实验一 TINY语言的词法分析

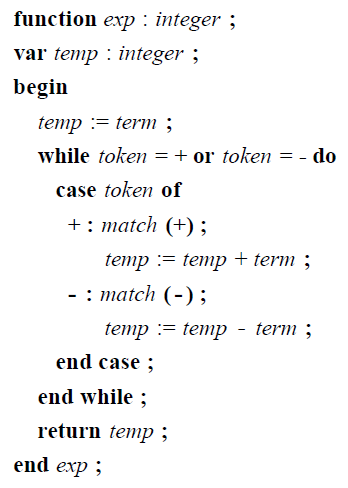
实验目的

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。实验结果：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。

Experimental Requirement

Build the lexical parser (a scanner) of TINY language, using the Lex tool of third party. The experimental result Scanner can receive the sample program of Tiny language, and output is a sequence of tokes that are defined by regular expression.

输入：



预习要求：

Preview requirements:

1.查阅各种专业文献，针对任意二种现代程序设计语言编译器的词法分析技术，找出各自的标志性的特征点，并加以比较分析和比较，指出其优缺点，并试图给出自已的结论。

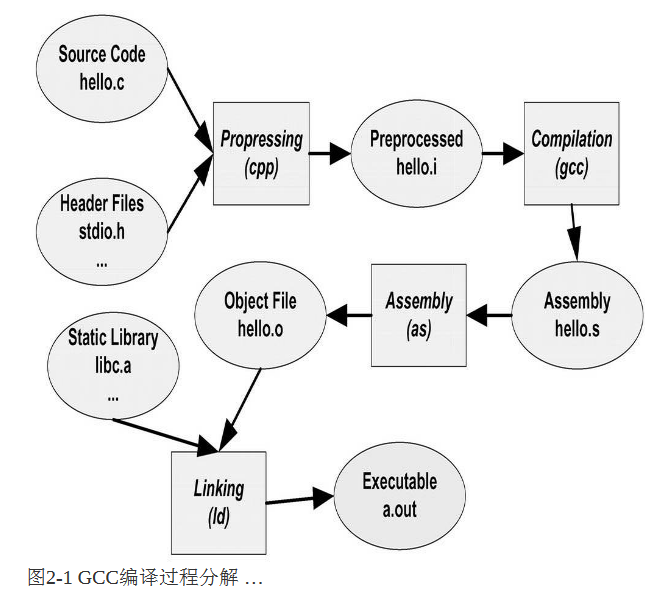
Referring to professional literatures, aiming at the lexical analysis technology of any two modern programming language compilers, finds out their respective features, compares them, points out their advantages and disadvantages, and tries to give your Evaluation and conclusions.

**(1)C++编译器**

c++编译器会将c++程序翻译成机器语言并保存到目标文件中。如果编译通过，就会把CPP转换成OBJ文件。

目标文件以机器码的形式包含了编译单元里所有的函数和数据、导出符号表、未解决符号表、地址重定向表等。

c++编译过程中，编译器会读取源程序的字符流，依次进行词法、语法、语义分析，将高级语言指令转换为功能等效的汇编代码,再转换为机器代码，生成目标文件（.obj）。



**细分过程**

Ⅰ.编译：

①预处理阶段：

宏#define

条件编译指令，如#ifdef，#ifndef，#else，#elif，#endif等

头文件包含，#include <iostream>

②编译优化阶段：

针对代码优化，不依赖具体计算机

针对计算机优化

Ⅱ.汇编：

把汇编语言代码翻译成目标机器指令，生成目标文件（.o文件、.obj文件）。此过程会依赖机器的硬件和操作系统环境。

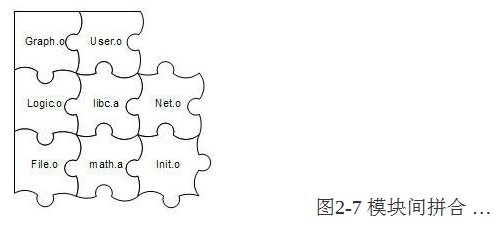
Ⅲ.链接：

将有关的目标文件（库文件、.o文件）彼此相连接，也即将在一个文件中引用的符号同该符号在另外一个文件中的定义连接起来，使得所有的这些目标文件成为一个能够被操作系统装入执行的统一整体。这个过程涉及地址的重定向。

