词法分析程序设计原理与实现

**16281208**

**熊超**

**2018.5.31**

目录

[词法分析程序设计原理与实现 1](#_Toc10187824)

[1.实验目的 3](#_Toc10187825)

[2.设计要求 3](#_Toc10187826)

[3.实验分析 3](#_Toc10187827)

[3.1为各种单词符号编号 3](#_Toc10187828)

[3.2预处理 4](#_Toc10187829)

[3.3使用条件语句实现DFM 4](#_Toc10187830)

[4.实验测试 6](#_Toc10187831)

[5.实验心得与体会 8](#_Toc10187832)

## 1.实验目的

以下为正则文法所描述的C语言子集单词符号的示例，请补充单词符号： ++，--， >>, <<, += , -= ,\*=, /= ，&&（逻辑与），||（逻辑或），！（逻辑非）等等，给出补充后描述C语言子集单词符号的正则文法，设计并实现其词法分析程序。

<标识符>→字母︱ <标识符>字母︱ <标识符>数字

<无符号整数>→数字︱ <无符号整数>数字

<单字符分界符> →+ ︱－ ︱\* ︱；︱, ︱(︱) ︱{︱}

<双字符分界符>→<大于>=︱<小于>=︱<小于>>︱<感叹号>=︱<等于>=︱<斜竖>\*

<小于>→< <等于>→= <大于>→> <斜竖> →/

<感叹号>→!

该语言的保留字 ：void、int、float、double、if、else、for、do、while 等等（也可补充）。

## 2.设计要求

（1）给出各单词符号的类别编码；

（2）词法分析程序应能发现输入串中的错误；

（3）词法分析作为单独一遍编写，词法分析结果为二元式序列组成的中间文件；

（4）设计两个测试用例（尽可能完备），并给出测试结果。

## 3.实验分析

### 3.1为各种单词符号编号

将各种需要的单词按类别放到不同的集合中

reserve\_word = { 'auto', 'break', 'case', 'char', 'const', 'continue',

'default', 'do', 'double', 'else', 'enum', 'extern',

'float', 'for', 'goto', 'if', 'int', 'long',

'register', 'return', 'short', 'signed', 'sizeof', 'static',

'struct', 'switch', 'typedef', 'union', 'unsigned', 'void',

'volatile', 'while'}

operatorORdelimiter = {"+","-","\*","/","<","<=",">",">=","=","==","+=","-=","/=",

"!=",";","(",")","^",",","\"","\'","#","&",

"&&","|","||","%","~","<<",">>","[","]","{","}","\\",".","?",":","!",'++','--','\*=','$'}

other = {'常数','标识符'}

创建一个字典，key为所有的符号，value则为1-n的编号，由于python中key的存放是乱序的，所以每次执行词法分析器符号表是不同的。

all\_signs = reserve\_word.union(operatorORdelimiter).union(other)

table = dict.fromkeys(all\_signs)

for i,value in enumerate(all\_signs):

table[value] = i+1

### 3.2预处理

#fliter the notes

source = re.sub('//.\*','',source)

source = re.sub('/\\*.\*\\*/','',source)

#fliter /n and /f

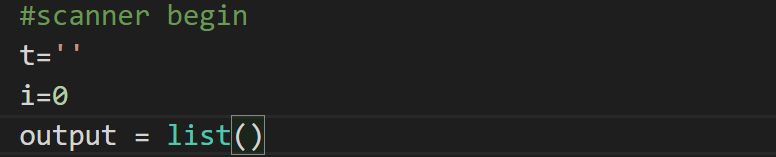
source = re.sub(r'\n|\f','',source) + '$'

