

目录

第零章 开始之前	1
0.1 从语言的发展了解计算机	1
0.2 内存、程序与数据	5
0.3 算法与算法的复杂性	7
0.4 数据结构	10
0.5 二进制	11
0.6 习题	11
第一章 代码中窥 C++	13
1.1 关于 C++	13
1.2 Hello IMUDGES!	15

第一章 代码中窥 C++

1.1 关于 C++

凡是语言，都一定有着自己的规则。而不同于英语、汉语这种自然语言，作为计算机所能识别的语言，它是需要在严格约束下避免二义性的，所以说我们将学到的 C++ 语言，是一门语法规则非常严格，但又“灵活多变”的语言。

所谓非常严格，是指 C++ 中不允许出现歧义，这种没有歧义的文法是经过严格的文法规则的限制、和一些约定来约束的。所以，如果在你的代码中出现了自己都会觉得有歧义的地方，往往就是程序中容易出错的地方。

而所谓灵活多变，就是指在掌握 C++ 严格的语法之后，可以写出无穷无尽充满创意的程序。从黑框 (控制台程序)，到带有图形界面 (GUI) 的程序，再到游戏，都可以利用 C++ 的语法加上你脑中的算法搞定！而所有的这些灵活多变的内容，都是在严格的文法约束下实现的。所以说 C++ 语言，是一门语法规则非常严格，但又灵活多变的语言！

C++ 是一门高级语言，所谓高级语言正如你们在前面章节中看到的，是不同于机器语言、汇编语言；能更加贴近人类思维的一门语言。C++ 是在 C 语言的基础上，增加了例如面向对象编程等诸多特性的程序设计语言。在程序设计竞赛中被广泛采用。

非常不严格地讲：我们常常所说的“C++”，包含了这门语言和一系列的工具。语言顾名思义就不多阐述。而工具实际上包含了很多内容：

我们将 C++ 语言所编写的代码叫做“源代码”。而源代码是不能直接被计算机执行的（回想我们在上一章中讲述的内容）。而是应该利用编译程序，将源代码编译成计算机能够识别的机器代码然后再执行。而执行编译

工作的工具就叫做“编译器”。

除此之外，仅仅有 C++ 语言，我们连基本的功能（例如屏幕输出）都实现不了。即便是高级语言，想要与硬件设备交互，一般都需要直接编写一定的机器语言或者汇编语言，所以才能让我们的程序操作硬件：屏幕打印、屏幕输入、文件操作等等。而我们想要通过一己之力实现这些最底层的操作是很困难的。好在 C++ 提供了一组可以被调用的工具。这些工具其实就是别人写好的代码，通过 C++ 语法能够接受的形式所调用，最终辅助我们完成底层调用的功能，让我们的精力能够集中在更重要的算法设计上而不是千篇一律的与底层的通讯上。我们称这种“别人的代码”叫做“库”。就如同仓库一样，我们可以调用仓库中的内容来实现我们的意图。

那么我们的程序是怎么与别人的程序产生关联的呢？这有很多种情况，我只讲述最常见的一种：静态链接。我们的程序在调用库的时候，编译器可以仅仅在相应的位置上做个标记，表明这里的程序是调用了库而不是我自己的。这么做的好处就是可以充分地提高编译效率。因为库中的代码往往经过严格的编写、测试，基本能保证没有问题，所以我们只需要保存该库编译完毕的状态，没有必要在编译自己的程序时还要将别人的库再编译一遍。我们只需要编译自己的程序就可以了。所以我们的程序与库之间是并列关系，而不是包含关系。但是想要让自己的程序能够变成可执行程序，仅仅是这种“做标记”的方法是不可接受的。因为你所需要的库文件在你的电脑上会有，但并不意味着别人的电脑上也会有。再加上可执行程序的复杂原理，我们必须将这些静态库与我们的程序绑定在一起。而将编译完的“目标代码 (Object Code)”与库文件链接形成“可执行文件”的工具叫做“链接器”。

也就是说，C++ 程序设计流程是：

1. 编写代码
2. 编译成目标代码
3. 链接成可执行文件

一般来说，想要实现上面的流程，需要利用操作系统提供的黑框 (Linux 下的终端、Windows 下的控制台)，再加上一大堆非常复杂的命令才能实现。编译命令甚至有可能比源代码还要复杂。如果需要对自己的代码进行调试，

那可是更加麻烦。所以聪明人们就为懒人发明了又一件工具：集成开发环境（Integrated Development Environment, IDE）。所谓开发环境，可以将其看作是写代码的一个工具。我们可以在这个开发环境中编写代码、代码着色、自动排查错误等。所谓“集成”就是指将“编译器”、“连接器”都集中在 IDE 中。程序员可以简简单单地点击按钮，就可以实现整套编译链接运行调试等功能。

在 ACM/ICPC 的比赛中，通常使用的编译器是 GCC(GNU Compiler Collection)。GCC 是一个支持很多种语言的开源的编译器的集合。它可以安装在 Windows（不推荐）和 Linux 等多种操作系统上。而在比赛中普遍使用的是安装了 Linux 操作系统和 GCC 编译器的开发环境，并且配置了某些 IDE 来让学生进行开发。

关于 IDE：对于 Windows 来说，我们推荐使用 Dev-C++。对于 Linux，我们推荐使用 Code-Blocks。

1.2 Hello IMUDGES!

其实吧，作为一名编程初学者，第一个编写的程序应该就是传说中的“Helloworld”。这个程序意图就是在屏幕上简简单单地打印出一个“Helloworld”的文字。目的是为了表示当前的开发环境没啥问题，可以继续工作了。

但是我觉得一个 Helloworld 太简单了，各种书籍都把这个例子用烂了，那我在这里就用一个稍微复杂的例子来实现这个“Helloworld”吧！

代码 1.1: Helloworld 升级版

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     cout<<"a"<<endl;
5     return 0;
6     //comment
7 }
```