普通高等教育“十三五”规划教材

C语言程序设计

贾志娟　主编

科学出版社

普通高等教育“十三五”规划教材

C语言程序设计

贾志娟　主编

贾遂民　张红艳　张　永　副主编

科学出版社

北京

内 容 简 介

本书是一本兼具趣味性和实用性的C语言程序设计教材，以程序设计过程为主线，以问题和案例引入内容，围绕问题的解决来讲解C语言。全书共分为11章，包括C语言概述、简单C程序设计、选择控制结构、循环控制结构、函数、数组、字符串、指针、结构体和共用体、编译预处理和文件等内容。

本书内容全面、知识点详细，既可作为普通高等学校各专业的C语言程序设计课程教材，也可作为从事计算机相关工作的人员的参考书。

图书在版编目(CIP)数据

C语言程序设计 / 贾志娟主编. —北京：科学出版社，2017.6
普通高等教育“十三五”规划教材
ISBN 978-7-03-053376-0

Ⅰ. ①C… Ⅱ. ①贾… Ⅲ. ①C语言－程序设计－高等学校－教材
Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2017)第134837号

责任编辑：于海云 / 责任校对：郭瑞芝
责任印制：霍 兵 / 封面设计：迷底书装

科学出版社出版
北京东黄城根北街16号
邮政编码：100717
http://www.sciencep.com

三河市骏走印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2017年6月第 一 版 开本：787×1092 1/16
2017年6月第一次印刷 印张：18
字数：427 000

定价：45.00元

(如有印装质量问题，我社负责调换)

前　言

本书是将编者多年的教学实践与学生在C语言程序设计学习过程中所遇到的各种问题和反馈意见相结合，编写而成的。该书力求对C语言程序设计中涉及的基本概念、基本理论、典型应用和语法规则等的表述更为详细、规范、科学和准确，在文字叙述方面也力求更加精炼、通顺，实验数据方面也做到更为准确有据。此外，本书还提供了大量的实例与习题，注重各部分知识的综合应用训练。

全书共11章，详细介绍了C语言程序设计的基本原理和方法。第1章为C语言概述，介绍C语言的背景及意义，并通过介绍简单的C语言程序建立程序设计的思想。第2章为简单C程序设计，主要介绍基本的数据类型和常用的算术运算符。第3章为选择控制结构，主要介绍如何使用if、if-else和switch语句以及条件运算符来控制程序的流程。第4章为循环控制结构，主要介绍while、do-while、for三种循环语句，并对由break、continue和goto语句控制的流程跳转语句进行了介绍。第5章为函数，介绍自顶向下、分而治之的模块化程序设计方法，通过例子介绍函数的概念、定义以及各种调用方法。第6章为数组，主要介绍了一维数组和二维数组的概念，并通过例子进行解说。第7章为字符串，介绍了字符串的概念、存储方式以及一些常用的函数，并通过例子介绍了字符串的应用。第8章为指针，介绍与指针相关的基础知识，并对指针作为参数、指针在数组和字符串以及指针的高级应用方面做了介绍。第9章为结构体和共用体，介绍了结构体、共用体、枚举以及链表的概念，并实例介绍了结构体在数组、指针及链表的应用。第10章为编译预处理，该章进一步阐述了编译预处理在C语言中的作用，并介绍了常用的编译预处理指令。第11章为文件，介绍如何使用文件将数据写入文件中以及从文件中读出。

本书由郑州师范学院的贾志娟教授提出编写思路并负责书稿的编写协调，郑州师范学院的贾遂民教授负责书稿的组织和审校工作。其中郑州师范学院的冯聪编写第1章；郑州师范学院的张玉编写第2章；郑州师范学院的张红艳编写第3章；南昌航空大学的赵靓编写第4章；郑州师范学院的马歌编写第5章；郑州师范学院的魏萌编写第6章；郑州师范学院的孔珊编写第7章；郑州师范学院的王宁编写第8章；郑州师范学院的孙陆鹏编写第9章；郑州师范学院的楚志刚编写第10章；郑州师范学院的赵靓编写第11章。南昌航空大学的蔡虹老师在本书编写过程中给予了帮助，在此一并表示感谢。

