



《编译原理速成课》

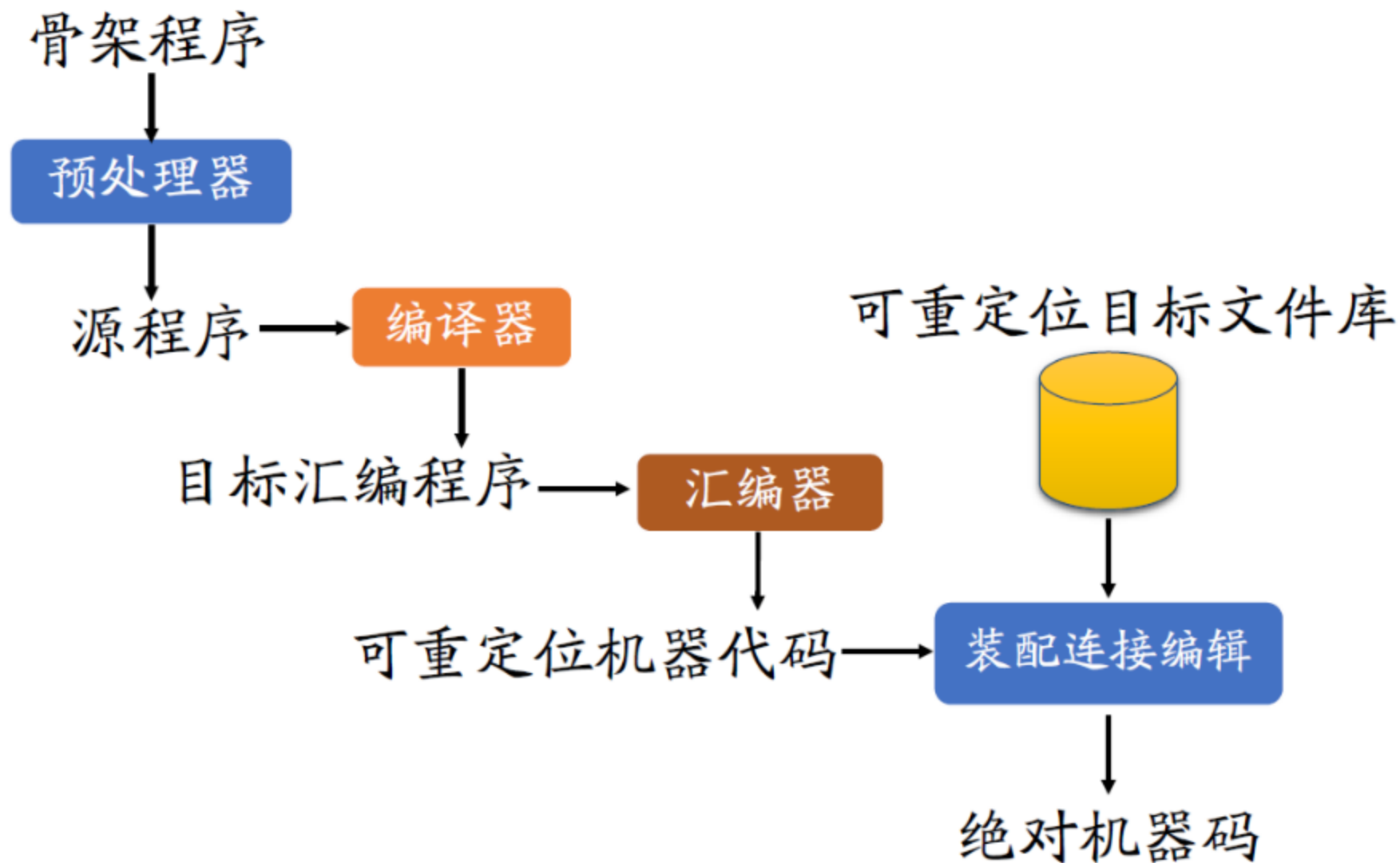
期末速成课



考点	重要程度	占分	题型
1. 编译程序	★★★	3~5	简答题、填空题
2. 编译过程	★★★★★	3~5	简答题、填空题
3. 编译程序结构	★★★★★	3~5	简答题

1.1 语言处理过程

一、语言处理过程



二、什么是编译

翻译程序

把某一种语言程序(称为源语言程序)等价地转换成另一种语言程序(称为目标语言程序)的程序。

编译程序(compiler)

把某一种高级语言程序等价地转换成另一种低级语言程序(如汇编语言或机器语言程序)的程序。(如C语言、PASCAL语言等是编译程序)

解释程序

把源语言写的源程序作为输入，但不产生目标程序，而是边解释边执行源程序本身。
(如BASIC语言是解释程序)

三、编译程序 vs. 解释程序



区别：是否产生目标代码



1.2 编译过程

编译程序的工作一般分为五个阶段:

