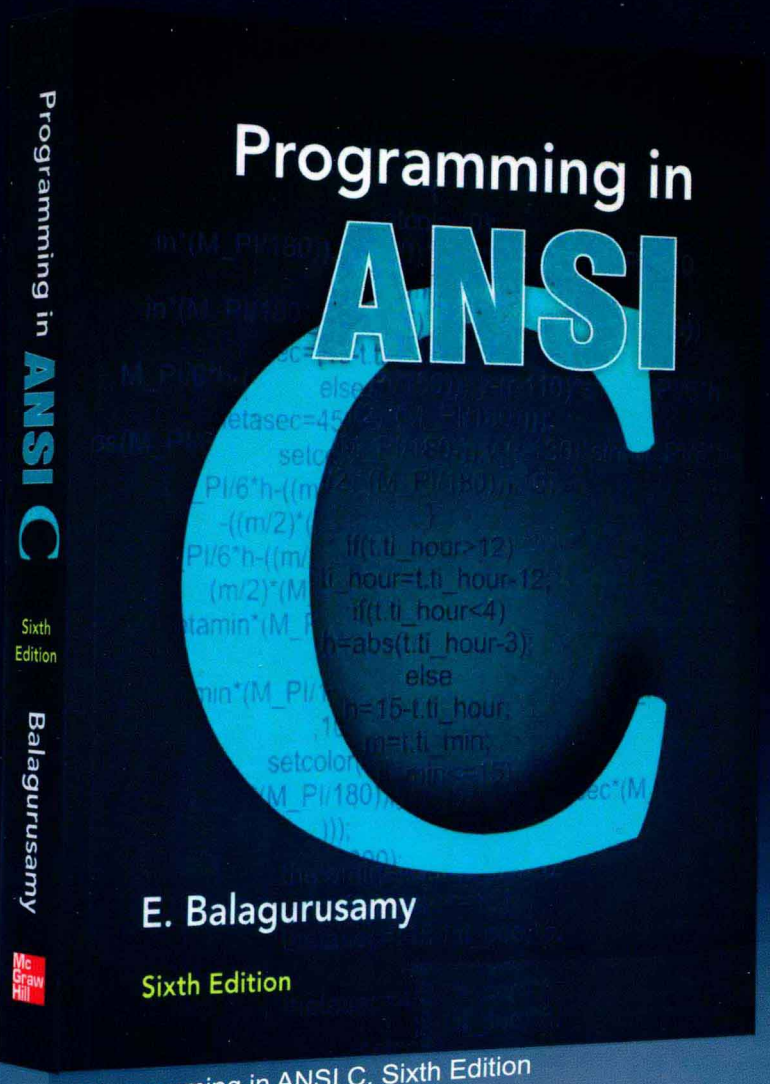
国外计算机科学经典教材

标准C程序设计

(第6版)

[印] E. Balagurusamy 著
王楚燕 鱼 静 译



Programming in ANSI C, Sixth Edition

Mc Graw Hill Education

清华大学出版社

国外计算机科学经典教材

标准 C 程序设计

(第 6 版)

[印] E. Balagurusamy 著

王楚燕 鱼 静 译

清华大学出版社

北 京

E. Balagurusamy
Programming in ANSI C, Sixth Edition
ISBN：978-1-25-900461-2

图书在版编目(CIP)数据

标准 C 程序设计：第 6 版/(印)巴拉古鲁萨米(Balagurusamy, E.) 著；王楚燕，鱼静 译.
—北京：清华大学出版社，2014
书名原文：Programming in ANSI C, Sixth Edition
(国外计算机科学经典教材)

ISBN 978-7-302-34666-1

Ⅰ.①标… Ⅱ.①巴… ②王… ③鱼… Ⅲ.①C 语言－程序设计－教材 Ⅳ.①TP312
中国版本图书馆 CIP 数据核字(2013)第 291204 号

责任编辑：王　军　李维杰
装帧设计：牛静敏
责任校对：曹　阳
责任印制：宋　林

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：清华大学印刷厂
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：32.5　　**字　　数**：832 千字
版　　次：2014 年 1 月第 1 版　　**印　　次**：2014 年 1 月第1 次印刷
印　　数：1～3500
定　　价：68.00 元

