OSBORNE

More Than 300,000 Copies Sold

C:The Complete Reference,Third Edition

〔美〕Herbert Schildt 著
王子恢 戴健鹏 等译
刘德贵 校

The One "Must Have"
Book on C —
A Comprehensive
Desktop Reference
Covers ANSI C

最新C语言精华
（第三版）

McGraw-Hill
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL:http://www.phei.co.cn

C: The Complete Reference, Third Edition

最新C语言精华

（第三版）

〔美〕Herbert Schildt 著
王子恢 戴健鹏 等译
刘德贵 校

電子工業出版社

内 容 简 介

《最新C语言精华(第三版)》是根据国际著名的C语言专家Herbert Schildt的有关新著翻译的。这是一本C语言的百科全书,包括C的命令、功能、编程和应用等方面的内容,是集专家及C语言编程人员的多方面专门经验之作。全书分五大部分,共28章。第一部分详细讨论了关键字、预处理、指令和C语言特性,内容包括表达式、程序控制语句、数组和串、指针、函数、数据类型、结构、联合、枚举和用户自定义类型、控制台I/O、文件I/O和预处理等;第二部分详细介绍了C标准程序库,包括K&R(UNIX)、ANSI、DOS环境下的I/O函数、串和字符函数、数学函数、系统函数、动态分配图形函数、杂函数、DOS调用等;第三部分介绍通用算法和应用程序,包括排序和搜索、队列、堆栈、链表和树、稀疏数组、表达式剖析和求值、人工智能问题求解及Windows 95框架的构造;第四部分用实例讨论了C软件的开发技术,包括汇编语言程序接口、效率、移植和调试等软件工程专题;第五部分介绍C解释程序。

本书内容全面,叙述清晰,为PC机及其广大用户和程序开发人员提供了一部很有用的工具书,是计算机应用人员、有关大专院校师生及PC机软件开发人员的必备参考书。

C: The Complete Reference, Third Edition
[美] Herbert Schildt 著
Osborne McGraw-Hill 1995年出版

最新C语言精华(第三版)

王子恢 戴健鹏 等译
刘德贵 校
责任编辑 路 石
特约编辑 王子恢

*

电子工业出版社出版(邮编100036,北京市173信箱)
电子工业出版社发行 各地新华书店经销
保定市印刷发行公司印刷厂印刷

*

开本:787×1092毫米 1/16 印张:36 字数:876千字
1997年2月第一版 1997年2月第一次印刷
印数:5000册 定价:58.00元
ISBN 7-5053-3974-5/TP.1730
著作权合同登记号图字:01-96-0155

译者的话

在生产和社会实践中，人们需要劳动的工具和交流的媒介。在计算机软件领域里，C 语言就是软件人员实现思想和经验交流的理想手段。

在过去几年的科研、开发、学习和交流中，我们在 DOS、Windows、NetWare、VMS 和 UNIX等环境下，使用 Sun、DEC、AT&T 和 PC 等机器，进行操作系统、数据库系统、编译程序和网络等系统软件方面的工作，也进行管理信息系统、人工智能、CAD 和实用工具等应用软件方面的工作。这些工作都是与 C 语言有关或基于 C 语言的。

由于 C 语言，我们在工作中得心应手，无论在理论上或技术上，都颇有收获。作为有经验的软件人员，我们由衷地感到，C 确实是在计算机上实现人们思想的基本手段，更是软件同行之间交流经验的通用语言。

《最新 C 语言精华(第三版)》把国际著名 C 语言专家 Herbert Schildt 的新作《C: The Complete Reference, Third Edition》介绍给广大读者，也体现我们的认识和经验，使一切尽在不言之中吧。

工作中，我们使用过 Watcom C，Borland C++，Microsoft C++，多种数据库的内嵌 SQL 的 C，还有 UNIX C 和几种操作系统提供的 C 语言环境。在实践中，本书都适于工作的许多方面。我们体会到，这本书确实是通用的入门书，同时又是有效的的教科书和实用的参考手册，各层次的读者都将通过它而受益匪浅。