由于编者水平有限，书中难免存在不足和疏漏之处，敬请读者批评指正。

编　者

2017年6月

目　录

第 1 章　C 语言概述

C 语言是一种功能强大、简洁的语言，本章主要介绍程序设计语言，以及 C 语言的起源和发展，并剖析一个简单的 C 语言程序。读完本章，你就可以编写出第一个 C 语言程序，其实 C 语言很简单。本章主要内容如下：

- 程序设计语言的发展
- C 语言的起源和发展
- C 语言的优缺点
- 一个简单 C 语言程序

当我们拿起这本书时，心里有一个疑问："为什么要学 C 语言？"当面向对象的程序设计思想风靡全球时，我们为什么还要先学习诞生于 40 多年前的 C 语言呢？那么，让我们来了解 C 语言的前世今生，并在以后的章节中逐步体会 C 语言的奇妙之处，也许我们就会明白为什么要学 C 语言了。

1.1　程序设计语言

从 20 世纪初，物理学和电子学科学家们就在争论——制造可以进行数值计算的机器应该采用什么样的结构。人们被十进制这个人类习惯的计数方法所困扰，所以那时以研制模拟计算机的呼声更为响亮和有力。

20 世纪 30 年代中期，冯·诺依曼大胆地提出，抛弃十进制，采用二进制作为数字计算机的数制基础。同时，他还说预先编制计算程序，然后由计算机按照人们事前制定的计算顺序来执行数值计算工作。冯·诺依曼和同事们设计出了一个完整的现代计算机雏形，并确定了存储程序计算机的五大组成部分和基本工作方法。冯·诺依曼的这一设计思想被誉为计算机发展史上的里程碑，标志着计算机时代的真正开始。冯·诺依曼成功将其理论运用在计算机的设计之中，根据这一原理制造的计算机被称为冯·诺依曼结构计算机，世界上第一台冯·诺依曼结构计算机是 1949 年研制的 EDVAC(Electronic Discrete Variable Automatic Computer)，由于冯·诺依曼对现代计算机技术的突出贡献，因此他又被称为"计算机之父"，存储程序控制原理被称为冯·诺依曼原理。

1.1.1　程序的概念

"程序"一词来自于生活，通常指完成某件事务的一种既定方式和过程。在日常生活中，可以将程序看成是对一系列动作的执行过程的描述。计算机程序是人们为了让计算机解决某个问题而编写的一系列有序指令的集合。

冯·诺依曼理论的要点：数字计算机的数制采用二进制，计算机应该按照程序顺序执行。

计算机之所以采用二进制编码，因为二进制只有 0 和 1 两个数码，可以表示 0、1 两种状

态的电子器件很多，如开关的接通和断开、晶体管的导通和截止、电位电平的高与低等都可表示 0 和 1 两个数码。早期的程序员将 0、1 数字编程的程序代码打在纸带或卡片上，1 打孔，0 不打孔，再将程序通过纸带机或卡片机输入计算机，进行运算。

存储程序思想：把计算过程描述为由许多指令按一定顺序组成的程序，然后把程序和数据一起输入计算机，计算机便可自动地从一条指令转到执行另一条指令，对已存入的程序和数据处理后，输出结果。

为实现存储程序思想，计算机必须具备五大基本组成部件，包括：

(1) 输入数据和程序的输入设备；

(2) 记忆数据和程序的存储器；

(3) 完成数据加工处理的运算器；

(4) 控制程序执行的控制器；

(5) 输出处理结果的输出设备。

1.1.2 程序设计语言的发展

虽然计算机技术发展很快，但存储程序思想至今仍然是计算机的基本工作原理。自计算机诞生的那一天起，这一原理就决定了人们使用计算机的主要方式——编写程序和运行程序。科学家们一直致力于提高程序设计的自动化水平，改进用户的操作界面，提供各种开发工具、环境与平台，其目的都是为了让人们更加方便地控制计算机，可以少编程甚至不编程来使用计算机。

