**编译原理**

实验一 TINY语言的词法分析

1. 实验目的

①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。

②掌握使用lex工具

③掌握构造词法分析器

④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

①将[0-9]+作为digit **匹配数字**

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

③将[a-zA-Z]+作为letter **匹配单词字母**

④将\{[^\}]\*\} 作为comment**匹配注释**

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

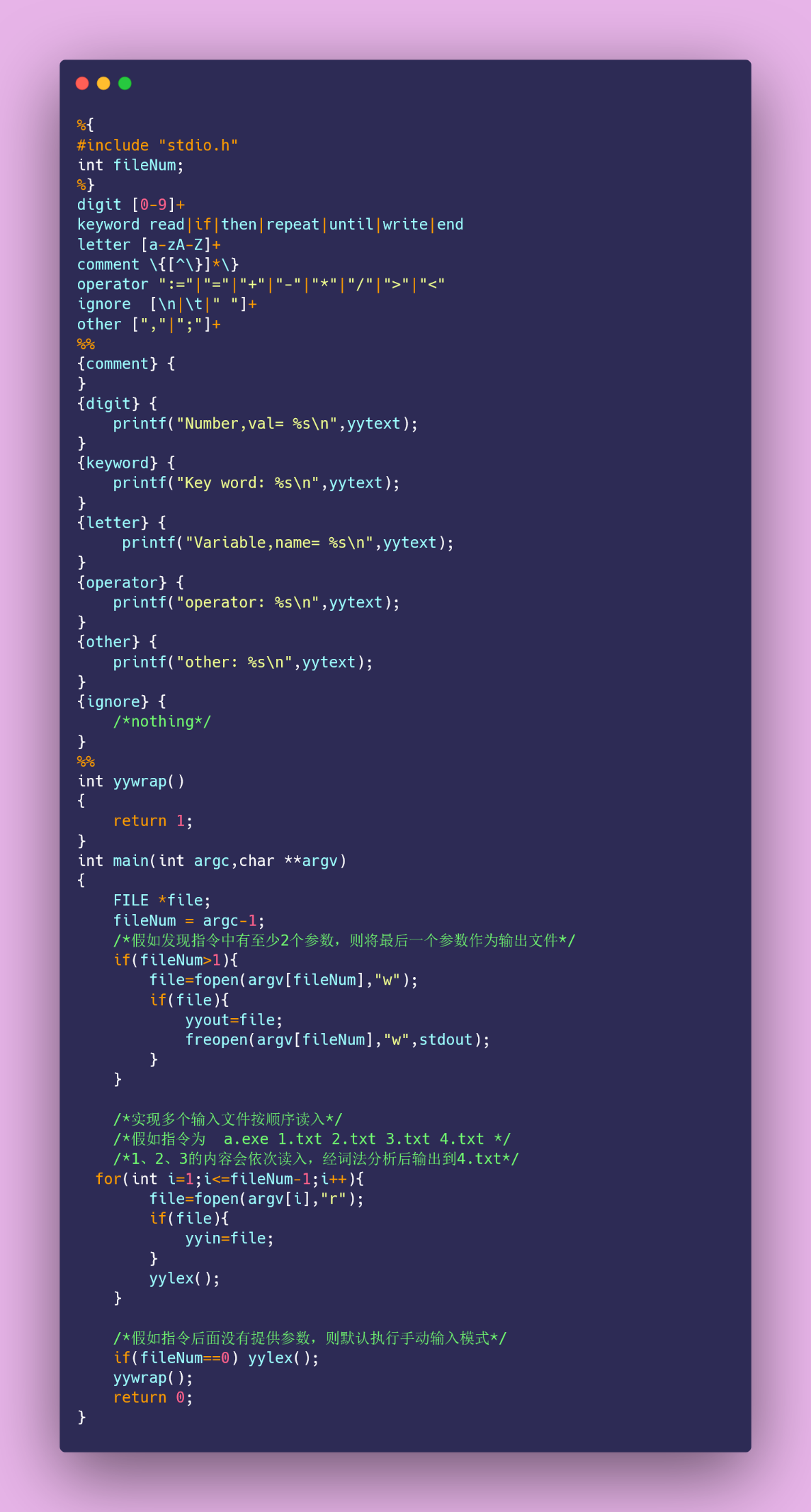
⑥将[" "|\t|\n]+ 作为ignore**匹配空格、换行、制表符 进行忽略**

