编译技术

School of Software,Yunnan University

云南大学软件学院

柳青

E-mail: liuqing@ynu.edu.cn

2017—2018学年下学期

课程的性质与任务

- “编译技术”是计算机类专业特别是计算机软件工程专业的一门重要专业课。
- 开设本课程的目的，在于系统地介绍编译系统的结构、工作流程及编译程序各组成部分的设计原理和实现技术。

课程的性质与任务

- 尽管“编译程序”是特指将高级程序设计语言翻译成低级语言的软件，但编译程序构造的基本原理和技术也广泛应用于一般软件的设计和实现。
- 通过学习本课程，既掌握编译理论和方法方面的基本知识，也具有设计、实现、分析和维护编译系统等大中型软件方面的初步能力。

* 本课程要求及成绩评定

第一章　编译程序概论

1.1 什么是编译程序

1. 程序设计语言与程序的翻译

(1)程序设计语言

- 机器语言: 由0、1代码构成，不需翻译就可直接执行其程序。
- 汇编语言: 机器指令助记符（伪代码）形式，汇编后才可执行其程序。 例如80386宏汇编语言。
- 高级程序设计语言: 类自然语言和数学公式形式，例如FORTRAN（1956，第1个高级语言），BASIC, ALGOL, COBOL, PL/1, PASCAL，Ada, C, SmallTalk,C++, Java, C#，PHP, Perl, Python 等。

(2) 基本术语

- 源程序(Source Program)：用源语言写的程序。源语言可以是汇编语言，也可以是高级程序设计语言。

- 目标程序(Target Program)：也称为“结果程序”，是源程序经翻译程序加工以后所生成的程序。目标程序可以用机器语言表示，也可以用汇编语言或其它中间语言表示。

- 翻译程序(Translating Program)：是指把一个源程序翻译成逻辑上等价的目标程序的程序。源程序为其输入，目标程序为其输出。

- 汇编程序(Assembler)：是指把一个汇编语言写的源程序转换成等价的机器语言表示的目标程序的翻译程序。

- 编译程序(Compiler)：若源程序是用高级程序设计语言所写，经翻译程序加工生成目标程序，则该翻译程序就称为“编译程序”，也可称为编译器。

- 运行系统(Running System)：目标程序执行时，需要有一些子程序（如一些连接
•装配程序及一些连接库等）配合进行工作，由这些子程序组成的一个子程序库称为运行系统。

- 编译系统(Compiling System)：编译程序和运行系统合称编译系统。

•解释程序：开发过程中不需要编译，边执行边解释。

(3) 程序的翻译 除机器语言程序外，用其它语言书写的程序都必须经过翻译才能被计算机识别。这一过程由翻译程序来完成。

·翻译程序:

