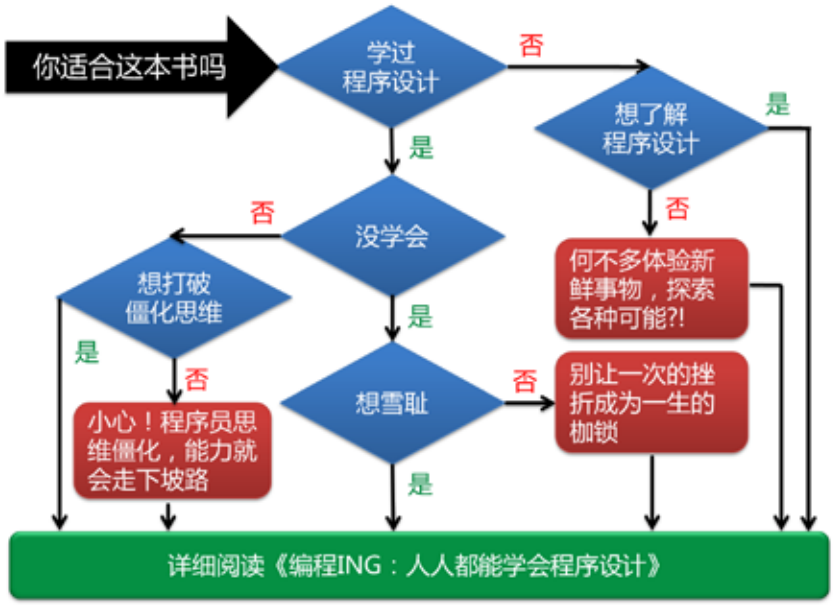
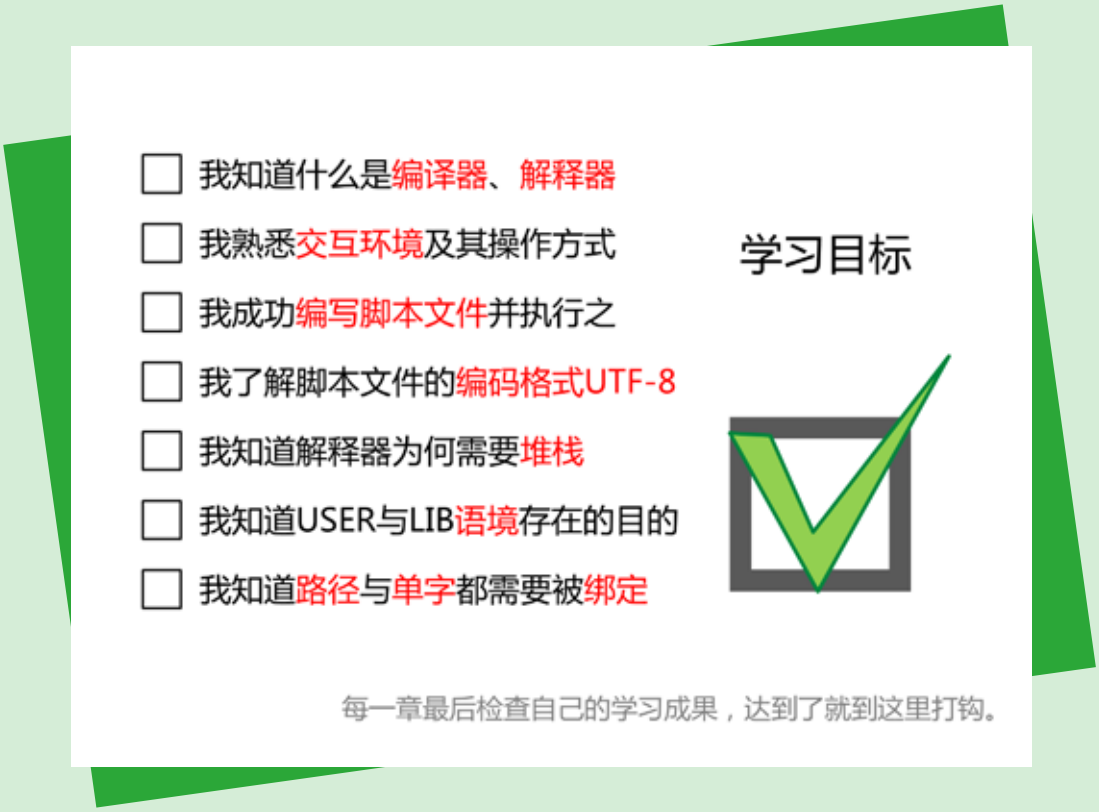


