



编译技术



邵兵
shaobing@buaa.edu.cn



课程内容

★ 课时:48学时 (1~13周, 上课教室: F117)

➤ 周一下午八九节: 15:50 ~ 17:25

➤ 周三上午三四节: 9:50 ~ 11:25

★ 分为两部分 (理论占60%, 实践占40%)

➤ 理论基础 (3学分): 课堂教学, 按时交作业。

- 平时作业, 10分
- 5次左右随堂考试, 共计30分 (开卷、不补)
- 期末闭卷考试, 60分
- 主动回答问题, 每次奖励0.5分, 5分封顶 (考前公布)

拒绝任何
不合理加
分请求!

➤ 实践部分 (1.5学分): 上机实践 (48机时, 10-17周)

- 12月底检查作业。具体由杨海燕老师统一安排。



教学要求

● 理论学习

- 理解高级程序语言的工作原理
- 学习不同语言表示的程序之间的等价转化方法
- 编译程序的构造和工作原理
- 编译相关技术和常用优化技术

● 能力培养

- 理解系统的概念，完整设计一个不同难度的编译系统



● 教材

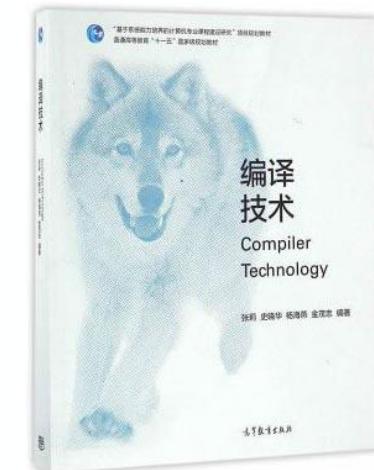
- 张莉等，《编译技术》，高等教育出版社，2016年9月。

● 参考书

- Alfred V. Aho等，《编译原理》（龙书）第2版：本科教学版，机械工业出版社，2009年5月。
- 刘春林等，《编译原理——学习指导与典型题解析》，国防工业出版社，2004年8月。

● 要求：

- 提前预习，上课认真听讲。
- 课后及时复习，认真完成作业。
- 独立完成实验内容。





助教 (TA) : 2人

姓名	学号	手机	邮箱
张彧铮	20376305	13653669621	20376305@buaa.edu.cn
杨振炜	20377408	15807550266	20377408@buaa.edu.cn



第一章 概论

(介绍名词术语、了解编译系统的结构和编译过程)

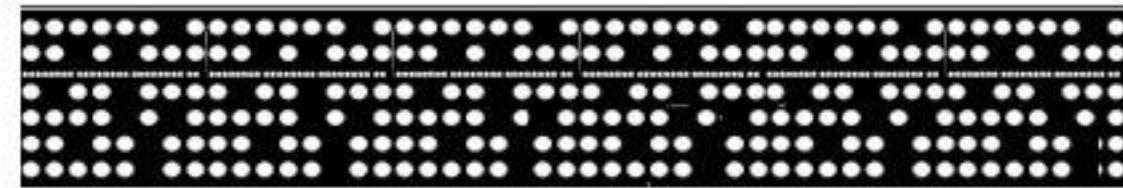


内 容

- 1.1. 编译的起源：程序设计语言的发展**
- 1.2. 基本概念**
- 1.3. 编译过程和编译程序构造**
- 1.4. 编译技术的其它应用**



1.1 基本概念



- 低级语言 (Low level language)
 - 字位码、机器语言、汇编语言
 - 特点：与特定的机器有关，功效高，但使用复杂、繁琐、费时、易出错。
- 高级语言
 - C、Fortran、Pascal、Java语言等
 - 特点：不依赖具体机器，移植性好，对用户要求低，易使用，易维护等。

```
8 #include <stdio.h>
9
10 int main(int argc, const char * argv[]) {
11     int iResult=0;
12     int i;
13     printf("由1加到100的结果是: ");
14
15     for (i=1;i<100;i++)
16     {
17         iResult=i+iResult;
18     }
19
20     printf("%d\n",iResult);
21
22     return 0;
23 }
24
25 }
```



