

编译技术

课程简介

- 什么是编译程序？
- 什么样的角色进行学习？
- 编译程序怎么设计和实现？
- 为什么学习编译原理和技术？
- 怎么学习编译原理和技术？

什么是编译程序

高级程序

```
int b=3;
int main()
{
    int a=2,sum;
    sum=a+b;
    return 0;
}
```

汇编程序

```
...
pushq    %rbp
movq     %rsp, %rbp
movl     $2, -8(%rbp)
movl     b(%rip), %edx
movl     -8(%rbp), %eax
addl     %edx, %eax
movl     %eax, -4(%rbp)
movl     $0, %eax
popq     %rbp
ret
...
```

编译程序

机器程序

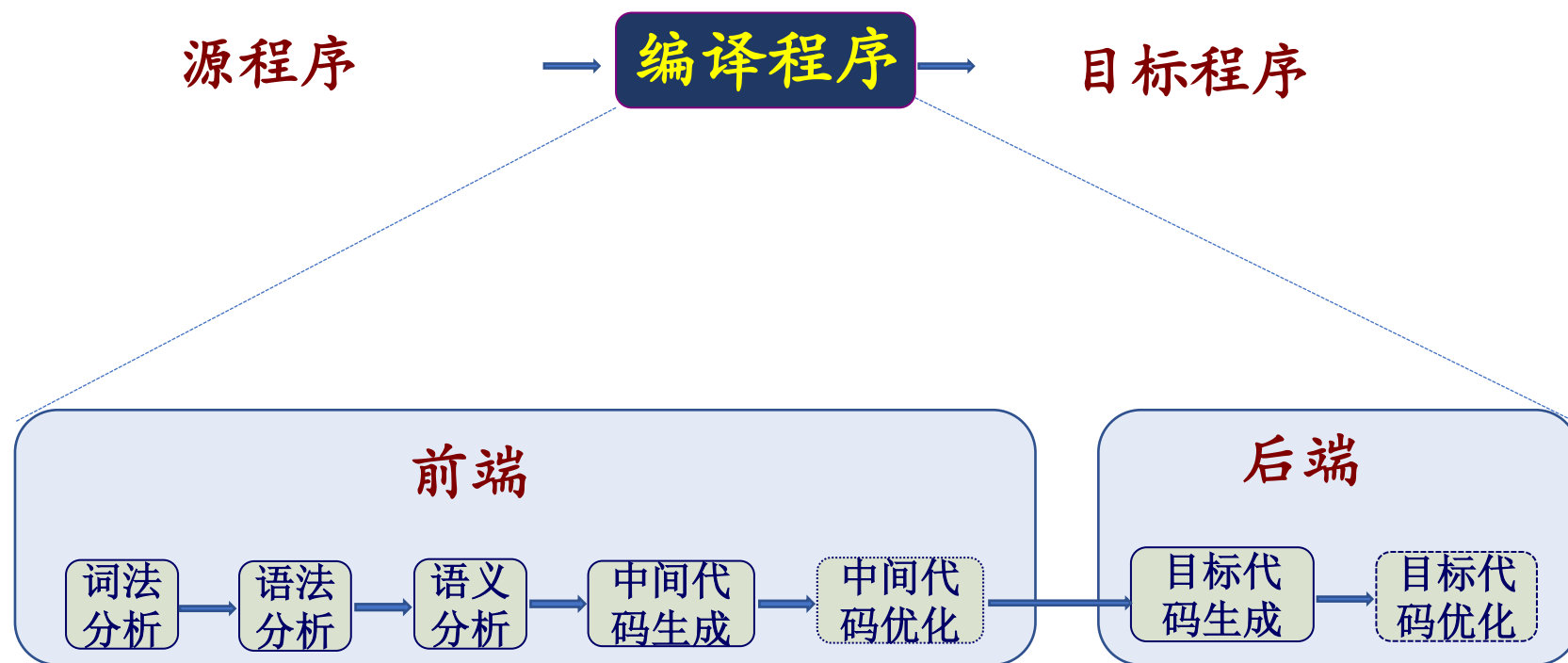
```
...
0000030 0000 0000 0040 0000 0000 0040 000c 0009
0000040 4855 e589 45c7 02f8 0000 8b00 0015 0000
0000050 8b00 f845 d001 4589 b8fc 0000 0000 c35d
0000060 0003 0000 4700 4343 203a 5528 7562 746e
0000070 2075 2e35 2e34 2d30 7536 7562 746e 3175
0000080 317e 2e36 3430 312e 2932 3520 342e 302e
0000090 3220 3130 3036 3036 0039 0000 0000 0000
...
```