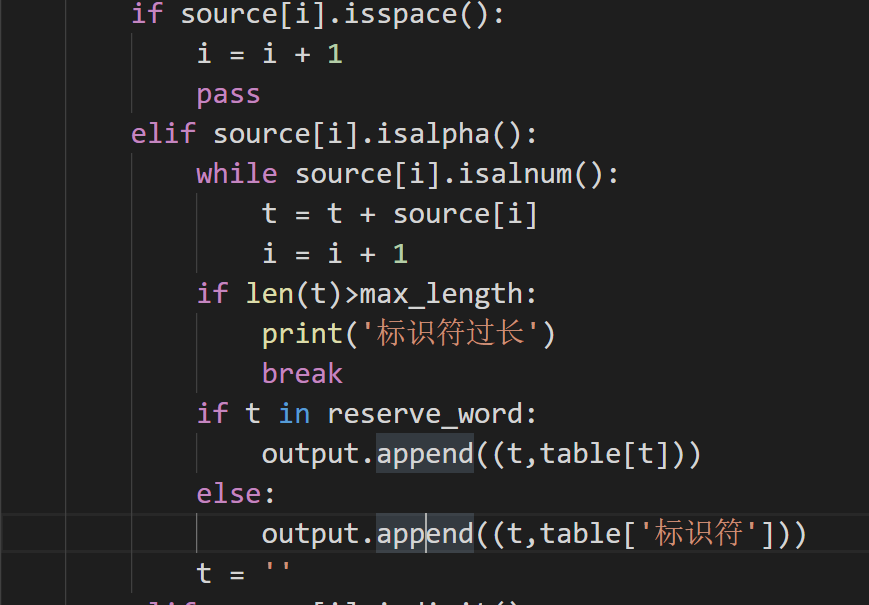
#scanner begin

使用正则表达式高效去除// 和/\*\*/型注释和换行符回车符，并在尾部添加结束符号

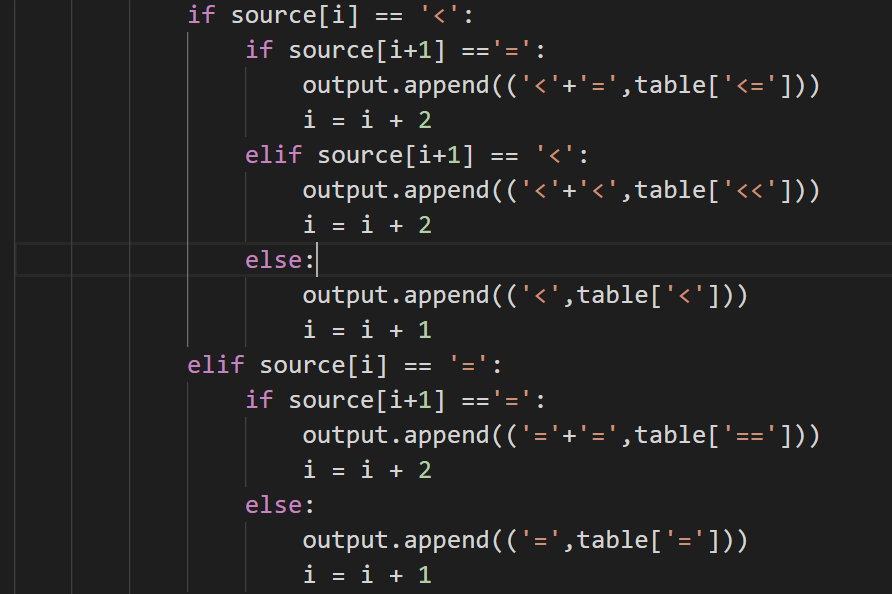
### 3.3使用条件语句实现DFM

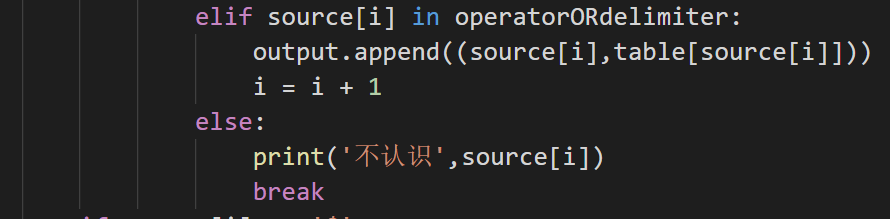
首先t表示每次读标识符、保留字和常数的临时变量，i为预处理后字符串的指针。Output为存放二元组输出的数组，二元组格式为（单词，单词编号）





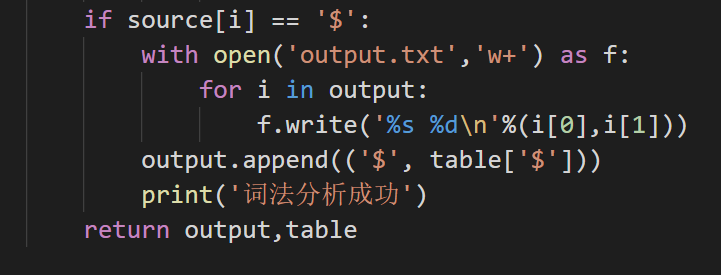
碰到空格指针前进，碰到字母数字就超前读取，直到下个不是字母数字的符号，并做长度限制判断，并按类别存入输出中，存放后清空t为下次做准备。