链接分类：

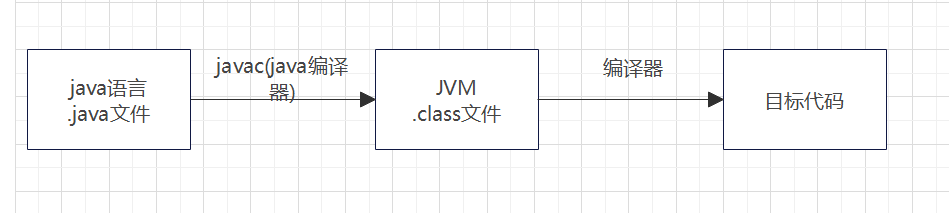
静态链接:在链接过程中就将库的内容加入到可执行程序中。缺点是生成的可执行文件太大，需要更多的系统资源，在装入内存时也会消耗更多的时间。

动态链接:等到程序运行时再按需求进行链接，能够将程序和模块区分开来。



**(2)JAVA编译器**

java语言是基于java虚拟机(JVM)而实现的一种高级语言，它需要通过java编译器编译成JVM识别的语言，最后由JVM实现到目标语言的转换。



javac是把java高级语言转变成JVM识别的一种二进制代码，可以将.java文件转 变为.class文件。

JVM识别的.class文件中存储的时字节码，转变的过程由JVM提供相关语法。

Javac的主要过程

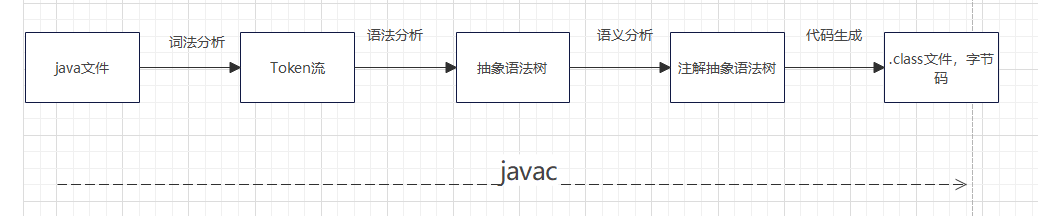
从java语言到字节码的转变要经过四个过程：

①java语言到Token流的过程，称为词法分析；

②Token流到抽象语法树的过程，称为语法分析；

③解析复杂的树节点，如语法糖的解析等，称为语义分析；

④抽象语法树到字节码的过程，称为代码生成。



javac的任务就是将java源代码语言转换成jvm能够识别的语言，然后jvm将jvm 语言再转化成当前机器能够识别的语言。这样做的好处是可以实现跨平台，不同平台上 只要拥有JVM就能编译对方的java代码

2.请描述自已拟定的实验计划和步骤，以及对输入和输出的设计。

Please design your experimental plan and procedures, as well as the design of input and output of the SCANNER

**计划：**

掌握词法分析器的构造和使用，学会使用 flex 工具来构造词法分析程序

**步骤：**

1. 学会使用flex工具
2. 构造tiny语言的词法分析器（扫描器），要求
3. 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。

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

输出：

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

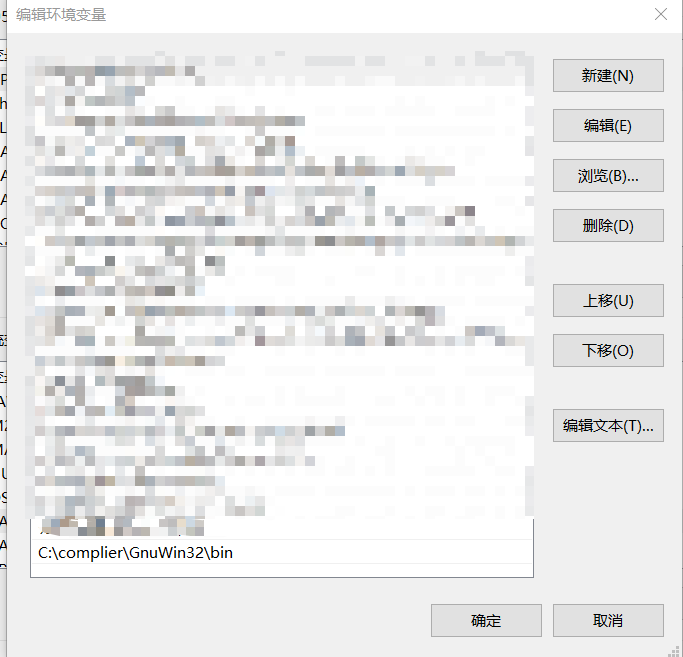
3.选择三方工具，阅读其使用说明，并尝试建立起工作环境。

Choose the LEX tools, read their instructions, and try to establish a working environment.