最后，译者感谢自始至终关心本书的刘德贵老师，并向路石编审致意。

参加本书翻译的还有李波、张蔚、崔娜、张烨、李峻、王义、李瀛、武助会、田洪志、韩飞麟、陈善荣、方在庆、范振国等。

译者　1996 年 10 月于北京

前　言

本书是《C：The Complete Reference》的第三版。在第一版问世以来的几年中，程序设计领域已经发生了很大变化。譬如，在编写第一版时，C 语言尚未标准化，许多竞争的工具都在努力争取主导地位，各个编译程序之间彼此略有差异。在着手准备第二版之际，ANSI C 标准刚刚被采纳。而今，使用标准的 C 代码简直就是一种习惯（而且，通过标准化可以完全实现真正的可移植 C 代码）。在本书第一版写成之际，DOS 是拥有强大用户基础的唯一操作系统，而 Windows 不过是一种稀奇的尚未出现生机的事物。今天，Windows 和 DOS 环境之多超乎任何想象。在本书第一版出版之际，C++还是试验性语言，仅仅吸引了为数不多的程序设计人员的注意，他们试图扩展程序设计的限制。在改编好第二版之时，C++已基本定型，但面向对象的范例仍未得到广泛接受或使用。在改写第三版之际，C++已成为计算机语言的主流，并且是 C 最有竞争力的对手。

目前，C 语言既标准又成熟，而且还是世界上最普及的专业程序设计语言。不仅如此，广泛使用 C 语言的环境还在继续扩展，例如，为 DOS、Windows、OS/2 及 UNIX 等环境生成的应用程序所选择的语言即是 C 语言。

必须承认，在本人编写《C：The Complete Reference》第一版时，并没有预料到后来所发生的所有变化与改进（譬如 C++的成功，尽管从一开始就很明显）。然而，无论当时还是现在，我始终认为：C 语言是我所遇到的最杰出的语言。它优美、稚致、连贯且（更重要的是）功能强大。我始终如一地喜爱 C 语言即源于它的不断成功。

读者对象

本书是为经验和水平各不相同的所有程序设计人员编写的，当然读者至少应能编制简单的 C 语言程序。对正在学习 C 语言的读者而言，这本书正是任何 C 语言教程的绝好配套教材，能够回答各种有关的问题。

由于 C++（C 的面向对象的改进型）是建立在 C 的基础上的，所以，本书还适合希望详细了解 C++构造基础的 C++程序员来阅读。

因此，无论读者用 C 语言编程还是用 C++语言编程，无论读者是程序设计的初学者还是成熟的专业人员，均会发现本书很有价值。

第三版的特点

本书绝大部分保留了前两版的基本结构。由于 C 语言是稳定的、标准化的语言，所以没有理由进行任何较大的改变。本书第三版的主要变化反映两个方面：使用 C 语言的环境的改变以及作为主流语言的 C++的出现。

在本书第二版的 29 章中，有 26 章在第三版中没有什么改变，另外 3 章已经替换了新内容。取消了第二版的低级的基于 DOS 的接口技术、基于 DOS 的图形函数以及 C++概貌，新增加了 Windows 95 程序设计和 C 解释程序。自从本书第二版问世以来，Windows 已从相对无名转变为事实上的主力环境，而 C 语言又是 Windows 程序设计最通用的语言，因此，将这

一重要的C编程环境列入本书似乎很有必要。

本书的第一版和第二版均以对C++的概括而结束。在着手准备本书的前两版之时，C++在很大程度上一直属于试验性语言。然而，在本书的第二版问世以后转眼间已经过去的五年中，C++得到了广泛的应用，而今已是与C语言并驾齐驱的主要程序设计语言。因此，对这一重要课题仅作简单概括似乎不再具有什么价值(C++这一课题简直变得太庞大了，它自己就需要一本完整的专著来介绍)。有兴趣的读者可以参考本人的《C++：The Complete Reference，Second Edition》(《最新C++语言精华(第二版)》，电子工业出版社，1997年翻译出版)一书，其中包含C++的全部内容。原来的C++那一章内容，已由对C解释程序的描述来替代。我想读者会发现，编制C解释程序是本书最有趣的章节。

