**编译原理**

实验一 TINY语言的词法分析

1. 实验目的

构造tiny语言的词法分析器（扫描器），要求利用第三方的lex工具进行构造。构造出的扫描器，能够读入教材样例中给出的tiny语言的示例代码，分解成token输出。

掌握使用lex工具

掌握构造词法分析器

掌握LEX的输入文件的编写

1. 实验设计

掌握词法分析器的构造和使用，学会使用 fex 工具来构造词法分析程序。

1.选择教材《程序清单2-3 TINY语言中的样本程序》作为输入

2.选择教材中《程序清单2-4》作为输出

这两个程序清单的教材截图放在报告最后的**附件1中**

3.我使用的是mac系统，所以是可以直接编译flex .l的文件的，借此构建了工作环境

4.程序设计

4-1：将[0-9]+作为digit **匹配数字**

4-2：将read |if| then| repeat | until |write | end 作为keyword **匹配关键字**

4-3：将[a-zA-Z]+作为letter **匹配单词字母**

4-4：将\{[^\}]\*\} 作为comment**匹配注释**

4-5；将":="|"="|"+"|"-"|"\*"|"/"|">"|"<"作为operator **匹配操作符**

4-6：将[" "|\t|\n]+ 作为ignore**匹配空格、换行、制表符 进行忽略**

4-7：将[","|";"]+ 作为other **正常输出**

4-8：将匹配到的token，根据书本给出的格式进行输出到屏幕上

1. 内容和步骤

1代码：

**tiny.l**

%{

#include "stdio.h"

int lineNum;

%}

digit [0-9]+

keyword read|if|then|repeat|until|write|end

letter [a-zA-Z]+

comment \{[^\}]\*\}

operator ":="|"="|"+"|"-"|"\*"|"/"|">"|"<"

ignore [" "|\t|\n]+

other [","|";"]+

%%

{comment} {

}

{digit} { printf("NUM,val= %s\n",yytext);

}

{keyword} { printf("reserved word：%s\n",yytext);

}

{letter} { printf("ID,name= %s\n",yytext);

}

{operator} { printf("%s\n",yytext);

}

{other} { printf("%s\n",yytext);

}

{ignore} {

/\*nothing\*/

}

%%

int yywrap()

{

return 1;

}

int main()

{

lineNum = 0;

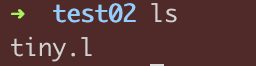
yylex();

return 0;