产品编号：054213-01

译者序

C 语言是经典的编程语言，凭借其自身简洁、灵活和功能强大等特点，自诞生以来就牢牢占据着流行编程语言的榜首。C 语言是强大的编程语言，可以进行从操作系统到设备驱动的各层次开发，可以应用在从大规模传统应用到新兴移动应用的各种领域。C 语言也是一门非常优秀的学习程序设计入门语言，其简洁性使得初学者无须学习太多语法就可以开始编写真正的应用程序，很多程序员的职业生涯就是从握手 C 语言开始的。

作为教材，本书最新版对内容进行了全面修订，以紧跟 C 程序设计领域的发展和当今教学大纲的需要。本书一如既往地强调通过示例进行学习的思想。在深入讲解 C 语言的每个主要特性后，通过一个完整程序示例来演示其用法。示例程序既简单，又很有启发性。按照本书结构认真学习，读者将可以很好地理解 C 语言。

编程语言的学习十分枯燥，学习的过程也特别艰辛，但是学成之后所获得的成就感也是无与伦比的。C 语言的灵活性在很大程度上得益于指针，而指针也往往是让初学者头疼的地方，甚至成为梦魇。我深知其学习的艰辛，对于自己在学习过程中的彷徨犹豫，挫折困顿，至今还历历在目。本书循序渐进、深入浅出的讲解，曾经让自己大惑不解的地方，从本书看来则是如此的理所当然，水到渠成。

纸上得来终觉浅，绝知此事要躬行。本书的作者深谙此理，在每章都有练习和习题供读者动手实践。学习一门语言没有比动手实践更快、更好的方法了。所以建议读者在每读完一章的时候亲自动手完成练习，而不是“读”过练习，如此方能成为理论和行动上的“巨人”。有人问大师，如何能技近乎道？大师曰：读书，读好书，然后实践之。万事无他，惟手熟尔！

在这里要感谢清华大学出版社的李阳和李维杰编辑，她们为本书的翻译投入了巨大的热情并付出了很多心血。没有你们的帮助和鼓励，本书不可能顺利付梓。

对于这本经典之作，译者本着“诚惶诚恐”的态度，在翻译过程中力求“信、达、雅”，但是鉴于译者水平有限，错误和失误在所难免，如有任何意见和建议，请不吝指正。感激不尽！本书由王楚燕、鱼静翻译，参与翻译活动的还有孔祥亮、陈跃华、杜思明、熊晓磊、曹汉鸣、陶晓云、王通、方峻、李小凤、曹晓松、蒋晓冬、邱培强、洪妍、李亮辉、高娟妮、曹小震、陈笑。

最后，希望读者通过阅读本书能早日步入 C 语言编程的殿堂，领略 C 语言之美！

作者简介

E. Balagurusamy 目前是位于 Coimbatore 的 EBG Foundation 公司的董事长。之前，他是位于 New Delhi 的 Union Public Service Commission 的委员，以及位于 Chennai 的 Anna University 的校长。他做过信息技术和管理领域的教师、培训师和顾问。他从 Indian Institute of Technology 的 Roorkee 分校获得了电子工程荣誉学士和硕士学位，以及系统工程博士学位。他的研究领域包括面向对象的软件工程和电子治理：技术管理、业务流程重组和全面质量管理。

E. Balagurusamy 撰写了大量研究论文和多本图书。他最畅销的图书包括(但不只是如下这些)：

- *Fundamentals of Computers*
- *Computing Fundamentals and C Programming*
- *Programming in C#*, 3/e
- *Programming in Java*, 4/e
- *Object-Oriented Programming with C++*, 5/e
- *Programming in BASIC*, 3/e
- *Numerical Methods*
- *Reliability Engineering*

他获得了多种荣誉，并被收录在 Directory of Who's Who of Intellectuals 和 Directory of Distinguished Leaders in Education 中。

