

初窥门径——C 语言和程序设计



项目概述

如同人和人的交流需要语言一样，我们与计算机的交流也需要语言；编程语言是人与计算机打交道的桥梁，它是人类的翻译官。众所周知，C 语言是较早的几门编程语言之一，它至今仍服务于现代社会。从开辟“鸿蒙”到“人工智能”，C 语言从一诞生就开始了它的风行世界之旅，放眼现在与未来：华为的自研操作系统就是主要用 C 语言开发的，万物皆可互联、机器拥有智能的时代，也依然离不开 C 语言的身影。从 TIOBE 世界编程语言排行榜可以看出，C 语言一直都名列前茅，让我们跟随本书一起来开启 C 语言学习之旅吧！

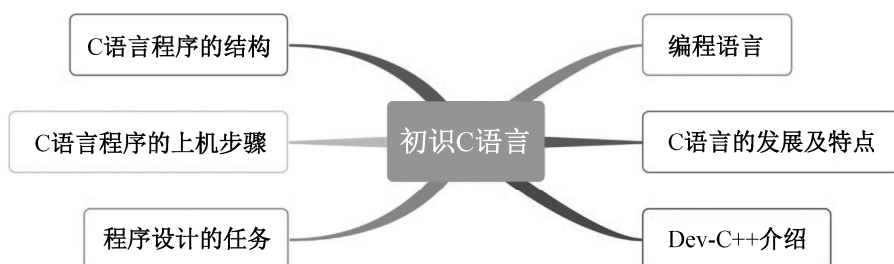


学习目标

了解编程语言的发展和特点；掌握 C 语言程序的运行步骤与方法；能简单地阅读 C 语言程序。



知识框架





任务1

Hello world! ——初识 C 语言



任务描述

编程语言是人与计算机打交道的桥梁，它充当人类的翻译官，请在 C 环境中输出“Hello world!”。



任务分析

本任务通过在 C 环境中输出“Hello world!”，了解编程语言及 C 语言的发展和特点，初步认识 Dev-C++ 的运行环境及运行的基本步骤。



知识准备

或许在此之前你没接触过 C 语言，但你肯定听说过它，因为对于其他编程语言来说，C 语言确实是一个“老古董”了。但 C 语言之所以能长盛不衰，自有它的优势和特点。

1.1 编程语言

编程语言是干什么的？为什么需要学习编程语言？简单地说，编程语言是人类与计算机打交道的桥梁，它充当人类的翻译官。计算机语言经历了以下几个发展阶段。

机器语言 计算机工作基于二进制，从根本上说，计算机只能识别和接受由 0 和 1 组成的指令。计算机能直接识别和接受的二进制代码称为机器指令。机器指令的集合就是该计算机的机器语言，在语言的规则中规定各种指令的表示形式及它的作用。显然，机器语言与人们习惯用的语言差别太大，难学、难写，难记、难检查、难修改、难以推广使用，因此初期只有极少数的计算机专业人员会利用机器语言编写计算机程序。

汇编语言 为了克服机器语言的缺点，第二代编程语言——汇编语言很快就被开发出来了。在汇编语言中，引入了大量的助记符来帮助人们编程，然后由汇编编译器将这些助记符转换为机器语言，这个转化的过程称为编译。虽然汇编语言比机器语言简单好记一些，但仍然难以普及，只能由专业人员使用。不同型号计算机的机器语言和汇编语



言是互不通用的，用甲机器的机器语言编写的程序在乙机器上不能使用。机器语言和汇编语言是完全依赖于具体机器特性的，是面向机器的语言。由于它“贴近”计算机，或者说离计算机“很近”，故称为计算机低级语言。

C 语言 有需求就会有市场，有市场就会有研发的动力，以 C 语言为代表的第三代编程语言很快就被开发出来了。第三代编程语言称为高级语言，C++、C#、Java、Delphi、Python 等都属于第三代编程语言。

```
#include <stdio.h>
int main( )
{
    printf("Hello world!\n");
    return 0;
}
```

