**C语言词法分析器实验报告**

**1 实验题目**

C语言词法分析程序的设计与实现

**2 实验内容**

1. 识别单词符号并以记号形式输出，并标出该单词符号所在行数；
2. 能够识别并跳过注释；
3. 能够检查到错误的词法；
4. 能够统计行数、各个单词符号的类别数，以及词法错误数；

**3 文法定义及状态转换图**

**3.1 语言说明**

本实验的C语言定义有以下记号及单词：

1. 标识符（IDENTIFIER）：C语言中的标识符只能由字母、数字和下划线三种字符组成,且第一个字符必须是字母或下划线；
2. 关键字（KEYWORD）：标识符集合的子集。本实验包含常见的 32 个关键字

auto double int struct break else static long

switch case enum register typedef char extern return

union const float short unsigned continue for signed

void default goto sizeof volatile do if while

1. 数值常量（NUMBER）：由整数部分、可选的小数部分和可选的指数部分构成；

同时考虑整数中合法的不带后缀的八进制数与十六进制数，如：0X3abcde2，0x1ABCD2F，010

整数中合法的带后缀的十进制数，如：后缀u、U、l、L、ul、lu、llu等

完整的浮点型常量，如：.23、4.0E2F、5.0L、.001e2f

1. 运算符（OPERATOR）

实验涉及的运算符（OPERATOR）如下：

算术运算符 + - \* / % ++ --

关系运算符 == != > < >= <=

逻辑运算符 && || !