前 言

C 是功能强大、灵活性好、可移植且结构良好的程序设计语言。由于 C 集成了高级语言和汇编语言的优点，因此可适用于系统和应用程序的开发。今天，C 语言无疑是操作系统和嵌入式开发中使用最广的通用语言。在几乎所有现代程序设计语言中，都可以看到 C 语言的痕迹。自从 1989 年标准化以来，C 经历了一系列的修订和改进，以提高该语言的可用性。现在，融合了这些新特性的版本称为 C99。

第 6 版新增内容

C 语言的创建人是 Dennis Ritchie(1941 年 9 月—2011 年 10 月)，他“帮助塑造了数字时代”。在 2011 年 Ritchie 辞世后，为了向这位帮助改变了计算机程序设计世界的人致敬，本书开始准备再版。Ritchie 曾这样说过：“C 语言在很多方面都很奇特。但是与其他许多成功的事物一样，C 语言在处理问题上有自己的统一性，这是小团队进行开发的结果。”

本书结构

本书首先在第 1 章概要地介绍了 C 语言，包括 C 语言的历史、C 程序的基本结构和执行。第 2 章讨论了如何声明常量、变量和数据类型。第 3 章讨论了内置运算符以及如何使用它们来构建表达式。第 4 章介绍了输入输出操作。第 5 章介绍了决策判断与分支，包括 if-else、switch 和 goto 语句。第 6 章讨论了决策与循环，包括 while、do 和 for 循环。第 7 章和第 8 章介绍了数组和数据元素的有序排列，这些对所有程序设计语言都很重要。第 8 章还介绍了字符串。第 9 章和第 10 章介绍了函数、结构体和共用体。指针是 C 语言中最难理解和掌握的部分，第 11 章以对用户非常友好的方式介绍了指针。第 12 章和第 13 章分别介绍了文件管理和动态内存分配。第 14 章介绍了预处理器。最后，第 15 章介绍了如何开发 C 程序，为读者提供了关于程序开发的一些真知灼见。

如果读者按照本书结构认真学习，将可以很好地理解 C 语言。

本书内容进行了全面修订，以紧跟 C 程序设计领域的发展和现实的需要。本书一如既往地强调通过示例进行学习的思想。在深入讲解 C 语言的每个主要特性后，通过一个完整程序示例来演示其用法。示例程序既简单，又很有启发性。

每章的开头含有一小节，介绍本章的主要内容。如有必要，则使用图形来描述有关的概念，以提高描述的清晰性，方便读者理解。基本的语言技巧和其他需要特别考虑的内容以“注意”的形式突显出来。

本书特色

- 带注释的代码全书随处可见，这些注释说明了 C 语言的各种特性是如何集成在一起以实现特定任务的。
- 补充信息和“注意”对正文进行了必要的补充，但独立于正文之外。
- 每章末尾的案例学习演示了把 C 特性集成在一起的常用方式，并显示了一些实际的应用。
- 每章末尾的“谨记”列举了一些有用的提示和可能出问题的地方。
- 每章末尾的“复习题”和“编程练习”为读者复习所学概念和实际的应用开发提供机会。
- 附录 D 给出的“项目设计”展示了开发大型程序时如何集成 C 的各种特性。
- 从大学生常问的问题中专门挑选了有代表性的问题来更新程序和复习题。
- 单独用一章的篇幅来介绍遵循 C99 标准的最新编译器的特性。

网络资源

本书还提供配套网站(http://www.mhhe.com/balagurusamy/ansic6)，上面提供了以下资源：

- 本书各章程序的可运行代码。
- 两个编程项目：库存管理系统和登录记录系统。
- 可下载的小项目(链表和矩阵乘法)，以及各章的案例学习程序。
- 附加阅读材料，例如“UNIX Operating System”和“Differences between ANSI C, C++ & ANSI C++”，它们可帮助读者深入了解 C 语言及其应用。

本书读者对象

本书针对的是想要成为 C 程序员的读者，无论他们是否理解并实际编写过程序。本书以简单易懂的方式讲解了什么是 C 语言，为什么使用 C 语言，以及如何使用 C 语言进行程序设计。