从以上程序段中可看出，打印“Hello world!”，C 语言只用了 6 行代码，但汇编语言需要用 20 行代码，而机器语言则需要用上百行代码，无论在开发效率还是代码可读性方面，C 语言都有着极大的优势。大家普遍更喜欢用 C 语言进行编程，而不是用汇编语言或机器语言。

事实上使用 C 语言进行编程，编译器会将 C 语言的代码编译成汇编语言，再由汇编语言的编译器编译为机器语言，通常看到的可执行文件就是机器语言文件，进而让 CPU 理解和执行。

1.2 C 语言的发展及特点

1972 年，贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言。后来，又对 C 语言做了多次改进，随着 UNIX 的广泛使用，C 语言也迅速得到了推广。1978 年以后，C 语言版本先后移植到大、中、小、微型机上，已独立于 UNIX 了。自此 C 语言风靡全世界，成为世界上应用最广泛的程序设计高级语言。20 世纪 90 年代初，C 语言在我国得以推广并成为学习和使用人数最多的一种计算机语言。

C 语言问世以后之所以得到迅速推广，是因为它用途广泛、功能强大、使用灵活，既可以用于编写应用软件，又可用于编写系统软件。C 语言主要有列特点。

1. 语言简洁紧凑，使用灵活方便。C 语言只有 32 个常用关键字，9 种控制语句，程序书写形式自由，主要用小写字母表示。
2. 运算符丰富。C 语言有 34 种运算符，运算类型丰富，表达式类型多样，灵活使



用各种运算符可以实现在其他高级语言中难以实现的运算。

3. 数据类型丰富。C 语言提供的数据类型包括整型、浮点型、字符型、数组类型、指针类型、结构体类型和共用体类型等，尤其是指针类型数据，能实现各种复杂的运算。

4. 具有结构化的控制语句（如 if...else 语句、while 语句、do...while 语句、switch 语句和 for 语句）。C 语言是完全模块化和结构化的语言。用函数作为程序的模块单位，便于实现程序的模块化。

5. 语法限制不太严格，程序设计自由度大。例如，对数组下标越界不进行检查，对变量的类型使用比较灵活。C 语言允许程序员编程有较大的自由度，但是该特点既是优点也是缺点，它要求程序员仔细检查程序，保证其正确性，不能过分依赖编译器。

6. C 语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此 C 语言兼备高级语言和低级语言的许多功能，可用来编写系统软件。C 语言的这种双重性，使它既可作为系统描述语言，又可作为通用的程序设计语言。

7. 用 C 语言编写的程序可移植性好。由于 C 语言的编译系统相当简洁，因此很容易移植到新的系统。

8. C 语言程序生成代码质量高，程序执行效率高。

1.3 Dev-C++介绍

Dev-C++是一个非常实用的开发工具，多款著名软件均是使用它开发的，它在 C 语言的基础上增强了逻辑性。它既可以运行 C 源程序，也可运行 C++源程序，区别在于源程序的扩展名不同，若要运行 C 源程序，需将文件保存为.c 格式。

Dev-C++是在 Windows 环境下适合初学者使用的轻量级 C/C++集成开发环境（IDE）。它是一款很容易获取的免费自由软件，遵守 GPL 许可协议分发源代码。它集合了 MinGW 中的 GCC 编译器、GDB 调试器和 AStyle 格式整理器等众多自由软件。Dev-C++ 4.9.9.2 可运行在 32 位机上，Dev-C++ 5.10 可运行于 64 位中文版 Windows 7 操作系统。

1. Dev-C++ 5.11 界面

Dev-C++ 5.11 界面如图 1-1 所示，开发环境包括多页面窗口工程编辑器、调试器，在工程编辑器中集合了编辑器、编译器、连接程序和执行程序，提供高亮度语法显示，



