c++实现编译原理词法分析实验（含代码）

原创

nicec1 于 2020-05-14 17:42:58 发布

分类专栏： 编译原理 文章标签： c语言 字符串 自然语言处理

c++实现编译原理词法分析实验（含代码）

一、实验目的：
通过设计编制调试一个具体的词法分析程序，加深对词法分析原理的理解。并掌握在对程序设计语言源程序进行扫描过程中将其分解为各类单词的词法分析方法。
编制一个读单词过程，从输入的源程序中，识别出各个具有独立意义的单词，即基本保留字、标识符、常数、运算符、分隔符五大类。并依次输出各个单词的内部编码及单词符号自身值。（遇到错误时可显示“Error”，然后跳过错误部分继续显示）
二、程序思路（仅供参考）：
这里以开始定义的C语言子集的源程序作为词法分析程序的输入数据。在词法分析中，自文件头开始扫描源程序字符，一旦发现符合“单词”定义的源程序字符串时，将它翻译成固定长度的单词内部表示，并查填适当的信息表。经过词法分析后，源程序字符串（源程序的外部表示）被翻译成具有等长信息的单词串（源程序的内部表示），并产生两个表格：常数表和标识符表，它们分别包含了源程序中的所有常数和所有标识符。
0.定义部分：定义常量、变量、数据结构。
1.初始化：从文件将源程序全部输入到字符缓冲区中。
2.取单词前：去掉多余空白。
3.取单词后：去掉多余空白（可选，看着办）。
4.取单词：利用实验一的成果读出单词的每一个字符，组成单词，分析类型。（关键是如何判断取单词结束？取到的单词是什么类型的单词？）
5.显示结果。

~~以上都是废话，直接看代码吧~~

代码

```
#include "stdafx.h"
#include <iostream>
#include <stdlib.h>
#include <stdio.h>
#include<string>
#define M 20
using namespace std;
string keyword[9] = { "main","if","int","for","while","do","return","break","continue" };//关键字=1
  //标识符值=2
  //常数值=3
char arithmetic[4] = { '+','-','*','/' }; //算术运算符=4
char relation[6] = { '=','>','<','>=','<=','!=' };//关系运算符=4
char border[6] = { ';',',','{','}','(',')' };    //分隔符=5
```

```
/**判断是否为关键字**/
bool IsKeyword(string word) {
 for (int i = 0; i<9; i++) {
  if (keyword[i] == word) {
   return true;
  }
 }
 return false;
}
/**判断是否为算术运算符**/
bool IsArithmetic(char ch) {
 for (int i = 0; i<4; i++) {
  if (arithmetic[i] == ch) {
   return true;
  }
 }
 return false;
}
/**判断是否为关系运算符**/
bool IsRelation(char ch) {
 for (int i = 0; i<6; i++) {
  if (relation[i] == ch) {
   return true;
  }
 }
 return false;
}
/**判断是否为分隔符**/
bool IsBorder(char ch) {
 for (int i = 0; i<6; i++) {
  if (border[i] == ch) {
   return true;
  }
 }
 return false;
}


/**判断是否为字母**/
bool IsLetter(char ch) {
 if (ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z') return true;
 return false;
}
/**判断是否为数字**/
bool IsDigit(char ch) {
 if (ch >= '0' && ch <= '9') return true;
 return false;
}
// 常数
char DigitProcess(char ch, FILE* fp)
{
 int i = -1;
 char digit[M];
 while ((IsDigit(ch)))
 {
  digit[++i] = ch;
  ch = fgetc(fp);
 }
 digit[i + 1] = '\0';
```

```
 cout << "(3 , " << digit << " )" << "  常数" << endl;
 return(ch);
}
// 标识符、关键字
char AlphaProcess(char ch, FILE* fp)
{
 int i = -1;
 char alpha[M];
 while (IsLetter(ch) || (IsDigit(ch)))
 {
  alpha[++i] = ch;
  ch = fgetc(fp);
 }
 alpha[i + 1] = '\0';
 if(IsKeyword(alpha))
  cout << "(1 , " << alpha << " )" << "  关键字" << endl;
 else
  cout << "(2 , " << alpha << " )" << "  标识符" << endl;

 return(ch);
}
// 运算符 、界符
char OtherProcess(char ch, FILE* fp)
{
 int i = -1;
 char othertp[M];
 othertp[0] = ch;
 othertp[1] = '\0';
 if (IsArithmetic(ch))//算术运算符
 {
  cout << "(4 , " << othertp << " )" << "  运算符" << endl;
  ch = fgetc(fp);
 }
 if (IsRelation(ch))//关系运算符
 {
  ch = fgetc(fp);
  othertp[1] = ch;
  othertp[2] = '\0';
  if (IsRelation(ch))
   cout << "(4 , " << othertp << " )" << "  运算符" << endl;
  else
  {
   othertp[1] = '\0';
   cout << "(4 , " << othertp << " )" << "  运算符" << endl;
  }
 }
 if (IsBorder(ch))//分界符
 {
  cout << "(5 , " << othertp << " )" << "  分隔符" << endl;
  ch = fgetc(fp);
 }
 ch = fgetc(fp);
 return(ch);
}
int main()
{
 char ch;
 FILE *fp;
 if (fopen_s(&fp, "E:\\test.txt", "r") != NULL) // 判断源文件是否存在
     cout<<" 文件不存在 "<<endl;
```

