

计算机科学丛书

C语言解析教程

(原书第4版)

(美) Ai Kelley 著
Ira Pohl

麻志毅 译



机械工业出版社
China Machine Press

C语言在全世界的学术界和工业界都广泛地应用，同时它也是计算机科学教育编程课程的首选编程语言。本书通过应用作者首次提出的一种独特的教学方法——解析，对C语言的关键特征及编程过程做了细致全面的介绍。书中大量的练习和带有详细注释的工作程序会使读者从整体上提高编程能力、掌握编程技巧。本书的附录还给出了大量标准库函数供读者参考。

本书全面介绍了C语言的数组、指针、函数、串处理、文件处理和软件工具等方面的内容，本书是计算机专业本科生的极佳教材。

A1 Kelley, Ira Pohl: C by Dissection: The Essentials of C Programming, 4E.

Original edition copyright © 2001 by Addison Wesley Longman, Inc.

Chinese edition published by arrangement with Addison Wesley Longman, Inc.

All rights reserved.

本书中文简体字版由美国Addison Wesley公司授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书版权登记号：图字：01-2000-4101

图书在版编目（CIP）数据

C语言解析教程/（美）凯利（Kelley, A.），（美）波尔（Pohl, I.）著；麻志毅译。
—北京：机械工业出版社，2002.1

（计算机科学丛书）

书名原文：C by Dissection: The Essentials of C Programming, 4E

ISBN 7-111-09336-4

I. C… II. ①凯… ②波… ③麻… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字（2001）第065414号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：杨海玲

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2002年1月第1版第1次印刷

787mm × 1092 mm 1/16 · 27.25印张

印数：0 001- 5 000册

定价：48.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及收藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔抉外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图

书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzedu@hzbook.com

联系电话：(010) 68995265

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

专家指导委员会

(按姓名笔画顺序)

尤晋元
张立昂
周克定
高传善
石教英
邵维忠
郑国梁
裘宗燕

王 珊
李伟琴
周傲英
梅 宏
吕 建
陆丽娜
施伯乐
戴 葵

冯博琴
李师贤
孟小峰
程 旭
孙玉芳
陆鑫达
钟玉琢

史忠植
李建中
岳丽华
程时端
吴世忠
陈向群
唐世渭

史美林
杨冬青
范 明
谢希仁
吴时霖
周伯生
袁崇义

译者序

在过去的20年内，由于C语言具有丰富的运算符和数据类型、使用上灵活方便、生成的目标代码质量高和程序的运行效率高等特点，它受到了人们的普遍欢迎，全世界的学术界和工业界都在广泛地使用它，特别是大多数高校都把它作为编程的主要教学语言。C++在一些有意义的方面改善了C，与C是高度兼容的，特别是它增加了面向对象特征，而面向对象方法与技术在计算机科学技术领域已占据了主流地位。利用C语言进行程序设计与开发能锻炼程序员进行抽象程序设计的能力，也能为学习C++这种具有更为抽象的概念和技术的语言奠定基础，况且C程序设计特性在C++中得到了频繁的使用。

本书的英文版自问世以来很受欢迎，现已发行到了第4版，这完全是由于作者对书中内容的独具匠心的编写与组织。作者们用他们首次提出的“解析”这种独特教学法解释了新遇到的编程元素和程序代码中的惯用法。本书给出了一些经过详细注释的完整程序，使读者能不断地改善编程技巧。此外，书中还注重于说明如何从C转向C++，通过学习，读者会感到转向C++编程并不是什么难事。

本书覆盖的内容较为广泛，译者希望读者能从中获益，同时恳请广大读者对翻译中可能出现的疏漏和错误予以批评指正。

译者

2000年5月



麻志毅，男，北京大学计算机科学技术系副教授。1999年在东北大学获得博士学位，同年到北京大学计算机科学技术系作博士后，出站后留校工作。现已发表学术论文30余篇，主持或参加政府科研项目十余项。主要研究领域为软件工程、面向对象技术和计算语言学。

前 言

目前，全世界的学术界和工业界都在广泛使用ANSI C编程语言。在很多教育机构，它是编程课程的首选语言和用于计算机科学教育的语言。这主要是因为从高级的课程到介绍性的课程都可选择C。此外，C带有很多有用的库，并由复杂的集成环境所支持。ANSI C的改进弥补了一些在传统C中发现的不足，例如弱类型规则，这些改进以及C作为开发系统、图形和数据库的语言的广泛影响使得它成为讲授编程和计算机科学的首选。

通过详细地展开工作程序，本书对编程过程进行了全面的介绍，说明了C编程语言的关键特征。本书用一种循序渐进的方式对程序代码进行了全面说明。书中的代码已经在几个平台上测试过，你可从因特网网站aw.com获得这些代码。