然后人工判别为含有分支的符号添加如上的嵌套条件判断



剩下的无分支的就不人工判断了

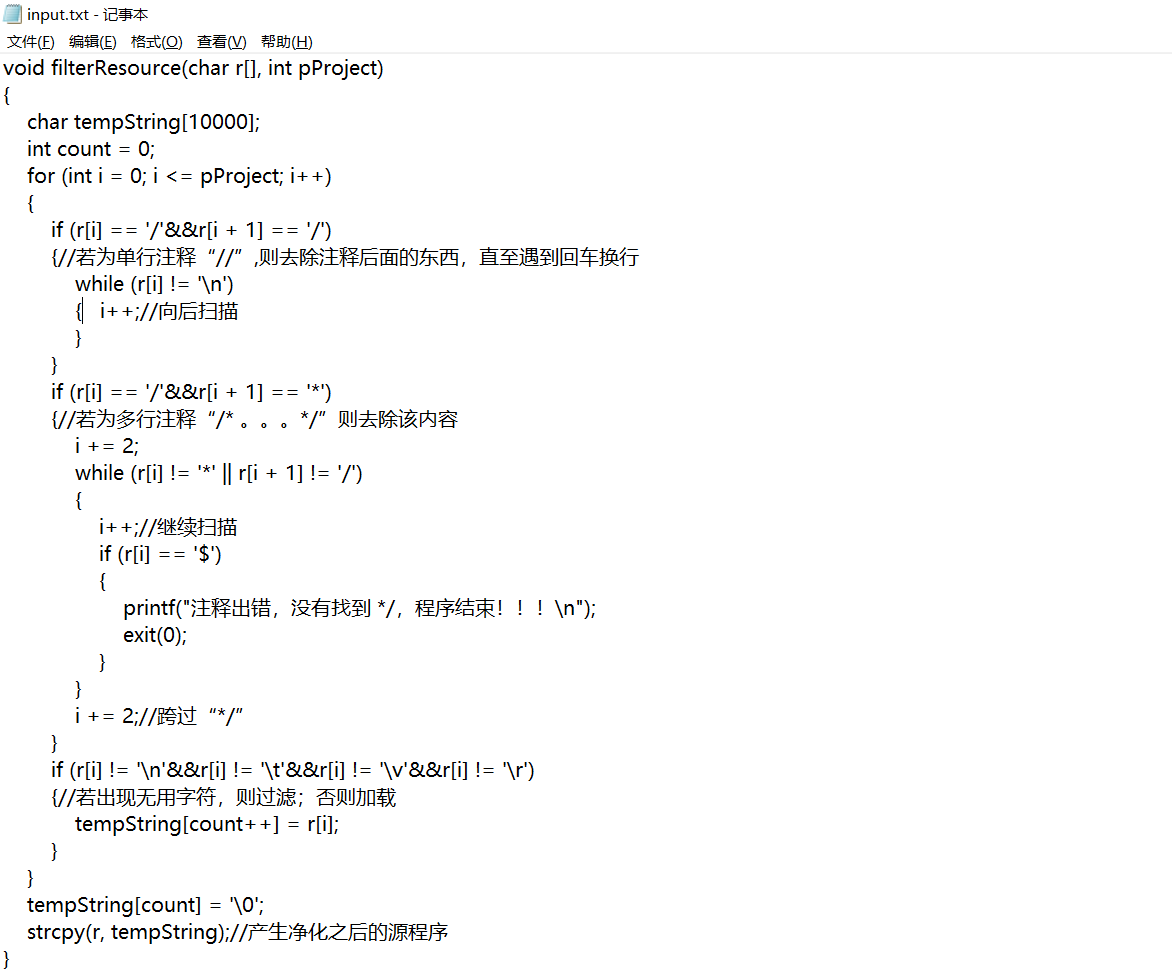
如果判断到最后了那肯定是出现了不在上述集合中的单词，不认识。



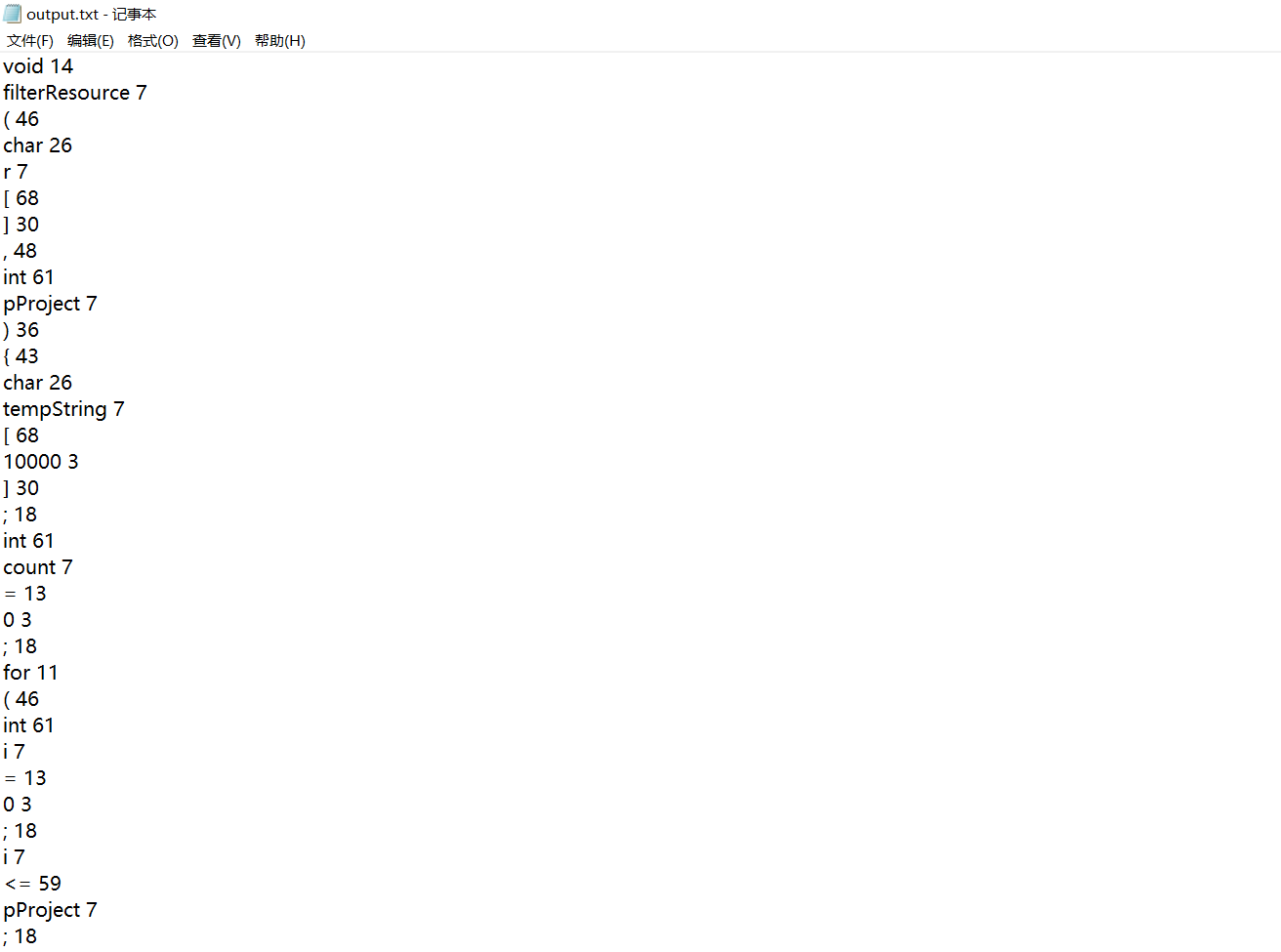
读到了结束符号DFM结束，把结果写入文件，向调用词法分析的返回二元组序列和符号表