位运算符 & | ^ ~ << >>

赋值运算符 = += -= \*= /= %= <<= >>= &= ^= |= 成员运算符 . ->

1. 分隔符（DELIMITER）

实验涉及的分隔符（DELIMITER）如下：

; , : ? ( ) [ ] { }

1. 注释标记：

以/\*开始，以\*/结束；

以//开始，以换行符结束

1. 字符串（STRING）及字符常量（CHARCON）

字符串：以”开始，以”结束；

字符常量：以’开始，以’结束

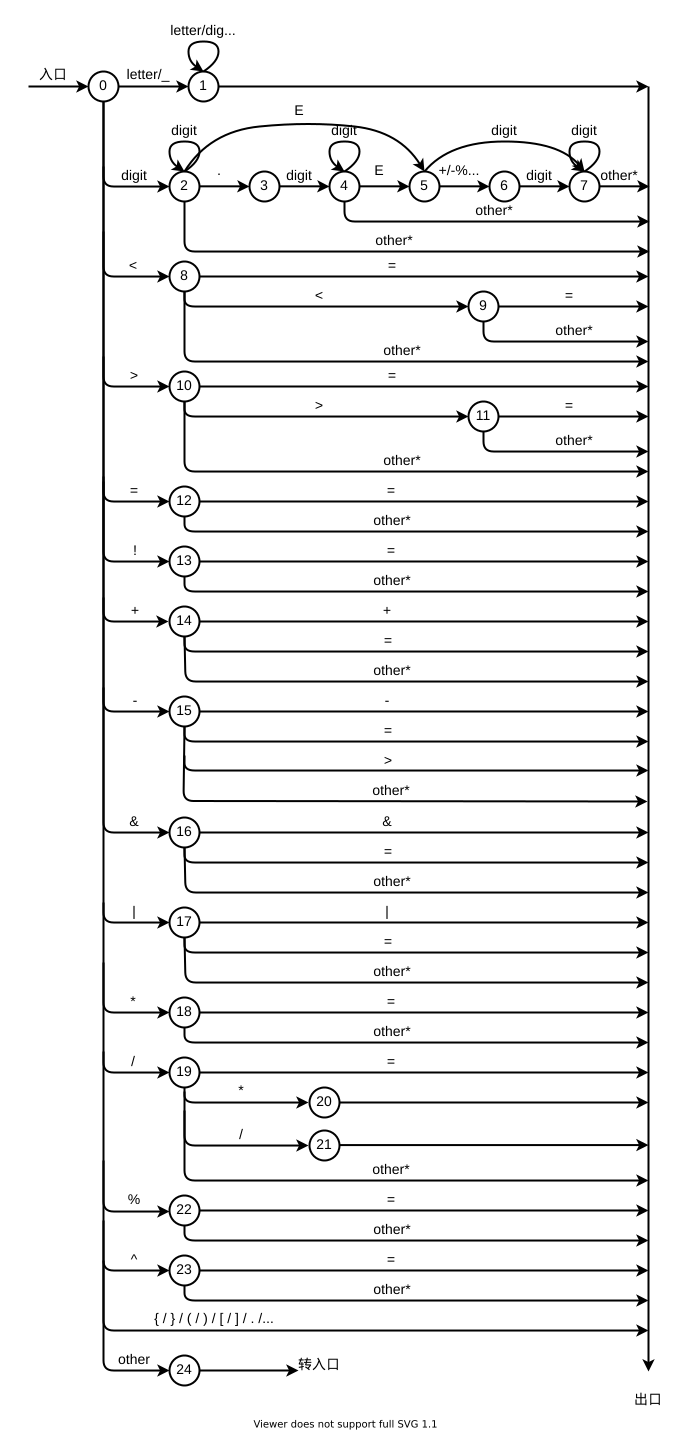
1. 错误类型（ERROR）：

①非法字符 @ 的报错——读到@就报错，并且规定不会出现在关键字、标识符、数字的识别过程中；

②未闭合的字符和字符串的报错——如果发现未闭合的单引号或双引号，将之后直到换行符之前的所有字符都认为是所属的错误的值(值里也包括开头的'或")；

③数字开头的非法标识符——规定这类错误的出现一定是在{空格符，制表符，换行符，分隔符，运算符}中的两者之间的。

**3.2 状态转换图：**

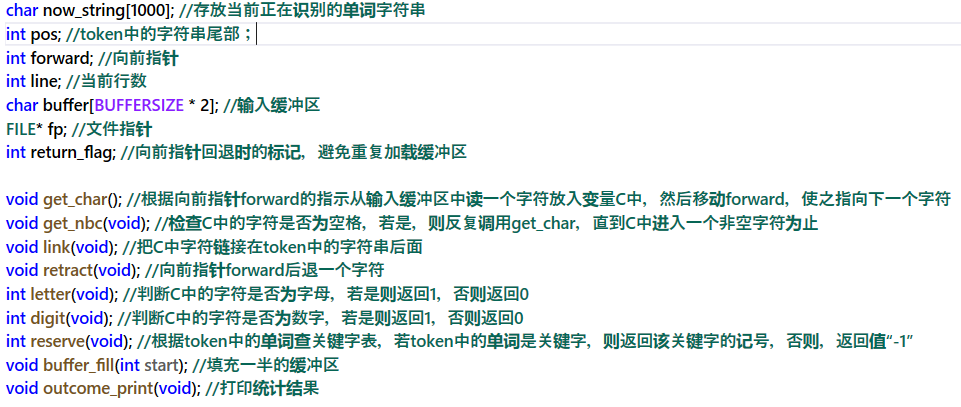
****

本词法分析程序的构造绝大部分以此状态转换图为基础

**4 词法分析程序的实现**

**4.1 定义全局变量**

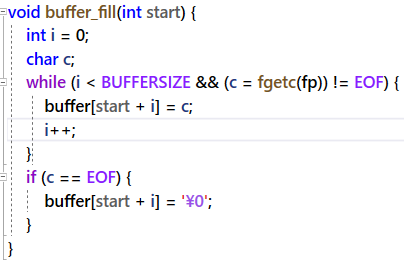
****



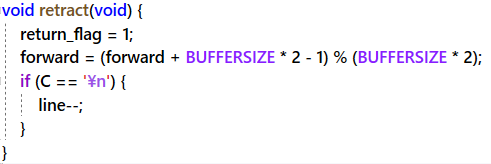
**4.2 主要函数说明**

**注：由于编译器字体原因，字符’\’都将显示为’¥’**

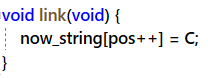
buffer\_fill更新缓冲区，并在文件结束时即时停止读入。为方便之后的判断，将文件结束符设置为'\0'。