你适合这本书吗？不管你是谁，我都希望答案是 Yes。我认为只要平常习惯使用计算机且有简单英文基础的人，就可以是本书的读者。我想这样的条件涵盖了现在社会大多数的人。如果你没有任何程序设计的经验，这本书就是专门为你写的。

如果你不是程序设计的初学者且惧怕程序设计——因为过去有不好的经验，我希望这本书能扭转你对程序设计的看法与理解，摆脱阴影。如果你写过程序，或者正在写程序，热爱写程序，这也会是一本适合你的书，因为书中提到的语言是新思维的语言，设计得聪明巧妙，做法不同于主流的语言。你可以从不同的编程思想中激荡出新的火花，避免思维的僵化。

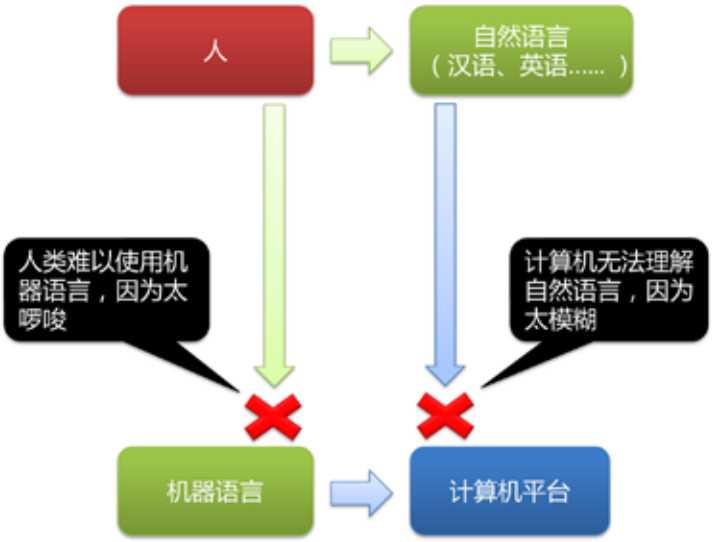
第1篇 编程原理



第 1 篇共有 7 章，对应 7 个学习目标。每当你完成一章内容的学习，可以自行评估是否完成了这一章的目标。若达到目标，就回到这里，在方框内打个钩。等 7 个钩都齐备了，就可以开始第 2 篇的学习。

第1章

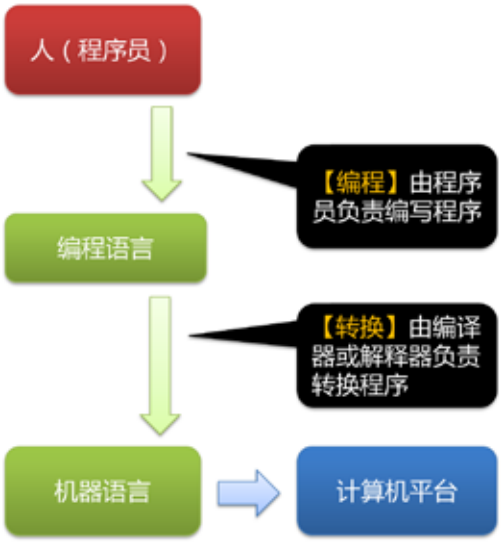
认识编程



人类使用的语言比如汉语、英语等，称为自然语言（natural language）；计算机使用的语言称为机器语言（machine language）。人类与计算机使用不同的语言，要如何沟通？