1. 机器语言

计算机能够直接识别和接收的二进制代码称为机器指令(machine instruction)，机器指令的集合就是机器语言(machine language)。机器语言是直接用二进制代码指令表达的计算机语言。机器语言的指令是用 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，如某台计算机字长为 16 位，即由 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。例如，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。

计算机可以直接识别机器语言，不需要进行任何翻译。每台机器的指令，其格式和代码所代表的含义都是硬性规定的，机器语言对不同型号的计算机来说一般是不同的。由于机器码是用许多二进制数表示的，用机器语言编程必然很烦琐，非常消耗精力和时间，难记忆，易弄错，并且难以检查程序和调试程序，工作效率低。因此初期只有极少数的计算机专业人员会编写计算机程序。

2. 汇编语言

为了简化使用机器语言编程的难度，人们进行了有益的改进：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，如用“ADD”代表加法，“SUB”代表减法等，这样就很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言就称为汇编语言(assembler language)或符号汇编语言(symbolic assembler language)。然而计算机是不认

识这些符号的，这就需要一个专门的程序，专门负责将这些符号翻译成二进制数的机器语言，这种翻译程序被称为汇编语言程序。

虽然汇编语言较机器语言已有很大的改进，但仍是面向机器的语言，主要缺点是：涉及太多机器资源的细节，依赖于机器硬件，移植性不好。

3. 高级语言

从最初与计算机交流的困难经历中，人们意识到，应该设计一种这样的语言，这种语言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，编出的程序能在所有机器上通用。经过努力，1954 年，第一个完全脱离机器硬件的高级语言——FORTRAN 问世了，这是程序设计语言发展史上的一个分水岭，人们把机器语言和汇编语言称为低级语言(它与计算机硬件的距离比较近，但不便于人的理解，不便于编写程序)，把以后发展起来的语言称为高级语言。

高级语言的编写方式更接近人们的思维习惯，比如可以用“+”来表示加法、用“-”表示减法，并使得编写的程序具有一定的通用性。低级语言涉及计算机硬件细节，所以不具有通用性。高级语言远离机器语言，与具体的计算机硬件关系不大，因而写出来的程序可移植性好，重用率高；要想在某一台计算机上运行用高级语言所编写的程序，该计算机只需要提供该语言的翻译系统即可。

计算机也不能直接识别高级语言程序，也要进行“翻译”，用一种称为编译程序的软件把用高级语言写的程序(称为源程序，source program)转化为机器指令的程序(称为目标程序，object program)，然后让计算机执行机器指令程序，最后得到结果。

有了高级语言后，广大计算机爱好者以及一般的科技人员、大中学生，甚至小学生，都能学习用高级语言编写程序，指挥计算机工作，不用知道机器指令，也可以不必深入学习计算机的内部构造和工作原理，就能利用计算机进行各种工作，从而为计算机的推广和普及创造了良好的条件。人们称高级语言的出现是计算机发展史上的“惊人的成就”。

目前，全世界有上千种高级语言，每种高级语言都有特定的用途，其中应用比较广泛的有上百种，语言热门程度可参看 TIOBE 排行榜。

1.2 C 语言的发展

C 语言从诞生之初就备受程序员的青睐，成为使用最为广泛的编程语言之一。目前，C 语言编译器普遍存在于各种不同的操作系统中，例如 UNIX、Microsoft Windows 及 Linux 等。C 语言的设计影响了许多后来的编程语言，例如 C++、Java、C#等。

1.2.1 C 语言的起源

20 世纪，人类有四分之三的时间在为贝尔实验室的发明欢呼，那里诞生了 7 位诺贝尔奖获得者。Ken Thompson 和 Dennis M. Ritchie 从大学毕业后就进入贝尔实验室工作直到退休，传奇的 C 语言和 UNIX 操作系统也由此诞生。

1964 年，Thompson 参与了贝尔实验室与麻省理工学院以及通用电气公司联合开发的一套多使用者分时作业系统，名叫 Multics。在开发 Multics 的期间，Thompson 创造了名为 Bon

