一、实验目的

设计并实现一个词法分析器，深刻理解编译原理中词法分析器的原理。

二、实验内容

通过使用自己熟悉的语言设计并实现一个词法分析器，是此法分析器按要求的格式输出经过分析的程序段。

要求分析一下程序片段：

const a=10;

var b,c;

procedure p;

begin

c:=b+a;

end;

begin

read(b);

while b#0 do

begin

call p;writeln(2\*c);read(b);

end

end.

该程序片段存放在example.txt中。

三、实验步骤

 1、对PL\0文法中各类单词分类：

  （1）保留字：const、var、procedure、begin、end、if、then、while、do、read、call、write、writeln

  （2）常数：由0-----9这几个数字组成

  （3）标识符：由字母打头的字母和数字的字符串

  （4）运算符：+，-，\*，/，:=，>=，<=，#

  （5）分界符：，、.、(、)、；

  2、将各类单词对应到lex中：

  （1）保留字：

reservedWord [const|var|procedure|begin|end|if|then|while|do|read|call|write|writeln]，由于在lex中不区分大小写，所以将保留字写成：

reservedWord [cC][oO][nN][sS][tT]|[vV][aA][rR]|[pP][rR][oO][cC][eE][dD][uU][rR][eE]|

[bB][eE][gG][iI][nN]|[eE][nN][dD]|[iI][fF]|[tT][hH][eE][nN]|[wW][hH][iI]

[lL][eE]|[dD][oO]|[rR][eE][aA][dD]|[cC][aA][lL][lL]|[wW][rR][iI][tT][eE]

[wW][rR][iI][tT][eE][lL][nN]

  （2）常数：

constant ([0-9])+  /\*0---9这几个数字可以重复\*/

  （3）标识符：

      identfier [A-Za-z]([A-Za-z][0-9])\*

  （4）运算符：

      operator \+|-|\\*|\/|:=|>=|<=|#|=  /\*在lex中，有特殊意义的运算符要加转义字符\，如+、\*及\*、/

  （5）分界符：

      delimiter [,\.;]

  3、PL/0的语言的词法分析器要求跳过分隔符（如空格，回车，制表符），对应的lex定义为：

     delim [""\n\t]

whitespace{delim}+

  4、为lex制定一些规则：

     {reservedWord}{count++;printf("\t%d\t(1,‘%s’)\n",count,yytext);}   /\*对保留字定规则，输出设置\*/

     {operator} {count++;printf("\t%d\t(2,‘%s’)\n",count,yytext); }

     {delimiter}{count++;printf("\t%d\t(3,‘%s’)\n",count,yytext);}

{constant}{count++;printf("\t%d\t(4,‘%s’)\n",count,yytext);}

{identfier}{count++;printf("\t%d\t(5,‘%s’)\n",count,yytext);}

{whitespace} {/\* do    nothing\*/ }         /\*遇到空格，什么都不做\*/

  5、写子程序：

   voidmain()

{

   printf("词法分析器输出类型说明:\n");

         printf("1：保留字\n");

         printf("2：运算符\n");

         printf("3：分界符\n");

         printf("4：常  数\n");

         printf("5：标识符\n");

         printf("\n");

         yyin=fopen("example.txt","r");

             yylex(); /\* start the analysis\*/

         fclose(yyin);

         system("PAUSE");/\*暂停\*/

}

 intyywrap()

 {

        return 1;

}

当lex 读完输入文件之后就会调用函数 yywrap 。如果返回1 表示程序的工作已经完成了，否则，返回 0。