- 如果要人类学习计算机的机器语言，对人类来说太困难，因为机器语言都是01010011这样的二进制格式。即使要计算机做一件很简单的事，也需要不可思议地啰唆叙述。
- 如果要计算机学习人类的自然语言，对计算机来说太困难，因为自然语言都太不精确（比如双关语），而且很多与语气或上下文相关，涵盖的知识领域也太广，这些都会让计算机无法理解人类语言。

如何跨越这样的鸿沟呢？



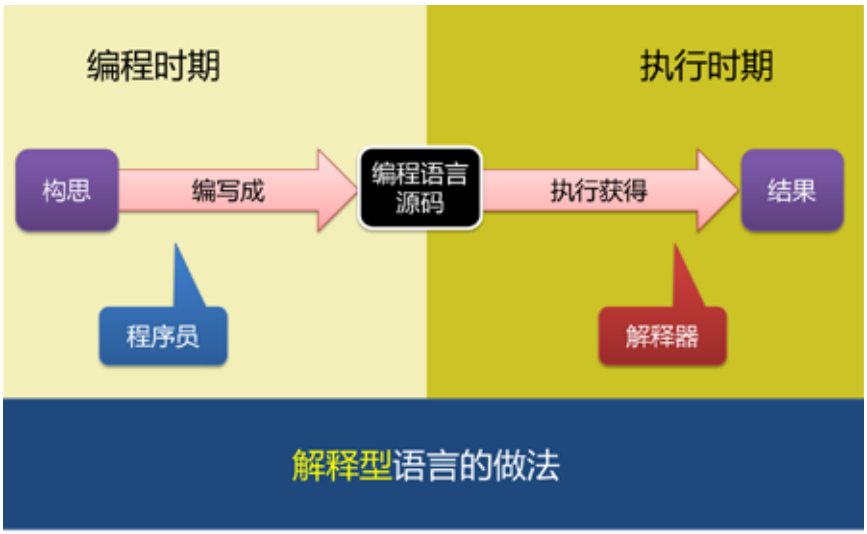
可行的方法是设计一套编程语言（programming language）。编程语言很容易学习与使用，因为它结合了机器语言的精准，并使用一些人类语言的符号（例如 `if`、`while`），让计算机与人类都能接受。人类只要经过一段时间的学习，就能够使用编程语言；而这个语言因为相当精准，所以可以通过一种转换软件（编译器或解释器，稍后说明），转换成机器语言让计算机执行。

能使用编程语言写程序，并以此为职业的人，称为程序员（programmer），或者程序设计师。程序员写出来的原始程序（未经任何转换处理）称为源代码（source code），或代码（code），或源码。

编译器是一种软件，用来在编译时期将源码转换成机器码

程序员负责将构思通过编程语言编写成源码，再将源码交给编译器（compiler），转换成机器码（machine code），然后就可以在计算机上执行。编译器将源码编译成机器码的这个过程，称为编译时期（compile-time）。机器码在计算机内执行的阶段，称为执行时期（run-time）。需要用到编译器的编程语言，称为编译型语言。