- 源程序

用汇编语言或高级语言编写的程序称为源程序。

- 目标程序

用目标语言所表示的程序。

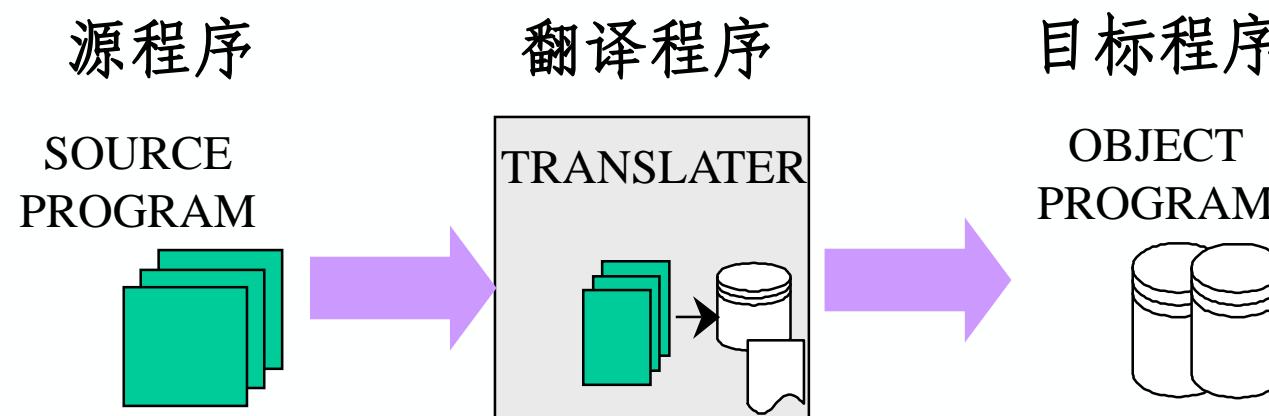
目标语言：可以是介于源语言和机器语言之间的“中间语言”，可以是某种机器的机器语言，也可以是某种机器的汇编语言。

- 翻译程序

将源程序转换为目标程序的程序称为翻译程序。它是指各种语言的翻译器，包括汇编程序和编译程序，是汇编程序、编译程序以及各种变换程序的总称。



源程序、翻译程序、目标程序 三者关系：



即源程序是翻译程序的输入， 目标程序是翻译程序的输出。



• 汇编程序

若源程序用汇编语言书写，经过翻译程序得到用机器语言表示的程序，这时的翻译程序就称之为汇编程序。这种翻译过程称为“汇编”（Assemble）。

• 编译程序

若源程序是用高级语言书写，经加工后得到目标程序，上述翻译过程称“编译”（Compile）。

汇编程序与编译程序都是翻译程序，主要区别是加工对象的不同。由于汇编语言格式简单，常与机器语言之间有一一对应的关系，所以汇编程序所要做的翻译工作比编译程序简单得多。

