


# 第 1 章

## 初识 C 语言

( 视频讲解：14 分钟)

本章不介绍 C 语言的具体技术，只介绍和 C 语言相关的一些内容。计算机语言有机器语言、汇编语言和高级语言三种，C 语言属于高级语言。C 语言不仅在过去非常流行，现在在编程语言的排行榜中也是稳居前三，发展前景非常可观。正是因为 C 语言有使用方便、功能强大、应用广泛、移植性高等优点，所以 C 语言不仅可以编写应用软件，而且非常适合编写系统软件。

当读者了解到 C 语言的历史和特性，就会对 C 语言产生浓厚的兴趣，从心里喜欢上这门语言。本章致力于让读者对 C 语言有个初步的认识和了解。

本章能够完成的主要范例（已掌握的在方框中打勾）

- 了解编程语言的发展史
- 认识 C 语言的发展趋势
- 掌握 C 语言的特点及应用领域
- 如何学好 C 语言
- 输出“hello world!”
- 计算 10+20



## 1.1 编程语言发展史

计算机程序设计语言，通常简称为编程语言，是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧，用来向计算机发出指令。一种计算机语言让程序员能够准确地定义计算机所需要使用的数据，并精确地定义在不同情况下所应当采取的行动。计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的历程，如图 1.1 所示。



图 1.1 计算机语言的发展历程

### 1.1.1 机器语言

机器语言是直接由二进制代码指令表达的计算机语言，指令是用 0 和 1 组成的一串代码，它们有一定的位数，并分成若干段，各段的编码表示不同的含义，例如某台计算机字长为 16 位，即有 16 个二进制数组成一条指令或其他信息。16 个 0 和 1 可组成各种排列组合，通过线路变成电信号，让计算机执行各种不同的操作。如某种计算机的指令为 1011011000000000，它表示让计算机进行一次加法操作；而指令 1011010100000000 则表示进行一次减法操作。它们的前 8 位表示操作码，而后 8 位表示地址码。从上面两条指令可以看出，它们只是在操作码中从左边第 0 位算起的第 6 和第 7 位不同，这种机型可包含 256 (2 的 8 次方) 个不同的指令。图 1.2 所示为一种存放机器语言指令的 CPU 芯片。



图 1.2 存放机器语言指令的 CPU 芯片

由于机器语言是 CPU 直接使用的语言，与人类平日使用的语言差异太大，因此被称为“低级语言”。机器语言如下所示：

```

00110100 10100001 10001000 11011001
11000101 10100100 11010101 11101010
11001001 10101100 10010001 10100001
.....
  
```

机器语言具有灵活、直接执行和速度快等特点，缺点是程序长、难记、难理解、不易查错。不同型号的计算机的机器语言是不相通的，照着一种计算机的机器指令编制的程序，不能在另一种计算机上执行。

### 1.1.2 汇编语言

机器语言虽然执行速度快，但难记、不易理解。为了提高编程效率，人们开始对机器语言进



## Note

行改进,用一些简洁的英文字母、符号串来替代一个特定指令的二进制串,比如上面提到的机器码指令“101101100000000”在汇编语言中用“ADD”代替,代表加法,机器码指令“1011010100000000”在汇编语言中用“SUB”代替,代表减法。这种用助记符描述的指令系统的语言,就是汇编语言(Assembly Language)。汇编语言相比机器语言更容易理解,例如,要计算  $c=3+5$ ,可以用几条汇编命令来实现,如:

标号	指令	说明
START	GET 3;	把 3 送进累加器 ACC 中
	ADD 5;	把累加器 ACC+5 送进累加器 ACC 中
	PUT C;	把累加器 ACC 送进 C 中
END	STOP;	停机

其中 ACC 表示累加器中的值等。

汇编语言是一种功能很强的程序设计语言,属于第二代计算机语言,也是利用计算机所有硬件特性并能直接控制硬件的语言。

使用汇编语言编写程序或阅读已经编写好的程序比起机器语言来要简单和方便多了。人们使用这种助记符编写程序后,若要计算机能够接受,还必须把编好的程序逐条翻译成二进制编码的机器语言,这个过程需要借助“汇编程序”来完成。汇编程序的功能就是把由汇编语言编写的程序翻译成机器语言程序,计算机才能执行该程序。这个翻译过程称为汇编。

