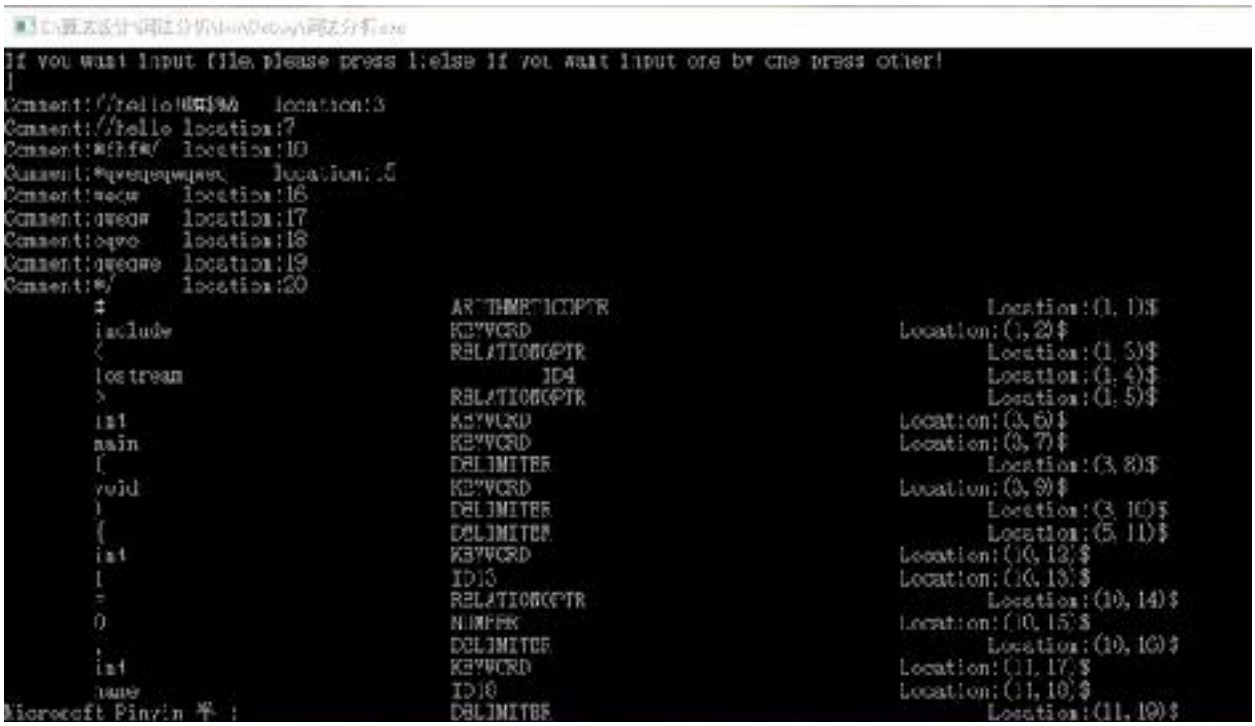


实验效果:

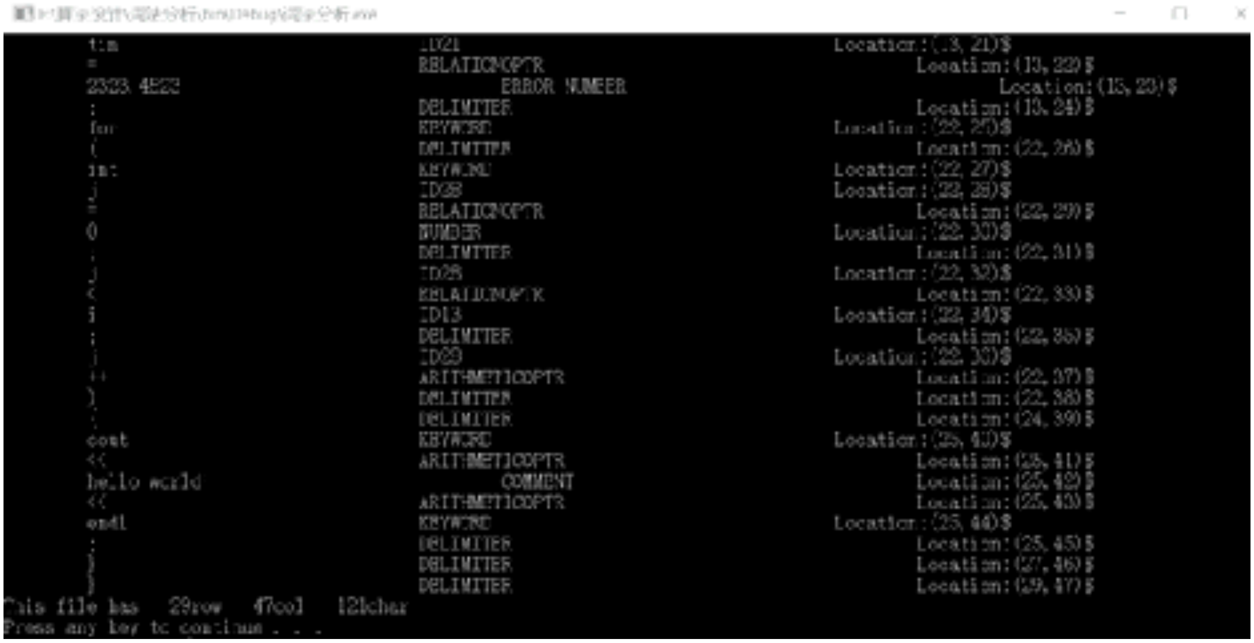
输入:

```
1   #include <iostream>
2   int main(void)//hello!@#$%&
3   {
4         //hello
5
6         int i=0;/*fhf*/int name;
7         float tim=2323.4523;
8         /*qweqeqwqweqweqw
9         qweqweqwe
10        qweqwe
11         */
12        for(int j=0;j<i;j++)
13        {
14            cout<<"hello world!"<<endl;
15        }
16         return 0;
17  }
```

输出



...



```
    tim                 ID21                    Location:(13,21)$
    =                   RELATIONOPTR                Location:(13,22)$
    2323.4E23               ERROR NUMBER                Location:(13,23)$
    ;                   DELIMITER                   Location:(13,24)$
    for                 KEYWORD                 Location:(22,25)$
    (                   DELIMITER                   Location:(22,26)$
    int                 KEYWORD                 Location:(22,27)$
    j                   ID28                    Location:(22,28)$
    =                   RELATIONOPTR                Location:(22,29)$
    0                   NUMBER                  Location:(22,30)$
    ;                   DELIMITER                   Location:(22,31)$
    j                   ID28                    Location:(22,32)$
    <                   RELATIONOPTR                Location:(22,33)$
    i                   ID13                    Location:(22,34)$
    ;                   DELIMITER                   Location:(22,35)$
    j                   ID29                    Location:(22,36)$
    ++                  ARITHMETICOPTR              Location:(22,37)$
    )                   DELIMITER                   Location:(22,38)$
    ;                   DELIMITER                   Location:(24,39)$
    cout                KEYWORD                 Location:(25,40)$
    <<                  ARITHMETICOPTR              Location:(25,41)$
    hello world             COMMENT                 Location:(25,42)$
    <<                  ARITHMETICOPTR              Location:(25,43)$
    endl                KEYWORD                 Location:(25,44)$
    ;                   DELIMITER                   Location:(25,45)$
    }                   DELIMITER                   Location:(27,46)$
    }                   DELIMITER                   Location:(29,47)$
This file has   29row   47col   121char
Press any key to continue . . .
```