目 录

第 1 章

C 语言概述

关键术语：
printf、程序

1.1 C 语言的历史

作为一种程序设计语言，字母“C”看上去是一个奇怪的名字。但是这个奇怪而好听的语言却是现今最流行的计算机语言之一，因为它是一种结构化的、高级的、与机器无关的语言。它允许软件开发人员开发程序时无须担心实现这些程序的硬件平台。

所有现代语言的起源都是 20 世纪 60 年代提出的 ALGOL 语言。ALGOL 语言是最先使用块结构的计算机语言。尽管它从来没有在美国流行开来，但在欧洲却被广泛使用。ALGOL 语言给计算机科学界带来了结构化程序设计的概念。20 世纪 60 年代，计算机科学家，如 Corrado Bohm、Guiseppe Jacopini 和 Edsger Dijkstra 使这一概念大众化了。随后，又有几种计算机语言被开发出来。

1967 年，Martin Richards 开发了一种称为 BCPL(Basic Combined Programming Language，基本组合程序设计语言)的计算机语言。该语言主要用于系统软件的开发。1970 年，Ken Thompson 创建了一种计算机语言，该语言继承了 BCPL 的很多特性，就称为 B 语言。在贝尔实验室，B 语言用来开发 UNIX 操作系统的早期版本。BCPL 和 B 语言都是“无类型”的系统程序设计语言。

C 语言是 Dennis Ritchie 于 1972 年在贝尔实验室从 ALGOL、BCPL 和 B 语言的基础上发展而来的。C 语言利用了这些语言的很多概念，并添加了数据类型的概念以及其他功能强大的特性。由于它是与 UNIX 操作系统一起被开发出来的，因此它与 UNIX 有着很强的关联。UNIX 操作系统(也是在贝尔实验室开发出来的)几乎完全是用 C 语言编码的。UNIX 是现今使用最为流行的网络操作系统，也是 Internet 数据超高速路的心脏。

多年以来，C 语言主要用于科研环境下，但最终，随着多种商用 C 编译器的发布，以及 UNIX 操作系统的不断流行，在计算机专业人士中也开始获得广泛支持。今天，C 语言可以运行在多种操作系统和硬件平台下。

20 世纪 70 年代，C 语言发展为现在所谓的“传统 C 语言”。自 1978 年 Brian Kernigham 和 Dennis Ritchie 的 *The C Programming Language* 一书出版后，C 语言成了最流行的语言。该书很受欢迎，以至于在程序设计界，C 语言被认为是“K&R C”。C 语言的快速发展导致了不同版本的语言出现，这些语言相互类似但往往互不兼容。对系统开发人员来说，这是一个严重的问题。

为了确保 C 语言的标准，1983 年，美国国家标准局(American National Standards Institute，ANSI)任命了一个技术委员会来定义 C 语言的标准。该委员会于 1989 年 12 月批准了一个 C 语言版本，这就是现在的 ANSI C。该版本又于 1990 年被国际标准组织(International Standards Organization，ISO)批准。C 语言的这个版本又称为 C89。

20 世纪 90 年代，一种完全基于 C 的语言——C++语言，经历了大量的改进和变化，于 1977 年 11 月成为一种获得 ANSI/ISO 批准的语言。C++在 C 的基础上添加了一些新特性，使之不仅成为一种真正的面向对象的语言，而且是一种更通用的语言。与此同时，美国的 Sun 公司创造了一种新的语言——Java 语言，它是以 C 和 C++为模型的。

所有流行的计算机语言在本质都是动态的。它们通过加入新特性来不断提高其功能和使用范围，C 语言也不例外。尽管 C++和 Java 语言都是从 C 发展而来的，但 C 语言标准委员会认为，如果将 C++/Java 语言的一些特性加入到 C 中，会提高 C 语言的性能，于是就产生了 C 语言的 1999 标准。这个版本的 C 语言通常称为 C99。C 语言的历史和发展如图 1-1 所示。