汇编语言比起机器语言在很多方面都有很大的优越性,如编写容易、修改方便、阅读简单、程序清楚等,但在计算机语言系统中,仍然把汇编语言列入“低级语言”的范畴,它仍然是属于面向机器的语言,也就是说,不同的计算机可以有不同的指令集。

汇编语言的应用情况如下:

- 70%以上的系统软件是用汇编语言编写的。
- 某些快速处理、位处理、访问硬件设备等高效程序是用汇编语言编写的。
- 某些高级绘图程序、视频游戏程序是用汇编语言编写的。
- 80%的病毒来自汇编语言。

虽然汇编语言相比机器语言更熟了一步,但汇编语言与机器语言很像,都是直接面向机器的,仍然是符号形式的机器语言,在执行效率和易理解程度上都很差。

### 1.1.3 高级语言

无论是机器语言还是汇编语言,编程过程都是非常枯燥、费时的。这时人们意识到,如果采用接近于数学语言或人的自然语言,同时又不依赖于计算机硬件,并且编出的程序能在所有机器上通用,这样编程就容易多了。经过努力,第一个现代意义上的编程语言 Fortran 于 1954 年出现,因其不再过度地依赖某种特定的机器或环境,所以这种语言被叫做高级语言。到目前为止,共有几百种高级语言出现,有重要意义的有几十种。

面向电子计算机的高级语言的出现是计算机发展史上的里程碑式事件。目前的高级语言分为面向过程和面向对象两种,如下所示:

- 面向过程的高级语言



- 面向过程的高级语言，又称算法语言，如 Fortran、Basic、Pascal 和 C 等。
- 优点：灵活，易于理解，容易查错。
- 缺点：维护性较差，难以扩充、修改。
- ☑ 面向对象的高级语言
  - 面向对象的高级语言是非面向过程的语言，如 C++、Java、PHP、C#和 Delphi 等。
  - 优点：灵活，易于理解，易维护、修改和扩充。
  - 缺点：掌握难度较大。

使用高级语言编写出的可供人阅读的程序叫做源程序（Source Program），也可以称为源代码（Source Code），以文件的形式存储在磁盘上的程序通常称为源文件（Source File）。高级语言使用编译器对源程序进行编译，生成可执行文件，可执行文件的扩展名是.exe，如图 1.3 所示。

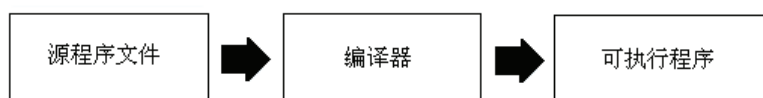


图 1.3 可执行文件的生成



Note

## 1.2 C 语言的发展

### 1.2.1 C 语言的产生

1970 年，UNIX 的开山鼻祖——美国贝尔实验室的 Ken Thompson（见图 1.4）设计出了既简单又很接近硬件的 B 语言（取 BCPL 的第一个字母），并用 B 语言写第一个 UNIX 操作系统。1972 年，Dennis Ritchie（见图 1.5）在 B 语言的基础上设计出了 C 语言（取 BCPL 的第二个字母）。C 语言既保持了 BCPL 和 B 语言的优点（精练、接近硬件），又克服了它们的缺点（过于简单、数据无类型等）。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的。



图 1.4 Ken Thompson



图 1.5 Dennis Ritchie

最初，C 语言运行于 AT&T 的多用户、多任务的 UNIX 操作系统上。后来，Ritchie 用 C 语言改写了 UNIX C 的编译程序，UNIX 操作系统的开发者 Ken Thompson 又用 C 语言成功地改写了



UNIX, 从此开创了编程史上的新篇章。UNIX 成为第一个不是用汇编语言编写的主流操作系统。

1983 年, 美国国家标准委员会 (ANSI) 对 C 语言进行了标准化, 于 1983 年颁布了第一个 C 语言草案 (83ANSI C), 后来又于 1987 年颁布了另一个 C 语言标准草案 (87ANSI C), 最新的 C 语言标准 C99 在 1999 年颁布, 并在 2000 年 3 月被 ANSI 采用。但是由于未得到主流编译器厂家的支持, C99 也并未广泛使用。

