

黑盒20%很有意思，50%很难不好理解，80%也就这么回事

编译器：源程序转换成目标程序

保持语义等价

- 狭义：程序设计语言到机器代码

编译器：一次编译多次运行，每次执行目标程序

翻译书的人

解释器：直接运行解释器，不生产目标程序，根据源程序语义直接运行。边解释边执行，错误诊断好

同声传译

编译器结构

分析部分

编译器的前端，与机器无关，和语言有关

- 源程序，语法结构，句法结构，中间表示
- 记录源语言信息的符号表
- 纠错

综合部分

编译器的后端，和机器相关

- 根据符号表和中间表示生成目标程序

词法分析

基于词素，产生词法单元，得到词素序列的语法结构

- 一个变量就是词素
- token-name由词法分析步骤使用
- attribute-value指向相应符号表，由语义分析&代码生成使用

语法分析

创建语法树

语义分析 (semantic)

也是树，经常和语法分析在一起做

- 类型检查，类型转换
- 用于代码生成

中间代码生成

把树用代码计算，对应指令

代码优化

改进中间代码，得到更好目标代码

- 运行快，占存少

需要保守，优化不好就变得不等价

代码生成

生成汇编

- 寄存器分配
- 指令选择
- 内存分配