的程序设计语言(简称 B 语言)。身为优秀的设计师，同时又是一名游戏爱好者，Thompson 设计了一款电子游戏——“Space Travel”，该游戏可执行于 Multics 之上。1969 年，贝尔实验室从 Multics 计划退出。为了能继续玩这款游戏，Thompson 只好找到一台老式 PDP-7 机器，花了一个月的时间为它开发了全新的操作系统，UNiplexed Information and Computing System(UNICS)，后来改称为 UNIX，并重写了他的“Space Travel”游戏。

UNIX 的出现开始并不为大家所看好，但是却引起了贝尔实验室另一位同事的注意，这就是 Dennis M. Ritchie，他主动加入进来，共同完善这个系统。从此一场轰轰烈烈的 UNIX 的传奇时代才真正的拉开了序幕。1972 年，他们联手将 UNIX 移植到当时最先进的大型机 PDP-2 上，由于 UNIX 是如此的简洁、稳定与高效，以至于当时大家都放弃了 PDP-2 上自带的 DEC 操作系统而完全改用 UNIX，这时的 UNIX 已经开始走向成熟。随着 UNIX 的需求量日益增加，Thompson 与 Ritchie 决定将 UNIX 进一步改写，以便可以移植到各种不同的硬件系统上。由于 UNIX 的原码中不少是用汇编完成，不具备良好的移植性，1973 年 Dennis 在 B 语言的基础上开发出了 C 语言，并用 C 语言重写了 UNIX。C 语言灵活、高效性、与硬件无关，并且不失其简洁性，正是 UNIX 移植所需要的法宝，于是 UNIX 与 C 语言完美结合在一起产生了新的可移植的 UNIX 系统。随着 UNIX 的广泛使用，C 语言成为了当时最受欢迎的编程语言并延续至今。

1983 年，因为 UNIX 和 C 语言的巨大成功，Thompson 和 Ritchie 共同获得当年度计算机界最高奖——图灵奖。

1.2.2 C 语言的发展

1978 年，Brian Kernighan 和 Dennis M. Ritchie 合著的 *The C Programming Language* 出版，成为 C 程序员必读的“圣经”，是各种 C 语言版本的基础。

随着 C 语言的不断发展扩充，1983 年，美国国家标准化协会开始制定新的 C 语言标准，1989 年完成，称为 C89 标准；到了 1995 年 C 语言又发生了一些变化，1999 年推出新标准，称为 C99 标准。2011 年，ISO 正式发布了 C 语言的新标准 C11，新的标准提高了对 C++的兼容性，并增加了一些新的特性。

无从考证究竟有多少软件是用 C 语言编写的，但很多重量级软件确实都是用 C 语言编写的。可以这么说，几乎没有不能用 C 语言实现的软件，没有不支持 C 语言的计算机系统。在世界编程语言排行榜中(www.tiobe.com)，C 总是排在数一数二的位置。图 1.1 和图 1.2 分别是 2017 年 2 月的 TIOBE 的编程语言排行情况和 2002～2016 年编程语言 TIOBE 指数走势图。

C 语言对现代编程语言有着巨大的影响，许多现代编程语言都借鉴了大量 C 的特性。在众多基于 C 的语言中，以下几种非常具有代表性。

- C++：包括了所有 C 特性，增加了类和其他特性以支持面向对象编程。
- Java：基于 C++，所以也继承了 C 的许多特性。
- C#：是综合 C++和 Java 而发展起来的一种较新的语言。
- Perl：是一种强大的脚本语言，在发展过程中采用了 C 的许多特性。
- PHP：是一种 HTML 内嵌式脚本语言，其语法混合了 C、Java 和 Perl。

编程语言排行榜　TOP20榜单

Fob 2017	Fob 2018	Change	Programming Language	Ralings	Change
1	1		Java	16.676%	–4.47%
2	2		c	8.445%	–7.15%
3	3		c++	5.429%	–1.48%
4	4		c#	4.902%	+0.50%
5	5		Pyhon	4.043%	–0.14%
6	6		PHP	3.072%	+0.30%
7	9	▲	JavaSorip	2.872%	+0.67%
8	7	▼	Visual Basic NET	2.824	+0.37%
9	10	▲	Delphi/Object Pascal	2.479%	+0.32%
10	8	▼	Perl	2.171%	–0.08%
11	11		Ruby	2.153%	+0.10%
12	16	▲	Swift	2.125%	+0.75%
13	13		Assembly language	2.107%	+0.28%
14	38	▲	Go	2.105%	+1.81%
15	17	▲	R	1.922%	+0.73%
16	12	▼	Visual Basic	1.875%	+0.02%
17	18	▲	MATLAB	1.723%	+0.63%
18	19	▲	PL/SQL	1.549%	+0.49%
19	14	▼	Objective-C	1.538%	+0.13%
20	23	▲	Scratch	1.500%	+0.71%

