NOIP C++ 编程语言

# 先行课---计算机工作原理

计算机现在几乎无处不在，改变着我们生活的方方面面。虽然我们每天都在用着这项革命性的技术，但很少有人知道“计算机是如何工作的？”

我们学习信息比赛的同学都将是未来信息化社会的栋梁，为了学习计算机程序设计，我们首先需要了解计算机的工作原理。

位于美国西雅图的计算机科学非营利组织，code.org，作为它的计算机科学发现系列的一部分，录制了一个名为“[计算机是如何工作的](https://www.bilibili.com/video/av35082389?from=search&seid=16791620252283005022)”系列共六个小视频，由微软帝国的创始人、计算机软件领域的传奇人物比尔盖茨领衔做宣传介绍，建议大家在学习本课程时首先要完成该系列视频的学习！



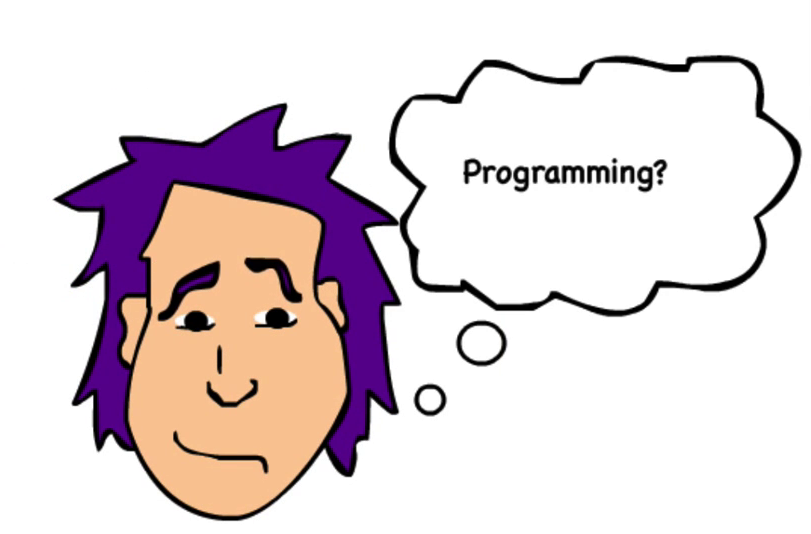
事实上，不管你是对每天用的设备感到好奇，或者想在未来做一些创新的东西，都应该从学习“计算机是如何工作的”开始。

这里是视频链接：[计算机是如何工作的](https://www.bilibili.com/video/av35082389?from=search&seid=16791620252283005022)