⑦将[","|";"]+ 作为other **正常输出**

将匹配到的token，根据书本给出的格式进行输出到txt文件中

1. 实验步骤

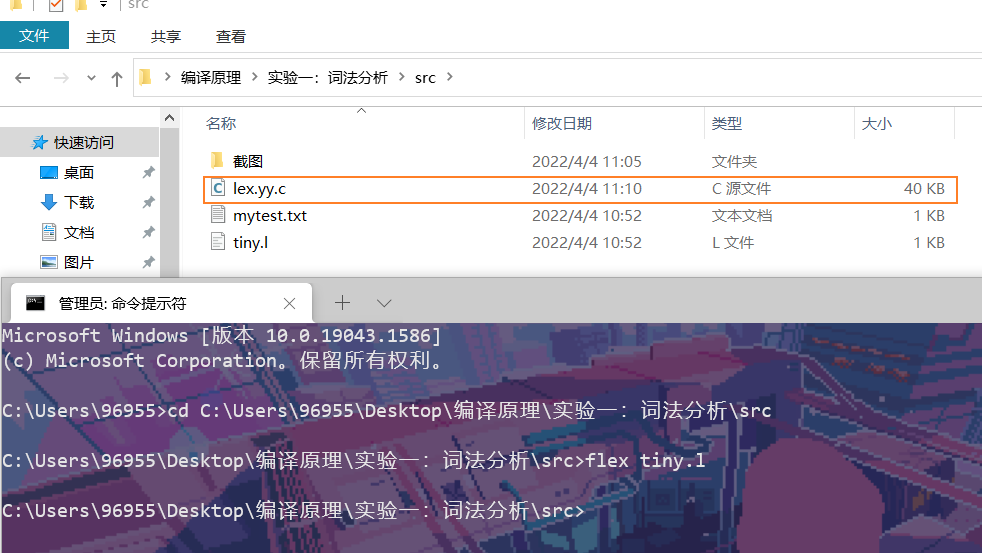
1、代码 tiny.l



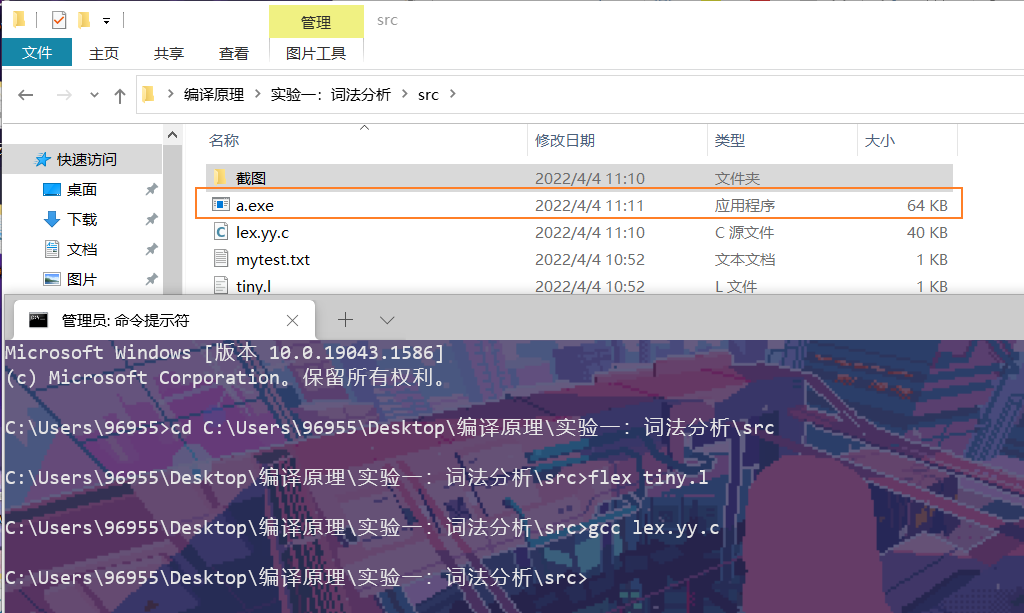
2、使用过程

①跳转到tiny.l所在的目录

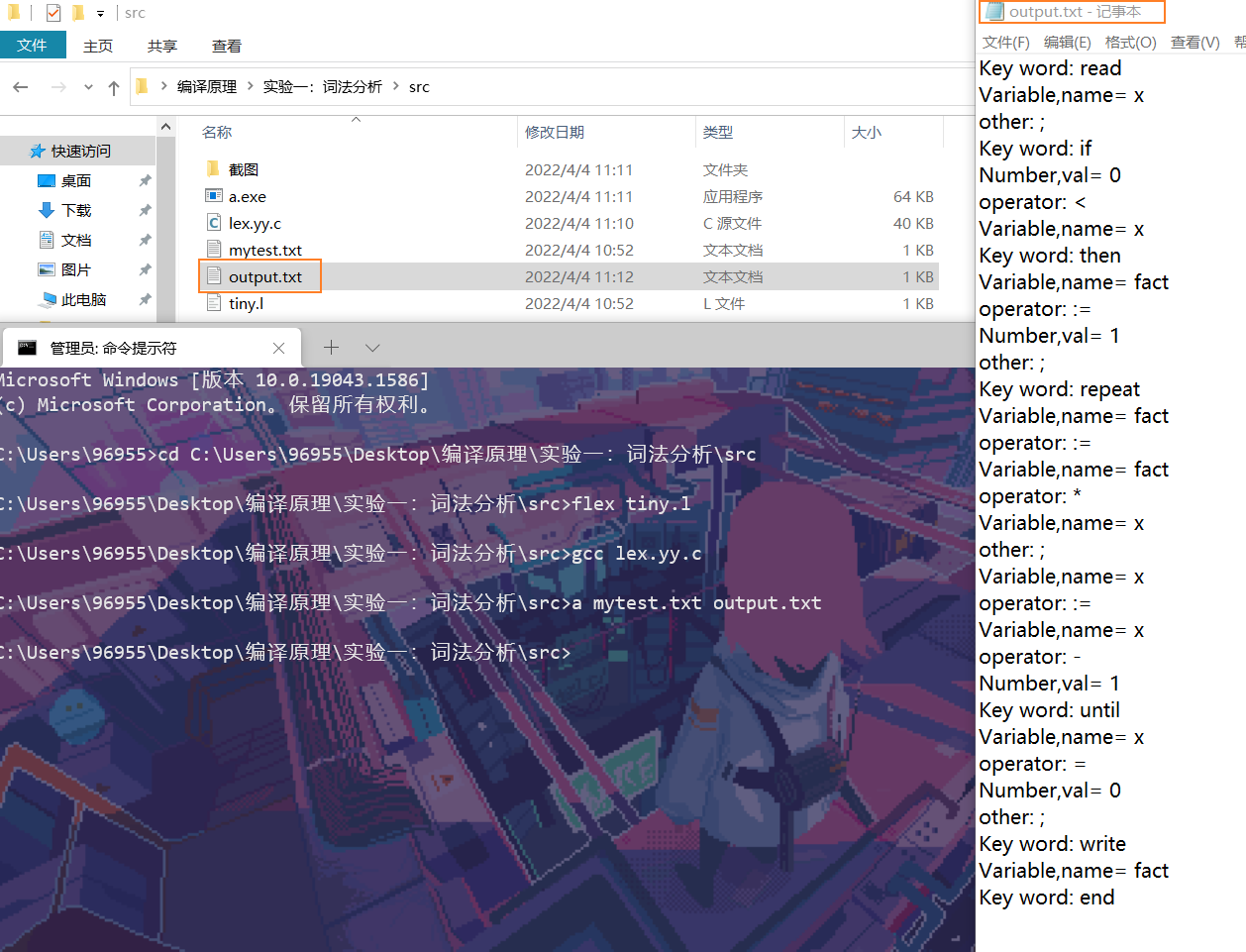
②执行 flex tiny.l 生成 lex.yy.c



③执行 gcc lex.yy.c 生成可执行文件 a.exe



④执行 a.exe mytest.txt output.txt 实现从mytest.txt中获取输入，在output.txt中打印输出



3、输入 mytest.txt

{ 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

4、输出结果 output.txt

Key word: read

Variable,name= x

other: ;

Key word: if

Number,val= 0

operator: <

Variable,name= x

Key word: then

Variable,name= fact

operator: :=

Number,val= 1

other: ;

Key word: repeat

Variable,name= fact

operator: :=

Variable,name= fact

operator: \*

Variable,name= x

other: ;

Variable,name= x

operator: :=

Variable,name= x

operator: -

Number,val= 1

Key word: until

Variable,name= x

operator: =

Number,val= 0

other: ;

Key word: write

Variable,name= fact

Key word: end

1. 实验结论:

1、相关理论

①lex的使用方法

②lex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。这3 个部分由位于新一行第1列的双百分号分开

③tiny语言的基本关键字和结构

Tiny中的单词记号分为 保留字、特殊符号和“其他”单词 三种。

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（”=”和”<”），以及括号(“(”)、分号(“;”)和赋值号(“:=”)。