甲语言编写的程序 → 翻译程序 → 等价的乙语言程序

其中：甲语言称为该翻译程序的源语言，用源语言书写的程序称为源程序，
乙语言称为该翻译程序的目标语言，目标语言程序称为目标程序。

·编译方式与编译程序：

编译方式是一种分阶段进行的方式，包括翻译和运行两部分。

前一阶段：翻译

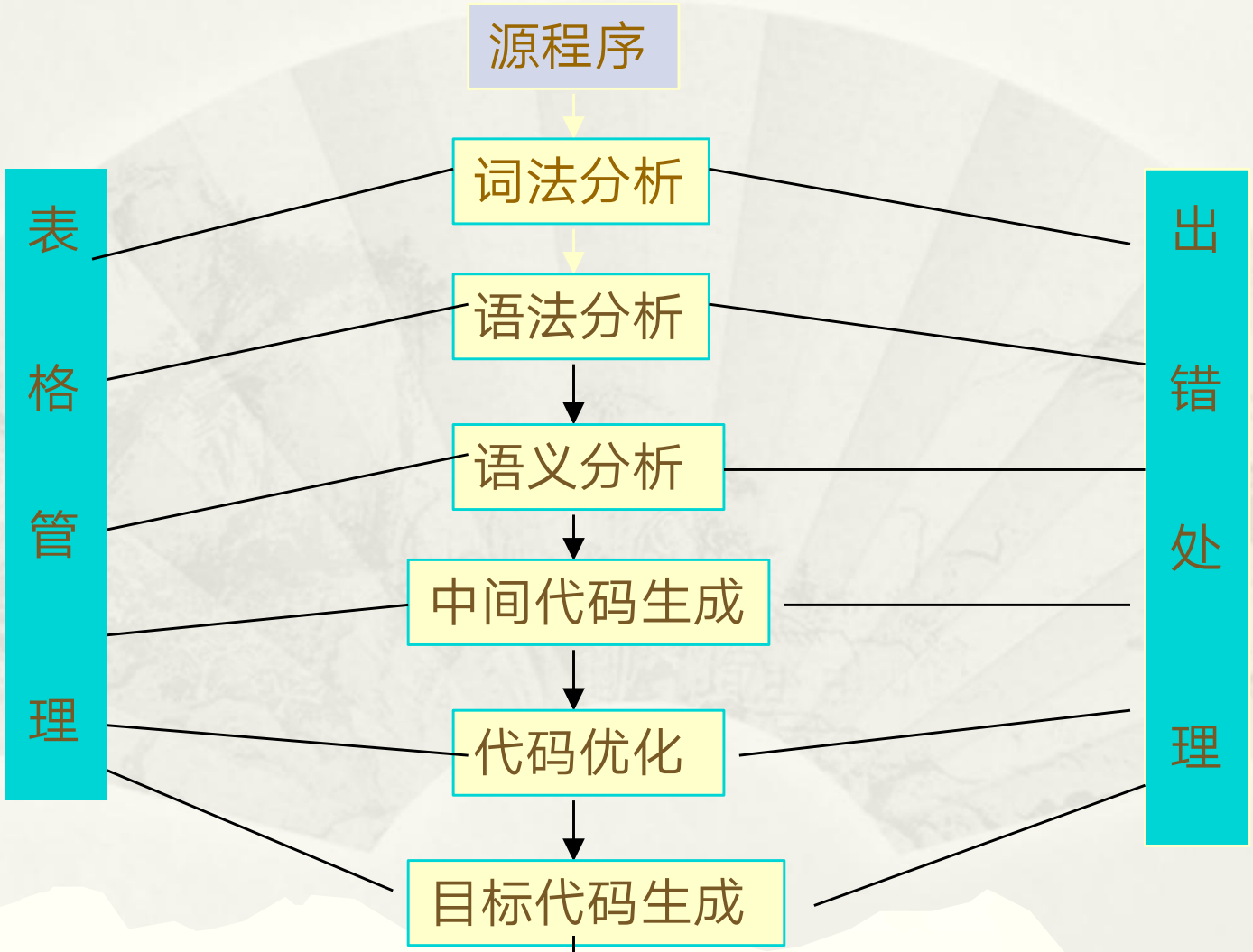
某高级语言书写的程序 → 编译程序 → 等价的某计算机的汇编语言程序或机器语言

后一阶段：运行

由运行系统配合完成。

1.2 编译过程概述

编译程序完成从源程序到目标程序的翻译工作，是一个复杂的整体的过程。从概念上来讲，一个编译程序的整个工作过程是划分阶段进行的。下图给出了一个编译过程的6阶段（8个部分），这是一种典型的划分方法。



简述其中各阶段的任务：

1、词法分析阶段

这个阶段的任务是从左到右一个字符一个字符地读入源程序，对构成源程序的字符流进行扫描和分解，从而识别出一个个单词（也称单词符号或符号TOKEN）。

例如某源程序片断如下：

begin var sum, first, count: real; sum:=first+count*10 end.

词法分析阶段将构成这段程序的字符组成了如下单词序列：

1. 保留字 begin　　2. 保留字 var
3. 标识符 sum　　4. 逗号 ，
5. 标识符 first　　6. 逗号 ，
7. 标识符 count　　8. 冒号 ：
9. 保留字 real　　10. 分号 ；
11. 标识符 sum　　12. 赋值号 ：=
13. 标识符 first　　14. 加号 +
15. 标识符 count　　16. 乘号 *
17. 整数10 10　　18. 保留字 end
19. 界符 .

2、语法分析阶段

是编译过程的第二个阶段。语法分析的任务是在词法分析的基础上将单词序列分解成各类语法短语，如“程序”，“语句”，“表达式”等等。一般这种语法短语，也称语法单位，或语法成分，或语法范畴。

语法分析所依据的是语言的语法规则，即描述程序结构的规则。通过语法分析确定整个输入串是否构成一个语法上正确的程序。

3、语义分析阶段

依据语言的语义规则，对语法分析得到的语法结构分析其含义以及应进行的运算，审查源程序中有无语义错误，为代码生成阶段收集类型信息。

4、中间代码生成

在进行了上述的语法分析和语义分析阶段的工作之后，有的编译程序将源程序转变成一种内部表示形式，这种内部表示形式叫做中间代码。

所谓“中间代码”是一种结构简单，含义明确的记号系统，这种记号系统可以设计为多种多样的形式。

重要的设计原则：一是容易生成；二是容易将它翻译成目标代码。

很多编译程序采用了一种近似“三地址指令”的“四元式”中间代码，这种四元式的形式为：（运算符，运算对象1，运算对象2，结果）。

比如：源程序 sum：= first + count * 10 可生成四元式序列如下，其中t_i(i=1,2,3)是编译程序生成的临时名字，用于存放运算结果的。

(1) (inttoreal 10 - t_1)