****

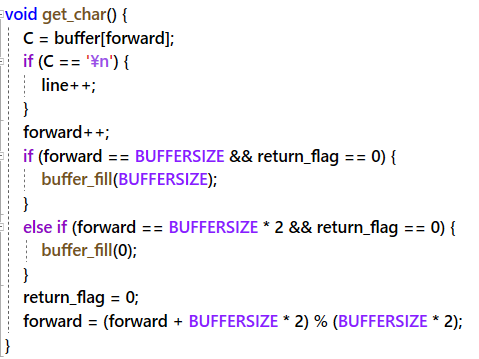
该函数向后回退一个向前指针，并将回退标记置1，防止重复更新缓冲区；同时若超前扫描时导致行数计数器增加，在这里一起回退



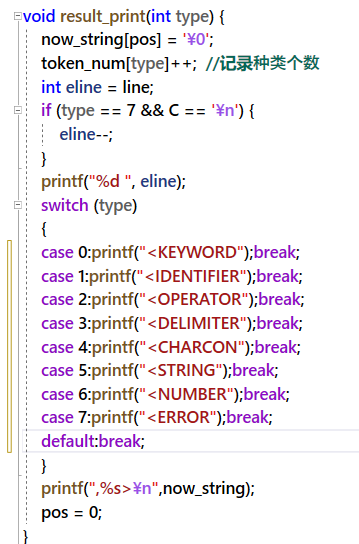
link函数将当前识别的字符C链接在now\_string中的当前识别的字符串后面



get\_char函数在从缓冲区获取下一个字符的同时，前移向前指针，统计程序的总行数，并在向前指针到达缓冲区边界时更新缓冲区。

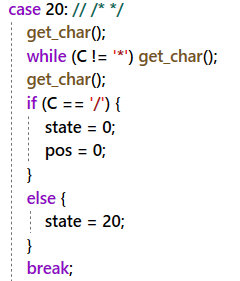


result\_print函数将当前识别的字符串now\_string以“行数 <记号类别,属性>“的形式输出，同时记录出现的记号类别的数量。并且对可能错误的字符恰好是换行符导致line中储存的行数不正确的情况，用局部变量eline解决这个问题。

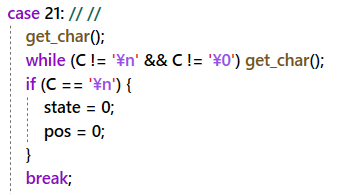


lexicalAnalyzer函数为本词法分析程序的主题函数，其程序逻辑比较简单，大部分为用switch-case实现上述状态转换图，大致上每一个状态就是一个case，状态的转换就是case的转换。针对注释识别、字符字符串识别、错误处理以及拓展要求部分的特殊处理在下方加以说明

* 1. ***注释识别***

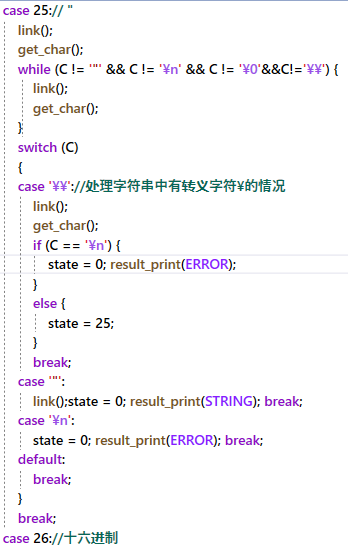


首先由/状态识别到\*状态进入/\*注释状态，此状态下只有识别到\*后再识别/才能跳出/\*注释状态回到初始状态0，同时pos=0清空读取注释前加入now\_string字符串中的字符以跳过输出注释部分，否则将一直识别注释部分

