KH北京科海
25年·科技图书出版专家

21世纪大学计算机系列教材

C语言程序设计

夏玮 潘钧 主编
李静 刘莹 副主编

中国人民大学出版社
北京科海电子出版社
www.khp.com.cn

21世纪大学计算机系列教材

C语言程序设计

夏 玮 潘 钧 主 编
李 静 刘 莹 副主编

中国人民大学出版社
·北京·

北京科海电子出版社
www.khp.com.cn

图书在版编目(CIP)数据

C 语言程序设计/夏玮，潘钧主编.
北京：中国人民大学出版社，2009
（21 世纪大学计算机系列教材）
ISBN 978-7-300-11030-1

Ⅰ. C…
Ⅱ. ①夏… ②潘…
Ⅲ. C 语言—程序设计—高等学校—教材
Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2009）第 128949 号

21 世纪大学计算机系列教材
C 语言程序设计
夏玮　潘钧　主编

出版发行	中国人民大学出版社　北京科海电子出版社		
社　　址	北京中关村大街 31 号	**邮政编码**	100080
	北京市海淀区上地七街国际创业园 2 号楼 14 层	**邮政编码**	100085
电　　话	（010）82896594　62630320		
网　　址	http: //www.crup.com.cn		
	http: //www.khp.com.cn（科海图书服务网站）		
经　　销	新华书店		
印　　刷	北京市鑫山源印刷有限公司		
规　　格	185 mm×260 mm　16 开本	**版　　次**	2009 年 10 月第 1 版
印　　张	20	**印　　次**	2009 年 10 月第 1 次印刷
字　　数	547 000	**定　　价**	29.80 元

内容提要

C 语言使用灵活、可移植性好，是绝大部分程序设计人员和计算机爱好者学习程序设计的首选语言。本书共分为 11 章，内容包括：C 语言概述，数据类型、运算符与表达式，顺序结构程序设计，选择结构程序设计，循环结构程序设计，数组，函数，指针，结构体与共用体，文件等内容，最后一章是实验，有助于提高读者在实际应用中的编程能力。

全书阐述清晰、层次分明、通俗易懂，完整描述了 C 语言及其语法特性。本书的一个鲜明特色是结合大量示例阐明了 C 语言结构的正确使用和语法，可操作性强，便于读者掌握并应用到实际工作中去。

本书适合作为各类高等学校、计算机培训学校等相关专业的教材，也可以作为程序设计爱好者的参考用书。

前　言

C 语言的功能非常强大。它使用灵活，可移植性好，既具有高级语言的优点，又可以实现低级语言的许多功能；既可以编写系统软件，也可以编写应用软件。因此，C 语言是绝大部分程序设计人员和计算机爱好者学习程序设计的首选语言。

本书的目的是使读者轻松、愉快地完成 C 语言程序设计课程的学习，使其不仅能掌握一门编程语言，同时也能掌握一种编程思想。全书采用循序渐进的方式推进讲解，由浅入深地对关键知识点进行分析，并辅以大量具有代表性和实用性的案例，使读者在掌握 C 语言程序设计的同时，也为以后学习数据结构、编译原理等课程打下良好的基础。

本书导读

1. 第 1 章首先介绍了 C 语言发展简史、 C 语言的构成和书写规则及 C 语言的运行环境，最后简单介绍了 C 语言的算法及其特性，引导读者进入 C 语言程序设计的学习。

2. 第 2 章主要介绍了 C 语言的基本数据类型、运算符与表达式。

3. 第 3 章~第 5 章分别介绍了 C 语言的顺序结构、选择结构和循环结构三种控制结构程序设计。

4. 第 6 章~第 10 章介绍了 C 语言的应用，如数组、函数、指针、结构体与共用体、文件等。其中，函数的使用最能代表 C 语言结构化的程序设计思想，是学习的重点之一。指针是 C 语言的精华，是 C 语言程序设计的典型特性之一，也是学习的难点，正是指针使 C 语言既具有高级语言的特点，又可以完成低级语言的功能。