内容提要

本书详细讨论了C语言及其函数库的各个方面，主要强调ANSI标准C语言，但也适当地介绍较老的标准，即事实上的"K&R"标准。

全书分成五部分：

- C语言(1～10章)
- C标准库(11～18章)
- 算法和应用(19～24章)
- C语言软件开发(25～27章)
- C解释程序(28章)

第一部分详细讨论关键字、预处理指令和C语言的特性。

第二部分讨论标准C函数库，这一部分不仅描述了ANSI C标准指定的所有函数，而且还描述了一些通用但非标准的函数。

第三部分介绍某些通用算法及应用程序，所有程序员都应在自己的工具包中包括这些内容。该部分还包含对人工智能问题求解以及Windos 95程序设计的讨论。

第四部分介绍C语言编程环境，其中包括汇编语言程序接口、效率、移植和调试等等。

第五部分通过编制C解释程序来说明C语言，这无疑是本书中最令人兴奋，最富有挑战性，同时也是最有趣的内容。对于大多数C程序员而言，将无法抑制自己考察、改进第五部分的C解释程序并对其进行其它方面修改的欲望。为了理解C语言的优美和雅致，再也没有比为之建立解释程序更好的方式了。

目　录

第一部分　C语言

第二部分 C 标准库

第三部分 算法和应用

第四部分 C语言软件开发

第五部分 C解释程序

第一部分　C 语言

本书第一部分全面讨论程序设计语言 C。第一章是对 C 语言的简要综述，有经验的程序设计人员可以直接跳到第二章。第二章研究 C 的固有数据类型、变量、操作符和表达式。第三章讲解程序控制语句。第四章讨论数组和字符串。第五章探讨指针。第六章讲解函数。第七章研究结构、联合、枚举和用户自定义的数据类型。第八、九两章分别讲述控制台 I/O 和文件 I/O。第十章讨论 C 语言的预处理程序和 C 程序中的注释。

这部分是本书的主要内容，反映了 ANSI C 标准，因此适于目前所有的 C 编译程序。

第 1 章　C 语言概述

本章描述程序设计语言 C 的概貌、起源、应用以及构成原则，主要适于初学者。

1.1　C 语言的起源

C 语言是由 Dennis Ritchie 创造并首先在配备 UNIX 操作系统的 DEC PDP-11 计算机上实现的。C 语言是早期计算机语言 BCPL（目前在欧洲还有应用）发展过程的产物。BCPL 由 Martin Richards 开发，影响了 Ken Thompson 首创的 B 语言。B 语言导致了 C 语言在 70 年代的发展。

多年来，UNIX 系统 V 配备的 C 语言一直是 C 的公认标准，由 Brian Kernighan 和 Dennis Ritchige 在《C 语言程序设计》（The C Programming Language，Prentice-Hall，1987）一书中描述。随着微型计算机的普及，出现了大批 C 语言系统。令人吃惊的是，其中多数系统接受的源程序都能高度兼容（即，为其中一台计算机编写的程序可以成功地用另一台去编译）。然而，没有统一标准，必然存在差异。为了克服这种不利局面，ANSI 于 1983 年初夏成立了一个委员会，由该委员会制订一劳永逸的定义 C 语言的标准。标准化过程花了六年时间（比任何人想象得都要长）。ANSI C 标准于 1989 年被完全采用，并于 1990 年推出第一个范本。今天，所有主要 C 语言编译程序都实现了 ANSI C 标准。本书完全涵盖 ANSI C 标准并以此为重点，同时又包括有关 UNIX C 的内容。换言之，使用各种 C 编译程序的人，在任何环境下，都可以在本书中找到适用的内容。

1.2　C 是中级语言

C 语言通常被称为中级计算机语言。这并没有贬意，不意味着它功能差或难以使用，或者比 BASIC、Pascal 等高级语言原始，也不意味着它有汇编语言不方便的特性（及其相应的麻烦）。相反，C 语言之所以被称为中级语言，是因为它把高级语言的最佳成份同汇编语言的控制和灵活性结合起来了。表 1-1 表明 C 在各种计算机语言中所处的地位。