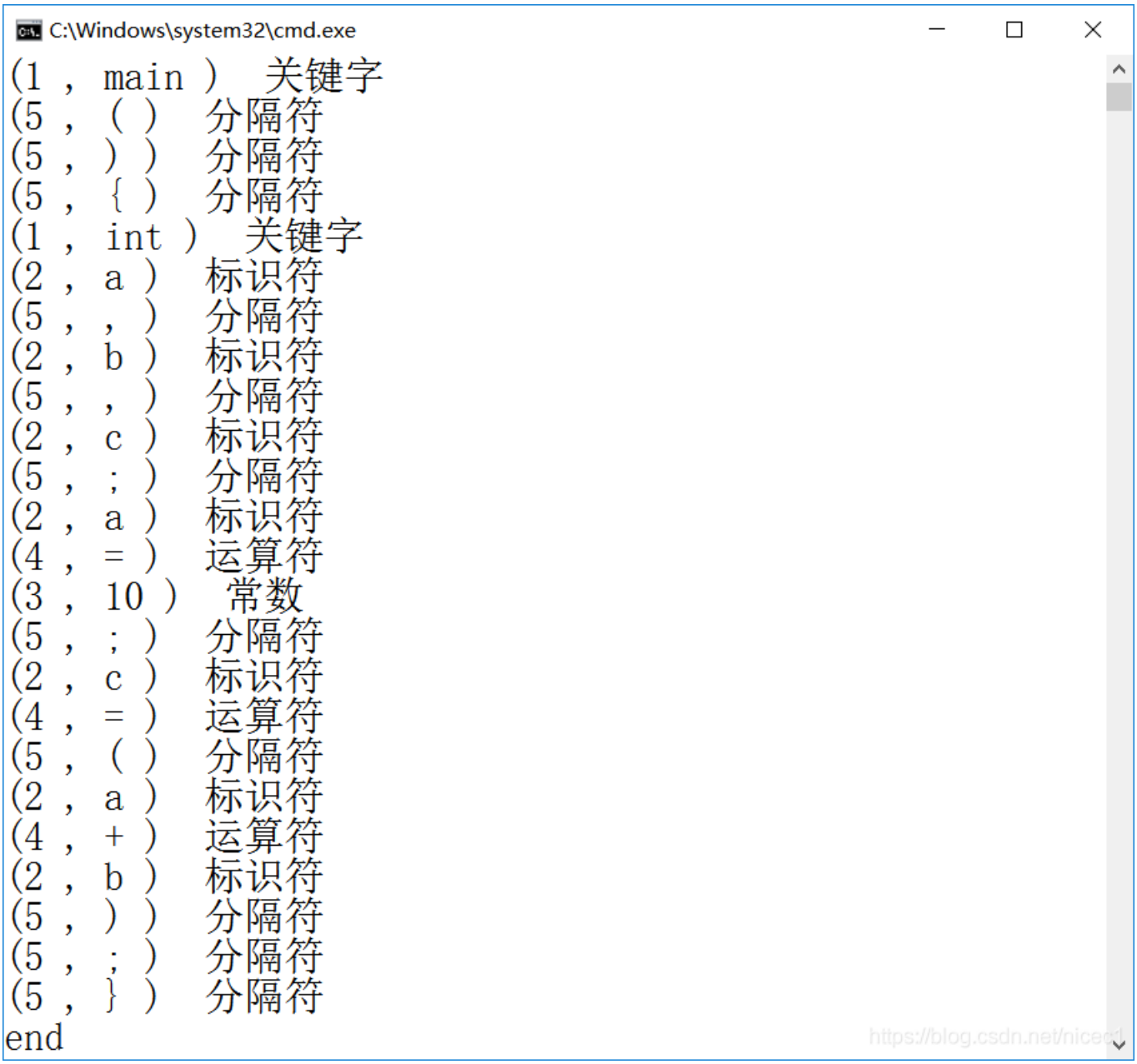
```
    cout<< "文件不存在" <<endl;
else
{
 ch = fgetc(fp); // 读入字符
 while (ch != EOF) // 如果文件没有结束 ,就一直循环
 {
  if (IsLetter(ch)) // 如果是字母
   ch = AlphaProcess(ch, fp);
  else if (IsDigit(ch)) // 数字
   ch = DigitProcess(ch, fp);
  else
   ch = OtherProcess(ch, fp);
 }
 cout << "end" << endl;
 getchar();
}
return 0;
}
```

不知道是不是代码bug，文件内容必须要用空格隔开

test.txt文件

```
main( )
{
int  a , b , c ;
a = 10 ;
c = ( a + b ) ;
}
```

运行结果



出现的一些问题

我用的vs，开始用`fp=fopen("E:\\test.txt","r"))`，会报

> error C4996: 'fopen': This function or variable may be unsafe. Consider using fopen_s instead. To disable deprecation, use _CRT_SECURE_NO_WARNINGS. See online help for details.

改成`fopen_s(&fp, "E:\\test.txt", "r") != NULL`就行了