以减少编辑错误。

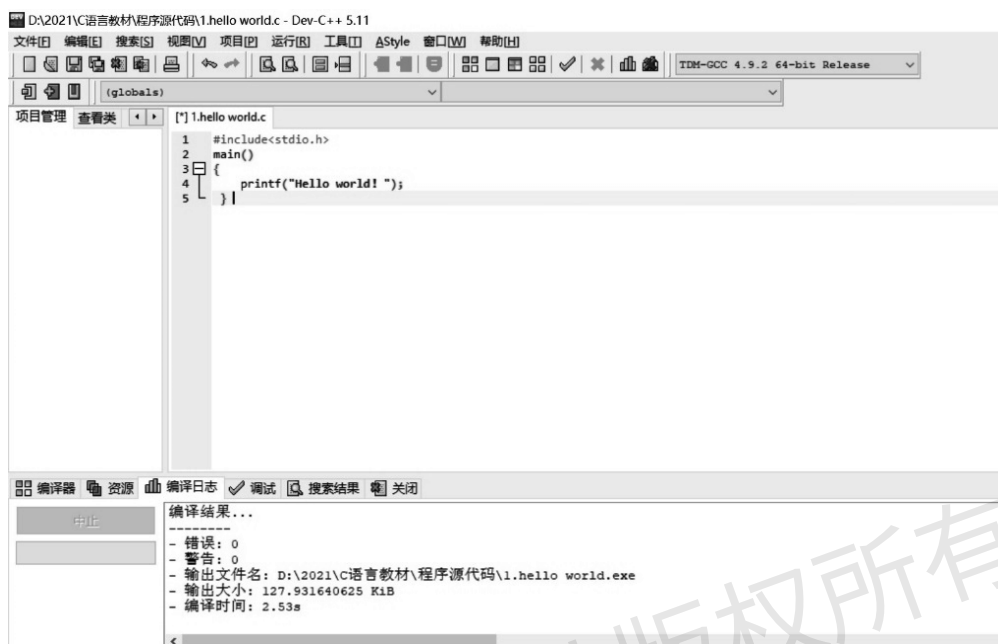


图 1-1 Dev-C++ 5.11 界面

Dev-C++ 5.11 中常见的按钮说明见表 1-1。

表 1-1 常见的按钮说明

按 钮	功 能	快 捷 键
	新建	Ctrl+N
	打开	Ctrl+O
	保存	Ctrl+S
	撤销	Ctrl+Z
	重作	Ctrl+Y
	编译	F9
	运行	F10
	编译运行	F11

2. 在 Dev-C++ 5.11 上运行 C 语言程序的基本步骤

(1) 新建或打开一个文件。

在 Dev-C++ 5.11 中新建或打开一个文件，使用按钮或，或者选择菜单“文件”→“新建”→“源代码”命令，如图 1-2 所示。



图 1-2 Dev-C++ 5.11 新建或打开一个文件

(2) 在编辑窗口中输入或修改程序。

```
#include<stdio.h>
main()
{
    printf("Hello world");
}
```

(3) 保存。

在“保存类型”下拉列表框中选择“C source file(*.c)”。

(4) 编译。

在消息窗口“编译日志”界面显示编译结果信息，如图 1-3 所示。

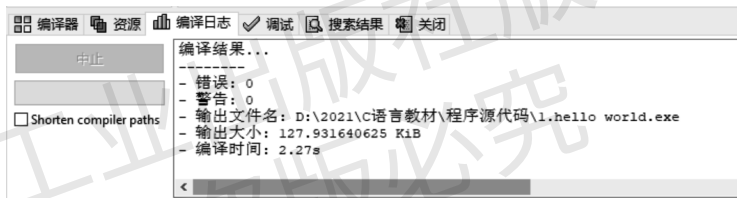


图 1-3 显示编译结果信息

(5) 运行。

若 C 源程序无编译错误，可单击  按钮运行目标程序，并弹出运行程序结果窗口，如图 1-4 所示。

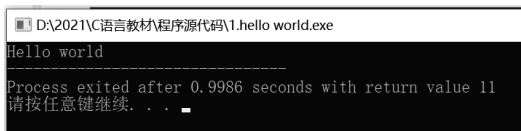


图 1-4 运行程序结果窗口

观察程序的运行结果，如果与预想的不同，应重新修改源程序，然后单击  按钮编译运行程序。重复上述步骤，直至出现满意的运行结果。



任务实施

本任务我们将初步认识 C 语言的运行环境及运行的基本步骤，并在 C 环境中输出“Hello world!”。



```
#include<stdio.h>
main()
{
    printf("Hello world!");
}
```

运行代码得到结果：

Hello world!