1101000000000001

✓ 编译程序

将一种语言表示的源程序**翻译**成另一种语言表示的目标程序

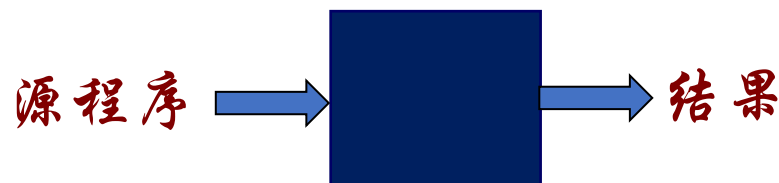
编译程序怎么设计和实现



为什么学习

➤ 更好地理解高级语言及其执行

- ✧ 打开从源程序到目标程序转换的黑箱
- ✧ 提高程序分析能力



➤ 解决复杂工程问题的基本能力

- ✧ 问题分析：抽象和表示
- ✧ 问题求解：分解、递归、启发式推理

编译原理修炼的内功

➤ 运用编译原理和方法构造实用工具

- ✧ 浏览器：HTML文件显示
- ✧ 文本语言检查

In this paper, we devise a
neural network model.
DeepIEP is an model to
integrate three datas.
←

为什么学习

编译学习的是一个方法学，而
不是一个纯粹的技术和原理

怎么学习

使用者 → 开发设计者 → 运用者

课程特点

➤ 基础和综合的课程

- ✧ 涉及整个计算机软硬件体系结构
- ✧ 融合计算机大部分课程的知识

➤ 解决的是一个复杂的工程问题

- ✧ 最难以挑战学习能力的课程之一

➤ 理论和实践完美结合的典范

- ✧ 经典的理论
- ✧ 典型的应用

课程特点

➤ 先修课程

- ✧ 程序设计语言
- ✧ 数据结构

➤ 参考书

- ✧ 《编译原理》第三版,王生原等编,清华大学出版社,2015年
- ✧ Alfred V. Aho / Monica S. Lam / Ravi Sethi / Jeffrey D. Ullman, Compilers, Principles, Techniques & Tools , Second Edition,(赵建华译,编译原理第二版,机械工业出版社)