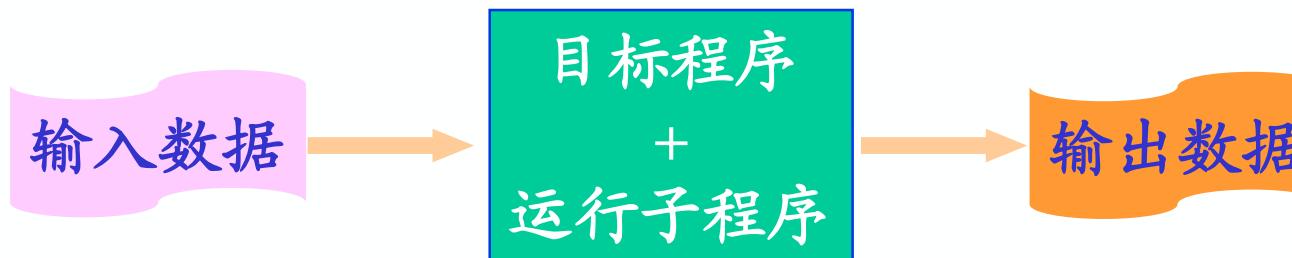


源程序的编译和运行

- 编译或汇编阶段



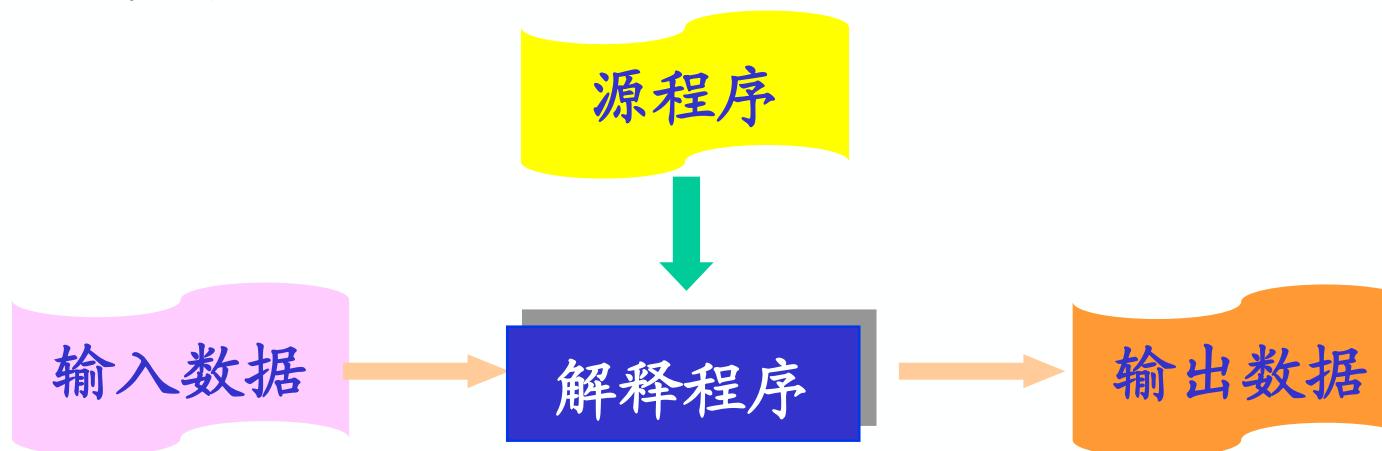
- 运行阶段





- 解释程序（Interpreter）
对源程序进行解释执行的程序。

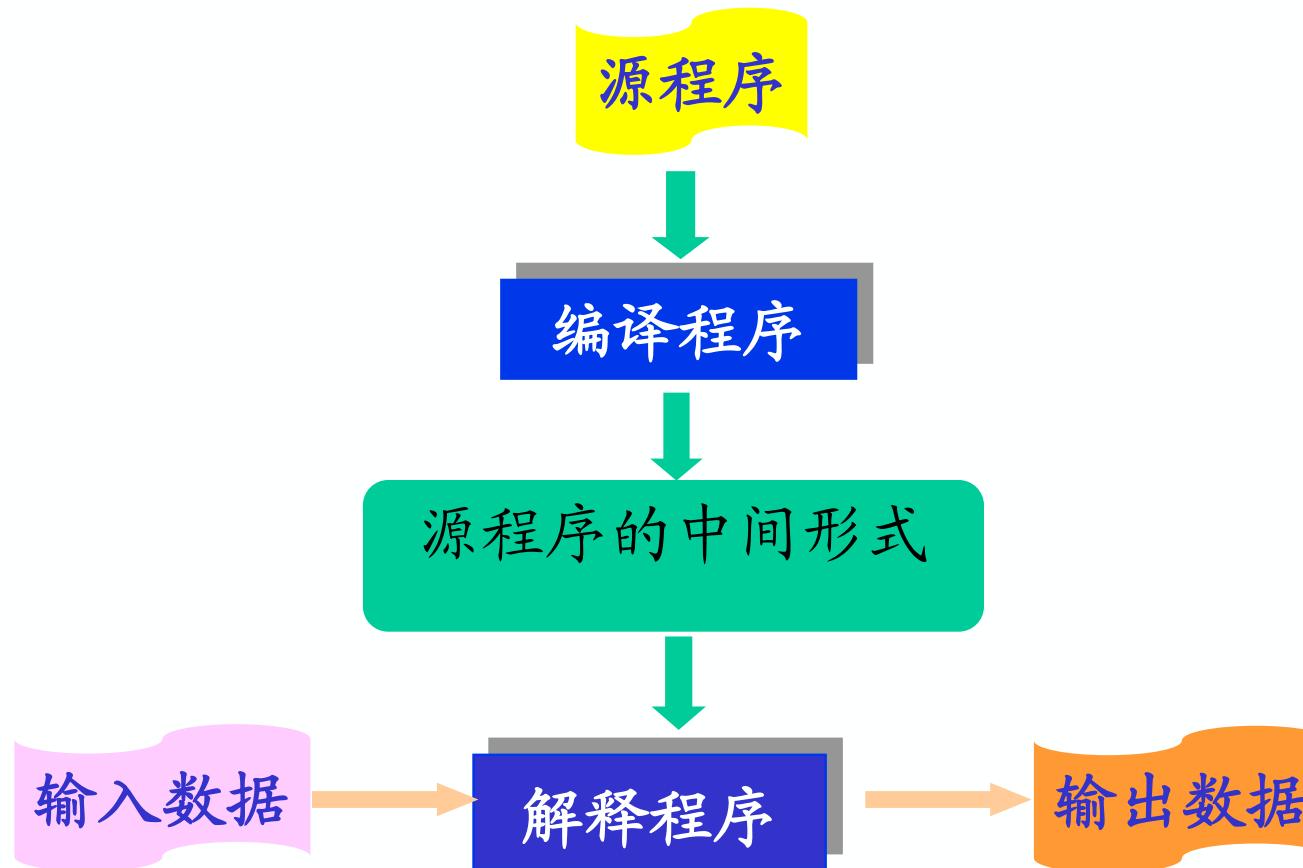
- 工作过程



- 特点、与编译程序比较



“编译-解释执行”系统





1.2 编译过程



所谓编译过程是指将高级语言程序翻译为等价的目标程序的过程。

习惯上是将编译过程划分为5个基本阶段：





一、词法分析



任务：分析和识别单词。

源程序是由字符序列构成的，词法分析扫描源程序（字符串），根据语言的词法规则分析并识别单词，并以某种编码形式输出。

- 单词：是语言的基本语法单位，一般语言有四大类单词
 - <1>**语言定义的关键字或保留字**（如BEGIN、END、IF）
 - <2>**标识符**
 - <3>**常数**
 - <4>**分界符**（运算符）（如+、-、*、/、；、（、）.....）



一、词法分析

任务：分析和识别单词。

源程序是由字符序列构成的，词法分析扫描源程序（字符串），根据语言的词法规则分析并识别单词，并以某种编码形式输出。

对于如下的字符串,词法分析程序将分析和识别出9个单词:

$\frac{X1}{1} \coloneqq \frac{(}{2} \frac{2.0}{3} \frac{+}{4} \frac{0.8}{5} \frac{)}{6} \frac{*}{7} \frac{\text{Constant1}}{8} \frac{}{9}$