(2) (* id_3 t_1 t_2)

(3) (+ id_2 t_2 t_3)

(4) (: = t_3 - id_1)

5、代码优化

任务：对前阶段产生的中间代码系列进行变换或改造。目的是使生成的目标代码更高效，即省时间省空间。例如上例四个四元式可优化为下面两个四元式。

(1) (inttoreal 10 - t_1)

(2) (* id_3 t_1 t_2)

(3) (+ id_2 t_2 t_3)

(4) (: = t_3 - id_1)

}

(* id_3 10.0 t_2)

(+ id_2 t_2 id_1)

6、目标代码生成

*任务：*将中间代码变换成特定机器上的绝对指令代码或可重定位的指令代码或汇编指令代码。它的工作与硬件系统结构和指令含义有关。

例如，使用两个寄存器（R_1和R_2），可将上面两条四元式生成下面的汇编代码。

	(1) MOV	id_3,		R_2
(* id_3 10.0 t_1)	(2) MUL	#10.0,		R_2
	(3) MOV	id_2,		R_1
(+ id_2 t_1 id_1)	(4) ADD	R_1,		R_2
	(5) MOV	R_2,		id_1

下一页 上一页

*另外两个重要的工作：*表格管理和出错处理

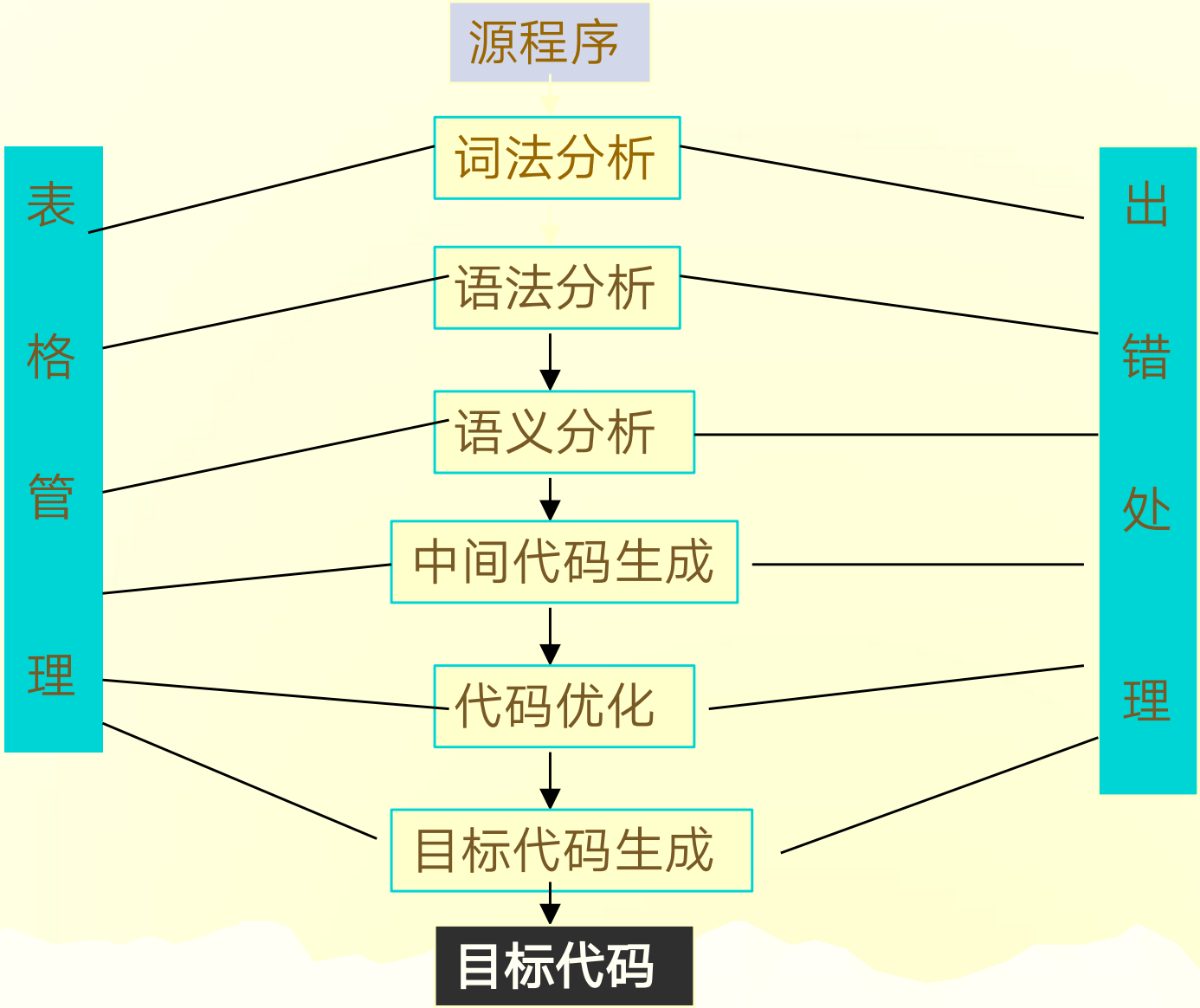
与上述六个阶段都有联系。

编译过程中源程序的各种信息被保留在种种不同的表格里，编译各阶段的工作都涉及到构造、查找或更新有关的表格，因此需要有表格管理的工作；

如果编译过程中发现源程序有错误，编译程度应报告错误的性质和错误发生的地点，并且将错误所造成的影响限制在尽可能小的范围内，使得源程序的其余部分能继续被编译下去，有些编译程序还能自动校正错误，这些工作称之为出错处理。

1.3 编译阶段的组合

1. 前端和后端



前端

其工作依赖于源程序而与目标机无关

后端

其工作依赖于目标机及中间代码而与源程序无关

若按照这种组合方式实现编译程序，有两种情况：

可以设想，某一编译程序的前端加上相应不同的后端可以为不同的机器构成同一个源语言的编译程序。

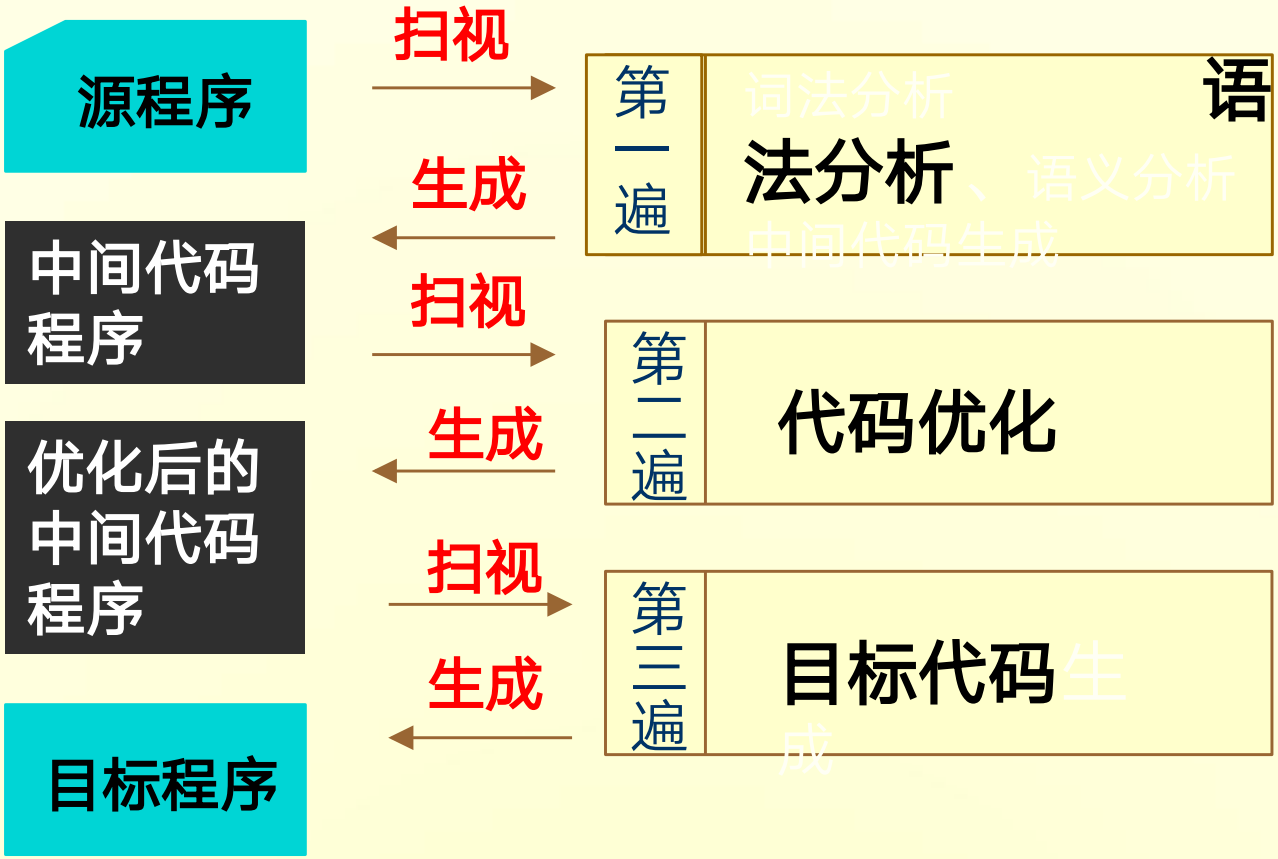
也可以设想，不同语言编译的前端生成同一种中间语言，再使用一个共同的后端，则可为同一机器生成几个语言的编译程序。