## 4.实验测试

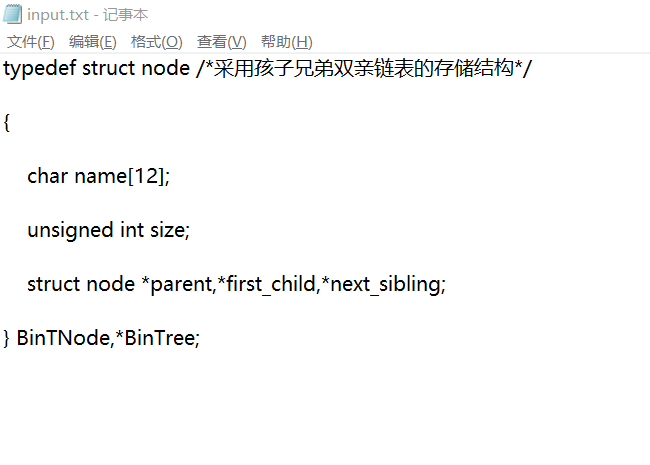
用例1



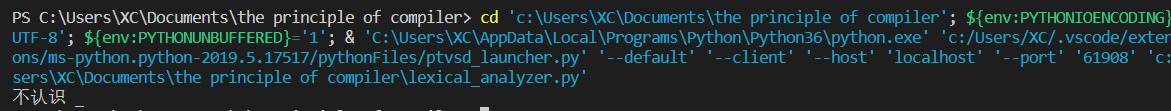
把上图c函数放入词法分析器 Output文件中，各个单词都被很好的识别了。



用例2



下划线\_我并未放在符号中



输出了不认识

## 5.实验心得与体会

通过本次专题实验，我更加深入的理解了词法分析器的构造，锻炼了了自己的编程能力。