尽管 C 语言发展于大型商业机构和学术界的研究实验室, 但是当开发者们为第一台个人计算机提供 C 编译系统之后, C 语言就得以广泛传播, 为大多数程序员所接受。对 MS-DOS 操作系统来说, 系统软件和实用程序都是用 C 语言编写的。Windows 操作系统大部分也是用 C 语言编写的。C 语言程序代码的编译和运行过程如图 1.6 所示。

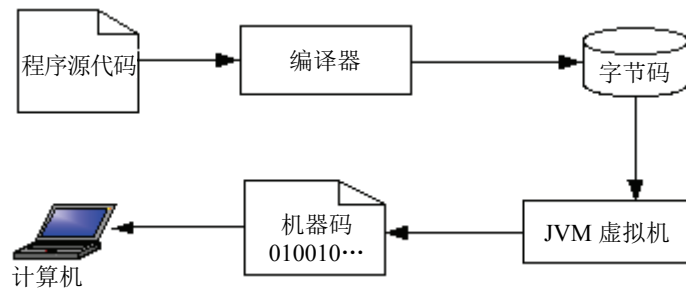


图 1.6 C 语言程序的编译和运行过程

C 语言是一种面向过程的程序设计语言, 同时具有高级语言和汇编语言的优点。C 语言可以广泛应用于不同的操作系统, 如 UNIX、MS-DOS、Microsoft Windows 及 Linux 等。本书介绍的 C 语言开发工具为 Turbo C2.0 和 Visual C++6.0, 目前最流行的 C 语言开发工具有以下 3 种, 如图 1.7 所示。

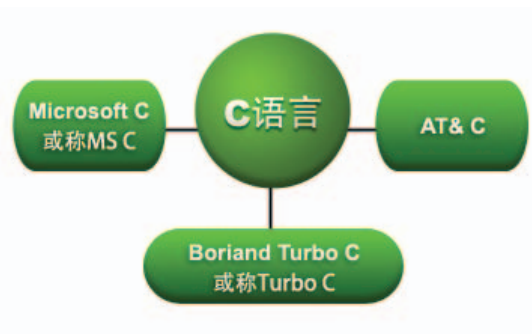


图 1.7 目前最流行的 3 种 C 语言开发工具

## 1.2.2 C 程序的特点

C 语言以其简洁、灵活、表达能力强、产生的目标代码质量高、可移植性好为其基本特点而著称于世。一种语言要具有长久的生命力, 总是有以下不同于其他语言的特点:

应用面广

- C 语言是一种面向过程的结构化程序设计语言, 既可编写系统软件 (如: DOS、



UNIX)，又可编写应用软件。

- ☑ 简洁、紧凑、方便、表达能力强
  - 32个关键字，9种控制语句，主要用小写字母。
- ☑ 运算符丰富（共有34种运算符）
  - 除了最基本的+、-、×、÷、%等运算符外，还将括号、赋值、类型强制转换等均作为运算符。
- ☑ 数据类型丰富，结构化程度高
  - 除基本类型外，还有指针，结构体、共同体等类型。
- ☑ 语法严格、灵活
  - 如数据类型可相互通用：整型、字符型通用。
- ☑ 可与机器硬件打交道
  - 直接访问内存地址，具有“高”、“低”级语言的功能。
- ☑ 生成目标代码质量高，执行效率高，语言简洁，可移植性好



#### 说明：

C语言的源文件可以分割成多个源程序，分别进行编译，然后连接起来构成可执行的目标文件，为开发大型软件提供了极大的方便。C语言还提供了多种存储属性，使数据可以按其需要在相应的作用域起作用，从而提高了程序的可靠性。



Note

### 1.2.3 C语言发展趋势

C语言是在1983年规定的标准化，颁布了第一份草案，自此以后，C语言就在编程语言中占据着龙头位置。如图1.8所示为2010年5月份的编程语言排行情况。

排名	语言	趋势	市场占有率
1	C	↑	18.058%
2	Java	↓	18.051%
3	C++	=	9.707%
4	PHP	=	9.662%
5	(Visual) Basic	=	6.392%
6	C#	↑	4.435%
7	Python	↓	4.205%
8	Perl	↑	3.553%
9	Delphi	↑↑	2.715%
10	JavaScript	↓↓	2.469%