5. 本书第 11 章是实验。该章基于 Turbo C 2.0 平台，对书中必须掌握的知识都要求学生上机操作，在实践中轻松掌握。这些上机指导有助于提高读者在实际应用中的编程能力，增强学习 C 语言编程的信心。

本书特点

1. 力求程序的正确性。本书从语法和程序结构两方面都力求规范，并且所有例题均在 Turbo C 2.0 上调试通过。

2. 易教易学。本书在内容编排上按照循序渐进、由浅入深的原则，突出重点内容，并运用文字、图表、小实例等相结合的方式，将抽象概念形象化，便于读者学习和掌握。

3. 内容新颖富有启发性。本教材在讲解程序设计的例题时，不是简单“教”读者用程序解决问题的方法，而是启发读者去分析解决问题的过程，从中找出可以归结为规则操作的方法，进而编写程序。在实例部分对已有的方法提出改进的可能，启发有能力的读者深入学习，旨在培养读者勤于思考的习惯。

4. 注重培养读者的应用能力。大量的实例是本书的一大特色，本书通过设置独立的上机实验指导以及综合应用实例来训练、提高读者在实际工作中的应用能力。

由于时间仓促，加之编者水平有限，书中不足之处在所难免，恳请广大读者不吝指正。

编 者

2009 年 8 月

目　录

第1章

C语言概述

C语言是国际上广泛流行的计算机高级程序设计语言，它集高级语言和低级语言的功能于一体，既可用于系统软件的开发，也适用于应用软件的开发。同时它还具有效率高和可移植性强等特点，因此被称为当代最优秀的程序设计语言。

本章主要内容

- C语言出现的历史背景
- C语言的特点
- C程序的结构
- C程序的上机执行过程
- 算法的概念、特点及表示方法
- 结构化程序设计方法

本章重点

- 掌握C程序的结构及上机执行过程
- 掌握算法的概念、特点及表示方法
- 掌握结构化程序设计方法

1.1 C语言的发展简史

从计算机诞生到今天，伴随着计算机技术的飞速发展，程序设计语言也在不断升级换代，主要经历了面向机器（机器语言和汇编语言）、面向过程（高级语言）和面向对象（高级语言）几个阶段。

1.1.1 程序设计语言

计算机的硬件提供了对数据进行计算的可能性，要使计算机按照人们的意图完成一项任务，就必须向它发出命令。能使计算机动作的命令叫指令。若干条指令组成程序。把解决一项任务的思路、方法和步骤最终落实为计算机程序的过程就是程序设计。用于书写计算机程序的语言叫程序设计语言，它是人与计算机之间进行信息交流的工具。

1. 机器语言

在计算机问世的初期，只能用二进制的0、1代码组成的指令来编写程序，这就是“机器语言”。它是CPU可以识别的一组由0和1序列构成的指令码，要求程序员对他所使用的计算机的硬件结构及其指令系统十分熟悉，因为编写的程序非常烦琐，操作过程也极易出错。其优点是程序可以直接在计算机上运行。

下面是8080 CPU要完成2+3操作的机器指令。

```
1011000000000010
1011001100000011
0000000011011000
```

要想看懂这段程序，必须熟悉Intel 8080 CPU的指令格式。显然，用0、1代码编程使人难理解、难辨认、难记忆、难排错，只能为少数专业人员所掌握，而且若CPU不同，则指令系统不同，导致所编程序根本无法移植。这种繁重的手工方式与快速发展的计算机硬件极不相称。

2. 汇编语言

为了减轻用机器语言编程的劳动强度，人们用一些简单而又形象的符号来代替每一条具体的机器指令，这就形成了“符号语言”。在此基础上，把一些子程序、寄存器等也用符号来表示，即为“汇编语言”。上面3条机器指令对应的汇编指令如下。

```
MOV a1,2
MOV b1,3
ADD a1,b1
```

