# 词法分析器

**一、问题描述**

本次实验的实验目的是设计与实现一个词法分析器，加深对词法分析原理的理解。

本次实验的实验内容为：

输入：源程序字符串（源程序存储在文本文件中，文件名作为命令行输入）；

输出：token序列（标准输出设备）。

**二、分析模型**

该词法分析器模型是要要求输入一个源程序字符串，输出这个字符串的token序列。包括：

①、关键字。如main、if、else、then、for、until、char、bool等；

②、专用符号。运算符：=、+、-、\*、/等，分隔符：，、；、：、{、}、（、）等

③、其他标记ID和NUM。通过以下正规式定义其它标记：

ID → letter(letter | digit)\*

NUM → digit digit\*

Letter → a | … | z | A | … | Z

Digit → 0|…|9

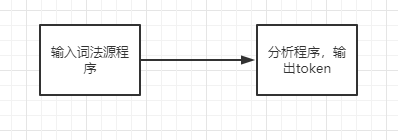
④、空白格由空格、制表符和换行符组成

空白一般用来分隔ID、NUM、专用符号和关键字，词法分析阶段通常被忽略。

除此之外还有单词符号种别码，需要自己定义。

**三、设计模型**

词法分析器的设计模型就是用c语言实现输入词法源程序，输出token。



重点在于如何通过源程序得到token序列。

**四、系统实现**

首先输入源程序字符串，将其用数组存储起来。然后通过词法分析程序对数组进行扫描，会出现如下情况：

①、如果遇到空格就略过；

②、遇到字符则可能是标识符或变量名，标识符不能以数字开头。为标识符分配唯一的种别码，变量名分配统一的种别码；

③、如果遇到数字则累加，对数字分配一个统一的种别码；

④、遇到运算符及分隔符要单独处理，为每一个运算符或分隔符分配一个唯一的种别码。

⑤、遇到‘\n’表示换行，遇到@即为结束标志。

总体结构如下：

对于变量名，因为不能以数字开头：

主要代码如下：

if(**isChar**(ch)){ *//可能是标识符或者变量名，不能以数字开头*

        m = 0;

        while(**isNum**(ch) || **isChar**(ch)){ *//首字符之后的可以是字符或者数字了*

            ……

        }

        ……

    else if(**isNum**(ch)){   *//如果是数字的话*

        while(**isNum**(ch)){

            …….

        }

       ……

    }

    else switch(ch){  *//给每一个运算符以及分隔符各自的标号*

        case '<':

            ……

            break;

        case '>':

            ……

            break;

        case '!': …

        case '+': …

        case '\*': …

        ……

        case '@': syn = 0;token[0] = ch;break; *//表示结束标志*

        case '\n': syn = -2;break;

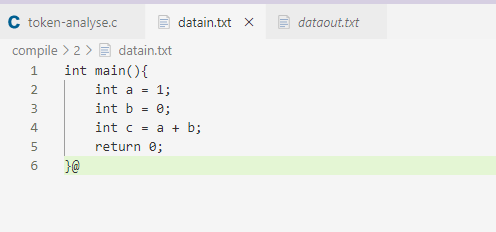
        default: syn = -1;break;

}

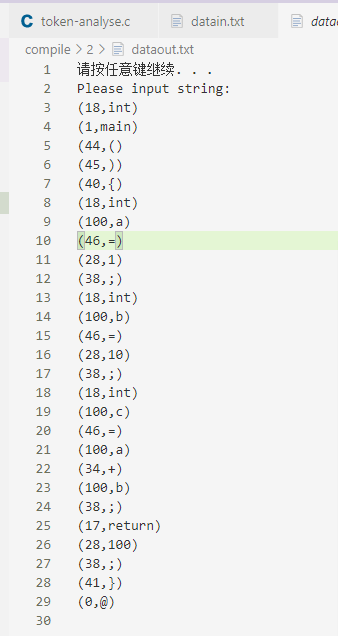
**五、测试数据及实验结果**

将文件保存为token-analyse.c;

datain文件内容：



通过token-analyse.exe<datain.txt>dataout.txt得到dataout.txt文件，内容为：



即输出将源程序字符串进行词法分析后得到的token序列。

**六、实验总结：**

1、在该实验的进行过程中，因为程序的基本思想在课本中也体现出来，因此上机时间还是能较快完成程序的编写的。当然在编写程序的过程中也出现了一些小小的bug，不过还是比较容易就被解决了。

2、遇到的问题就是对于设置一些运算符的种别编码时，对运算符的长度的确定判断有些偏差，导致程序不能良好运行，但是在debug之后还是比较容易能够找出来并加以改正的。

3、该程序不算很难，但我想一个真正的词法分析器还会有其他的东西，对于这部分的核心内容必须去掌握。只有这样，才能在逐步加深自己的理解，并能够自己真正地动手实践与设计。“纸上得来终觉浅，绝知此事要躬行”。