图 1.8 2010年5月份的编程语言排行



### 1.2.4 C 语言的应用领域

因为 C 语言具有高级语言的特点，又具有汇编语言的特点，所以可以作为工作系统设计语言编写系统应用程序，也可以作为应用程序设计语言，编写不依赖计算机硬件的应用程序。它几乎可用于所有领域，如嵌入式、便携式计算机、电视、电话、手机和其他大量设备上运行。C 语言的用途数不胜数，它拥有无可比拟的能力。C 语言应用领域主要包括：

- 操作系统平台开发
- 游戏软件开发
- 嵌入式系统开发
- 桌面应用系统开发
- 交互式系统开发
- 多媒体系统开发
- 网络系统开发
- 底层接口系统开发
- 软硬件系统设备开发

C 语言无处不在，它可应用于任何地方、任何领域，并且已拥有几百万个用户，其应用范围要强于任何一种计算机语言。如果仔细观察就会发现，C 语言就在我们身边，例如我们经常使用的 Windows、Linux 操作系统等，还有各种以 C 语言编写的游戏。

### 1.2.5 如何学好 C 语言

如何学好 C 语言，这是所有初学者共同面对的问题。其实每种语言的学习方法都大同小异，需要注意的主要有以下几点：

- 明确自己的学习目标和大体方向。选择并锁定一门语言，按照自己的学习方向努力学习和认真研究。
- 初学者不要看太多的书，先找本基础书系统地学习。很多程序开发人员工作了很久也只熟悉部分基础而已，没有系统地学习 C 语言。
- 不要死记语法。在刚接触一门语言，特别是 C 语言的时候，掌握好基本语法，并大概了解一些功能即可。借助开发工具的代码辅助功能，完成代码的录入，这样可以快速地进行学习状态。
- 多实践，多思考，多请教。必须亲手编写程序代码、运行程序并分析调试。
- 不要急躁。遇到技术问题，必须冷静分析，保持耐心。
- 遇到问题，首先尝试自己解决，这样可以提高自己的程序调试能力，并对常见问题有一定的了解，明白出错的原因，甚至举一反三，解决其他关联的错误问题。
- 多查阅资料。可以经常到 Internet 上搜索相关资料或者解决问题的办法，网络上有很多别人遇到的问题及其解决办法。



- ☑ 多阅读别人的源代码。不但要看懂别人的程序代码，还要分析编程者的编程思想和设计模式，并为己所用。

虽然读者还没有学习使用开发环境和如何编写程序，但是可以在互联网上搜索一下，看看能不能查到怎样才能输出“Hello World!”。



Note

## 实例 1 输出“Hello World!”

👉 实例位置：光盘\MR\sl\01\1.1

在 VC6.0 中新建一个工程，在工程中建立一个 C 源文件，输入如下代码，按 F7 快捷键编译，按 Ctrl+F5 快捷键运行程序。

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

程序运行如图 1.9 所示。

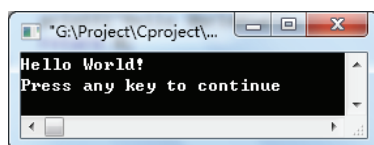


图 1.9 输出 Hello World!

## 实例 2 计算 10+20

👉 实例位置：光盘\MR\sl\01\1.2

建立工程，新建 C 源文件，编写如下代码，运行程序。

```
#include <stdio.h>
int main()
{
    int a = 10, b = 20;
    printf("10 + 20 = %d\n", a + b);
    return 0;
}
```

运行效果如图 1.10 所示。



Note



图 1.10 计算 10+20

## 1.3 本章小结

本章从 C 语言的产生背景、语言特点、发展趋势和应用领域等方面进行全面介绍。另外，本章在最后还为初学者在如何学好 C 语言方面提出了若干指导意见，相信读者读完本章，一定会爱上 C 语言。C 语言，是程序员最正确的选择。