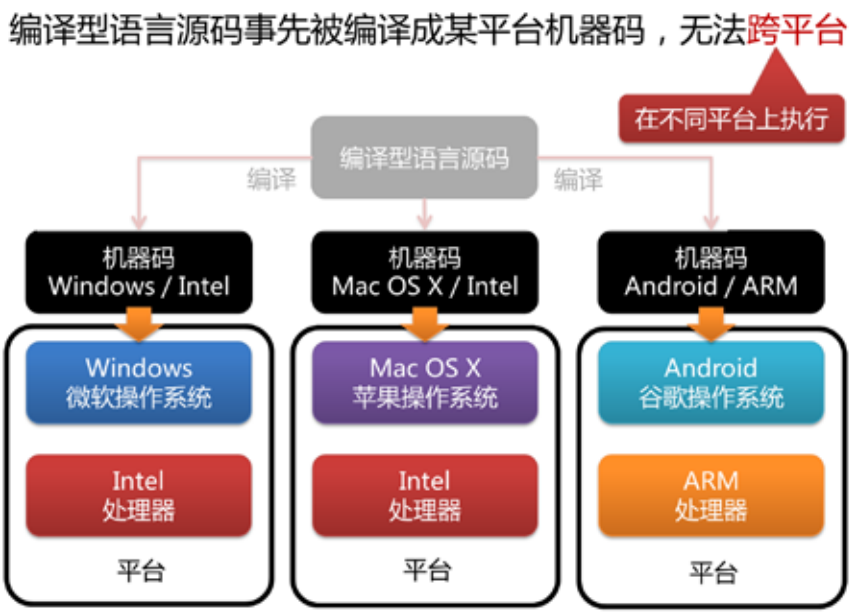
我们可以把这看成一次性翻译。事先翻译成机器码，以后每次执行都是直接执行机器码，不需要再转换。编译过的程序因为是机器码，所以执行效率很高，这是编译型语言一个很大的优点。



解释器是一种软件，用来在执行时期将源码转换成机器码

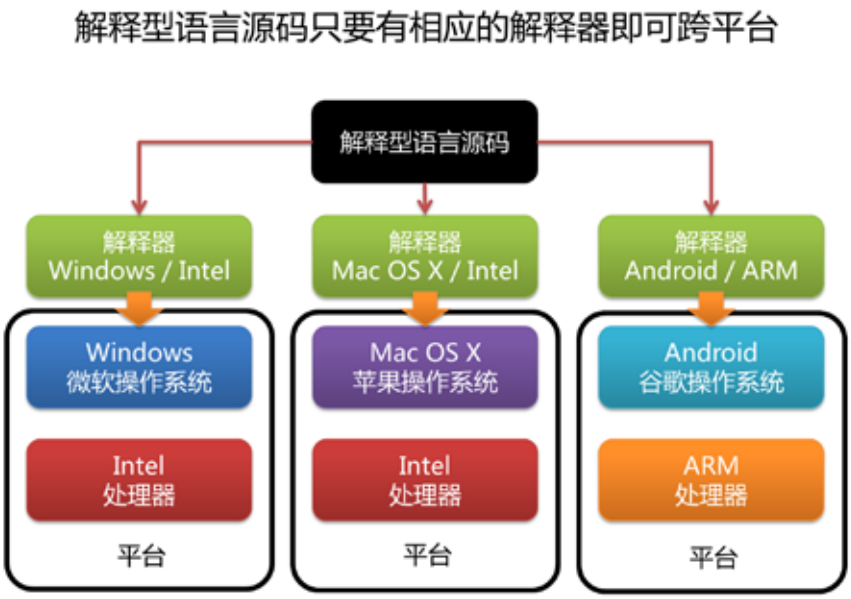
有些语言不需要编译器，而是在执行时由解释器（interpreter）一边翻译一边执行的。需要解释器的语言称为解释型语言。采用解释型语言写出来的代码常被称为脚本（script），所以解释型语言也常被称为脚本语言（scripting language）。