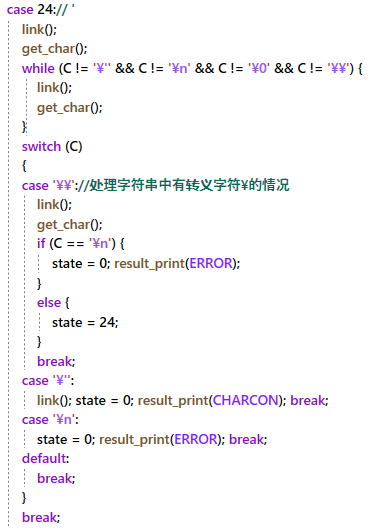


首先由/状态识别到/状态进入//注释状态，此状态下只有识别到换行符或文件读完才能跳出//注释状态回到初始状态0，同时pos=0清空读取注释前加入now\_string字符串中的字符以跳过输出注释部分，否则将一直识别注释部分

* 1. ***字符字符串识别***



对于字符“，其本身和之后字符串的内容都将被输出，故都link进now\_string中。针对字符串中转义符\的处理，识别到\后紧跟的“无法作为右引号结束字符串，故仍在字符串识别状态25中，而若\后面紧跟的换行符遇到错误3：未闭合的双引号，输出error并返回初态；针对换行符\n进行错误处理；针对”说明字符串闭合，将其正确输出并返回初态



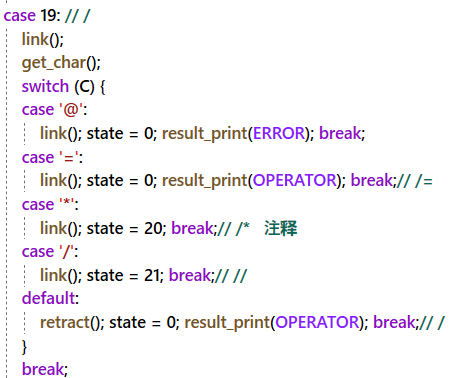
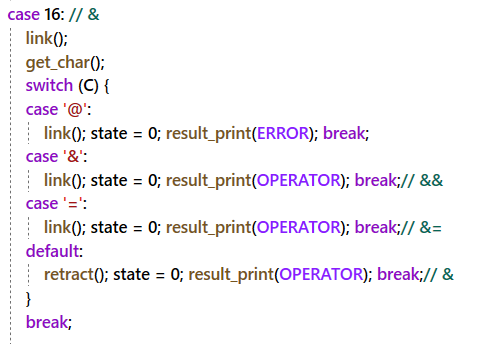
对字符‘的分析过程与“类似，在此不过多赘述

* 1. ***错误处理***

**错误1**：非法字符 @ 的报错——读到@就报错，并且规定不会出现在关键字、标识符、数字的识别过程中

题目此处并未说清识别到@该具体如何输出和读到@就报错是什么意思。观察测试集发现@错误不会出现在注释或字符字符串的识别过程中，故删除了此部分的@错误处理。而题目说明@错误不会出现在关键字、标识符、数字的识别过程中，那么只能在运算符和分隔符的识别过程中，因此本程序在运算符和分隔符的每个识别状态中加入只要识别到@就报错，同时跟前面正在识别的分隔符或运算符一同输出。同时在初态0时识别到@也直接报错输出。但是测试集中并未出现运算符/分隔符连着@一起输出的错误情况，故还是不太明白@错误的具体情况与样例应该是怎样的