图 1.1　2017 年 2 月 TIBOE 的编程语言排行情况

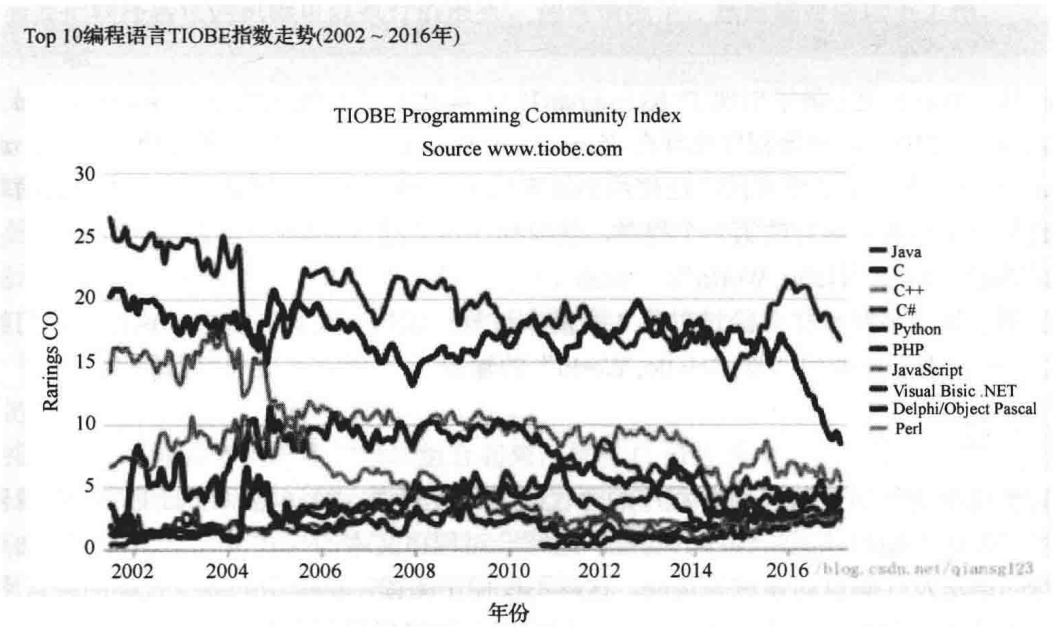

图 1.2　2002～2016 年编程语言 TIBOE 指数走势图

1.2.3 C 语言的特点

C 语言既有高级语言的特点，又具有汇编语言的特点。发明 C 语言是为了编写以往由汇编语言编写的应用程序，因此能够在有限的内存空间里快速运行就显得至关重要。

C 语言拥有一个庞大的数据类型和运算符集合，这个集合使得 C 语言具有强大的表达能力，寥寥几行代码往往就可以实现许多功能。

C 语言是一种包容性语言。C 语言不像其他语言那样为减少程序员犯错，提供了太多范式来约束程序员，C 语言假设用户知道自己在做什么，这种信任给程序员带来了自由，他们拥有最大的发挥空间，可以自由地编写代码。这些精心设计的代码运行效率高，可以极大地节约资源。可是从另一方面讲，这使得 C 程序更容易隐藏错误，C 的灵活性导致用它编程出错概率高，在用其他语言编程时可以发现的错误，C 编译器却不加限制。如果程序员不自我约束，代码将会存在隐患，因为它不安全、不稳定、不易于维护。

优缺点经常是同源的，C 语言更是如此，其优缺点主要来自于 C 语言与硬件的紧密结合及其赋予程序员的自由空间。C 语言中那些容易导致初学者出错的特性，往往也正是吸引编程老手们的特性。