# 第0章 C/C++快速介绍

## 程序，编程和编程语言（Programming & Programming Language）

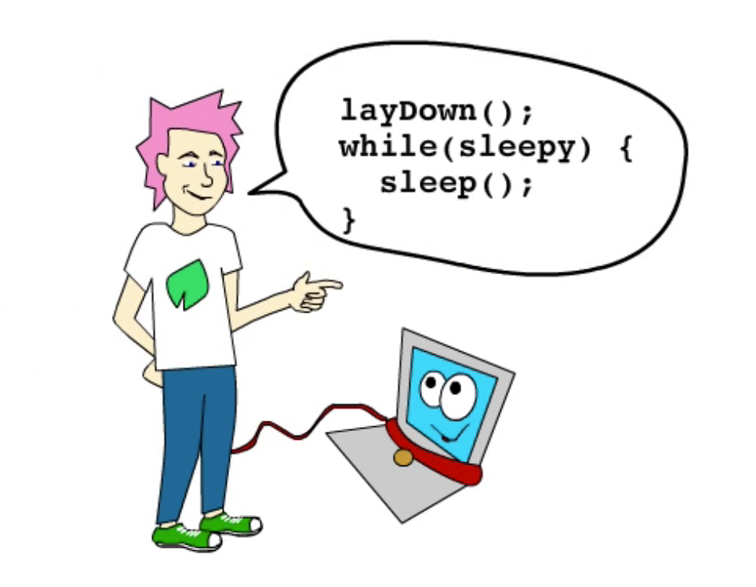
如果你之前没有学习过编程，你可能很想知道什么是编程（Programming）？



程序：Program，剧本，节目，计算机完成某种特定任务而执行的一系列指令。

编程：Programming，简言之，就是编写程序（Writing Program）。

当我们编写程序的时候，我们是使用一种看起来怪异格式的语言来给计算机下达一系列的命令（Command）。你可以认为计算机就是一个非常听话的小狗，不管你让它做什么它都会照你说的去做。



而给计算机下达命令，我们需要使用计算机能够听懂的语言，这就是编程语言（Programming Language）。

### 机器语言

计算机能够直接读懂和执行的语言，即CPU能够识别的指令，叫做机器语言，它包含控制和运算（算术（加减乘除）和逻辑运算（与或非））等基本指令，是计算机唯一能够理解的语言。

### 编程语言，编译器与解释器

人类生活中使用的语言叫做自然语言（比如我们使用的汉语普通话、英语等）。

为了能够编程（给计算机下达指令），我们需要使用双方都能理解的语言。比如我们可以直接使用机器语言进行编程（早期的编程就是直接使用机器语言，但是这样编程效率很低），或者让计算机能够理解自然语言，这样我们就可以直接使用汉语或者英语给计算机下达指令了。但自然语言过于复杂，而且有歧义，计算机很难理解（自然语言理解是目前很热门的一个人工智能研究领域）。

解决方案是引入一个中间人-翻译：为了便于人与计算机的交流（编程），人们设计了各种介入自然语言和机器语言的中间语言，即编程语言。

编程语言消除了自然语言的歧义性，容易翻译成机器语言。我们使用编程语言编写程序，这样的程序叫做**源代码**，源代码需要被翻译成机器语言代码才能被计算机执行。

机器语言，汇编语言，高级语言。

### 高级语言与低级语言

根据编程语言是更接近于机器语言还是更接近于自然语言，将编程语言分为低级语言和高级语言。

我们一般称接近于机器语言的为低级语言，比如汇编语言和C语言；更接近于自然语言的为高级语言，比如C++/Java/Python/Go语言等。

源代码翻译成机器语言代码的方式主要有两种：

事先翻译与现场翻译

### 编译与翻译

一种是编译：在源代码的执行前，事先使用编译器（compiler）将整个源代码翻译成机器语言程序后才能在计算机上执行。

另一种是解释：有一些语言，用它编写的程序不是事先翻译成机器程序，而是由一个解释器在程序运行时，一行行动态解释来翻译成机器语言指令来执行。

前一种需要事先编译成机器指令程序的语言称为编译型语言，后一种在运行时由解释器翻译成机器指令执行的语言称为解释性语言。

Pascal/C/C++语言属于编译型语言，而Python属于解释性语言。Java则是。。。有争议

## 操作系统与编程

操作系统，资源的管理者，1， 提高资源利用率2，方便使用。

Windows，Linux，macOS,Unix

OS 提供进程-虚拟计算机（i/o, cpu,memory）

OS的发展：无OS，单用户，字符，多用户分时，虚拟存储。

为虚拟计算机编程：认为我们拥有一个虚拟的计算机。

## UNIX与C语言

C语言最初是由贝尔实验室（美国）的丹尼斯·里奇 (Dennis Ritchie)和肯·汤普森（Kenneth Thompson）为开发UNIX操作系统而设计的，是一种通用的程序设计语言。其特点包括简洁的表达式、流行的控制流和数据结构、丰富的运算符集等。

一般说C语言是一种相对“低级”的语言。这种说法并没有什么贬义，它仅仅意味着C 语言更接近于机器语言，可以直接处理大部分计算机能够处理的对象，比如字符、数字和地址。由于它很适合用来编 写编译器和操作系统，因此被称为“系统编程语言”。