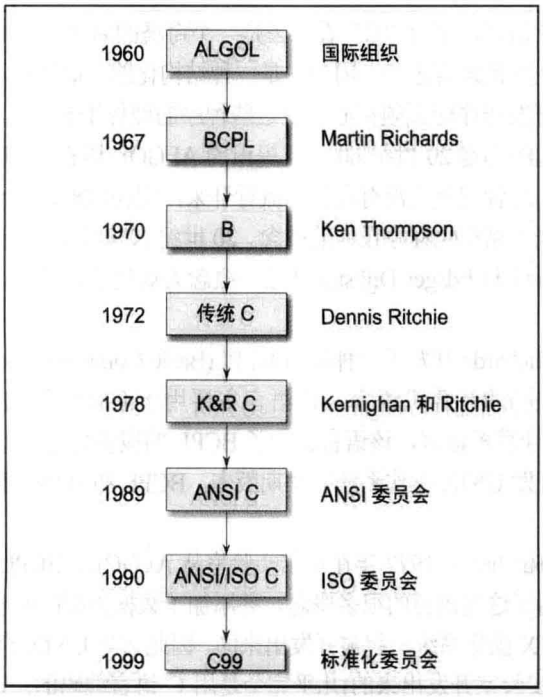


图 1-1　C 语言的历史

尽管 C99 是一个改进版本，但很多常用的编译器仍不支持 C99 的所有新特性。因此，我们将在附录 E 中单独介绍加入到 C99 的新特性。这样，感兴趣的读者就可以快速地参考这些新特性，并在可能的情况下使用它们。

1.2　C 语言的主要特性

C 语言之所以越来越受欢迎，是因为它具有很多可取的特性。它是一种健壮的语言，其丰富的内置函数集和运算符可用来编写任意复杂的程序。C 语言编译器融合了汇编语言的性能和高级语言的特性，因此很适合编写系统软件和商业软件。事实上，市场上很多的 C 编译器都是用 C 语言编写的。

用 C 语言编写的程序高效且运行速度快。这归功于它的各种数据类型和功能强大的运算符。其运行速度是 Basic 语言的好几倍。例如，一个计算 0~15 000 个变量的程序，用 C 语言编写，其运行时间为 1 秒，而用 Basic 语言编写的则要运行 50 秒。

ANSI C 只有 32 个关键字，内置函数是其强大之处。C 语言有很多标准函数可用来开发程序。

C 语言是高度可移植的。这意味着，为某一台计算机编写的 C 程序，只要稍作修改，甚至不用修改就可以在另一台计算机上运行。如果我们计划使用不同操作系统的新计算机，可移植性就很重要。

C 语言很适合结构化程序设计，因而要求用户以功能模块的方式来思考问题。这些模块的恰当结合就可以形成一个完整的程序。这种模块化的结构使得程序的调试、测试和维护更加容易。

C 语言的另一个重要特性是它的自我扩展能力。一个 C 程序基本上是各种函数的集合，这些函数由 C 函数库支持。我们可以不断地将自己的函数添加到 C 函数库中。由于有了这么大量的函数，程序设计就变得简单了。

在讨论 C 语言的具体特征之前，我们先来看一个 C 范例程序，分析并了解程序是如何工作的。

1.3　示例程序 1：显示一条消息

如下所示是一个很简单的程序：

```
main( )
{
/*·····printing begins·····*/
       printf("I see, I remember");
/*·····printing ends·····*/
}
```

该程序将产生如下输出：

```
I see, I remember
```

让我们来详细看看该程序。第一行告诉系统程序名是 main，从这一行开始执行。main()函数是 C 系统使用的一个特殊函数，用来告诉计算机程序的运行起点。每个程序必须正好有一个 main 函数。如果有多个 main 函数，编译器就无法知道哪一个是程序的运行起点。

紧跟在 main 后面的空括号表明 main 函数不带参数。关于参数的概念，我们将在第 9 章讨论函数时再详细介绍。

第二行的开始是“{”，表明 main 函数的开始，而最后一行的闭括号“}”则表示该函数的结束。在上面的范例程序中，闭括号还标志着整个程序的结束。这两个括号之中的所有语句就形成了函数体。函数体包含有一个指令集，从而完成指定的任务。

