**词法分析器的设计**

一.实验目的：

设计一个词法分析程序，理解词法分析器实现的原理，掌握程序设计语言中的各类单词的词法分析方法，加深对词法分析原理的理解。  
二.实验任务：

（1）设计词法分析器，使其能够识别简单语言的单词符号

（2）识别并输出简单语言的基本字，标识符，运算符，常数，界符。

三.实验内容：

（1）词法分析程序的算法思想

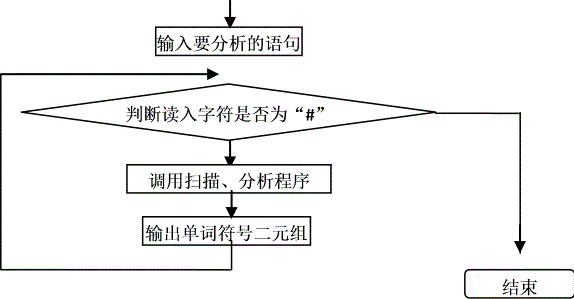
算法的基本任务是从字符串表示的源程序中识别出具有独立意义的单词符号，其基本思想是根据扫描到单词符号的第一个字符的种类，拼出相应的单词符号。

（2）C语言子集的单词符号及内码值

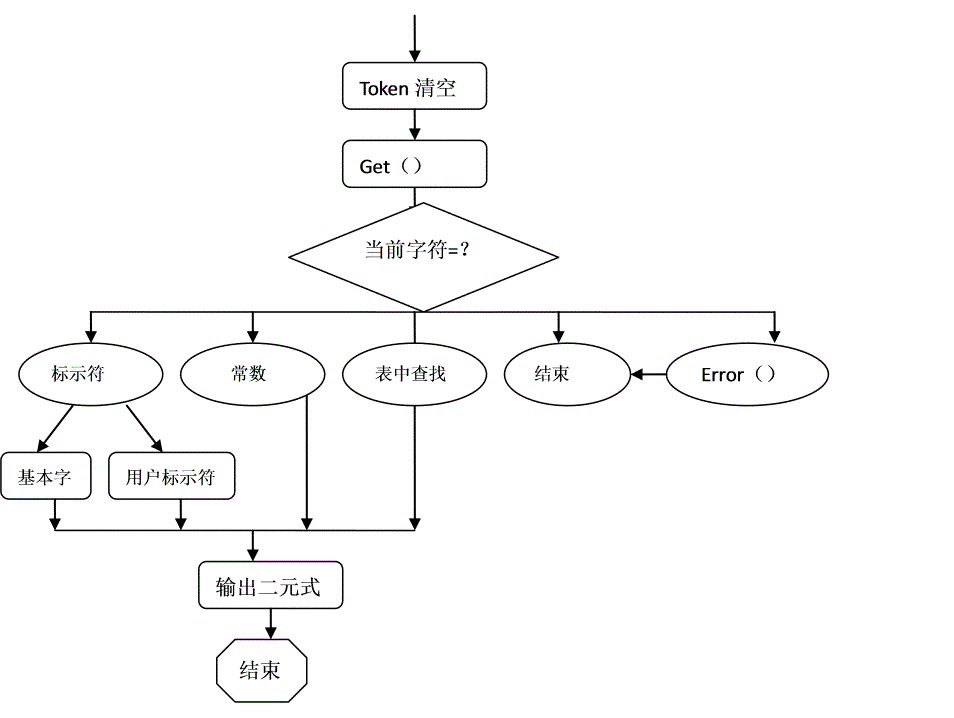
|  |  |  |  |
| --- | --- | --- | --- |
| 单词符号 | 种别编码 | 助记符 | 内码值 |
| while | 1 | while | — |
| if | 2 | if | — |
| else | 3 | else | — |
| switch | 4 | switch | — |
| case | 5 | case | — |
| 标识符 | 6 | id | id在符号表中的位置 |
| 常数 | 7 | num | num在常数表中的位置 |
| + | 8 | + | — |
| - | 9 | - | — |
| \* | 10 | \* | — |
| <= | 11 | relop | LE |
| < | 11 | relop | LT |
| == | 11 | relop | EQ |
| = | 12 | = | — |
| ; | 133 | ; | — |

