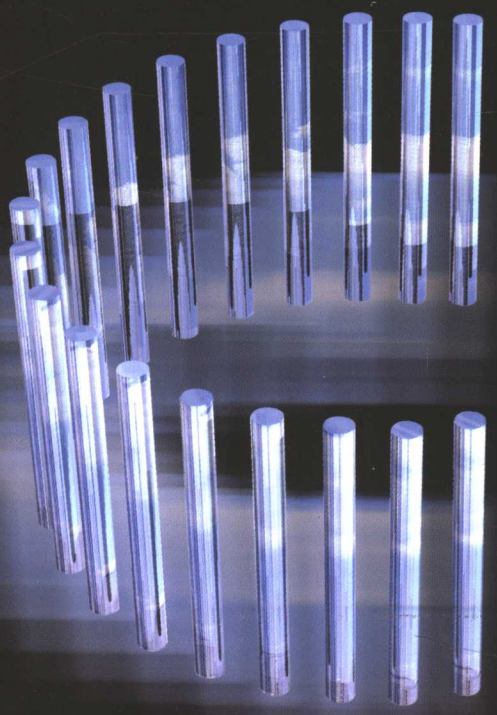
Mc Graw Hill Education
世界著名计算机教材精选

标准C程序设计
（第3版）

E Balagurusamy 著
金 名 张长富 等 译



PROGRAMMING IN ANSI C

Third Edition

清华大学出版社
Mc Graw Hill

世界著名计算机教材精选

标准C程序设计

（第3版）

E Balagurusamy 著

金 名 张长富 等 译

清华大学出版社

北 京

E Balagurusamy
Programming in ANSI C, 3e
EISBN:0-07-053477-2

图书在版编目(CIP)数据

标准C程序设计（第3版）/（印）巴拉古鲁萨米（Balagurusamy, E.）著；金名、张长富等译. —北京：清华大学出版社，2006.5
（世界著名计算机教材精选）
ISBN 7-302-12756-5

Ⅰ. 标…　Ⅱ. ①巴…　②金…　③张…　Ⅲ. C 语言－程序设计－教材　Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2006）第 025964 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
责任编辑：龙啟铭
印 装 者：北京国马印刷厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：29.75　**字数**：737 千字
版　　次：2006 年 5 月第 1 版　2006 年 5 月第1 次印刷
书　　号：ISBN 7-302-12756-5/TP・8131
印　　数：1～3000
定　　价：59.00 元

译 者 序

C 语言是所有本科生课程的一门核心课，也是现今使用最广泛的计算机语言。国内国外的 C 语言图书已经非常多了，但通过本书的翻译，我们觉得这本书还是很有引进价值，具体表现在它的以下几个特点上：

- 本书是基于最新的 C 语言标准。
- 在本书的最后给出了一个完整的应用程序开发示例 PHONE BOOK。
- 扩展讨论了 C 的指针。
- 每章后面的“谨记”一节给出了很有用的编程提示以及可能容易出错的问题。
- 每章后面的案例学习给出了 24 个真实的开发，展示了 C 程序的设计过程。
- 87 个程序设计范例向读者阐述了良好程序设计的基本原则。
- 还有 185 个练习题和 133 道实践编程题。

总之，本书的语言简洁易懂，示例非常丰富且具有很强的实际指导意义，是一本很好的 C 程序设计的教材。

本书的第 2 版被印度的很多学院和大学用作教材。在过去的几年中，本书还被一些重要的软件培训与开发公司用作培训教材。

由于本书的独特性、与众不同的学习方法以及简明的写作风格，相信本书也会受到国内学生和老师的欢迎。

本书由金名、张长富、冯华君、刘守燕、杨咏梅等译，张国清、潘华等人也参与了部分翻译和校对工作。欢迎广大读者指正。

第1版前言

C是一种通用的结构化程序设计语言，它功能强大、高效且简洁。C集成了高级语言的特性和汇编语言的某些元素，因此，对程序员和计算机都很贴切。用C语言进行程序设计很流行，也很有趣。

本书旨在教授读者如何使用C语言进行程序设计，不需要读者具有该语句的预备知识，适合于初学者和有经验的程序员。

全书贯彻了“用示例学习”的概念。在深入介绍了C语言的每个特性之后，给出了一个完整的示例程序，用于演示说明其应用。每章末尾的“案例学习”不仅介绍了把C语言的特性集成在一起的常用方法，而且还显示了它在实际生活中的应用。只要有必要，就用图形来描述，以便于读者更好地理解。最后一章介绍了开发高效、无错误的C程序的一些指导原则。

本书包含了100多个示例和程序。所有程序都使用与UNIX和MS-DOS操作系统兼容的编译器进行了测试，并适当地讨论了输出结果。这些程序还演示了良好编程风格的一般原则。

第3版前言

作为使用最广泛的计算机语言，C 是所有本科生课程的一门核心课。本书的第 2 版被很多学院和大学用作教材。在过去的几年中，本书还被一些重要的软件培训与开发公司用作培训教材。由于本书的独特性、与众不同的学习方法以及简明的写作风格，本书在学生和老师中很受欢迎。

本书融入了过去 10 年中采用本书的学生和老师的很多反馈意见。本版的特点包括：

- 根据读者的反馈，全书的内容和程序都做了重新评阅和修改。
- 每章末尾的练习分成了两部分，即“复习题”和“编程练习”。
- 每章都添加了针对本章内容的大量问答题和问题求解。
- 在很多章中添加了“谨记”一节。该节列举出了一些有用的提示和容易出现的问题。
- 新加的附录Ⅲ“电话簿程序”，有助于读者全面体验 C 语言的特性。