C 语言是编写操作系统的最好选择。因为它能直接与计算机底层打交道，精巧、灵活、高效。也正因为它的这种特性，在设计对运行效率要求较高的系统，比如设备驱动程序、高性能实时中间件、嵌入式领域、并发程序设计等时，C 语言也是首选。在需要继承和维护已有的 C 代码的地方，还需要 C 语言。在设计编程能力的考试环节，通常考的都是 C 语言。

1.3 C 程序初识

我们将从一个最简单的 C 语言程序开始，建立 C 语言程序的基本概念。我们的任务是编写一个 C 语言程序，输出如下一行文字：

```
Hello, World
```

Hello World 程序最早出现于 Kernighan1972 年编写的内部技术文档 *Introduction to the Language B* 之中，后来该程序出现在 Kernighan 和 Ritchie 的经典名著 *The C Programming Language* 一书中。因 C 语言的广泛使用而逐渐成为各种程序设计语言中最基本、最简单的程序，通常是初学者所编写的第一个程序，就像程序员们的“初恋”。

试试看：输出“Hello, World”。

C 程序从编写到运行要经过以下 4 个基本过程：编辑、编译、链接、执行。我们就从这 4 个过程一一来分析如何实现“Hello, World”的输出。

1.3.1 编辑

编辑过程就是创建和修改 C 程序的源代码。一般来说，编译器系统如果带有编辑器，就会提供很多便于编写和组织程序的功能。通常会对程序文本的格式进行自动排版，如高亮显示特殊的语法及代码自动缩进等功能，这样不仅便于阅读，也能帮助减少代码的错误率。

也可使用其他编辑器来创建源文件，但它们必须将代码保存为纯文本，而没有嵌入附件的格式数据。比如微软的记事本可以用来编写程序代码，但是 Word 就不适合编写程序代码。

在屏幕上输出“Hello, World”。

【例 1.1】 程序代码：

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

打开编辑器，输入以上程序。注意不要忘了圆括号()、花括号{}、双引号""、分号等特殊符号。1.4 节会对程序的要素进行详细说明，这里仅做简要介绍，程序第 1 行如下：

```
#include <stdio.h>
```

这是必不可少的，包含了 C 语言标准输入输出库函数的相关信息。每个 C 程序都由一个或多个函数构成，其中必须有一个 main()——因为每个程序总是从这个函数开始执行的。一对花括号内的代码块，称为函数体，它包含了定义函数功能的所有语句。这个例子中的 main()函数体非常简单：

```
printf("Hello, World\n");
```

printf 函数用来显示期望信息“Hello, World”。“\n”表示显示信息后要进行换行操作。

```
return 0;
```

表明程序终止时会向操作系统返回 0。

1.3.2 编译、链接和执行

我们编写了程序 1.1，并生成了一个名为 hello.c 的文件(C 程序的文件扩展名为.c)。接下来，就需要把程序转化为机器可以执行的形式。通常情况下，系统需要做以下工作。

1)预处理

首先程序会被交给预处理器。预处理器处理以#开头的命令，比如程序 1.1 中的#include <stdio.h>。

2)编译

预处理器处理后的程序就可以进行编译了，编译器会把源代码转换成机器语言(目标代码)。编译器可以在转换的过程中发现并报告错误。编译阶段出现错误，意味着必须重新编辑源代码。反之，如果编译成功，就会产生一个目标文件，该文件与源文件同名，但扩展名是.o 或.obj。但是，这时的程序还不能运行。

3)链接

链接器把由编译器产生的各种模块组合起来，再从 C 语言提供的库函数中添加必要的代码模块，将它们组合成一个可执行文件。链接器也可以检查和报告错误，例如遗漏了程序的某个部分，或者引用了一个根本不存在的库函数等。链接阶段出现错误，也意味着必须重新编辑源代码。如果链接成功，就会产生一个可执行文件(扩展名为.exe)。

4)执行

通过链接得到可执行文件。在大多数 IDE 中，都有一个相应的命令菜单，来运行编译、