①下载最新版的GnuWin32或MinGW，并解压。Bin目录下有**flex.exe**和**bison.exe**

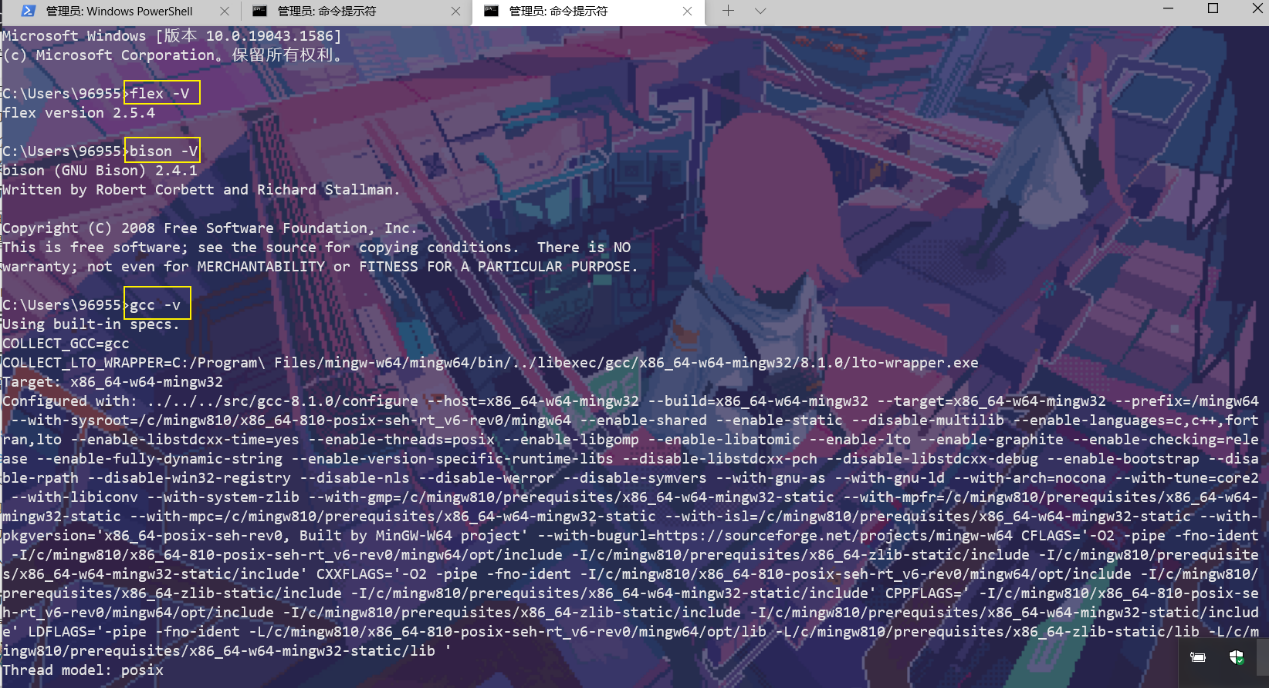


②将上面的路径配置进环境变量



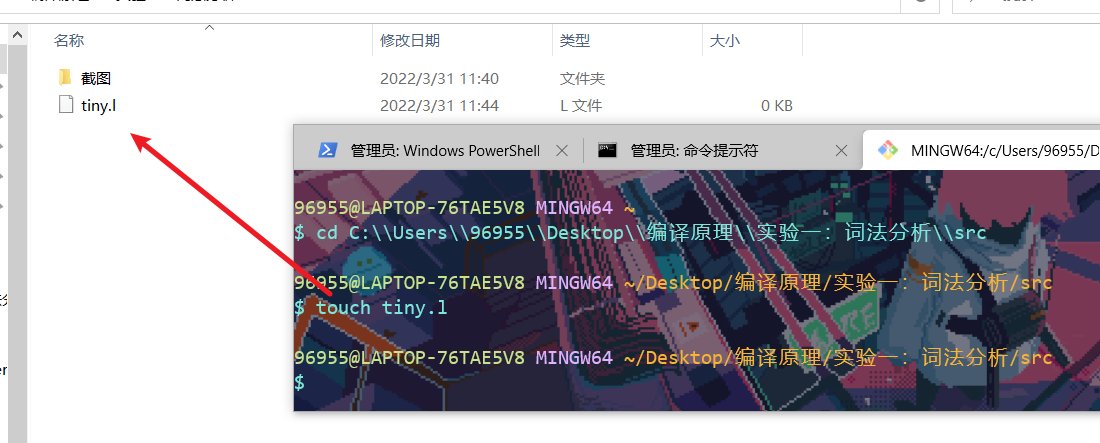
③以管理员方式打开cmd，分别输入flex -V、bison-V、gcc -v三个命令查看是否配置成功