作为中级语言，C 允许对位、字节和地址这些计算机功能中的基本成份进行操作。尽管如此，C 语言程序还是非常容易移植的。可移植性表示，易于把为某种计算机写的软件改编到另一种机器或操作系统上。例如，如果为苹果机写的一个程序，能够方便地改为可以在 IBM PC 上运行的程序，则称之为可移植的。

所有的高级语言都支持数据类型的概念。一种数据类型定义了一类变量的取值范围和在其上操作的一组运算。常见的数据类型是整型、字符型和实数型。虽然 C 语言有五种固有的基本数据类型，但与 Pascal 或 Ada 相比，却不是强类型语言。C 几乎允许所有类型转换，例如，字符型和整型数据能够自由地混合在某一表达式中。

表 1-1　C 在计算机语言中的地位

高级	Ada
	Modula-2
	Pascal
	COBOL
	FORTRAN
	BASIC
中级	C++
	C
	FORTH
低级	Macro-assembler
	Assembler

不象高级语言那样，C 几乎不进行运行时的错误检查，例如不检查数组边界是否溢出。检查运行时的错误是程序员的责任。

同样，C 并不严格要求参数和变元之间的类型兼容。正象我们从其它编程经验中了解的那样，高级计算机语言通常要求变量的类型要（或多或少地）与接收这一变量的参数的类型相一致，但 C 并不是这样。相反，C 的变量的类型可以是任意的，只要它能够转换成参数的类型即可。而且，C 可以自动完成这种转换。

C 语言的特色是：允许对位、字节、字和指针直接操作，非常适合经常进行上述）操作的系统程序设计。

C 语言的另一个重要特点是：仅有 32 个关键字（27 个来源于 Kernighan 和 Ritchie 的公认标准，另 5 个是 ANSI 标准化委员会增加的）。这些关键字就是构成 C 语言的命令。高级语言通常有几倍于此的关键字。与 BASIC 的大多数版本相比，后者的关键字超过 100 个。

1.3　C 是结构化语言

有编程经历的人，一定会听说过计算机语言中的块结构（block-structured）一词。严格地说，C 语言并不完全是块结构语言，一般称为结构化语言（structured language）。C 语言与 ALGOL、Pascal 和 Modula-2 等结构化语言非常类似。

注意：从技术上看，C 语言不是块结构语言的原因是，一个块结构语言允许在过程（procedure）和函数（function）中定义其它过程和函数。因为 C 不允许在函数内再创建函数，所以

不能正式称为块结构语言。

结构化语言的显著特征是代码和数据的封装(compartmentalization)。这种语言能够把执行某个特殊任务所需的指令和数据从程序的其余部分分离出去、隐藏起来。获得隔离的一个方法是,调用使用局部(临时)变量的子程序。通过使用局部变量,我们能够编制对程序其它部分没有副作用的子程序,容易编写共享代码段的程序。如果开发了一些分离良好的函数,在引用时我们仅需要知道函数做什么,不必知道它如何做。切记:过度使用全局变量(可以被全部程序访问的变量)会由于意外的副作用而在程序中引入错误,设计过 BASIC 程序的人对这个问题都有深刻体会。

结构化语言提供了大量程序设计功能,直接支持若干循环结构,如 while,do-while 和 for。在结构化语言中禁止或不提倡使用 goto 语句,例如,不能象标准的 BASIC 和传统的 FORTRAN 那样把它作为常用的程序控制语句。结构化语言允许将陈述语句放在某一行的任何地方,不要求遵守象较老的 FORTRAN 语言那种严格的程序格式。

下面是结构化语言和非结构化语言的例子:

非结构化语言	结构化语言
FORTRAN	Pascal
BASIC	Ada
COBOL	C++
	C
	Modula-2

现代语言一般都是结构化的,而陈旧的计算机语言的标志之一就是非结构性。结构化语言比非结构化语言更便于程序设计,用结构性语言编写的清晰程序更易于维护。这已是人们普遍接受的观点。