➤ 线上资源

- ✧ 国防科技大学建设的慕课资源

课程目标

➤ 知识目标

- ✧ 掌握文法和程序语言的形式化方法；
- ✧ 理解编译器的结构和各阶段的功能，掌握编译各阶段设计方法和技术；
- ✧ 理解程序在内存中的运行结构

➤ 能力目标

- ✧ 能够运用文法对程序语言进行抽象、表示和推导；
- ✧ 能够运用分解、迭代、启发式推理等；
- ✧ 具备编译系统分析、设计和实现初步能力；

➤ 素质目标

- ✧ 理论联系实践的行为方式，IT报国的情怀和理想
- ✧ 严谨求实和脚踏实地的工匠精神，团队合作精神和创新意识

名人轶事

➤ 2020年图灵奖

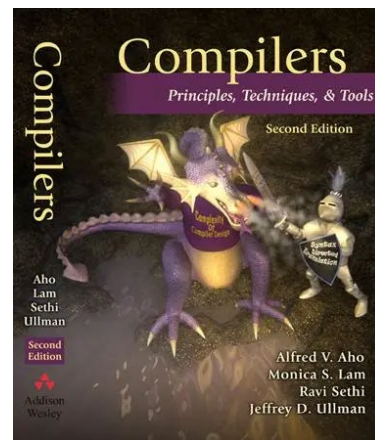
✧ Jeffrey David Ullman

✧ Alfred Vaino Aho



➤ 贡献领域/获奖理由

✧ 创造了全球数百万编程人员使用的工具和教材，推进编程语言实现的基础算法和理论，并在极具影响力的书籍中综述了这些研究成果



第一讲 引论

➤ 学习的目标

- ✧ 了解编译概念、过程和结构

➤ 讨论的主要内容

- ✧ 编译的相关概念;
- ✧ 编译的过程和结构;
- ✧ 编译程序的组织;