注：flex和biston的v必须大写

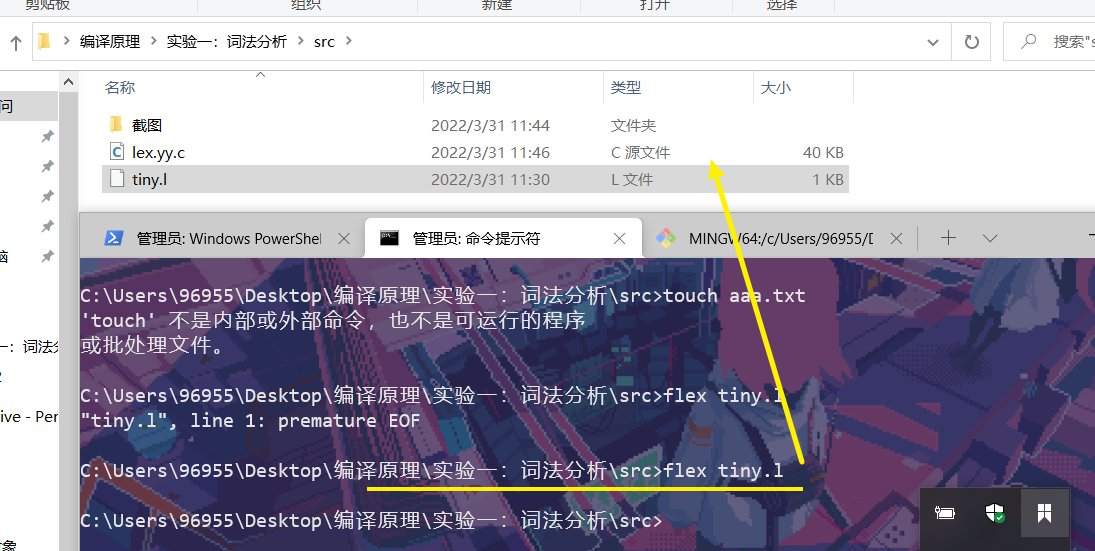


**使用例：**

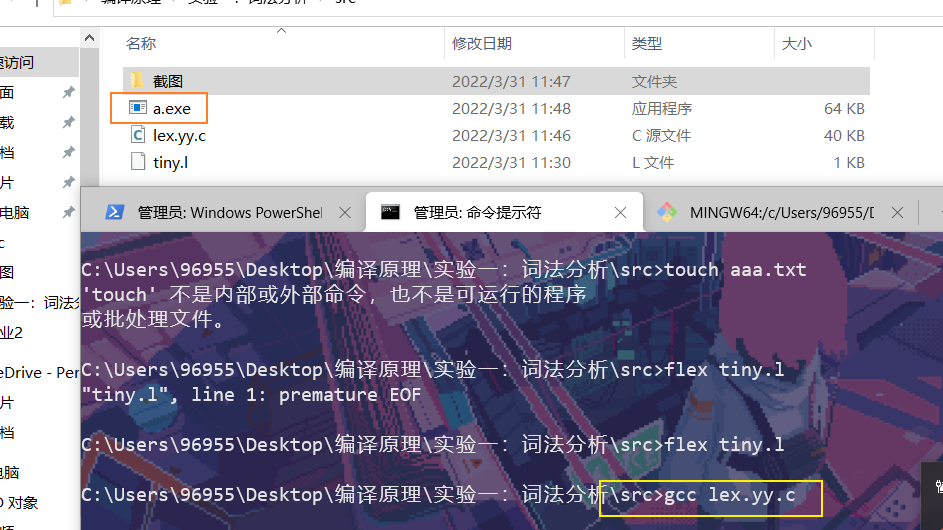
①新建tiny.txt文件。然后按照规定的格式写入代码



②通过cmd进入该目录。通过flex指令调用tiny.l 生成lex.yy.c



③通过gcc命令编译lex.yy.c生成可执行文件



④cmd在管理员模式下，输入这个可执行文件的文件名和对应的参数可以直接运行



查阅资料:

[1] <https://blog.csdn.net/weixin_44007632/article/details/108666375>

[2] <https://blog.csdn.net/weixin_30394333/article/details/95790165>

[3]《程序员的自我修养》俞甲子、石凡、潘爱民 电子工业出版社 2009.4 ISBN 7121085119

[4] <https://www.cnblogs.com/ericling/articles/11736681.html>

[5] https://blog.csdn.net/weixin\_41966991/article/details/81152490