数是一个或多个数字的序列。

注释应放在花括号{…}中，且不可嵌套。

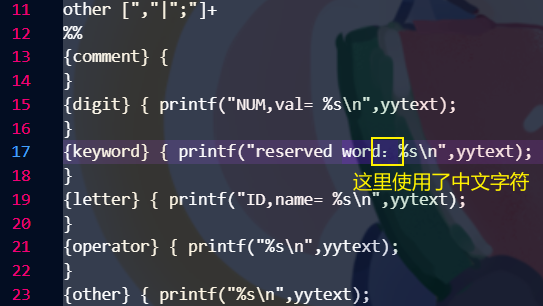
代码为自由格式，不强制要求缩进。

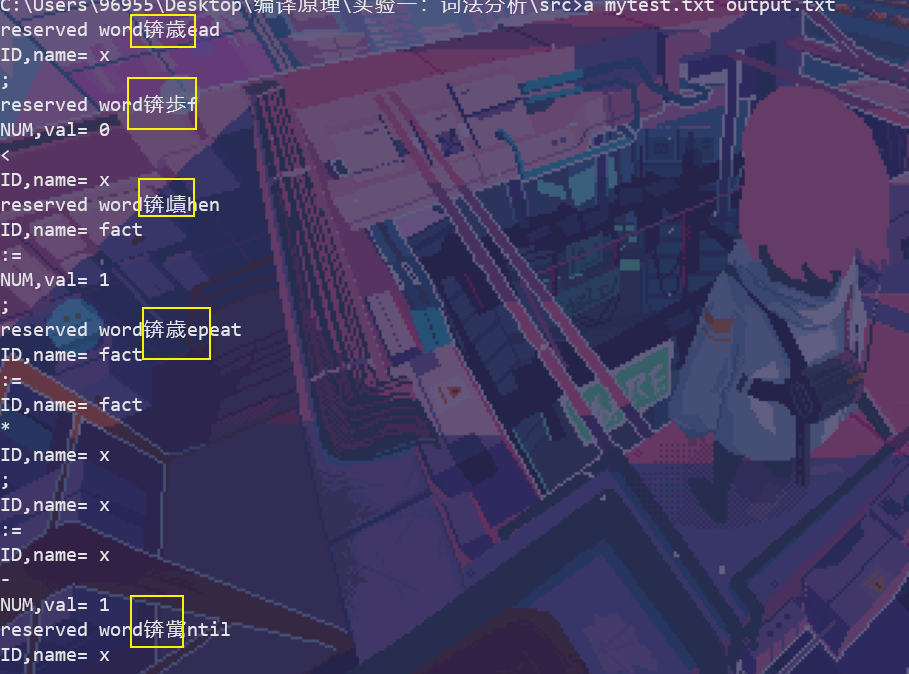
空白符由空格、制表位和新行组成。

④理解底层的词法分析

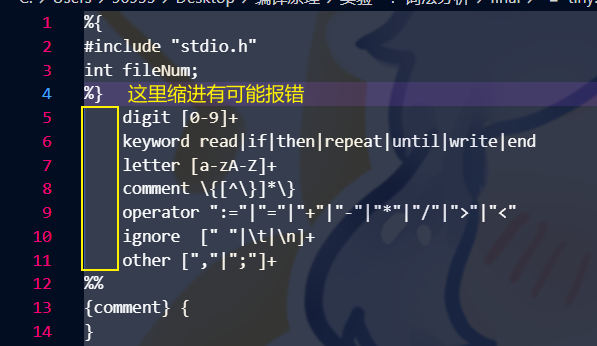
2、遇到的一些问题

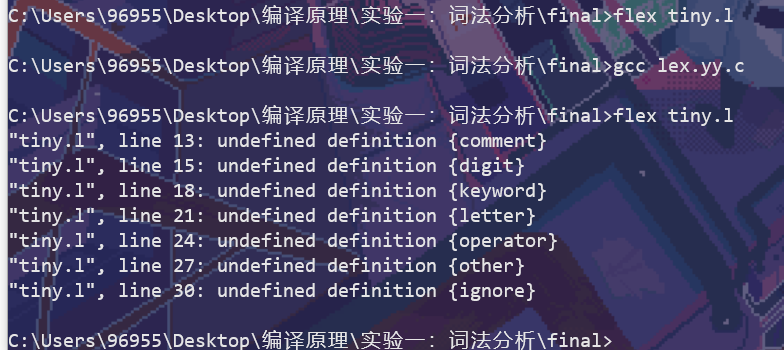
①tiny.l中最好不要使用中文字符，有可能会乱码





②tiny.l中，前面部分（%%区域）开头最好不要缩进，有可能报错





3、总结

通过本次实验，我进一步认识了词法分析。对词法分析的理解过程和计算机网络的学习有相近之处，因为二者都是短时间内完成的，平时很难关注的到，比如网络的“输入一个网址，回车，跳出网页的过程中发生了什么”，这一问题的模式就类似于编译过程中的“点击运行按钮到程序执行的过程中发生了什么”。再加上翻阅文献，我更好地了解了从高级语言到目标代码(或汇编指令)的过程，对今后的编程大有裨益。