2、遍 (Pass)

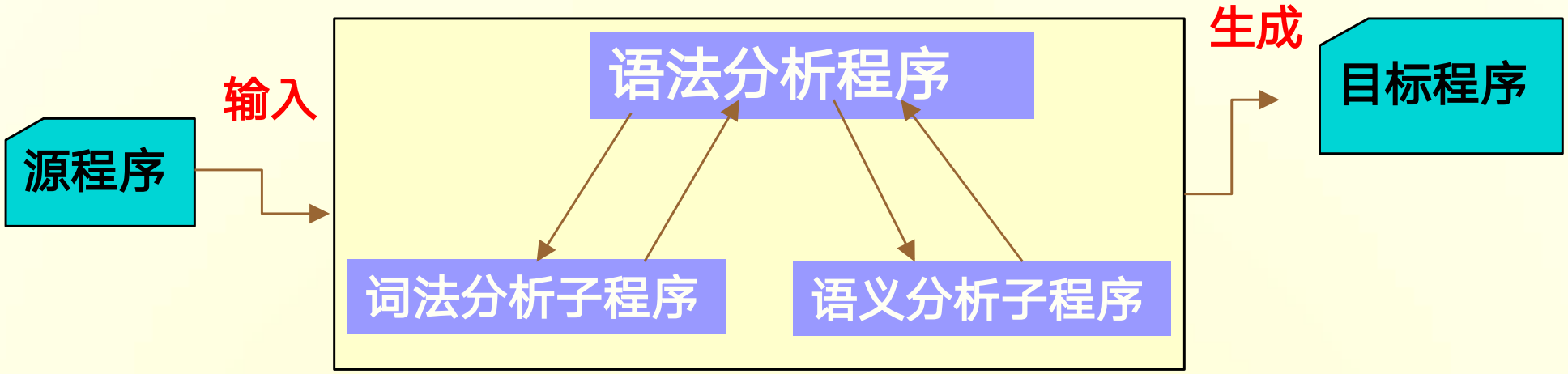
所谓遍，是对源程序或源程序的中间形式从头到尾扫视并完成规定任务的过程。

每一遍扫视可完成一个阶段或多个阶段的功能。

如： 三遍的编译程序：


源程序
扫视
第一遍
词法分析、语法分析、语义分析
中间代码生成
生成
中间代码程序
扫视
第二遍
代码优化
生成
优化后的中间代码程序
扫视
第三遍
目标代码生成
生成
目标程序

一遍的编译程序：以语法分析程序为核心 。



一个编译程序需要分成几遍，应视语言要求，机器情况而定。例如 PASCAL和C，用一遍扫描的编译程序去实现比较困难，宜于采用多遍扫描的编译程序结构。

1.4 编译技术和软件工具

1. 语言的结构化编辑器
2. 语言程序的调试工具
3. 语言程序的测试工具
4. 高级语言之间的转换工具
5. 并行编译技术

1.5 程序设计语言范型

- 从支持的计算模式来看有如下4种程序设计语言范型。

1. 强制(命令)式语言:
   也称过程式语言，如C,FORTRAN等;
2. 函数式语言：
   也称应用式语言，如ML和LISP等；
3. 基于规则的语言：
   也称逻辑程序设计语言，如PROLOG。
4. 面向对象语言：提供抽象数据类型，支持封装性、继承性和多态性。如C++和Java等。

作业：

1. 简要解释源程序、目标程序、翻译程序、汇编程序和编译程序的概念。
2. 编译过程包括哪几个阶段？各阶段的主要功能是什么？
3. 什么是编译程序的前端和后端？这样划分有何好处？
4. 什么是遍？多遍扫描方式有何优点和不足？
5. 程序设计语言有哪些范型？各有何特点？