第七章 高级语言与低级语言

学习目标

编程语言

- 高级编程语言
- 低级编程语言

转换器

- 编译器
- 解释器
- 汇编器

7.1 介绍

我们使用高级编程语言编写程序，就如你现在使用的 python 一样，这些代码我们是可以理解的，比如你写的代码我可以理解，但是机器是理解不了的。机器需要把这些代码转换成机器码（01 组成的机器码）。

7.2 编程语言

7.2.1 什么是编程语言

变成语言是一种方式：我们通过编写代码，告诉计算机我们要做什么，怎么做，什么时候做。

计算机程序是完成某些特定功能的一组指令。计算机程序有高级程序和低级程序，我们都可以编写。大部分程序员都是编写高级程序的。

7.2.2 高级编程语言

高级编程语言使我们主要关注问题本身，而不用关注计算机底层的实现。即我们只要编写代码就好，这个代码如何执行的我们并不需要关心。

好处：课本上。

不同语言可以相互转换，比如用 python 编写的代码可以转换成 c++的。

7.2.3 低级编程语言

低级编程语言是和机器架构和硬件相关的，比如电脑上和手机上的低级语言就有区别。低级语言分为机器码与汇编语言，执行的时候是高级语言->汇编语言->机器码。

有少数程序员是编写汇编语言的，原因如课本上。

例子。

一般没有人直接编写机器码，因为它都是 01 组成的代码太难读了，就算用 16 进制展示也不容易读。

7.3 转换器

转换器就是高级语言与低级语言转换的工具。

编译器

将整个高级语言程序转为机器码。一次编译，到处执行。产出一个可执行文件。

解释器

解释器，与编译器干同样的事，都是转换成机器码，但是方式不一样。解释器每次只转换一行。

汇编器

将汇编语言转换成机器码。产出一个可执行文件。

7.4 错误

语法错误

当有语法错误时，比如 print 写成了 pritn 编译型语言无法编译出可执行文件，就算没错的语句也无法执行。解释性语言可以执行没错的语句，直到碰到错误停下。

逻辑错误

逻辑错误只有靠运行时才能发现，比如你想算 c=a*b，写成了 c=a+b，只有观察运行结果发现错误改正才能发现。

7.5 交互开发环境

交互式和集成开发环境(IDE)

交互式开发环境 Interactive Development Environment

输入一行，回车后执行一行，方便测试某些功能，初学者使用。例如 python 中的 Ipython 就是 python 的交互式解释器
使用场景：用于较短的命令测试，在使用新的命令时可以快速测试出来
比如：输入 print 函数后回车下一行就出现要打印的东西

集成开发环境 Integrated Development Enviroment
能够完成编写程序，执行控制台，调试等功能的集成开发，例如 pycharm，spyder 可以认为是 python 的集成开发环境。
使用场景：多使用于代码长度较多和需要测试整体运行的地方

课后习题一起做一下。