二、语法分析

任务：根据语法规则（即语言的文法），分析并识别出各种语法成分，如表达式、各种说明、各种语句、过程、函数、程序等，并进行语法正确性检查。

例如，对于前面提到的例子 **X1 := (2.0 + 0.8) * Constant1** 我们可以根据语言赋值语句的文法来分析和识别该语句（单词串）。首先给定文法：

<赋值语句> → <变量><赋值操作符><表达式>

<变量> → <简单标识符>

<赋值操作符> → :=

<表达式> →

语法分析根据文法，将<变量>、<赋值操作符>、<表达式>识别出来，进而将赋值语句识别出来。在识别过程中进行语法检查；若有错误，则应输出出错信息。



三、语义分析、生成中间代码



任务：对识别出的各种语法成分进行语义分析，并产生相应的中间代码。

- 中间代码：一种介于源语言和目标语言之间的中间语言形式。
- 生成中间代码的目的：
 - <1> 便于做优化处理；
 - <2> 便于编译程序的移植（中间代码不依赖于目标计算机）。
- 中间代码的形式：编译程序设计者可以自己设计，常用的有四元式、三元式、逆波兰表示、LLVM IR等。



例： $X1 := (2.0 + 0.8) * C1$

由语法分析识别出为赋值语句，语义分析首先要分析语义上的正确性，例如要检查表达式中和赋值号两边的类型是否一致。

根据赋值语句的语义，生成中间代码。即用一种语言形式来代替另一种语言形式，这是翻译的关键步骤。（翻译的实质：语义的等价性）

下面介绍一种常用的中间代码来替换上述的赋值语句。



★ 四元式 (三地址指令)