这种程序可让人有“望文知义”的感觉，记忆起来相对容易。但是，计算机能直接识别和执行的只能是机器代码，它并不认识这些助记符号。必须先将汇编语言翻译成机器语言，才能被计算机接受并自动运行，该过程是由计算机系统软件中的汇编程序完成的，如图1-1所示。

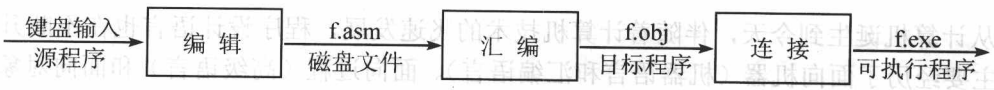

图1-1 汇编语言翻译过程

显然，与机器语言相比，用汇编语言编程在一定程度上减轻了编程的工作量，程序的编写效率大大提高。但是，由于汇编语言指令和机器语言指令几乎是一一对应的，因此对计算机硬件的依赖并无改观。机器语言和汇编语言统称为面向机器的语言，或低级语言。

利用面向机器的语言，可以编出质量极高的程序，但是在编程时，不仅要考虑解题思路，还要熟悉与计算机硬件密切相关的指令系统，甚至要手工分配存储器，特别是对于复杂的科学计算和大量的高精度浮点数的处理更为困难。对外设的使用也很麻烦，需要知道硬件的端口地址及其操作顺序（先送控制码，再送信息码等）。人们急切盼望着形如a=sin(x)+2cos(x)的接近自然语言和数学算法的新型语言出现。

3. 高级语言

为了避开具体的计算机，用一些符号来描述解题意图，尽量接近于数学公式的自然描述，使书写的程序能够通过各类计算机对应的翻译程序，即可在各类计算机上运行，于是出现了高级语言。世界上第 1 种高级语言是 1954 年出现的 FORTRAN 语言，主要用于科学计算。随后在 1960 年出现了算法语言 ALGOL（体现了程序的嵌套结构）和 COBOL（主要用于数据处理），1964 年出现了 BASIC 语言（主要面向初学者），1971 年出现了 Pascal 语言（第 1 个结构化的程序设计语言），1972 年以后出现了 C 语言。经历了残酷的优胜劣汰过程，最后保留下来并继续被普遍使用的是那些比较优秀的高级语言，C 语言便是其中之一。

用 BASIC 语言完成 2+3 计算的程序如下：

```
A=2
B=3
A=A+B
END
```

显然，用高级语言描述问题更接近人们的习惯，且具有通用性。不必了解实际计算机的机型、内部结构及其 CPU 的指令系统，只要掌握某种高级语言本身所规定的语法和语义，便可直接用该语言来编程，这样做大幅度降低了编程的劳动强度，提高了编程效率。当然，计算机也不能直接识别和执行用高级语言编写的程序，必须将高级语言翻译成机器语言后，才能被计算机接受并运行。这个翻译过程是由计算机系统软件中的翻译程序完成的。翻译程序有两种：解释程序和编译程序。编译程序是先编译，后执行；解释程序是边解释，边执行，如图 1-2 所示。

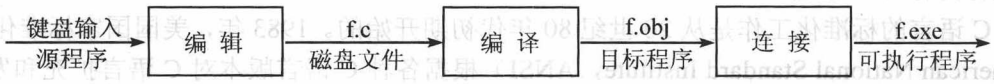

图 1-2 高级语言的编译过程

前些年，因为计算机硬件成本和使用费用较高，程序设计的主要目标是力求编写出代码短、速度快的程序。近年来，计算机变得体积越来越小、速度越来越快、容量越来越大、价格越来越低，而开发、维护程序的费用却日趋上升，于是程序设计的目标也随之发生了变化，即在程序正确的前提下，提高程序的可读性、易维护性、可移植性。