词法分析

语法分析

语义分析和中间代码产生

代码优化

目标代码产生

一、词法分析

任务: 输入源程序，对构成源程序的字符串进行扫描和分解，识别出一个个单词符号。

依循的原则: 构词规则

描述工具: 正规式和有限自动机

二、语法分析

任务: 在词法分析的基础上，根据语言的语法规则把单词符号串分解成各类语法单位。

依循的原则: 语法规则

描述工具: 上下文无关文法

三、语义分析和中间代码生成

任务:对各类不同语法范畴（语句、过程、表达式、函数等）按语言的语义进行初步翻译。

依循的原则: 语义规则

中间代码:三元式，四元式，逆波兰式、树形结构等

四、代码优化

任务: 对于前阶段产生的**中间代码**进行加工变换，以期在最后阶段产生更高效的目标代码。主要包括：提取公共子表达式、合并已知量、删除无用语句、循环优化等。

依循的原则: 程序的等价变换规则

五、目标代码的生成

任务: 把中间代码变换成特定机器上的目标代码。

依赖于硬件系统结构和机器指令的含义。

目标代码三种形式:

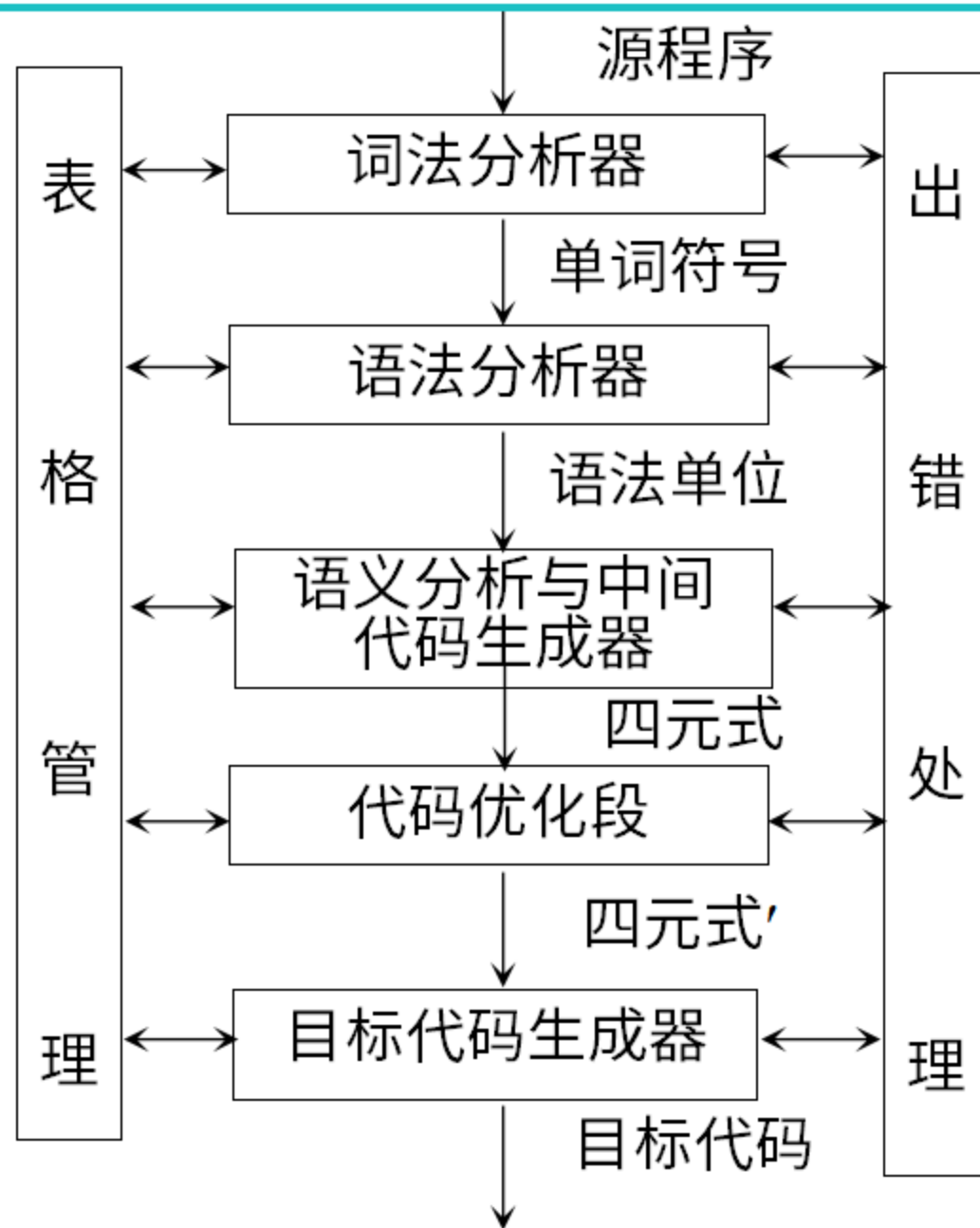
- 1.绝对指令代码: 可直接运行
- 2.可重新定位指令代码: 需要连接装配
- 3.汇编指令代码: 需要进行汇编