对于前面提到的例子 $X1 := (2.0 + 0.8) * C1$

运算符	左运算对象	右运算对象	结果
(1) +	2.0	0.8	T1
(2) *	T1	C1	T2
(3) :=	X1	T2	

其中T1和T2为编译程序引入的工作单元。

四元式的语义为: $2.0 + 0.8 \rightarrow T1$

$T1 * C1 \rightarrow T2$

$T2 \rightarrow X1$

这样所生成的四元式与原来的赋值语句在语言的形式上不同, 但语义上等价。



四、代码优化

任务：目的是为了得到高质量的目标程序。

例如：上面的四元式中第一个四元式是计算常量表达式值，该值在编译时就可以算出并存放在工作单元中，不必生成目标指令来计算，这样四元式可优化为：

编译时： $2.0 + 0.8 \rightarrow T1$

运算符 左运算对象 右运算对象 结果

* T1 C1 T2

:= X1 T2



五、生成目标程序

由中间代码很容易生成目标程序（地址指令序列）。这部分工作与机器关系密切，所以要根据具体机器进行。在做这部分工作时（要注意充分利用累加器），也可以进行优化处理。

注意：在翻译成目标程序的过程中，要切记保持语义的等价性。

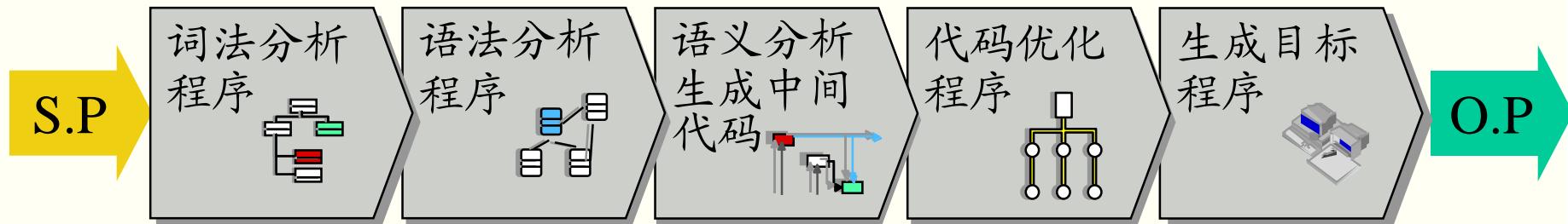


1.3 编译程序构造



1.3.1 编译程序的逻辑结构

按逻辑功能不同，可将编译过程划分为五个基本阶段。与此相对应，我们将实现整个编译过程的编译程序划分为五个逻辑阶段（即五个逻辑子过程）。





在上列五个阶段中都要做两件事：

- (1) 建表和查表 (2) 出错处理

所以编译程序中都要包括表格管理和出错处理两部分。

★ 表格管理 (符号表组织)

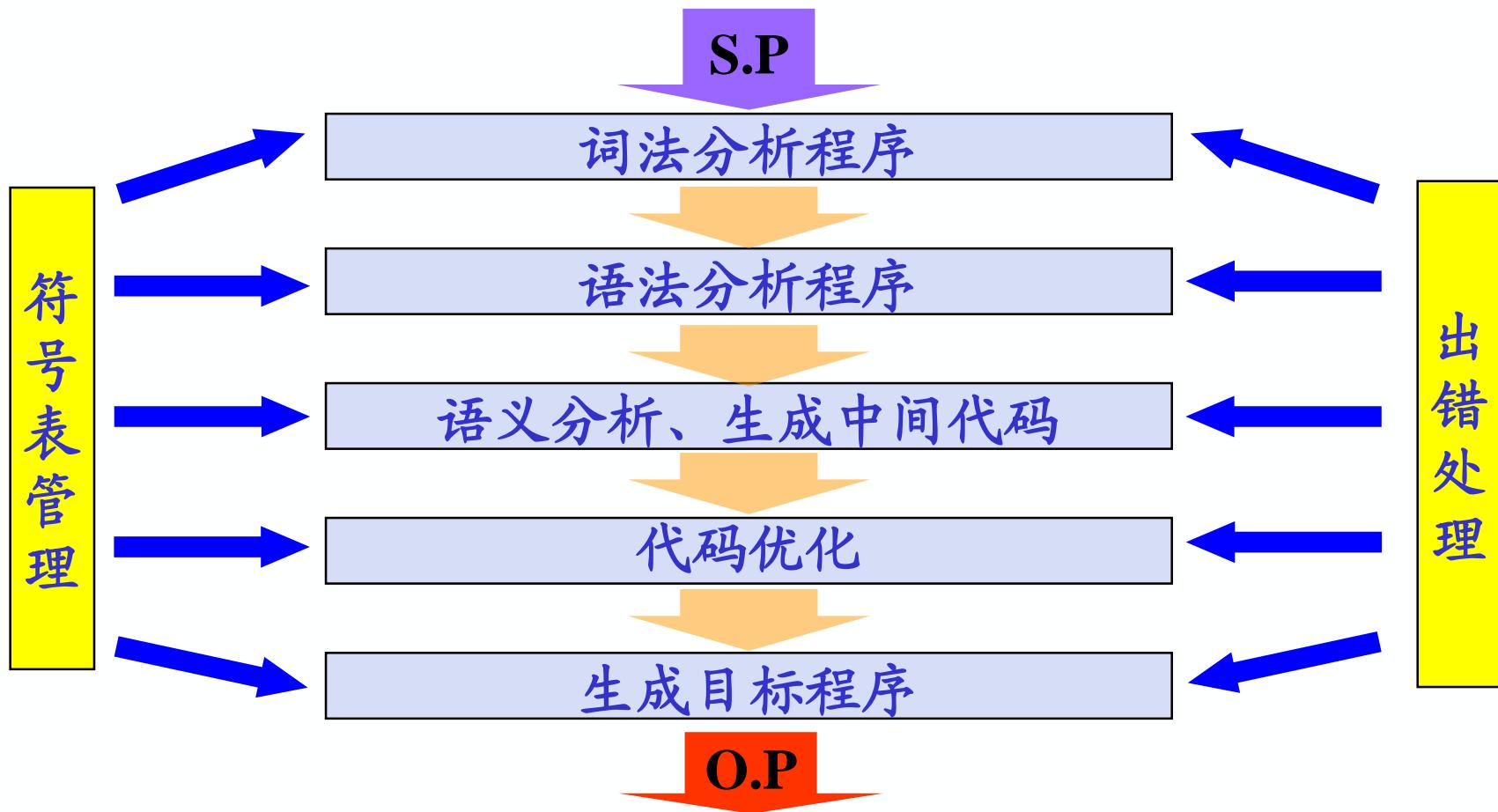
在整个编译过程中始终都要贯穿着建表（填表）和查表的工作。即要及时地把源程序中的信息和编译过程中所产生的信息登记在表格中，而在随后的编译过程中同时又要不断地查找这些表格中的信息。