任务总结

本任务清楚了 C 语言的特点、运行环境以及运行步骤，实现了简单语句的输出。



任务拓展 经典编程

请使用快捷键编辑运行输出“这是我的第一个 C 语言程序。”

任务 2

求两个整数之和——C 语言程序的结构



任务描述

求两个整数的和，熟悉 C 语言程序的结构及 C 语言程序运行的方法。



任务分析

熟悉 C 语言程序的结构及 C 语言程序运行的方法。本任务需设置 3 个变量，其中，a 和 b 用来存放两个整数，sum 用来存放两数之和。用赋值运算符“=”把相加的结果传送给 sum 并输出。



知识准备

1.4 C 语言程序的结构

C 语言程序的结构有以下几个特点。



(1) 一个 C 语言程序是由一个或多个函数组成的，其中必须且只能包含一个 `main` 函数，但可以包含若干个其他函数。几乎程序的全部工作都是由各个函数分别完成，函数是 C 语言程序的基本单位。使用函数既可以简化程序，又可以提高程序的可读性，实现程序的模块化。

(2) 程序总是从 `main` 函数开始执行。`main` 函数在程序中的先后位置不影响程序的执行过程。由 `main` 函数开始调用其他函数，其他函数间也可以相互调用，最终返回 `main` 函数结束程序。

(3) 一个函数由函数首部和函数体两部分组成。函数首部即函数的第 1 行，包括函数名、函数类型、函数参数（形式参数）名、函数参数类型。例如，下列 `max` 函数的首部为

<code>int</code>	<code>max</code>	<code>(int</code>	<code>x,</code>	<code>int</code>	<code>y)</code>
↓	↓	↓	↓	↓	↓
函数类型	函数名	函数参数类型	函数参数名	函数参数类型	函数参数名

一个函数名后面必须跟一对圆括号，括号内写函数参数名及函数参数类型。如果函数没有参数，可以在括号中写 `void`，也可以是空括号，如：`int main(void)`或 `int main()`。函数体即函数首部下面的花括号内的部分。如果在一个函数中包括有多层花括号，则最外层的一对花括号内的部分是函数体。函数体一般包括声明部分和执行部分。

(4) 程序中要求计算机完成的操作是由函数中的 C 语言语句完成的。在每个数据声明和语句的最后必须有一个分号。分号是 C 语言语句的必要组成部分，是不可缺少的。如：

```
c=a+b;
```

(5) C 语言程序书写格式比较自由。一行内可以写几个语句，一个语句可以分写在多行上，但为清晰起见，习惯上每行只写一个语句。

(6) 程序应当包含注释。注释行从“`//`”开始到本行结束，连续若干注释行可以用“`/*`”开始，并以“`*/`”结束。注释可以增加程序的可读性，但不影响程序的功能。