- ✧ 编译程序的伙伴程序

计算机理解高级语言的方式

➤ 编译

- ✧ 将A语言程序翻译为B语言程序的一种翻译程序(Translator)
- ✧ 称A语言为源语言
- ✧ 称B语言为目标语言

源程序 → 编译程序 → 目标程序

输入 → 目标程序 → 输出

计算机理解高级语言的方式

➤ 解释

- ✧ 不产生目标程序
- ✧ 不区别翻译阶段和执行阶段
- ✧ 程序执行期间一直有解释程序守候
- ✧ 常用于实现虚拟机



计算机理解高级语言的方式

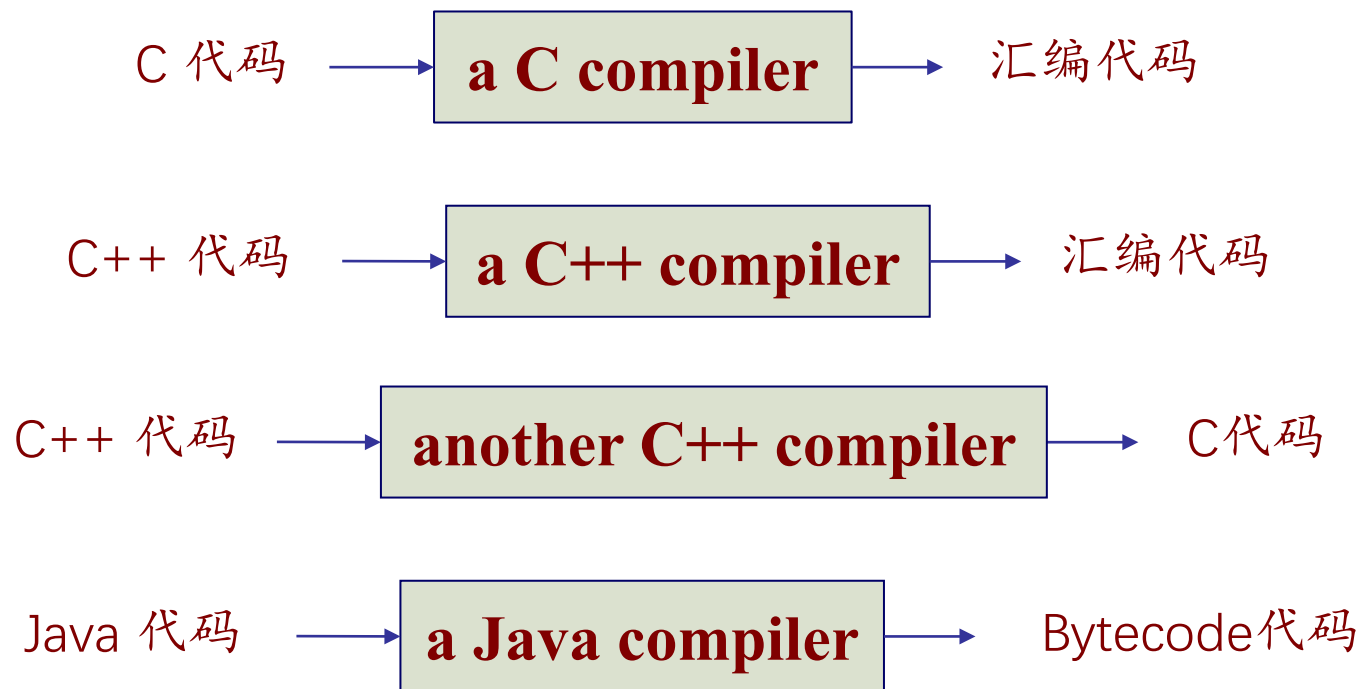
➤ 比较编译程序和解释程序

- ✧ 编译产生的目标程序执行通常比解释器快；
- ✧ 解释器的错误诊断效果通常比编译器好；

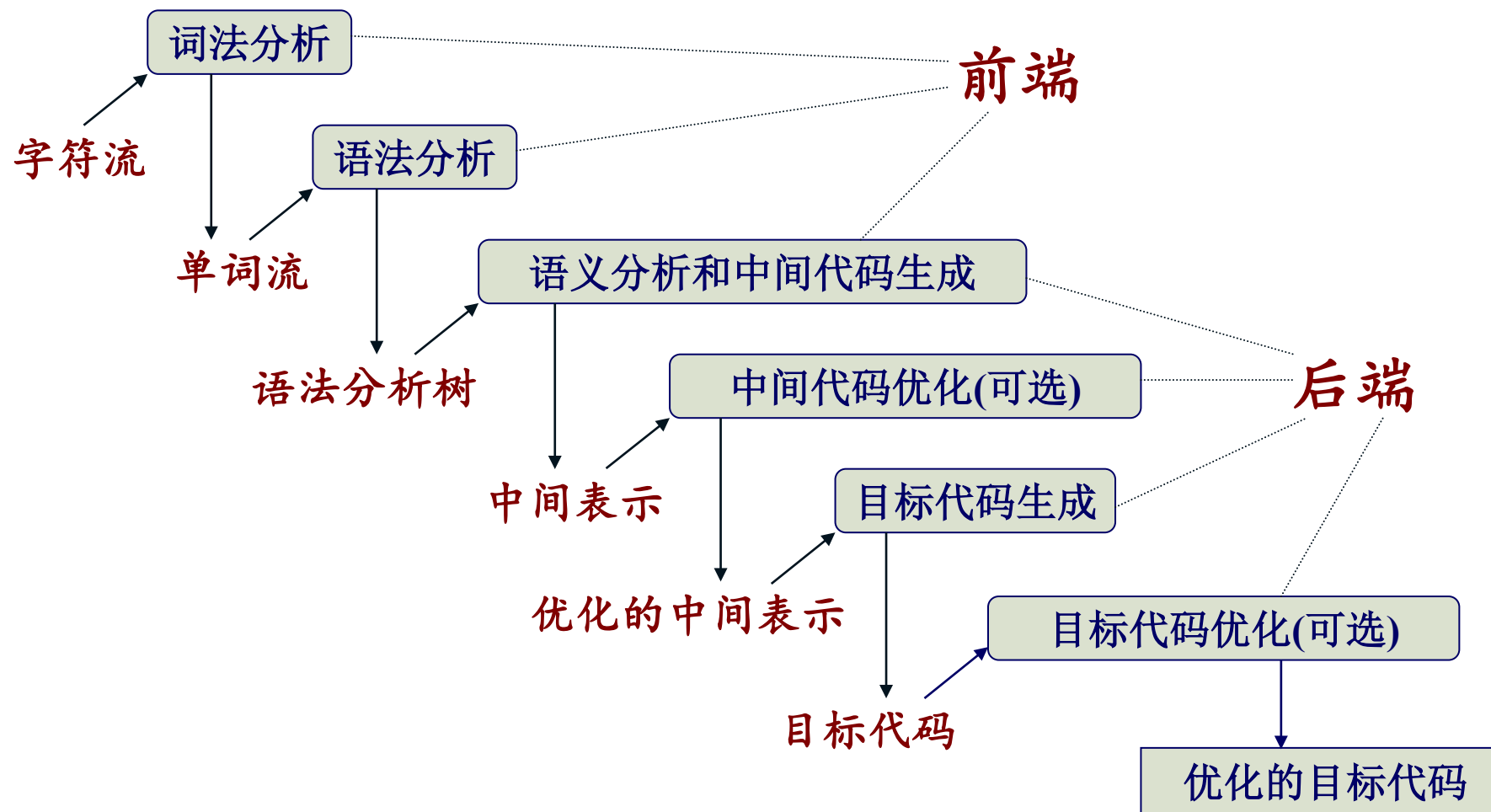
典型的编译器

➤ 翻译

✧ 编译程序通常从较高级语言的程序翻译至较低级语言的程序



典型编译程序的逻辑过程



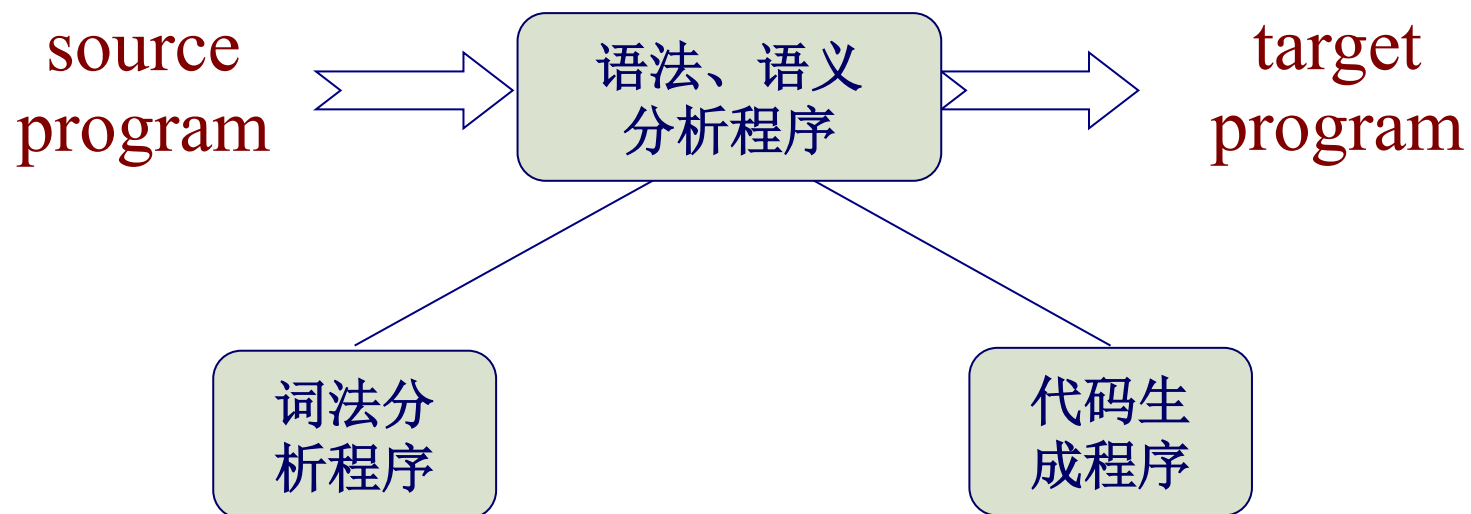
编译程序的组织

➤ 编译程序的遍(Passes)

- ✧ 可以将一个或多个逻辑步骤组织成一遍；