高级语言的结构化程序设计思想在过去的几十年里一直是程序开发的主流思想，但它毕竟是面向过程的，当解决一个实际问题时，需要把数据结构及其算法步骤详细告诉计算机，即不仅要考虑“做什么”，还要考虑“怎么做”，这种数据和方法分离的现象使程序代码的可重用性差、程序维护的一致性差。现在流行的面向对象的高级语言，如 Visual C++、Visual Basic、Java 等设计的一个关键是定义“类”，由“类”再生成“对象”。“类”的封装性可将数据和方法“打包”到一起，保证了程序维护的一致性；“类”的继承性减轻了编程负担，保证了代码的可重用性；“类”的多态性则体现了程序的灵活性，同一消息被不同对象接收将产生不同的动作。

1.1.2 C 语言的发展

在 C 语言产生之前，人们编写系统软件主要是使用汇编语言。由于用汇编语言编写的程序依赖于计算机硬件，其可读性和可移植性都比较差。而一般高级语言又不具备低级语言能够直观地对硬件实现控制和操作、程序执行速度快的特点。在这种情况下，人们迫切需要一种既有一般高级语言特性，又有低级语言特性的语言。于是 C 语言就应运而生了。

1. C 语言的发展历史

对 C 语言的研究起源于系统程序设计的深入研究和发展。1967 年，英国剑桥大学的 M. Richards 在 CPL（Combined Programming Language）语言的基础上，实现并推出了 BCPL（Basic Combined Programming Language）语言。

1970 年，美国贝尔实验室的 K.Thompson 以 BCPL 语言为基础，设计了一种类似于 BCPL 的语言，称为 B 语言。他用 B 语言在 PDP-7 机上实现了第一个实验性的 UNIX 操作系统。

1972 年，贝尔实验室的 Dennis M. Ritchie 为克服 B 语言的诸多不足，在 B 语言的基础上重新设计了一种语言，由于是 B 语言的后继，故称为 C 语言。1973 年，贝尔实验室的 K. Thompson 和 Dennis M. Ritchie 合作，首先用 C 语言重新改写了 UNIX 操作系统，在当时的 PDP-11 计算机上运行。此后，C 语言作为 UNIX 操作系统上标准的系统开发语言，伴随着 UNIX 操作系统的发展，越来越广泛地被人们接受和应用并被移植到其他计算机系统。

1978 年，Brian W. Kernighan 和 Dennis M. Ritchie（K&R）正式出版了著名的《The C Programming Language》一书，此书中介绍的 C 语言成为后来广泛使用的 C 语言版本基础，它被称为标准 C 语言。

C 语言的标准化工作是从 20 世纪 80 年代初期开始的。1983 年，美国国家标准化协会（American National Standard Institute，ANSI）根据各种 C 语言版本对 C 语言扩充和发展，颁布了 C 语言的新标准 ANSI C。ANSI C 比标准 C 有了很大的扩充和发展。

由于 C 语言的不断发展，1987 年，美国国家标准化协会在综合各种 C 语言版本的基础上，又颁布新标准，为了与标准 ANSI C 区别，称为 87 ANSI C。1990 年，国际标准化组织 ISO 接受了 87 ANSI C 作为 ISO C 的标准。这是目前功能最完善、性能最优良的 C 语言新版本，目前流行的 C 语言编译系统都是以它为基础的。本书讲述的内容基本上是以 ANSI C 为基础的，并参考 87 ANSI C。

2. C 语言是当代最优秀的程序设计语言

早期的 C 语言主要是用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们认识，到了 20 世纪 80 年代，C 语言开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到广泛的使用，成为当代最优秀的程序设计语言之一。

从 C 语言的发展历史可以看出，C 语言是一种既具有一般高级语言特性（ALGOL 60 带来的高级语言特性），又具有低级语言特性（BCPL 带来的接近硬件的低级语言特性）的程序设计语言。C 语言从一开始就是用于编写大型、复杂系统软件的，当然 C 语言也可以用来编写一般的应用程序。也就是说，C 语言是程序员的语言！