1.5 C 语言程序的上机步骤

用 C 语言编写的程序称为源程序，计算机不能直接识别和执行用高级语言写的指令，必须用编译程序（也称编译器）把源程序翻译成二进制形式的目标程序，然后再将



该目标程序与系统的函数库及其他目标程序连接起来，形成可执行的目标程序。在编写好一个源程序后，怎样上机编译和运行呢？

运行一个 C 语言程序的主要步骤是：编辑→编译→连接→运行，其完整步骤如图 1-5 所示。

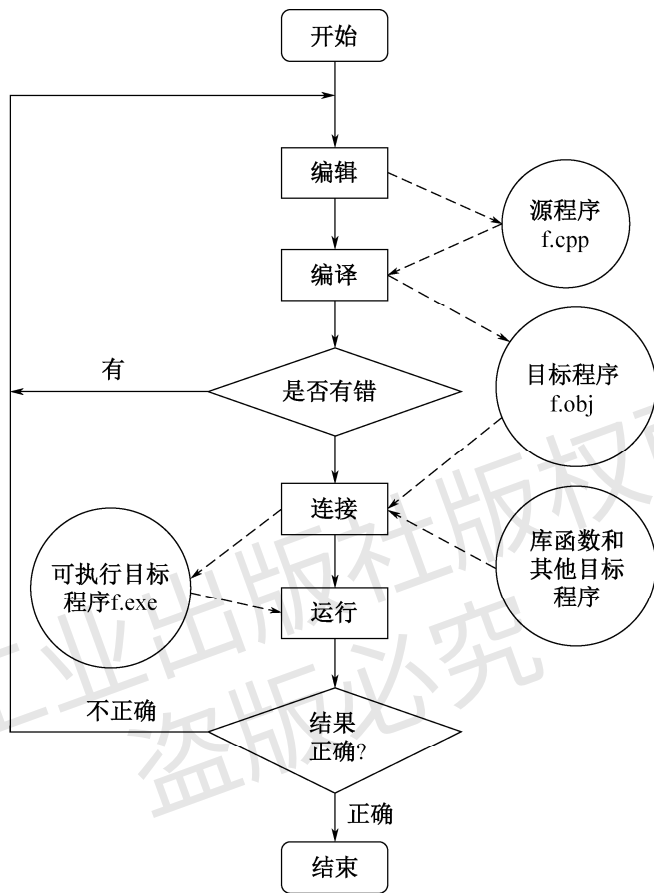


图 1-5 运行 C 语言程序的完整步骤

其中，实线表示操作流程，虚线表示文件的输入输出。例如，编辑后得到一个源程序 f.cpp 文件，然后编译时将源程序 f.cpp 文件输入，经过编译得到目标程序 f.obj 文件，再将所有目标模块输入计算机，与系统提供的库函数等进行连接，得到可执行目标程序 f.exe，并将其输入计算机使之运行，得到结果。

一个程序从编写到运行得到预期结果，不能保证一次就成功，往往要经过多次反复修改。编写好的程序并不一定能保证正确无误，除了用人工方式检查外，还需借助编译系统来检查有无语法错误。如图 1-5 可知，如果在编译过程中发现错误，应当重新检查源程序，找出问题，修改源程序，并重新编译，直到无错为止。有时编译过程未发现错误，能生成可执行目标程序，但是运行的结果不正确。一般情况下，这不是语法方面的错误，而可能是程序逻辑方面的错误，如计算公式不正确、赋值不正确等，应当返回检查源程序，并改正错误。



由于运行每个源程序都需要经过上述步骤，而这些步骤需要通过各种应用程序来实现，这就给开发者带来了不便。目前有许多集成开发环境（IDE）将程序的编辑、编译、连接和运行等操作集成在一个界面上，使用十分方便，如 Dev-C++、Visual Studio 2017、Turbo C 等软件。

1.6 程序设计的任务

如果只是编写和运行一个很简单的程序，上面介绍的步骤就已经足够。但是实际上需要考虑和处理的问题比上面的例子复杂得多。程序设计是指从确定任务到得到结果、写出文档的全过程。

从确定任务到最后完成任务，一般经历以下几个工作阶段。

（1）问题分析。对于接手的任务要进行认真的分析，研究所给定的条件，分析最后应达到的目标，找出解决问题的规律，选择解题的方法。

（2）设计算法。即设计出解题的方法和具体步骤。例如，要解一个方程式，就要选择用什么方法求解，并且把求解的每一个步骤清晰无误地写出来。一般用流程图来表示解题的步骤。