（3）程序流程图：

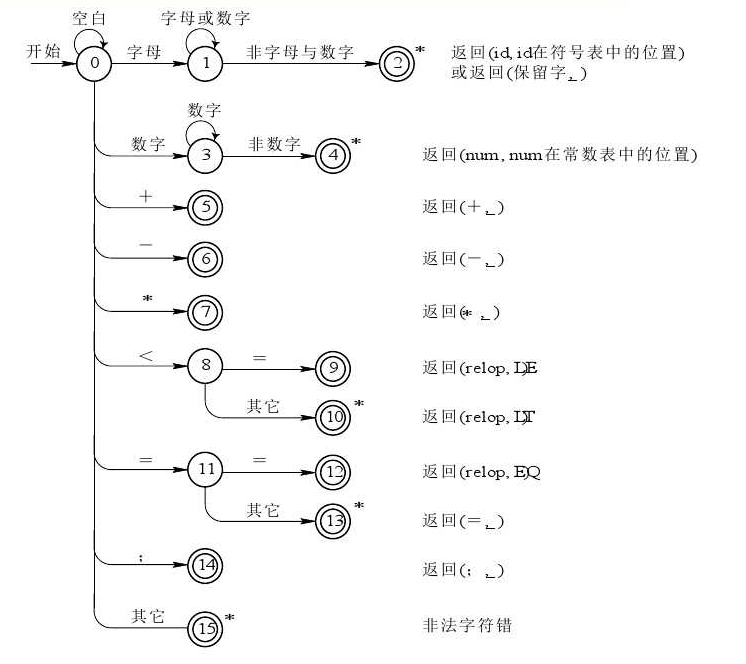
主程序：



扫描子程序：



（4）状态转换图：



（5）源程序：

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include<stdlib.h>

#include <stdio.h>

#include <math.h>

#include <string.h>

int i, j, k;

char c, s, a[20], token[20] = { '0' };

int letter(char s){

if ((s >= 97) && (s <= 122)) return(1);

else return(0);}

int digit(char s){

if ((s >= 48) && (s <= 57)) return(1);

else return(0);}

void get(){

s = a[i];

i = i + 1;}

void retract(){

i = i - 1;}

int lookup(char token[20]){

if (strcmp(token, "while") == 0) return(1);

else if (strcmp(token, "if") == 0) return(2);

else if (strcmp(token, "else") == 0) return(3);

else if (strcmp(token, "switch") == 0) return(4);

else if (strcmp(token, "case") == 0) return(5);

else return(0);

}

void main(){

printf("please input string :\n");

i = 0;

do{

i = i + 1;

scanf("%c", &a[i]);

} while (a[i] != '#');

i = 1;

j = 0;

get();

while (s != '#'){

memset(token, 0, 20);

switch (s){

case 'a':case 'b':case 'c':case 'd':case 'e':case 'f':

case 'g':case 'h':case 'i':case 'j':case 'k':case 'l':

case 'm':case 'n':case 'o':case 'p':case 'q':case 'r':

case 's':case 't':case 'u':case 'v':case 'w':case 'x':

case 'y':case 'z':

while (letter(s) || digit(s)){

token[j] = s;

j = j + 1;

get();

}

retract(); k = lookup(token);

if (k == 0)

printf("(%d,%s)\n", 6, token);

else printf("(%d,-)\n", k);

break;

case '0':case '1':case '2':case '3':case '4':

case '5':case '6':case '7':case '8':case '9':

while (digit(s)){

token[j] = s;

j = j + 1;

get();

}

retract();

printf("(%d,%s)\n", 7, token);

break;

case '+':printf("('+',null)\n"); break;

case '-':printf("('-',null)\n"); break;

case '\*':printf("('\*',null)\n"); break;

case '<':get();

if (s == '=') printf("(relop,LE)\n");

else{

retract();

printf("(relop,LT)\n");

}

break;

case '=':

get();

if (s == '=')

printf("(relop,EQ)\n");

else{

retract();

printf("('=',null)\n");

}

break;

case ';':

printf("(;,null)\n");

break;

case ' ':break;

default:printf("非法字符！\n");

}

j = 0;