C 语言的主要结构成分是函数——C 的独立子程序。在 C 语言中,函数是一种构件(程序块),程序的所有操作都在其中发生。函数允许一个程序中的诸任务被分别定义和编码,使程序模块化。可以确信,设计好的函数能在各种情况下正确工作,不会对程序的其它部分产生副作用。能否设计出独立的函数在大型项目中是至关重要的。在这种项目中,一个程序员的代码不得意外地影响其他人的程序。

在 C 语言中,实现结构化和代码隔离的另一种方法是,使用复合语句(或称分程序)。一个复合语句是作为一个语句处理的、在逻辑上相互关联的一组语句。在 C 语言中,复合语句就是处于一对大括号之间的语句序列。如:

```
if(x<10){
  printf("too low,try again\n");
  scanf("%d",&x);
}
```

在上例中,如果 x 小于 10,位于 if 句之后、两个大括号之间的两个语句都被执行。这两个句子连同大括号表示一个代码块,是一个逻辑单位,其中所有语句必须一起执行。注意,每一个语句或是一个单句或是一个复合句。代码块不仅能使许多算法得到清晰、优美而有效的实现,而且更有助于程序员抓住正在实现的算法的本质。

1.4 C是面向程序员的语言

令人吃惊的是，计算机程序语言并非都是为程序员设计的。我们以典型的非程序员语言COBOL和BASIC为例。COBOL的设计目标不是改善程序员的环境，不是增加代码的可靠性，也不是提高编程的速度。它的部分目标是使非程序员能够阅读并理解(基本不可能)程序。BASIC主要是供非程序员编制计算机程序、解决简单问题的。

与此形成鲜明对照的是C语言。C是由实际工作中的程序设计人员创造、影响并测试的，最终向程序员提供了程序员期望的一切：很少限制、很少强求、块结构、独立函数和简洁扼要的关键字集合。通过C，程序员可以基本达到汇编码的效率，程序又有ALGOL和Modula-2的结构。因此，C语言自然深受第一流专业程序设计人员的欢迎。

程序设计人员广泛使用C语言的一个主要因素是，常能用C代替汇编语言。汇编语言使用的汇编指令，是能够直接在计算机上执行的二进制代码的符号表示，汇编语言的每个操作都映射为计算机执行的单一任务。虽然汇编语言给予程序员达到最大灵活性和最高效率的潜力，但开发和调试汇编语言程序的困难是难以忍受的。进一步说，由于汇编语言是非结构化的，最后完成的程序肯定象一团意大利面条——纠缠不清的跳转、调用和变址。非结构性使汇编语言难于阅读、改进和维护。也许更重要的是，汇编语言程序不能在使用不同中央处理器(CPU)的机器之间移植。

最初，C语言主要用于系统程序设计(systems programming)。系统程序是计算机操作系统及操作系统支持的实用程序(utilities)的一部分。下列常称为系统程序：

- 操作系统
- 编译程序
- 翻译程序
- 数据库
- 编辑程序
- 电子表

随着C语言的普及，由于其可移植性和高效率，许多程序员开始用C设计各类程序。几乎所有计算机上都有C语言编译程序，使我们可以很少改动、甚至不加改动地将为一种机器写的C语言源程序在另一种机器上编译执行。可移植性节省了时间和财力。各种C编译程序均可产生非常紧凑、执行快捷的目标码，比各种BASIC编译程序的目标码都紧凑、快速。

也许C语言被用于所有各类程序设计的主要原因在于程序员的偏爱。C提供了汇编语言的速度和FORTH的可扩充性，而很少有Pascal和Modula-2的限制。C程序员都可以创建并维护一个专门的函数库，库中的函数根据个人需要配置，可以在各种程序中使用。由于允许(更确切地说是鼓励)分别编译，C语言可以使程序员方便地管理大型项目，最大限度地减少重复劳动。

1.5 编译和解释

计算机语言定义程序的属性而不是程序执行的方式，理解这一点是很重要的。程序执行一般有两种方式，即编译的(compiled)或解释的(interpreted)。尽管用任何计算机语言编写的程序都可以编译或解释，但有些语言主要是为一种执行方式设计的。例如，BASIC是设计