**错误2**：未闭合的字符和字符串的报错——如果发现未闭合的单引号或双引号，将之后直到换行符之前的所有字符都认为是所属的错误的值(值里也包括开头的'或")；

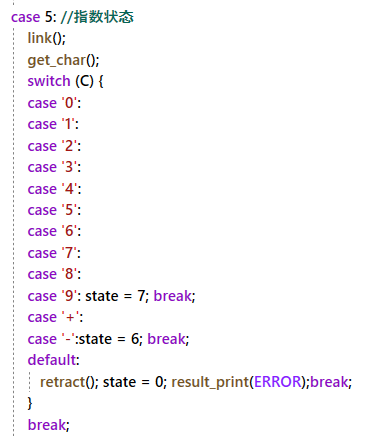
此错误的分析处理已在字符串和字符常量的识别过程中进行了说明，在此不再赘述

**错误3**：数字开头的非法标识符——规定这类错误的出现一定是在{空格符，制表符，换行符，分隔符，运算符}中的两者之间的

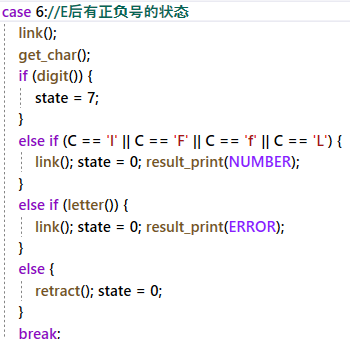
对于先识别到一个数字进入常数状态，识别到E/e则可能为指数的识别故先进入指数状态5，识别到u/U/l/L可能为十进制数后缀而进入状态27，而若出现其他字母则说明为非法标识符。因为查询的是非法标识符，而.本身就不能出现在标识符中，故不分析字母出现在小数点和E/e之间的错误情况

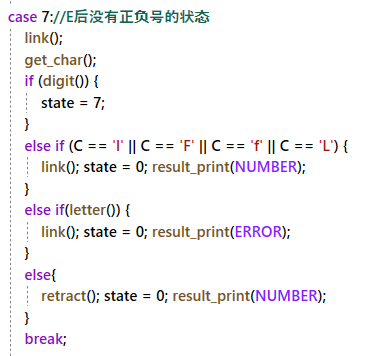


而对于指数状态5，若E/e后面没有紧跟数字或正负号，也可以看作为数字后面跟着E/e的非法标识符



对E/e后的状态，l、f、L、F可作为浮点数的后缀，其余出现的字母可判断为非法标识符。由于测试集输入保证为合法常数，故一些特殊情况不予考虑（如15EL等）

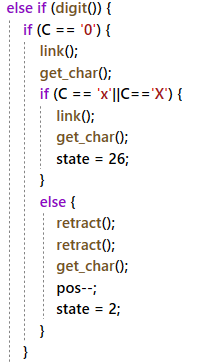




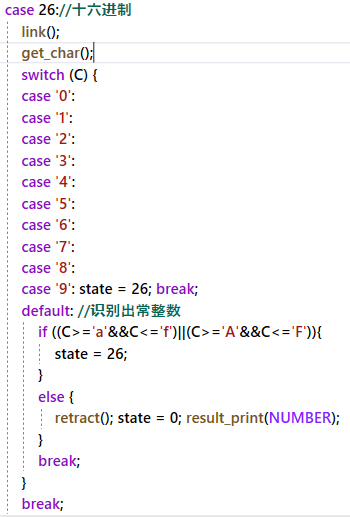
* 1. ***拓展要求***

**要求1**：考虑整数中，对合法的不带后缀的八进制数与十六进制数的识别；

对初始状态识别到数字，若为字符0，则link到当前字符串中，同时分析下一个字符是否为x/X，若是则说明为0x/0X前缀进入十六进制数识别状态。创建十六进制数识别状态26；否则则回退两个字符并重新识别到字符0进入数字识别状态2，当前字符串也回退一个字符。



