**电子科技大学 信息与软件工程 学院**

**标 准 实 验 报 告**

**（实验）课程名称 编译技术**

**电子科技大学教务处制表**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名:李富浩 学号:2014220301023 指导教师:周尔强**

**实验地点：信软楼305 实验时间：2015.9.20**

**一、实验室名称：信软 305/306**

**二、实验项目名称：词法分析器**

**三、实验学时：4学时**

**四、实验内容及步骤：**

1.用C语言编写词法分析器,为语法语义分析提供单词，使之能把输入的字符串形式的源程序分割成一个个单词符号传递给语法语义分析，并把分析结果(整数、浮点数、字符串常量、符号、标识符及关键字）输出.

实验步骤:1：清楚实验目的，即从外部读入文件，判断文件中的内容；2：清楚程序的流程，列举需要的函数；3：编写函数，包括各种判断函数，主函数；4：运行测试，出错修改；

1. 用 flex 完成1中的内容

实验步骤:1:在前面实验的基础上了解正则表达式，了解flex的用法；2：编写auto\_lex.l文件，并通过flex工具生成lex.yy.c文件；3：测试修改，这花费了很长的时间，因为flex并不能报c语言的一些错误，flex生成的c程序很长很长，所以调试只能一段一段地注释，然后编译看看哪里有错误。

**五、实验运行结果及测试：**

只用C语言实现的代码:

#include<stdio.h>

#include<stdlib.h>

#include "ctype.h"

#include "string.h"

#define KEY\_NUM 32;

char string\_temp[15] = {" "};

int string\_index = 0;

void show(char cha[])

{

int i = 0;

while (cha[i] != ' ')

//while(i<=20)

{

printf("%c", cha[i]);

i++;

}

printf(">\n");

}

//识别字母

int word(char ch)

{

int ch\_exist = 0;

if (isalpha(ch))

ch\_exist = 1;

return(ch\_exist);

}

//识别数字

int digit(char ch)

{

int dig\_exist = 0;

if (isdigit(ch))

dig\_exist = 1;

return (dig\_exist);

}

//识别空格

int space(char ch)

{

int space\_exist = 0;

if (ch == ' ')

space\_exist = 1;

return space\_exist;

}

//识别符号

int delimiter(char ch)

{

int del\_exist = 0;

int i;

char delimiters[13] = { '+','-','\*',';','(',')','{','}','<','>','!','=','#' };

for (i = 0; i<13; i++)

{

if (ch == delimiters[i])

{

del\_exist = 1;

break;

}

}

return(del\_exist);

}

//识别注释

int note(char ch)

{

int note\_exist = 0;

if (ch == '/')

note\_exist = 1;

return (note\_exist);

}

//识别字符串常量

int constant\_string(char ch)

{

int constant\_string\_exist = 0;

if (ch == '"')

constant\_string\_exist = 1;

return (constant\_string\_exist);

}

int keyword(char key[])

{

char \*keys []= { "auto","break","case","char","const","continue","do","double","else","enum","else","extern","float","for","goto","if","int","long","register","return","short","signed","sizoef","static","struct","switch","typedef","union","unsigned","void","volatile","while"};

int i;

for (i = 0; i < 32; i++)

{

if (strcmp(key, keys[i]) == 0)

{

return 1;

break;

}

}return 0;

}

void show\_word(char cha[])

{

if (keyword(cha))

{

printf("<关键字,");

show(cha);

}

else

{

printf("<标识符,", cha);

show(cha);

}

}

void cleararray(char cha[])

{

int i;

for (i = 0; i<40; i++)

cha[i] = ' ';

string\_index = 0;

}

void main()

{

FILE \*fpr;

char ch, file\_path[20];

int status\_index = 0;

printf("请输入要分析的文件的目录:");

scanf("%s", &file\_path);

if ((fpr = fopen(file\_path, "r")) == NULL)

{

printf("打开指定文件失败!");

printf("打开默认文件失败!");

system("pause");

exit(0);

}

printf("文件已成功打开\n");

while (!feof(fpr))

{

ch = fgetc(fpr);

if (status\_index == 0)

{

cleararray(string\_temp);

if (digit(ch))

{

status\_index = 1;

string\_temp[string\_index++] = ch;

}

else if (word(ch))

{

status\_index = 3;

string\_temp[string\_index++] = ch;

}

else if (delimiter(ch))

{

printf("<符号,%c>\n", ch);

status\_index = 0;

cleararray(string\_temp);

}

else if (note(ch))

{

status\_index = 4;

}

else if (constant\_string(ch))

{

status\_index = 7;

string\_temp[string\_index++] = ch;

}

}

else if (status\_index == 1)

{

if (digit(ch))

{

string\_temp[string\_index++] = ch;

}

else if (ch == '.')