感谢给本书提供建议的老师和学生，这些建议极大地提高了本书的质量。

E Balagurusamy

目　录

第1章 C语言概述

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为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 ALGOL 语言，该语言是 20 世纪 60 年代引入的。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，又宣布开发了好几种计算机语言。

1967 年，Martin Richards 开发了一种称为 BCPL（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，且就称为 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，也是因特网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 年由 Brian Kernigham 和 Dennis Ritchie 著作的 The C Programming Language 一书的出版，C 语言成了最为流行的语言。该书很受欢迎，以至于在程序设计界，C 语言就认为是“K&R C”。C 语言的快速发展导致了不同版本的语言出现，这些语言类似但往往不兼容。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 Institate，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 年批准了一个 C 语言版本，这就是现在的 ANSI C。该本版又于 1990 年被国际标准组织（International Stardards Organiyation，ISO）批准。该标准于 1999 年进行了更新。C 语言的历史如图 1.1 所示。

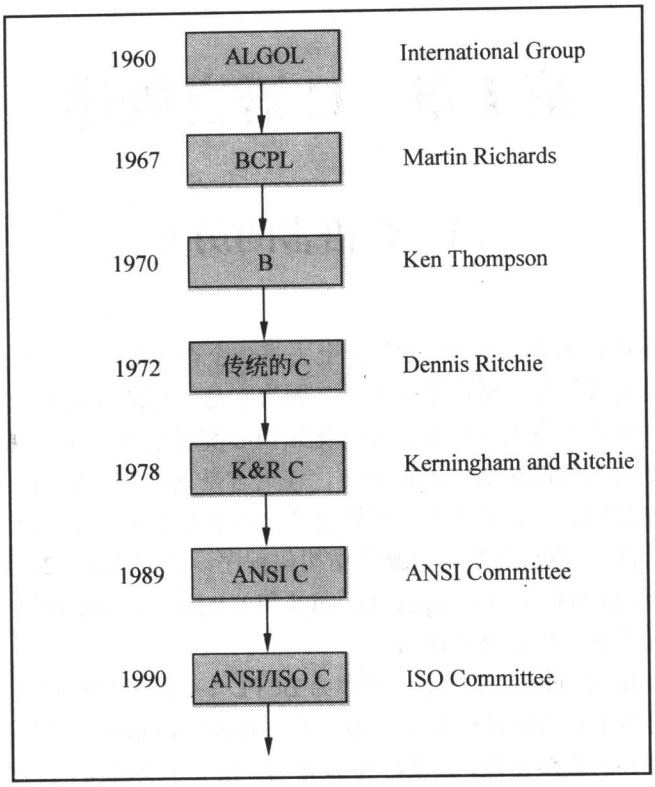


图 1.1　C 语言的历史

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 0~1500 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。

C 语言有 32 个关键字，其长度取决于其内置函数。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时，可移植性很重要。

C 语言很适于结构化程序设计，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 函数库中去。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

请看图 1.2 所示的一个很简单的程序。

```
main( )
{
/*.....printing begins.....*/

    printf("I see, I remember");

    /*.....printing ends.....*/
}
```

图 1.2　显示一行文字的程序

该程序将产生如下的输出：

```
I see, I remember.
```

让我们来详细看看该程序。第一行告诉操作系统程序名是 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。紧跟在 main 后面的空括号对表明 main 函数不带参数。关于参数的概念我们将在讨论函数（第 9 章）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，函数体包含有 3 个语句，其中只有 printf 一行是可执行的语句。以/*开始，以*/结尾的行称为注释行。程序中恰当地使用注释行，可以提高程序的可读性。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。一旦编译器发现了注释的开始标志，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。

假如我们要如下地将输出显示成两行：

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

这可以通过添加两个 print 函数来实现，具体如下：

```
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”，第二个的是“I remember!”。这些参数只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包含有字符\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。

如果省略掉第一个 printf 语句的新行字符，那么输出仍为一行，即如下所示：

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

这类似于图 1.2 中的程序的输出。但是，注意，在“,”和“I”之间没有空格。

只使用一个 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! ");
I
   see,
       I
           remember!
```

注意，有些作者推荐在所有程序的开头加上如下的输入输出库函数：

```
# 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请参见第 4 章。

在我们继续讨论其他更多的示例之前，必须注意到很重要的一点：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 1.3 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)

空括号对表示该函数不带参数。这也可以在括号中使用关键字 void 来明确表示不带参数。我们还可以在 main 之前指定 int 或 void。关键字 void 表示该函数不给操作系统返回任何信息，而 int 表示函数将返回一个整数值给操作系统。如果指定为 int，那么程序的最后一行必须是“return 0;”。为了简单起见，我们在本书的所有示例程序中使用第一种形式的 main 函数。

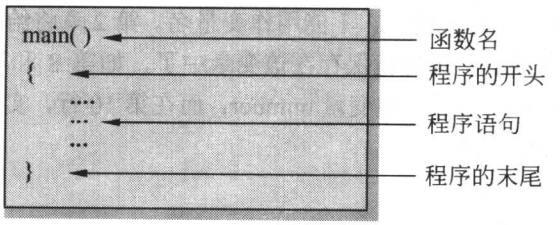


图 1.3　简单程序的结构