

编译原理 第一章作业

2154312 郑博远

1. 高级程序设计语言有哪两种执行方式？其特点是什么？

高级语言程序设计采用解释方式和编译方式两种方式运行。

解释程序边解释边执行源语言程序，不产生目标语言程序。其优点是易于查错、启动速度快、跨平台可移植性好；且由于不产生目标语言程序，其通常使用较少的内存空间。其缺点是每次执行程序都需要重新解释源代码，因此执行速度通常比编译执行慢。

编译程序能够把某种语言的程序转换成另一种语言的程序，而后者与前者在逻辑上是等价的。其优点是执行速度通常比解释执行快，因为代码已经被转换成底层机器代码，不需要每次都进行解释和分析。其缺点是在运行中发现的错误必须查找整个程序确定且移植性不好。此外，编译执行通常需要更多的内存空间，因为需要保存生成的目标代码。

2. 什么是编译程序？它的功能是什么？

编译程序是一种工具，它可以将某种高级编程语言的程序转换成另一种语言的程序，而这两种程序在逻辑上是等价的。编译程序的主要功能是将高级语言源代码翻译成另一种目标语言的代码，通常是机器代码或中间代码。这个转换过程确保了程序的功能和逻辑在两种语言中是相同的，但代码形式不同。编译程序能够帮助程序员在不同平台上运行他们的程序，而无需手动编写每个平台的代码，从而提高了开发效率。

3. 一个编译程序由哪几个阶段构成？

编译程序通常可以分为以下五个主要阶段：

1. 词法分析：在词法分析阶段，编译器将源代码分解成词法单元或标记，例如关键字、标识符、运算符和常量。这些标记用于构建程序的基本语法结构。

2. 语法分析：语法分析阶段检查标记序列是否符合编程语言的语法规则。它将标记组织成语法树或语法分析树，以表示程序的结构和嵌套关系。

3. 语义分析和中间代码生成：在语义分析和中间代码生成阶段，编译器执行更深层次的分析，检查程序是否符合语义规则，如类型匹配、作用域规则等。同时，它将语法树或语法分析树转化为中间代码的抽象表示，以便后续的优化和目标代码生成。

4. 优化：优化阶段用于提高程序性能。编译器可以对生成的中间代码进行各种优化，例如常量折叠、循环优化和内联函数，以减少执行时间和资源消耗。

5. 目标代码生成：最后，编译器将中间代码翻译成目标平台的机器代码或虚拟机代码，即计算机硬件或虚拟机可以直接执行的代码。通过这一阶段，编译器实现了源代码在目标平台上的功能。

这五个阶段构成了编译程序的主要过程，其目标是将高级编程语言源代码转换成目标平台上可以执行的等效代码，同时保持程序的功能和逻辑正确性。不同编译器和编程语言的实现可能会在这些阶段中有所不同，但通常都涵盖了这些基本步骤。