{

status\_index = 2;

string\_temp[string\_index++] = ch;

}

else if (delimiter(ch))

{

printf("<整数,");

show(string\_temp);

cleararray(string\_temp);

printf("<符号,%c>\n", ch);

}

else if (constant\_string(ch))

{

printf("<整数,");

show(string\_temp);

cleararray(string\_temp);

status\_index = 7;

}

else if (note(ch))

{

printf("<整数,");

show(string\_temp);

cleararray(string\_temp);

status\_index = 4;

}

else if (space(ch))

{

printf("<整数,");

show(string\_temp);

cleararray(string\_temp);

status\_index = 0;

}

else if (word(ch))

{

printf("这是什么?");

}

}

else if (status\_index == 2)

{

if (digit(ch))

{

string\_temp[string\_index++] = ch;

}

else if (delimiter(ch))

{

printf("<浮点,");

show(string\_temp);

cleararray(string\_temp);

printf("<符号,%c>\n", ch);

}

else if (constant\_string(ch))

{

printf("<浮点,");

show(string\_temp);

cleararray(string\_temp);

status\_index = 7;

}

else if (note(ch))

{

printf("<浮点,");

show(string\_temp);

cleararray(string\_temp);

status\_index = 4;

}

else if (space(ch))

{

printf("<浮点,");

show(string\_temp);

cleararray(string\_temp);

status\_index = 0;

}

else if (word(ch))

{

printf("这是什么?");

}

}

else if (status\_index == 3)

{

if (digit(ch))

{

string\_temp[string\_index++] = ch;

}

else if (delimiter(ch))

{

string\_temp[string\_index++] = '\0';

show\_word(string\_temp);

cleararray(string\_temp);

status\_index = 0;

printf("<符号,%c>\n", ch);

}

else if (constant\_string(ch))

{

string\_temp[string\_index++] = '\0';

show\_word(string\_temp);

cleararray(string\_temp);

status\_index = 7;

}

else if (note(ch))

{

string\_temp[string\_index++] = '\0';

show\_word(string\_temp);

cleararray(string\_temp);

status\_index = 4;

}

else if (word(ch))

{

string\_temp[string\_index++] = ch;

}

else if (space(ch))

{

string\_temp[string\_index++] = '\0';

show\_word(string\_temp);

cleararray(string\_temp);

status\_index = 0;

}

}

else if (status\_index == 4)

{

if (ch == '\*')

{

status\_index = 6;

}

else if (ch == '/')

{

status\_index = 5;

}

else

{

printf("<符号,/>\n");

status\_index = 0;

cleararray(string\_temp);

}

}

else if (status\_index == 5)

{

printf("<注释符,//>\n");

printf("<注释,");

do {

ch = fgetc(fpr);

printf("%c", ch);

} while (ch != '\n');

printf(">");

status\_index = 0;

printf("status 5");

}

else if (status\_index == 6)

{

printf("<注释符,/\*>\n");

printf("<注释段,");

do {

ch = fgetc(fpr);

printf("%c", ch);

} while (ch != '\*');

if (ch = fgetc(fpr) == '/')

{

printf(">\n");

printf("<注释符,\*>\n");

status\_index = 0;

}

else

printf("\*");

}

else if (status\_index == 7)

{

printf("<符号,\">\n");

printf("<字符串常量,");

do {

printf("%c", ch);

ch = fgetc(fpr);

} while (ch != '"');

printf(">\n");

printf("<符号,\">\n");

status\_index = 0;

}

}

fclose(fpr);