本书中的代码能用于大多数C语言系统，例如MacOS、MS-DOS、OS/2、UNIX和Windows操作系统中的C系统。

解析 (Dissection)

通过用解析的方法详细地展开工作C程序，本书向读者提供了清晰而完整的编程过程。解析是作者于1984年研制的一种独特的教学法，用于说明工作代码的关键特征。解析类似于结构化的代码预排。其意图是向读者解释新遇到的编程元素和工作代码中含有的习惯语法。本书以一种按部就班的易于遵循的方式对程序和函数进行解释。在不同上下文中的应用充分强化了关键思想。

无背景假设

本书假设没有任何编程背景，学生和初学的计算机用户都可使用，不熟悉C的有经验的程序员也会受益于C语言细致的结构化表达。对于学生而言，本书可作为计算机科学或编程的第一门教程。

对于其他学科，本书也适合于CS1课程或初始编程课程。本书的每章都给出了一些带有详细注释的程序，以用整体的方式使学生不断地提高编程技巧。本书一开始就向学生介绍完整的程序，在本书的前面部分引导学生编写作为结构化编程主要特征的函数。函数与程序的关系就像段落和文章一样。能胜任编写函数是熟练程序员的标志，因而要强调这点。书中的例子和练习是富有内容和难度的，因而教师可精选适合于学生的作业。

专有特征

《C语言解析教程》(《C by Dissection: The Essentials of C programming》)的第4版增加了一些新的专有特征：

- 全部运用最新的ANSI C，例如函数原型。
- 每章末尾的“转向C++”一节有助于转换到面向对象的C++。
- 早期的对多文件程序的解释使得程序员能正确地编制模块代码和产品，并能正确地使

用库。

- 早期的对简单递归的解释反映出在早期阶段的计算机科学课程中对递归的提前介绍。
- 介绍了程序的正确性和输入的安全性。
- 对函数和指针进行深入的解释，这些概念通常是初学者的绊脚石。
- 对二维数组的讨论反应了科学家和工程师对C越来越多的使用。
- 对递归的深入研究反应了计算机科学家为了实现复杂算法对C的越来越多的使用。
- 可选的C++和Java练习。
- 附录中的位运算符改善了对主流语言特征表示。

章节特征

每章都含有如下的教学法元素：

解析 每章都有几个重要的示例程序。本书用解析方法解释这些程序的主要元素。这种新的编程思想的讨论，有助于读者在首次遇到这些思想时对它们的理解。

编程风格和方法 本书到处强调编程风格和方法。书中较早地提出了诸如结构化分支语句、嵌套控制流、自顶向下设计和结构化编程这样的重要概念。从一开始的详细解释就采用了一致和正确的编码风格，以强调编码风格的重要性和合理性。本书使用的编码风格是使用各种C语言编程的专业人员常用的。因为C支持函数原型和强类型检查，这种风格已被全面采用。

工作代码 从一开始，本书就向学生介绍完全可执行的程序。使用可执行的代码，学生能更好地理解 and 评价正在讨论中的编程思想。很多程序和函数都通过解析来解释。对编程思想的变化经常会体现在习题中。

常见的编程错误 书中描述了很多典型的编程错误以及避免这些错误的方法。学习编程语言遇到的大多数挫折都是由于遇到含糊的错误引起的。很多书讨论正确编码，而让读者用反复实验的方法去找出错误。本书说明了C中典型的错误是如何产生的以及应该怎样来改正错误。

系统考虑 在几乎任何的计算机和大多数操作系统中C都是可用的，但从一个系统到另一个系统在行为上偶然会存在不同。本书描述了这些不同。在ANSI C和传统C之间也存在不同，这些不同在本书中也做了描述。所有的程序通常都在一些不同的环境下进行了测试。本书强调编写可移植的独立于系统的代码。

转向C++ 每章的末尾有一个可选的节，它描述了转向C++所需的编程元素。本书对这些内容也给出了练习。就绝大部分而言，C是C++编程语言的子集，在转向C++之前很多学生先学习C，本书有助于这种自然过渡。对于希望完全精通C++的读者来说，可以将本书与Ira Pohl所著的《*C++ for C Programmers, Third Edition*》(Addison Wesley Longman, Inc., Reading, MA 1999, ISBN 0-201-39519-3)配合起来使用。

小结 在每章的末尾，我们给出了一个简洁的这一章的要点列表，对读者起复习的作用，以加深理解这一章提出的新思想。

练习 各章末的练习测试读者的语言知识。很多练习要与阅读正文交互式地进行，这样可以促进读者自我调整学习进度。还有一个练习特征：一些练习非常详细地着眼于一个主题，另外一些练习能扩充读者的知识以达到使用的高级境界。