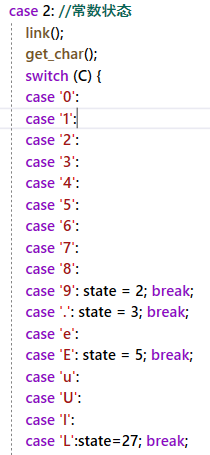
十六进制数的识别过程



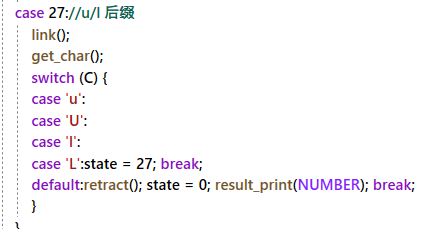
八进制的识别在状态2中已可处理，不再加以说明

**要求2**：考虑整数中，对合法的带后缀的十进制数的识别

整数中合法十进制的后缀为u、U、l、L的组合，故创建u/l后缀识别状态27，在整数识别状态2中若识别到u/l/U/L则跳转到状态27

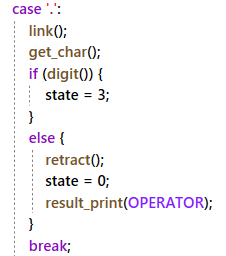


后缀为u、U、l、L的组合，故识别到这个四个字符就仍在状态27中，直到识别到其他字符退出（测试集输入保证合法，故不考虑特殊错误情况）

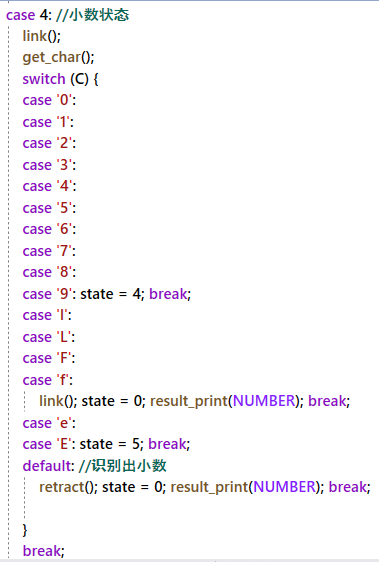


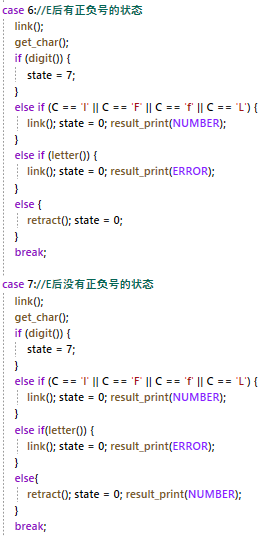
**要求3**：考虑完整的浮点型常量的识别

针对浮点型常量可能以小数点开头的情况，对字符.的识别进行修改，若.后紧接数字则进入小数点状态3。否则正常输出并返回状态0



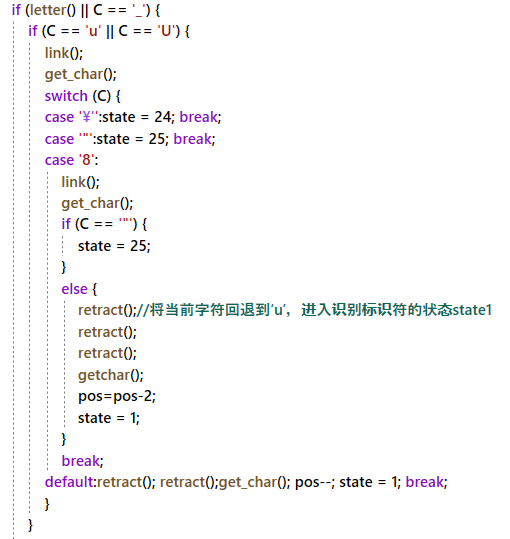
针对l/f/L/F后缀在浮点数识别的过程中都有处理，识别到后加入当前字符串并输出



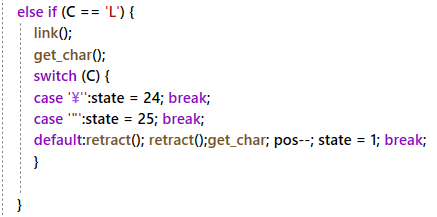


**要求4、5**：考虑带前缀的字符串常量和完整的字符常量的识别（以及转义符用于转义序列）

对于前缀u8/u/U，在初始状态0识别字母时，若连续识别到u‘/U‘则为字符常量的前缀进入字符常量识别状态24；若连续识别到u“/U“则为字符串的前缀进入字符串识别状态25，否则回退到识别字符u/U进入标识符识别状态1。若连续识别到u8‘/U8‘则为字符常量的前缀进入字符常量识别状态24，否则连续回退到识别字符u/U进入标识符识别状态1



对于前缀l/L，在初始状态0识别字母时，若连续识别到l‘/L‘则为字符常量的前缀进入字符常量识别状态24；若连续识别到l“/L“则为字符串的前缀进入字符串识别状态25，否则回退到识别字符l/L进入标识符识别状态1



对于转移字符的处理在字符串与字符常量的识别过程中有说明，在此不再赘述

**5 词法分析程序测试**

见文件夹中测试文件及结果

结果均正确