3. 目前在我国 PC 系列兼容机上常用的 C 语言版本

目前在我国 PC 系列兼容机上常用的 C 语言版本有以下几种。

- **Borland 公司**：Turbo C（V2.0，V3.0）、Turbo C++、Borland C++和 C++Builder（Windows 版本）。
- **Microsoft 公司**：Microsoft C（V5.0，V6.0，V7.0）和 Visual C++（Windows 版本）。
- **Computer Innovasions 公司**：C 86（V2.3）。
- **Lattice 公司**：Lattice C（V4.0）等。

1.2 C 语言的特点

C 语言是一种“中级语言”，既具有高级语言的特点又具有低级语言的特点，既适合于开发系统软件又适合于编写应用程序，被广泛应用于事务处理、科学计算、工业控制、数据库技术等领域。概括起来，C 语言主要具有以下几个特点。

1. 语言简洁、紧凑，使用方便、灵活

C 语言只有 32 个关键字和 9 种控制语句，书写紧凑，压缩了一切不必要的程序组成成分。

C 语言程序书写自由，可以一行一个语句，也可以一行多个语句，同时还可使用任何用户自己熟悉的文本编辑器来输入源程序，语法限制不太严格，很多语句都有多种书写形式，程序设计自由度大。

C 语言的 32 个关键字如表 1-1 所示。

表 1-1 C 语言关键字

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

2. 数据结构丰富

C 语言的基本数据类型有整型、实型、字符型等，在此基础上还可创建数组、指针、结构体和共用体等复杂数据类型。C 语言的灵活性和应用能力优于其他计算机语言，如使用指针，可以作为函数的参数传递、动态分配内存空间、简化数组处理等。

使用 C 语言可以很方便地实现复杂的数据结构（如链表、树、栈等运算），丰富的数据结构极大地增强了 C 语言的处理功能。

3. 运算符丰富

C 语言共提供 34 种运算符，按优先级大小划分为 15 个等级，其中包括一些 C 语言特有的运算符，如自增/自减运算符、逗号运算符、求字节运算符、强制类型转换运算符、赋值运算符等。使用这些运算符，可以使 C 语言实现在其他高级语言中难以实现的运算。

4. 结构化的语言

C 语言具有结构化的控制语句，如 if-else 语句、while 语句、switch 语句、for 语句等。使用这些语句，可以方便地控制程序流程，实现顺序、选择、循环三种结构化程序的基本结构。

5. 模块化的语言

开发一个较大的程序，需要很多人经过较长时间的努力才能完成。一般来说，一个较大的应用程序往往被分为若干个模块，对于较大的模块还可以细分为较小的模块，每个模块对应一个函数或过程，实现特定的功能。相同功能的模块可用同一函数或过程来实现，因此用函数或过程来实现程序的模块化，可以大大减少工作量。只要善于利用函数或过程，就可提高编程效率。

C 语言是模块化的语言，它以函数作为程序的基本单位。在进行 C 语言程序设计时，可将一些常用的功能模块编写成函数，放在函数库中供其他函数调用。

6. 程序可移植性好、代码执行效率高

汇编语言一般要与某种机器硬件对应，难以移植，而 C 语言在不同机器上的编译程序大约有 80%的代码是公共的，可以很方便地移植到各种型号的计算机和各种操作系统中。

同时，C 语言程序所生成的目标代码的质量高于其他高级语言，执行效率高，一般只比汇编语言程序所生成的目标代码的执行效率低 10%～20%。

C 语言的优点很多，但也存在一些缺点和不足。例如，运算符优先级太多，尤其是有个别运算符的优先级还与我们常规约定的有所不同，不便记忆和掌握；自由转换虽然比较方便，但类型检查不严，增加了不安全的因素；C 语言使用灵活、变化多样，也增加了学习的难度，初学者较难掌握。