作为学生的教材

本书作为教学生编程的教材，在第一学期的课程中可以使用第1章到第10章，其中包含了C编程语言的数组、指针和串的全部用法。第二学期课程致力于高级数据类型、文件处理和软件工具，这些内容包含在第11章到第15章。对于为已经具有一些编程知识（不一定是C语言的知识）的学生而设立的课程，教师也可以使用到本书的所有主题。本书也能用作需要学生使用C的其他计算机科学课程的教材。

交互式环境

本书可用于交互式环境。通过键盘和屏幕的实验完全受到鼓励。对PC机而言有很多厂商提供了交互式C/C++系统，例如Borland、IBM、Metroworks、Microsoft和Symantec。

作为专业程序员的教材

虽然本书是为初级程序员准备的，但对于有经验的程序员来说，本书也是对C语言较好的介绍。结合Al Kelley和Ira Pohl所著的《*A Book on C, Fourth Edition*》(Addison Wesley Longman, Inc., Reading, MA 1998, ISDN 0-201-18399-4)，计算机专业人员会获得对该语言的全面理解，包括在MS-DOS和UNIX下有关使用它的关键之处。作为一种结合，这两本书提供了对C编程语言的完整分析以及在别处难以获得的用法。此外，本书与Ira Pohl所著的《*C++ for C Programmers, Third Edition*》和Ira Pohl所著的《*Object-Oriented Programming Using C++, Second Edition*》(Addison Wesley Longman, Inc., Reading, MA 1997, ISDN 0-201-89550-1)相结合，还向学生或专业人员提供了对面向对象语言C++的完整分析。

ANSI C标准

ANSI 是美国国家标准协会 (American National Standards Institute) 的英文首写字母的缩写，该协会致力于包括编程语言在内的多种标准的制订。特别是ANSI委员会X3J11负责制订编程语言C的标准。在20世纪80年代后期，该委员会创建了称为“ANSI C”或“标准C”的标准草案。到1990年，该委员会完成了这项标准，国际标准化组织 (International Standardization Organization, ISO) 也批准了这项标准。因而，ANSI C或ANSI/ISO C标准在国际上获得承认。

ANSI C标准规定了用C编写的程序的格式，并确定了如何解释这些程序。该标准的目的是促进C语言程序在各种机器上的可移植性、可靠性、可维护性和有效执行。所有主要的C编译器都遵循ANSI C标准。

Al Kelley

Ira Pohl

University of California, Santa Cruz

目 录

出版者的话	
专家指导委员会	
译者序	
前言	
第1章 编写ANSI C程序	1
1.1 准备编程	1
1.2 第一个程序	2
1.3 变量、表达式和赋值	5
1.4 初始化	8
1.5 #define和#include的用法	8
1.6 printf()和scanf()的用法	9
1.6.1 printf()的用法	10
1.6.2 scanf()的用法	11
1.7 while语句	13
1.8 问题求解: 计算总和	14
1.9 风格	15
1.10 常见的编程错误	16
1.11 系统考虑	17
1.11.1 编写和运行C程序	17
1.11.2 中断程序	18
1.11.3 输入文件尾标识	18
1.11.4 输入和输出的重定向	18
1.12 转向C++	19
小结	20
练习	20
第2章 词法元素、运算符和C系统	26
2.1 字符和词法元素	27
2.2 注释	28
2.3 关键字	29
2.4 标识符	29
2.5 常量	30
2.6 串常量	31
2.7 运算符和标点符号	31
2.8 运算符的优先级和结合性	32
2.9 增量运算符和减量运算符	33
2.10 赋值运算符	34
2.11 例子: 计算2的幂	36
2.12 C系统	36
2.12.1 预处理器	37
2.12.2 标准库	37
2.13 风格	39
2.14 常见的编程错误	40
2.15 系统考虑	41
2.16 转向C++	42
小结	42
练习	43
第3章 控制流	48
3.1 关系、等式和逻辑运算符	48
3.2 关系运算符和表达式	49
3.3 等式运算符和表达式	50
3.4 逻辑运算符和表达式	50
3.5 复合语句	53
3.6 空语句	53
3.7 if和if-else语句	53
3.8 while语句	56
3.9 问题求解: 找最大值	57
3.10 for语句	58
3.11 问题求解: 组合数学	59
3.12 问题求解: 布尔变量	61
3.13 逗号运算符	61
3.14 do语句	62
3.15 goto语句	63
3.16 break和continue语句	64
3.17 switch语句	65
3.18 嵌套的控制流	65
3.19 条件运算符	66
3.20 风格	67
3.21 常见的编程错误	68