视频讲解更清晰

1.3 编译程序结构

一、编译程序总体框架



二、表格管理

常见的表格：符号名表，常数表，标号表，入口名表，过程引用表。

格式：

名字	信息
----	----

三、出错处理

出错处理程序：发现源程序中的错误，把有关错误信息报告给用户

语法错误

语义错误

注：编译程序一般很难检测出逻辑错误！

【题1】 PASCAL程序段

```
PROCEDURE INCWAP(M, N:INTEGER);  
  LABEL START;  
  VAR  
    K:INTEGER;  
  BEGIN  
    START:  
    K:=M+1;  
    M:=N+4;  
    N:=K;  
  END.
```

表 1 符号名表 SNT

NAME	INFORMATION
M	形式参数, 整型, 值参数
N	形式参数, 整型, 值参数
K	整型, 变量

表 2 常数表 CT

	值 (VALUE)
(1)	1
(2)	4

表 3 入口名表 ENT

	NAME	INFORMATION
(1)	INCWAP	二目子程序, 入口 四元式: 1

表 4 标号表 LT

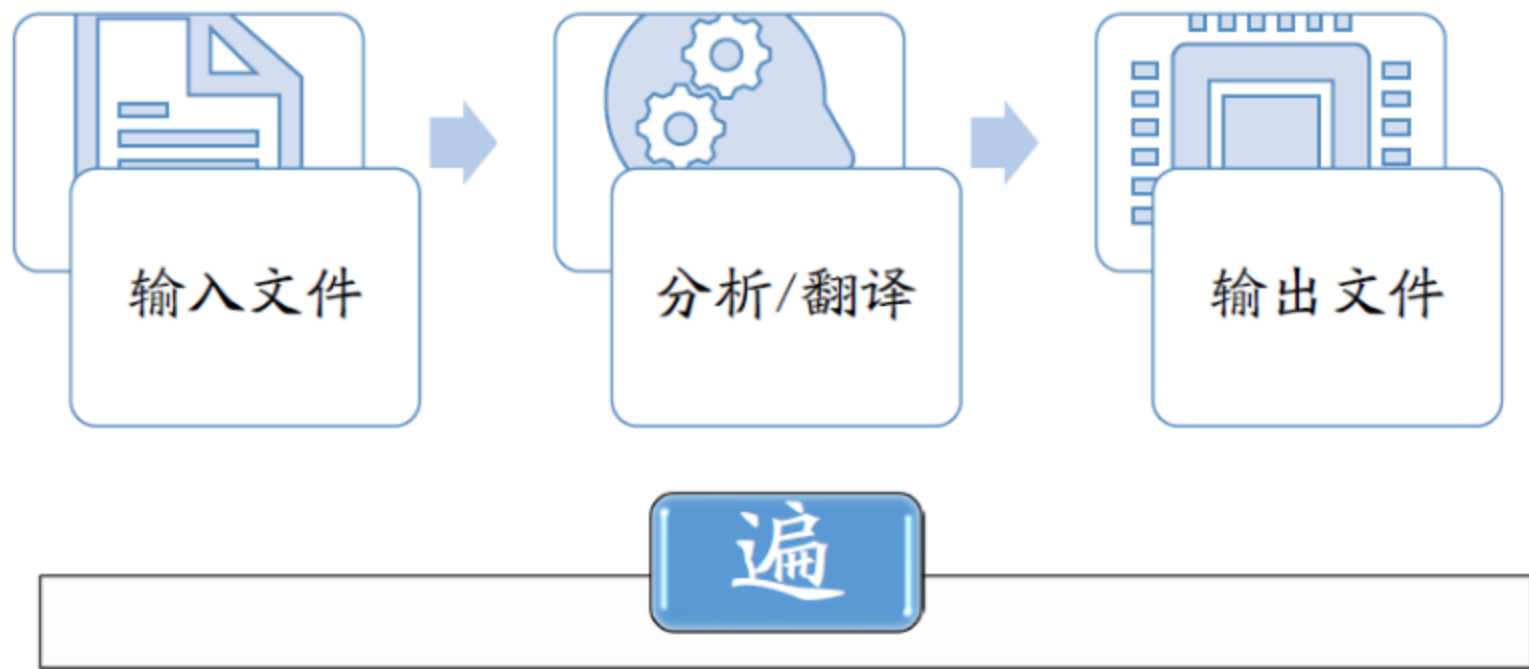
	NAME	INFORMATION
(1)	START	四元式: (4)

注: 词法分析时, 先建好标号a表, 等到中间代码生成时, 再将相关信息填入。
(维护管理)

四、遍 (pass)

对输入文件(源程序或其等价的中间形式)从头到尾扫视，完成预定的处理。

阶段与遍是不同的概念。一遍可以由若干段组成，一个阶段也可以分若干遍来完成。



五、编译前端与后端

编译前端：与**源语言**有关，如词法分析，语法分析，语义分析与中间代码产生，
与机器无关的优化

编译后端：与**目标机**有关，与目标机有关的优化，目标代码产生



优点：减少对内存容量的要求，程序逻辑结构清晰；优化更充分，有利于移植。

不足：编译程序运行的效率低。