在上面的范例程序中，函数体包含 3 条语句，其中只有 printf 一行是可执行的语句。以/*开始，以*/结尾的行被称为注释行。在程序中恰当地使用注释行，可以提高程序的可读性，使其更容易让人理解。由于注释行不是可执行语句，因此/*与*/之间的内容全部被编译器忽略掉。通常，一个注释可以插入到程序的任何空白处——行的开始、中间或结尾处——但不能插入到一个词的中间。

尽管注释行可以出现在程序的任意地方，但在 C 语言中它们不能嵌套。这就意味着，不能在注释行中再插入注释行。一旦编译器发现注释的开始标志，它就将忽略掉后面的所有内容，直到再发现一个结束标志为止。下面的注释行：

```
/* = = = =/* = = = = */ = = = = */
```

是不合法的，因此将产生一个错误。

由于注释行不会影响程序的运行速度以及编译后程序的大小，因此我们应大方地在程序中使用注释。注释有助于开发人员和其他用户理解程序的不同函数和运算，对程序的调试和测试也有帮助。我们将在后面的范例程序中体会到注释行的作用。

现在让我们来看 printf 函数，这是该范例程序中唯一可执行的语句：

```
printf("I see, I remember");
```

printf 是预定义的标准的 C 函数，用于显示输出。预定义的含义就是，该函数已编写好并已编译。在链接时，与我们的程序链接在一起。编译和链接的概念将在本章的后面介绍。printf 函数将两个引号之间的内容显示出来。在本范例程序中，其输出为：

```
I see, I remember
```

注意，打印行以分号结尾。C 语言的每条语句都必须以分号结尾。

假如，我们要把输入打印为如下所示的两行：

```
I see,
I remember!
```

可以通过添加两个 printf 函数来实现，如下所示：

```
printf("I see, \n");
printf("I remember !");
```

两个括号之间的信息称为该函数的参数。第一个 printf 函数的参数是“I see, \n”，第二个 printf 函数的参数是“I remember!”。这些参数是要显示出来的字符串。

注意，第一个 printf 函数的参数在字符串的末尾含有字符\和 n 的组合。该组合称为新行字符。新行字符命令计算机切换到下一行(即新行)。在概念上，它类似于打字机的回车键。在显示了逗号字符后，新行字符\n 的出现将使“I remember!”显示在下一行。字符\和 n 之间不允许

有空格。

如果省略掉第一条 printf 语句的新行字符，那么输出仍为一行，如下所示：

```
I see, I remember !
```

这类似于本节开头程序的输出。但是，注意“,”和“I”之间没有空格。

只使用一条 printf 语句，只要在适当的地方使用新行字符，也可以生成两行或多行输出，例如语句：

```
printf("I see,\n I remember !");
```

的输出为：

```
I see,
I remember !
```

而语句

```
printf("I\n.. see,\n··· ··· ··· I\n··· ··· ··· remember !");
```

的输出为：

```
I
.. see,
··· ··· ··· I
··· ··· ··· remember !
```

注意，有些作者推荐在所有程序的开头加上如下输入/输出库函数：

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

但是，对 printf 和 scanf 并没有必要，因为这两个函数已经定义为 C 语言的一部分了。关于输入输出函数的更多内容，参见第 4 章。

在我们继续讨论其他更多的示例之前，必须注意很重要的一点：C 语言是区分大小写字母的。例如，printf 和 PRINTF 并不相同。在 C 语言中，通常都是写成小写字母。而大写字母用作表示常量的符号名。我们也可以在输出字符串中使用大写字母，例如“I SEE”和“I REMEMBER”。

上面介绍的用于显示“I see, I remember!”的示例是最简单的程序之一。图 1-2 列举了这类简单程序的一般格式。所有 C 程序都需要一个 main 函数。

main 函数：

main 函数是每个 C 程序的一部分。C 语言允许有如下多种形式的 main 函数声明：

- main()
- int main()
- void main()
- main(void)
- void main(void)
- int main(void)