用解释型语言写出来的程序，每次执行时都要再次翻译，所以缺点是效率会低一点，但优点是跨平台（后面将说明原因）。

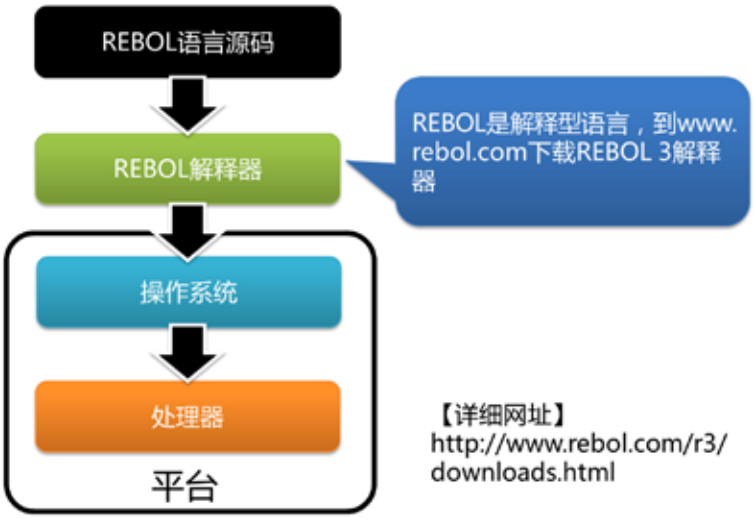


所谓的跨平台是指程序可以不经处理就在不同平台上执行。而“平台”一词有很多定义，在本书中是指操作系统与硬件（处理器）的组合。

用编译型语言写出来的程序，必须先编译成机器码。而机器码是与底下的平台息息相关的，所以用编译型语言写出来的程序，无法跨平台（也就是说，无法在不同的平台上执行）。



解释型语言则很容易跨平台，因为它的可执行代码就是源码（不需要编译），所以代码中没有与平台相关的部分。不管平台是微软 PC（Windows + Intel），或者苹果 Mac（Mac OS X + Intel），或者安卓手机（Android + ARM），或者其他平台，只要该平台上有对应的解释器，就可以顺利执行解释型语言写出来的程序。



本书使用的语言是 REBOL，这是语法最简单的一种语言，只有少数规则必须记，所以相当适合初学者。但 REBOL 可不是玩具，它是很有用的工具，威力很强大，能做许多实际的事。而且 REBOL 有许多创新的做法，可以让其他编程语言老手感到惊喜。简单、实用、创新，这三个理由让我选择 REBOL 当这本编程启蒙书的主角。

REBOL 是解释型语言，用 REBOL 语言写出来的程序并不能马上运行，必须有 REBOL 解释器才行。REBOL 解释器就像一个翻译员，它协助我们把 REBOL 程序翻译成底下平台能理解的形式。没有 REBOL 解释器，我们的 REBOL 程序就无法运行，所以必须在计算机中准备好一个 REBOL 解释器。

你可以到 www.rebol.com 网站免费下载 REBOL 解释器。目前 REBOL 解释器有两大版本，一个是 2.7.8 版，一个是 3.0 版。本书使用的是最新版本 3.0 版。

也请各位关注一个名为 Red 的新语言，它是开放版本的 REBOL 语言。网址在 www.red-lang.org。

用浏览器访问 http://www.rebol.com/r3/downloads.html，根据你的操作系统，下载正确版本的 REBOL 解释器。如果你的操作系统是微软 Windows，你可能需要下载的是 r3-a111-3-1.exe；如果你的操作系统是苹果 Mac OS X，你可能需要下载的是 r3-a111-2-5.tar.gz。a111 是本书出版时 REBOL 的最新版本。当你阅读此书时，如果有更新版本的 REBOL 解释器，请使用最新版。

为了加快下载的速度，有些文件会用压缩格式提供下载。如果你下载的文件的扩展名是 .gz 或 .zip，表示此文件是压缩格式，下载后，还有一个解压缩的步骤。解压缩的方式很简单，用鼠标双击就可以解压缩到一个新的文件夹，新文件夹内的文件就是 REBOL 解释器。

你可以为解压缩后的 REBOL 解释器文件重命名，取个简洁清楚的名称（例如 REBOL3），但必须维持原来的扩展名（例如 .exe）不变。你可以在“用户”目录下建立一个 REBOL 主目录，专门放置 REBOL 相关的文件。以我为例，我在 Windows 的 REBOL 主目录是 C:\Users\Jerry\REBOL，在苹果 Mac OS X 上的 REBOL 主目录是 /Users/Jerry/REBOL。

最后，将下载的（且解压缩过）REBOL 可执行文件复制到 REBOL 主目录内。