★ 出错处理

规模较大的源程序难免有多种错误。编译程序必须要有出错处理的功能，即能诊察出错误，并向用户报告错误性质和位置，以便用户修改源程序。出错处理能力的优劣是衡量编译程序质量好坏的一个重要指标。



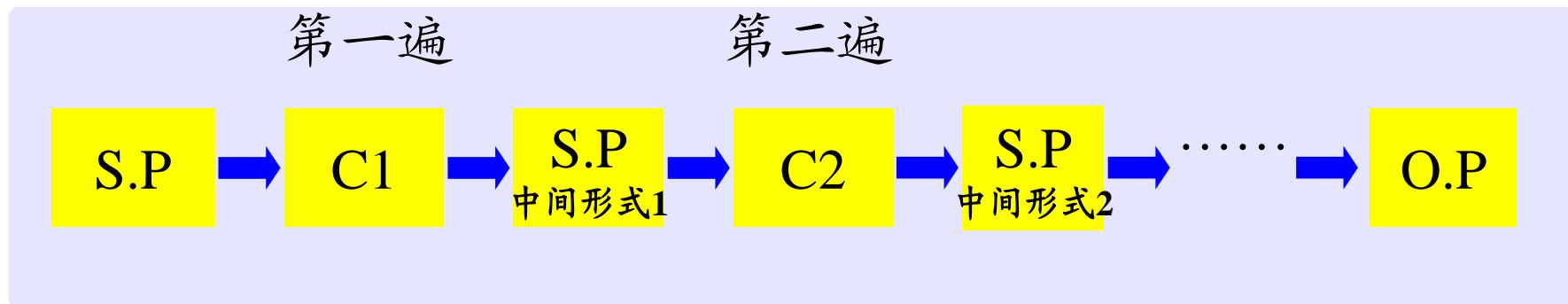
典型的编译程序具有7个逻辑部分





1.3.2 遍 (PASS)