图一 丹尼斯·里奇 (右)和肯·汤普森（左）

**由于他们“研究发展了通用的操作系统理论，尤其是实现了UNIX操作系统”，1983年丹尼斯·里奇与肯·汤普逊一起获得了计算机界最负盛名、最崇高的、被称为计算机界的诺贝尔奖”之称的图灵奖。1999年两人为发展C语言和Unix操作系统一起获得了美国国家技术奖章。一般称丹尼斯·里奇为C语言之父，肯·汤普森为Unix系统之父，或者并称C语言与Unix系统之父。**

## C++ 语言

C++语言是由同样来自于贝尔实验室的本贾尼·斯特劳斯特卢普（Bjarne Stroustrup）在C语言的基础上引入并扩充了面向对象的概念，发明的一种新的程序语言。为了表达该语言与C语言的渊源关系，它被命名为C++。而本贾尼·斯特劳斯特卢普博士被尊称为C++语言之父。



图二 本贾尼·斯特劳斯特卢普

C语言是一门面向过程（Procedure Oriented）的编程语言，C++语言在C语言的基础上增加了面向对象（Object Oriented）和泛型编程（Generic Programming)的支持。

面向过程的编程思想认为程序是由数据以及对数据的处理（即过程）组成的，而面向对象的编程思想认为，程序，就像我们的自然世界一样，是由对象（Object）以及对象间的交互组成的。（Object翻译成“对象”并不是很好理解，翻译成客体、物体更合适）

C语言是一门简洁优雅的“低级语言”，但C++则是拥有众多高级特性的“高级语言”。事实上，由于拥有众多的高级特性，C++语言是最难于学习掌握的一门编程语言。幸运的是，信息比赛里只会使用到C++语言的一个非常小的子集，学习掌握起来还是相对容易的。不过，适当地掌握C++语言的一些高级特性，特别是标准库的使用，可以极大方便我们程序的编写，提高编程的效率，达到事半功倍的效果。

## 编程工具：编辑器（editor），编译器（compiler）以及集成开发工具（IDE）

源程序是使用某种编程语言（比如C++）编写的文件，你可以使用任何你熟悉的编辑器来编写源程序。

Windows操作系统上的写字板，记事本，word等，MacOS/Linux上的vi，vim，emac等通用编辑器都可以用来编写C++程序。

比如下面是一个C++源程序：



我们说，计算机能够直接执行的程序是机器语言程序，需要使用编译器，将C++源程序翻译成机器语言程序，这个处理过程称为编译（compiling）。

现在我们最常使用的是使用GNU编译工具集（GCC，GNU Compiler Collection)，其中用于执行C++源程序编译的编译器叫做g++。

比如将上面的源程序保存为helloworld.cc,下面的命令将它编译为可执行的机器语言程序，在Mac和Linux，生成的可执行文件名字默认为a.out：

$ g++ helloworld.cc

下面是使用hexdump工具显示的上面C++程序编译成的机器语言程序的开头几行（这是在Mac电脑上的可执行文件，Windows电脑上跟这个不一样）：



为了方便软件开发工作，人们设计了集成开发环境（IDE，Integrated Development Environment）。IDE将编辑器、编译器以及用于排除源代码错误的调试器以及其它辅助开发的工具集成在一起，特别是，这些工具可以为某种语言的程序开发高度定制，比如，编辑器中的语法高亮显示、自动代码补齐、边编辑时边进行语法检查等，图形化的调试器等，极大地方便了开发人员的工作，提高了开发的效率，建议初学者在学习编程时都要选用一款IDE来使用。

下面分别介绍下各种操作系统上推荐初学者使用的IDE：Windows系统上的Code::Blocks，Mac 系统上的Xcode， Linux平台的同学也建议Code::Blocks,不过这里不介绍它的安装。

注意，安装IDE对于初学者还是一项很有挑战性的工作，建议请求学长或者老师帮忙安装。此外，这里还会介绍一个在线练习C++开发的网站，初学者使用它是非常方便的。