（3）编写程序。根据得到的算法，选择一种高级语言编写源程序。

（4）对源程序进行编辑、编译和连接，得到可执行程序。

（5）运行程序，分析结果。运行可执行程序，得到运行结果。能得到运行结果并不意味着程序正确，要对结果进行分析，看它是否合理。例如，把“ $b=a;$ ”错写为“ $a=b;$ ”，程序不存在语法错误，能通过编译，但运行结果显然与预期不符。因此要对程序进行调试（debug）。调试的过程就是通过上机发现和排除程序中故障的过程。经过调试，得到了正确的结果，但是工作不应到此结束。不要只看到某一次结果是正确的，就认为程序没有问题。例如，求 $c=b/a$ ，当 $a=4$ ， $b=2$ 时，求出 c 的值为 0.5，是正确的，但是当 $a=0$ ， $b=2$ 时，就无法求出 c 的值。说明程序只能对某些数据得到正确结果，对另外一些数据却得不到正确结果，即程序还有漏洞，因此，还要对程序进行测试（test）。所谓测试，就是设计多组测试数据，检查程序对不同数据的运行情况，从中尽量发现程序中存在的漏洞，并修正程序，使之能适用于各种情况。作为商品提供使用的程序，是必须经过严格测试的。

（6）编写程序文档。许多程序是提供给用户使用的，如同正式的产品应当配有产品说明书，正式提供给用户使用的程序也必须配有程序说明书（也称为用户文档）。内容



应包括程序名称、程序功能、运行环境、程序的装入和启动、需要输入的数据，以及使用注意事项等。程序文档是软件的一个重要组成部分，软件是计算机程序和程序文档的总称。现在的商品软件下载包中，既包括程序，也包括程序文档，有的程序文档则在程序中以帮助（help）或 readme 形式提供。



任务实施

设置 3 个变量，其中，a 和 b 用来存放两个整数，sum 用来存放两数之和。用赋值运算符“=”把相加的结果传送给 sum 并输出。

```
#include<stdio.h>
main()                /*求两个整数之和*/
{
    int a,b,sum;       /*这是声明部分，定义a,b,sum为整型变量*/
    a=1;               /*将1赋给a，从这行开始的4行是C语言语句*/
    b=2;               /*将2赋给b*/
    sum=a+b;           /*将a+b的和赋给sum*/
    printf("%d",sum);  /*输出sum的值*/
}
```



任务总结

在本任务中通过综合运用 C 语言程序的结构特点，依据 C 语言程序的运行方法，实现了简单的两个整数求和功能。



任务拓展 经典编程

请查找资料并了解函数的调用功能，求出两个整数中的较大者。



项目总结

本项目介绍了编程语言的发展及特点，并通过两个任务实例分析了 C 语言程序的格式、构成和基本要求，最后介绍了 C 语言程序的上机步骤。学习 C 语言应注意以下几个问题。

- (1) 编写程序应该规范，养成良好的程序设计风格。
- (2) C 语言是目前世界上使用较广泛的计算机程序设计语言之一，它效率高、灵



活度高、可移植性强、功能强大。

(3) 一个 C 语言程序由一个或若干个函数构成，必须且只能包含一个 main 函数。程序从 main 函数开始执行。

(4) 函数由函数首部和函数体两部分组成。在函数体内可以包括若干条语句，语句以分号结束，一行内可以写几条语句，一条语句也可以分多行写。

(5) 上机运行一个 C 语言程序需经过四个主要步骤：编辑→编译→连接→运行。



达标检测

一、填空题

1. C 语言程序的基本组成单位是_____。
2. 一个 C 语言程序必须且只能包含一个_____。
3. 一个 C 语言程序的执行是从_____开始，到_____结束。
4. 程序中要求计算机完成的操作是由函数中的 C 语言语句完成的，_____是 C 语言语句的必要组成部分。
5. 上机运行一个 C 语言程序必须经过四个主要步骤：_____→_____→_____→_____。

二、编程题

编写程序，输出以下信息：

```
* * * * *  
* Dev-C++ *  
* * * * *
```