遍：对源程序（包括源程序中间形式）从头到尾扫描一次，并做有关的加工处理，生成新的源程序中间形式或目标程序，通常称之为一遍。



★要注意遍与基本阶段的区别

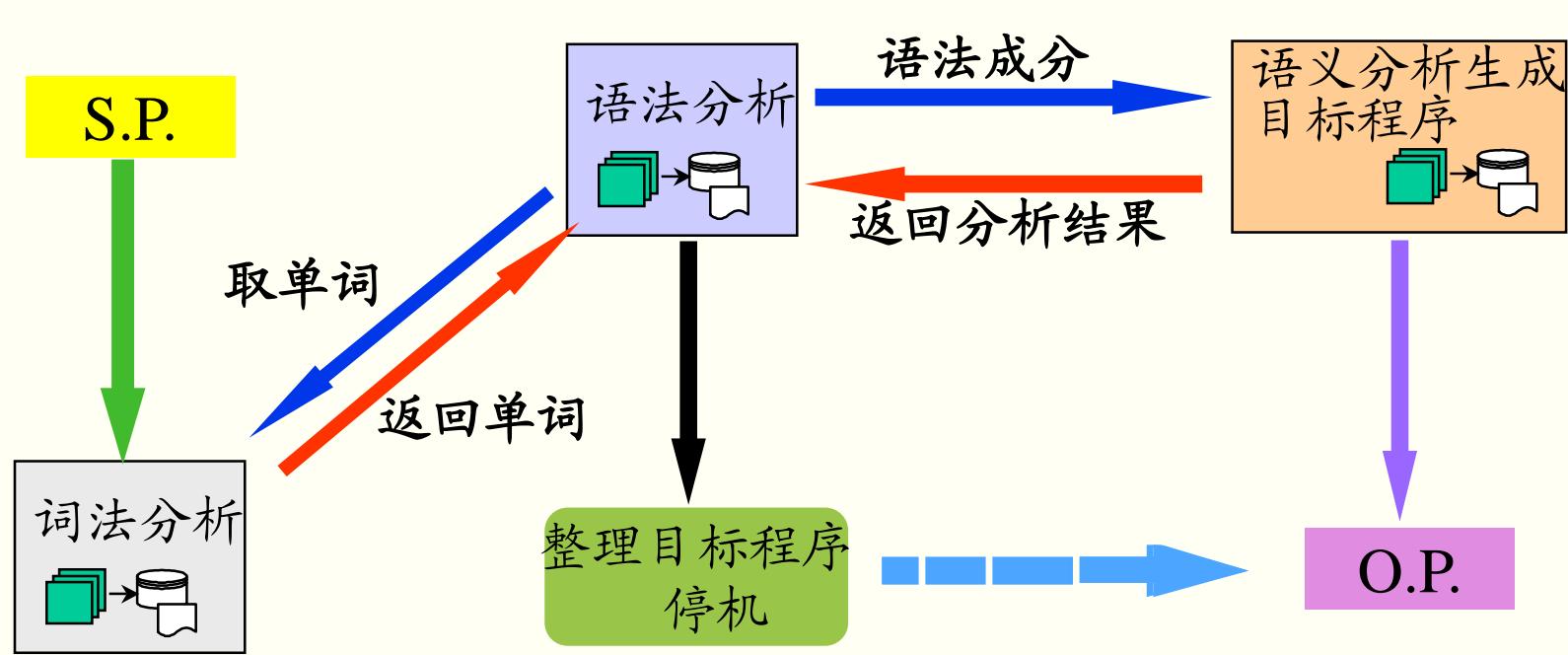
五个基本阶段：是将源程序翻译为目标程序在逻辑上要完成的工作。

遍：是指完成上述5个基本阶段的工作，要经过几次扫描处理。



一遍扫描即可完成整个编译工作的称为一遍扫描编译程序。

其结构为：





1.3.3 前端和后端

根据编译程序各部分功能，将编译程序分成前端和后端。

前端：通常将与源程序有关的编译部分称为前端。

词法分析、语法分析、语义分析、中间代码生成、
代码优化

————分析部分

特点：与源语言有关