- ✧ 每遍读入一个输入文件并产生一个输出文件；
- ✧ 每遍仅对一种代码形式从头到尾扫描一趟；
- ✧ 每遍将一个代码空间变换到另一个代码空间；
- ✧ 代码空间 = 代码 + 符号表 + 其他有用信息。

编译程序的组织

➤ 单遍扫描编译程序的组织



编译程序的组织

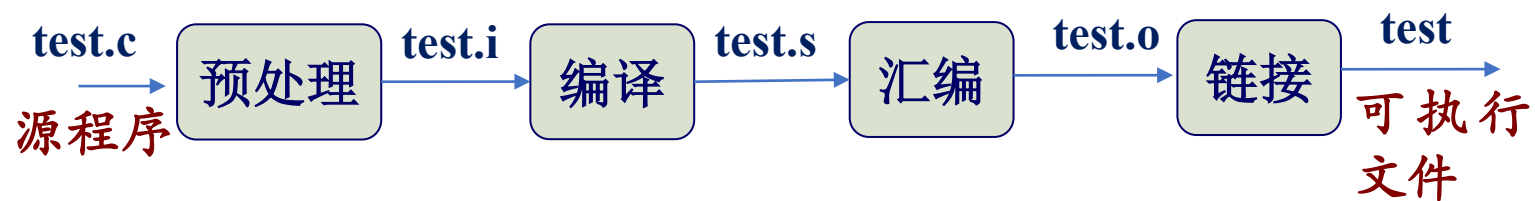
➤ 多遍扫描编译程序的组织

- ✧ 一组精心设计的中间表示;
- ✧ 可以将特定语言的前端和特定语言的后端相结合;
- ✧ GCC



编译程序的伙伴程序

- 预处理器
- 汇编器
- 链接器
- 加载器



编译程序的伙伴程序

➤ 预处理器

✧ 支持宏定义

如C源程序中 #define 行的处理

✧ 支持文件包含

如C源程序中 #include 行的处理

✧ 支持其他更复杂的源程序扩展信息



编译程序的伙伴程序

➤ 汇编器

- ✧ 将汇编语言程序翻译为可重定位的机器语言程序

➤ 链接器

- ✧ 合并多个可重定位机器语言程序文件到同一个程序

➤ 加载器

- ✧ 将可重定位机器语言程序的相对地址变换为绝对地址
- ✧ 装入和连接程序产生最终可执行的机器语言程序