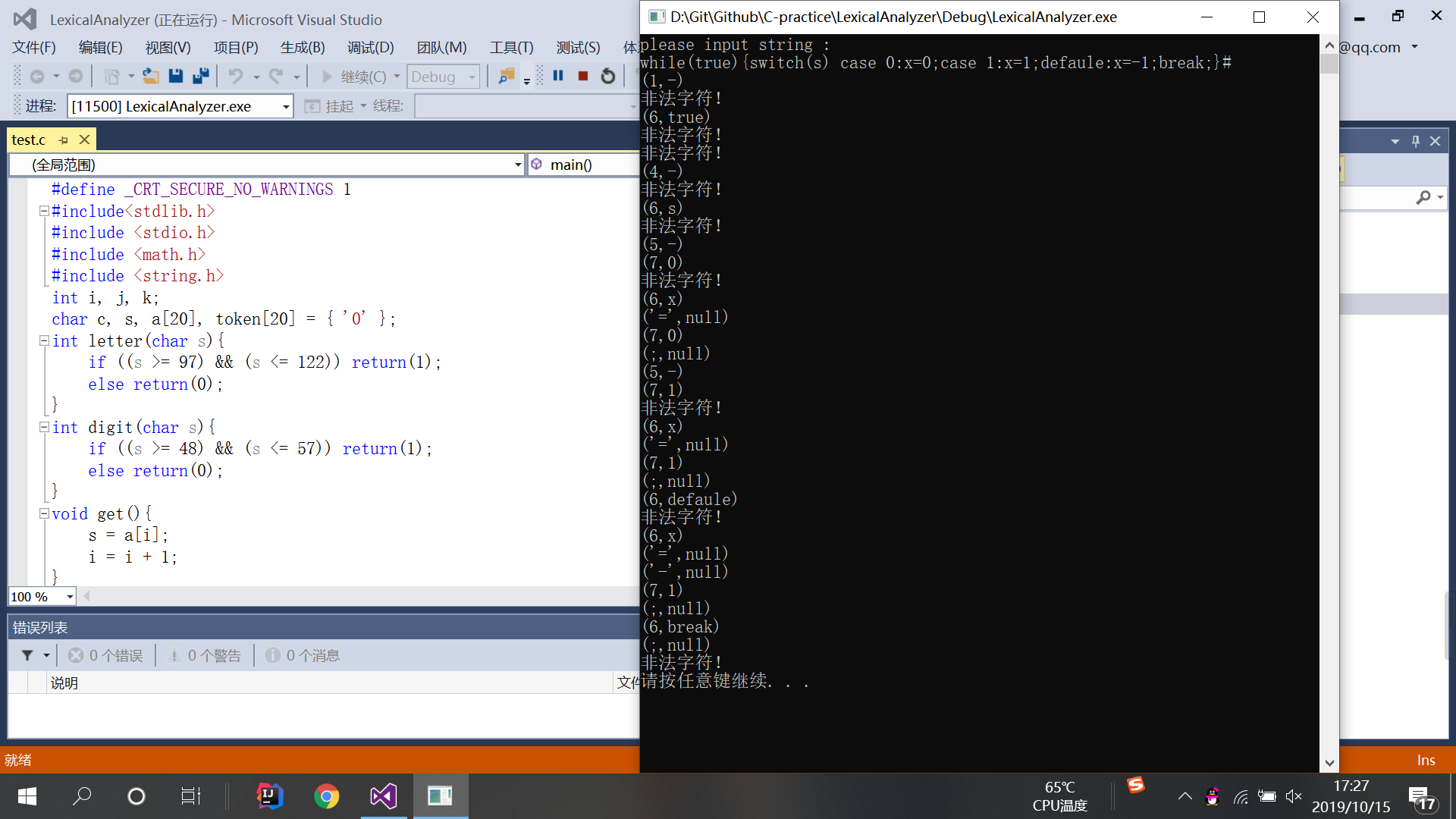
get();

}

system("pause");

}

实验结果：



四.实验结果分析及总结：

该程序实现了简单的词法分析，词法分析的基本任务是从字符串表示的源程序中识别出具有独立意义的单词符号，其基本思想是根据扫描到单词的第一个字符的种类，拼出相应的单词符号。通过本次实验的完成，更加加深了我对词法分析的理解。