后端：与目标机有关的部分称为后端。

目标程序生成（与目标机有关的优化）

————综合部分

特点：与目标机有关

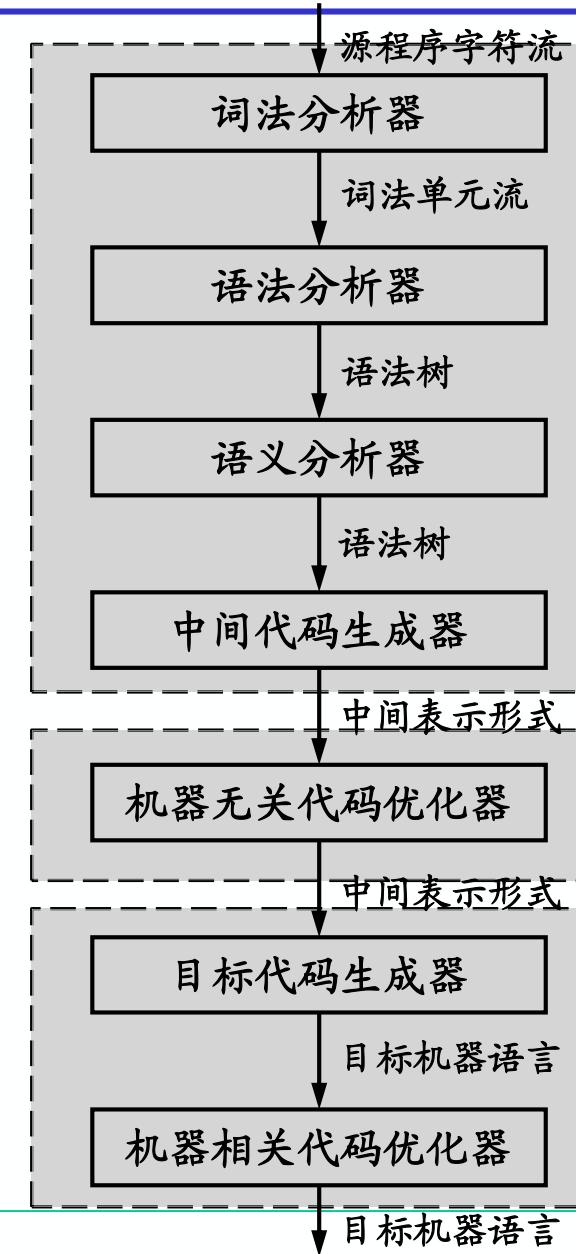


1.3.3 前端和后端

前端

中端

后端

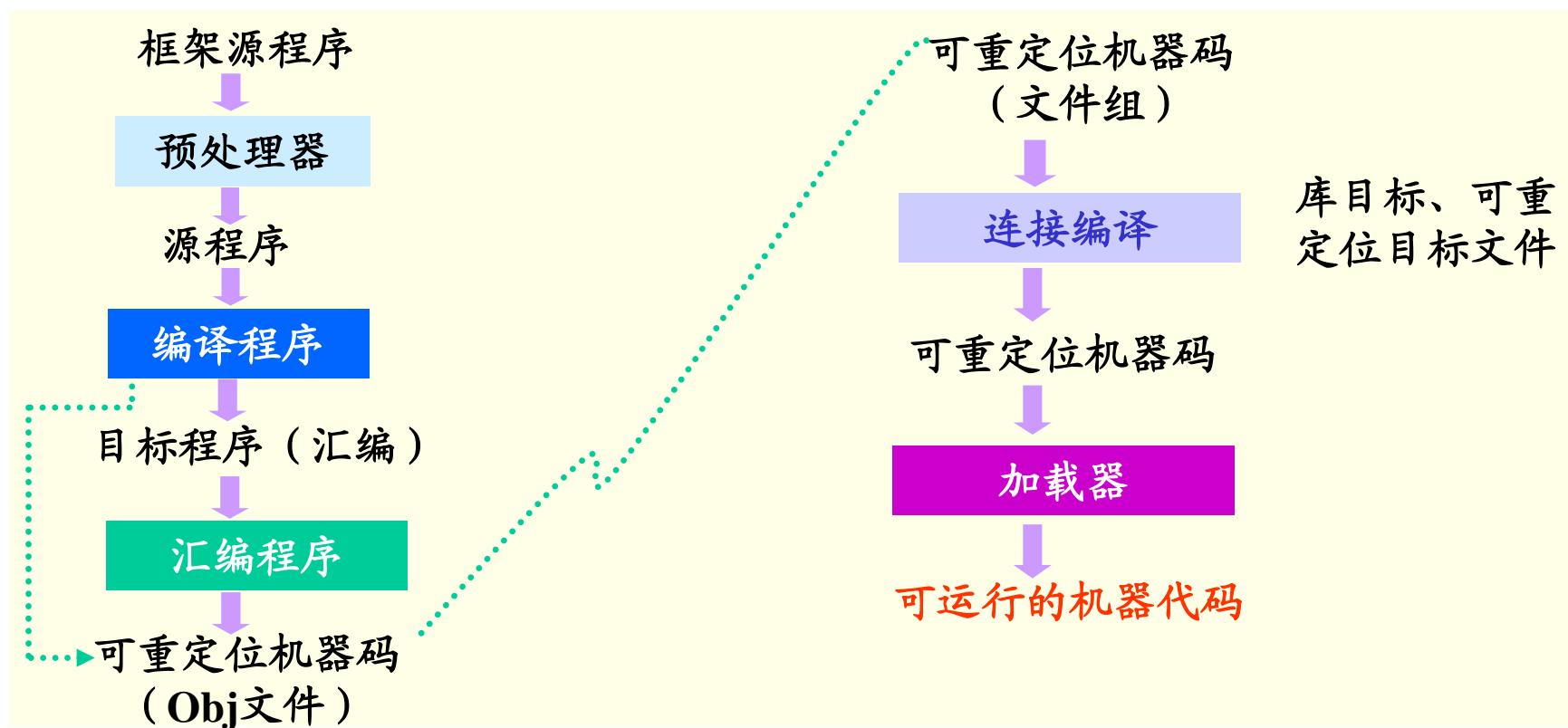




1.4 编译程序的前后处理器

源程序：多文件、宏定义和宏调用，包含文件

目标程序：一般为汇编程序或可重定位的机器代码





1.5 编译技术的应用

- 语法制导的结构化编译器
- 程序格式化工具
- 软件测试工具
- 程序理解工具
- 高级语言的翻译工具
- 等等。

作业：第15页1、2、3、4、5题