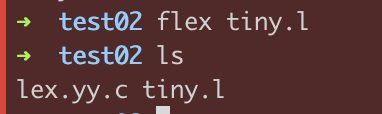
}

2、步骤：

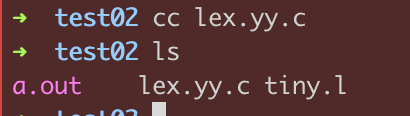
(1)在一个空文件夹内vim一个tiny.l 文件，并键入代码



(2)使用flex命令，对tiny.l文件进行编译。生成lex.yy.c文件

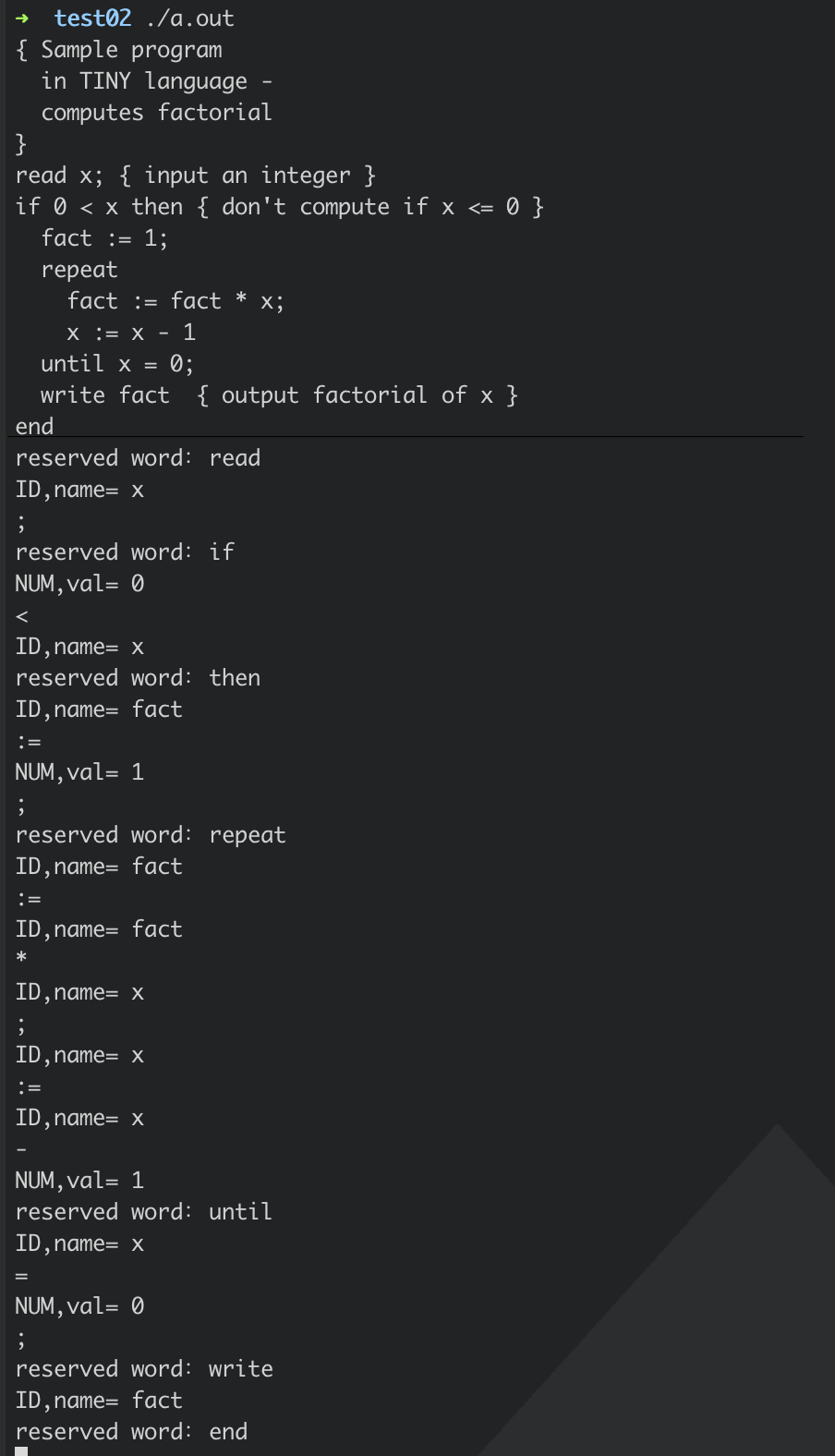


(3)使用cc命令 编译lex.yy.c文件，生成a.out 可执行文件



3、结果：

**图示：**end上方是输入下方是输出

****

**输入**

{ Sample program

in TINY language -

computes factorial

}

read x; { input an integer }

if 0 < x then { don't compute if x <= 0 }

fact := 1;

repeat

fact := fact \* x;

x := x - 1

until x = 0;

write fact { output factorial of x }

end

**输出**

reserved word：read

ID,name= x

;

reserved word：if

NUM,val= 0

<

ID,name= x

reserved word：then

ID,name= fact

:=

NUM,val= 1

;

reserved word：repeat

ID,name= fact

:=

ID,name= fact

\*

ID,name= x

;

ID,name= x

:=

ID,name= x

-

NUM,val= 1

reserved word：until

ID,name= x

=

NUM,val= 0

;

reserved word：write

ID,name= fact

reserved word：end

1. 实验结论:

1 理论基础

1. 关于LEX工具的使用方法
2. 掌握LEX输入文件的格式

LEX输入文件由3个部分组成：定义集、规则集、以及辅助程序集或用户程序集。这3个部分由位于新一行第1列的双百分号分开

(3)了解词法分析器的内部工作原理

(4)了解TINY语言的基本关键字和结构

TINY 的单词记号分为三种典型类型：保留字、特殊符号和“其他”单词。保留字一共 8 个，特殊符号包括运算符和界符：分别是四种基本的整数运算符号，两种比较符号（等号和小于），以及括号、分号和赋值号。除赋值号是两个字符的长度以外，其余均为一个字符 TINY 的标识符是一个或多个字母的序列。数是一个或多个数字的序列。除了单词之外，TINY 还要遵循以下词法规则：注释应放在花括号{…}中，且不可嵌套；代码应是自由格式；空白符由空格、制表位和新行组成。

2、分析和总结

**问题：**在编写词法分析器时，我发现在cc lex.yy.c文件的时候mac终端会报错，上网查询相应资料后了解到，mac的Xcode更新后并没有把全部的c语言库预装，所以报错，自己手动安装之后便解决了这个问题。

**结果：**该词法分析器功能良好，可以完成预定的要求。实验结果按教材中给出的例子的格式，把输入的TINY语言代码分解成token输出，并按字母、数字、关键字、操作符、进行标注，忽略了空格、换行、制表符。

**总结：**通过这次实验，我对词法分析器有了进一步的了解，把理论知识应用于实验中。通过这次语义分析的实验, 我对高级语言的学习有了更深的认识 ，了解得更透彻。 同时我了解了高级语言转化为目标代码或汇编指令的过程,。对今后的学习将起很大的作用,对以后的编程有很大的帮助。

3、对工具的评价

Mac上的LEX的确挺方便的，不需要安装其他的东西就可以直接使用了，缺点就是在终端下执行的，功能比较单一，只能简单的编译执行.l文件，并且mac上使用LEX的时候有可能遇到缺少头文件库的问题，因为mac的Xcode没有将c语言的部分库预装到电脑中，使用的时候可能需要自己安装。

1. 附件

附件1:《程序清单2-3》和《程序清单2-4》