1.3 C 语言程序的构成和书写规则

一个 C 程序可以由若干个扩展名为.c 的源文件组成（通常只含一个.c 文件），C 语言是

以.c 文件为单位进行编译的。编译后，一个.c 文件会生成一个扩展名为.obj 的目标代码文件。.c 文件主要由若干个函数组成（至少包含一个函数）。C 语言的函数可以理解为满足一定格式的、能完成特定功能的一段独立程序，它相当于其他语言中的子程序，所以可以是系统提供的库函数，也可以是用户自己编制的函数。

1.3.1 一个 C 语言示范程序

用 C 语言语句编写的程序称为 C 程序或 C 源程序。本节从两个简单的 C 程序中分析 C 程序的特性。

【边学边练】 一个简单的 C 语言程序。

在屏幕上显示“Welcome to the C world!”语句，并且显示一个由“*”号组成的大写字母 C。

实现此功能的 C 语言程序代码如下：

```
#include <stdio.h>                          /*文件包含*/
main( )                                     /*定义主函数*/
{
    printf("Welcome to  the C  world!\n");      /*输出欢迎语句*/
    printf("****\n");                           /*以下几行输出大写字母 C*/
    printf("*\n");
    printf("*\n");
    printf("****\n");
}
```

说明：

① main 表示“主函数”。每个 C 语言程序都必须有一个 main 函数，它是每一个 C 语言程序的执行起始点（入口点）。main()表示“主函数”main 的函数头。

② 用{ }括起来的是“主函数”main 的函数体。main 函数中的所有操作（语句）都在这一对{ }之间，即 main 函数的所有操作都在 main 函数体中。

③ printf 是 C 语言的输出函数，功能是用于程序的输出（显示在屏幕上），双引号内的字符串原样输出。“\n”是换行符，即在输出完“Welcome to the C world!”或“*”后回车换行。

④ 每条语句用“;”号结束。

⑤ /*……*/括起来的部分是一段注释。注释只是为了改善程序的可读性，在编译、运行时不起作用（事实上编译时会跳过注释，目标代码中不会包含注释）。注释可以放在程序任何位置，并允许占用多行，只是需要注意“/*”和“*/”匹配。

1.3.2 C 语言程序的构成

通过【边学边练】可以看出 C 程序的基本结构。

1. C 程序由函数构成

一个 C 源程序至少包含一个 main 函数，也可以包含一个 main 函数和若干个其他函数。函数是 C 程序的基本单位。

被调用的函数可以是系统提供的库函数，也可以是用户根据需要自己编写设计的函数。C 是函数式的语言，程序的全部工作都是由各个函数完成。编写 C 程序就是编写一个个函数。

2. main 函数是每个程序执行的起始点

一个 C 程序总是从 main 函数开始执行，而不管 main 函数在程序中的位置。可以将 main 函数放在整个程序的最前面，也可以放在整个程序的最后，或者放在其他函数之间。

3. 一个函数由函数首部和函数体两部分组成

（1）函数首部是一个函数的第一行。

例如：

```
int max (int x, int y)
```

就是函数首部。

（2）函数体是函数首部下用一对{}括起来的部分。

例如：

```
{
    int z;
    if(x>y)  z=x;
    else z=y;
    return z;
}
```

就是函数 max 的函数体。

如果函数体内有多个{}，最外层是函数体的范围。函数体一般包括声明部分和执行部分两部分。

声明部分：在这部分定义本函数所使用的变量。

执行部分：由若干条语句组成的命令序列（可以在其中调用其他函数）。

4. C 程序书写格式自由

一行可以写几个语句，一个语句也可以写在多行上。

程序没有行号，也没有 FORTRAN、COBOL 那样严格规定书写格式（语句必须从某一列开始）。

每条语句的最后必须有一个分号“;”表示语句的结束。

5. 可以使用/*……*/对 C 程序中的任何部分作注释

注释可以提高程序可读性，使用注释是编程人员的良好习惯。