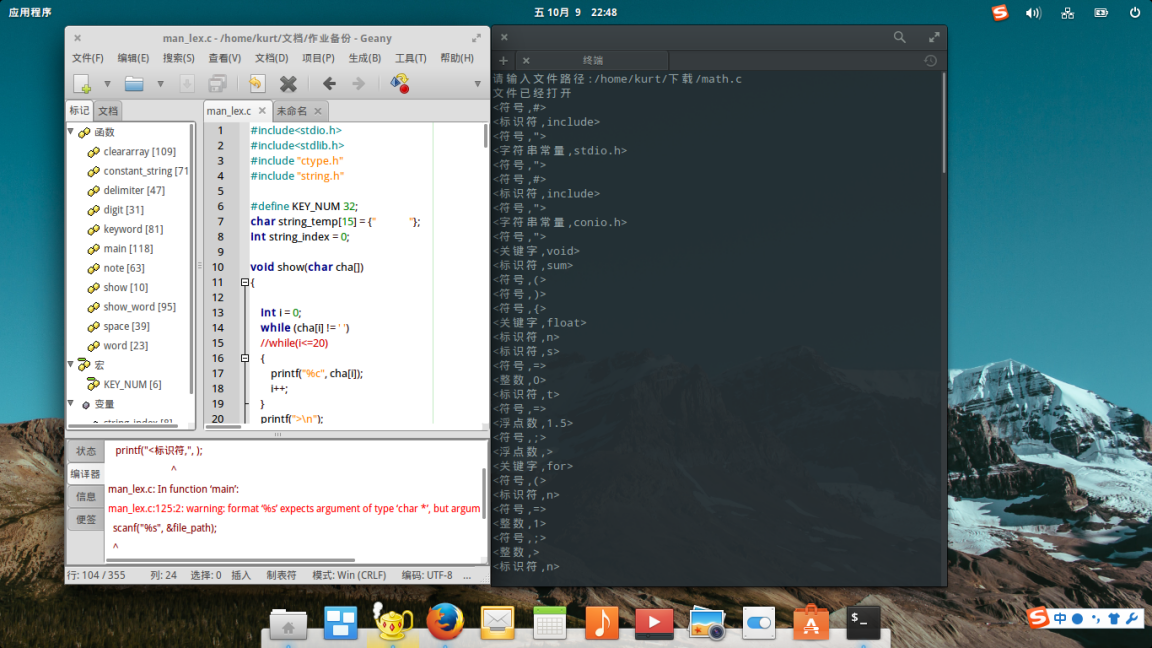
printf("分析完成!\n");

system("pause");

return 0;

}

运行结果



使用flex工具完成的代码：

%{

#include <ctype.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

%}

digit [0-9]

num {digit}+

space [ \t]+

enter [\n]+

%%

\/\\*[\w\W]\*?.\*|\/\/.\* {printf("<注释,%s>\n",yytext);}

\".\*\" {printf("<字符串常量,%s>\n",yytext);}

[ \t] {}

[\n] {}

{digit}+ {printf("<整数,%s>\n",yytext);}

{digit}+"."{digit}\* {printf("<浮点数,%s>\n",yytext);}

"int"|"else"|"return"|"void"|"if"|"while"|"float"|"else"|"for" {printf("<关键字,%s>\n",yytext);}

"#"|"("|")"|"{"|"}"|","|"="|";"|"+="|"&"|"!="|"%"|"/" {printf("<符号,%s>\n",yytext);}

[a-zA-Z\_][a-zA-Z0-9]\* {printf("<标识符,%s>\n",yytext);}

%%

int main(void)

{

char file\_path[20];

printf("请输入文件路径:");

scanf("%s",&file\_path);

FILE \*fpr;

if ((fpr = fopen(file\_path, "r")) == NULL)

{

printf("打开文件失败!");

system("pause");

exit(0);

}

yyin = fopen(file\_path,"r");

printf("分析完成 \n");

return yylex();

}

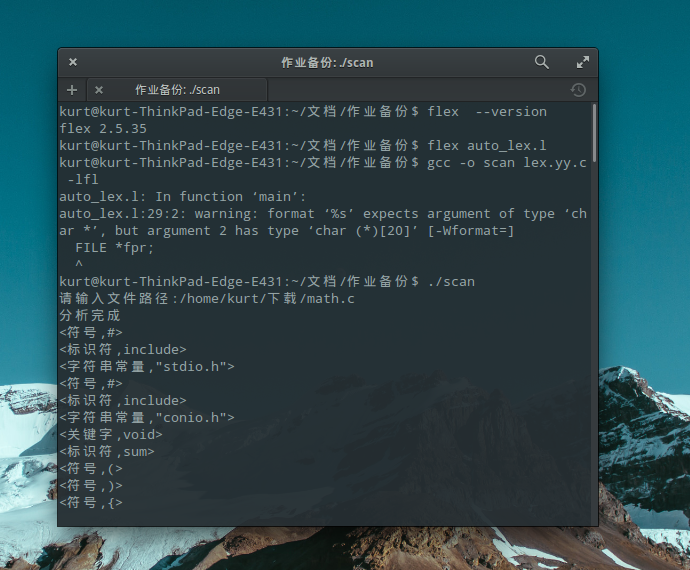
int yywrap()

{

return 1;

}

运行结果



1. **实验结论与总结：**

运行的结果都一样，但是使用flex工具代码只有几十行，而纯c语言写的代码有几百行，所以善用工具可以提高效率，我们要善用工具。

**报告评分：**

**指导教师签字：**