3.22 系统考虑	70	6.2 基本数据类型	128
3.23 转向C++	71	6.3 字符和数据类型char	129
小结	71	6.4 数据类型int	130
练习	72	6.5 整数类型short、long和unsigned	131
第4章 函数和结构化编程	78	6.6 浮点类型	132
4.1 函数调用	78	6.7 sizeof运算符	134
4.2 函数定义	78	6.8 数学函数	134
4.3 return语句	80	6.9 转换和类型转换	136
4.4 函数原型	82	6.9.1 整型提升	136
4.5 自顶向下设计	83	6.9.2 常用的算术转换	136
4.6 程序的正确性: assert()宏	86	6.9.3 类型转换	137
4.7 从编译器的角度来看函数声明	87	6.10 问题求解: 计算利息	138
4.8 问题求解: 随机数	88	6.11 风格	141
4.9 函数定义次序的可选风格	89	6.12 常见的编程错误	142
4.10 开发一个大程序	90	6.13 系统考虑	143
4.11 模拟: 正反面游戏	93	6.14 转向C++	143
4.12 调用和按值调用	95	小结	144
4.13 递归	96	练习	145
4.14 风格	97	第7章 枚举类型和typedef	151
4.15 常见的编程错误	97	7.1 枚举类型	151
4.16 系统考虑	99	7.2 typedef的用法	152
4.17 转向C++	100	7.3 例子: 石头、剪刀、布游戏	154
小结	102	7.4 风格	158
练习	103	7.5 常见的编程错误	159
第5章 字符处理	110	7.6 系统考虑	160
5.1 数据类型char	110	7.7 转向C++	160
5.2 getchar()和putchar()的用法	112	小结	160
5.3 例子: 大写	115	练习	161
5.4 ctype.h中的宏	117	第8章 函数、指针和存储类型	165
5.5 问题求解: 重复字符	117	8.1 指针声明和赋值	165
5.6 问题求解: 对单词计数	118	8.2 地址和间接访问	166
5.7 风格	120	8.3 指向void的指针	169
5.8 常见的编程错误	121	8.4 引用调用	170
5.9 系统考虑	122	8.5 作用域规则	171
5.10 转向C++	123	8.6 存储类型	172
小结	125	8.6.1 存储类型auto	172
练习	125	8.6.2 存储类型extern	172
第6章 基本数据类型	128	8.6.3 存储类型register	174
6.1 声明和表达式	128	8.6.4 存储类型static	174

8.7 静态外部变量	175	10.10 系统考虑	221
8.8 缺省的初始化	176	10.11 转向C++	221
8.9 例子：字符处理	176	小结	221
8.10 函数声明和函数定义	179	练习	222
8.11 类型限定符const和volatile	179	第11章 递归	225
8.12 风格	180	11.1 递归问题求解	225
8.13 常见的编程错误	181	11.2 例子：在屏幕上绘制图案	228
8.14 系统考虑	182	11.3 用递归处理串	229
8.15 转向C++	183	11.4 分而治之的方法	230
小结	184	11.5 例子：汉诺塔	231
练习	185	11.6 风格	237
第9章 数组和指针	190	11.7 常见的编程错误	238
9.1 一维数组	190	11.8 系统考虑	239
9.1.1 初始化	191	11.9 转向C++	239
9.1.2 下标	192	小结	240
9.2 例子：分别对每个字母计数	192	练习	240
9.3 数组和指针间的关系	194	第12章 结构和抽象数据类型	245
9.4 指针运算和元素尺寸	195	12.1 声明结构	245
9.5 把数组传递给函数	195	12.2 访问成员	246
9.6 排序算法：冒泡排序	196	12.3 运算符的优先级和结合性：总结	248
9.7 二维数组	197	12.4 结构、函数和赋值	249
9.8 多维数组	199	12.5 问题求解：学生记录	251
9.9 动态内存分配	200	12.6 结构的初始化	253
9.10 风格	201	12.7 typedef的用法	253
9.11 常见的编程错误	203	12.8 自引用结构	254
9.12 系统考虑	203	12.9 线性链表	255
9.13 转向C++	204	12.10 对链表的操作	256
小结	205	12.11 计数和查找	257
练习	206	12.12 插入和删除	258
第10章 串和指针	211	12.13 风格	259
10.1 串结束标志	211	12.14 常见的编程错误	260
10.2 串的初始化	212	12.15 系统考虑	260
10.3 例子：心情愉快	212	12.16 转向C++	261
10.4 用指针处理串	214	小结	266
10.5 问题求解：单词计数	217	练习	266
10.6 把参数传递给main()	218	第13章 输入/输出和文件	269
10.7 标准库中的串处理函数	218	13.1 输出函数printf()	269
10.8 风格	220	13.2 输入函数scanf()	272
10.9 常见的编程错误	220	13.2.1 控制串中的指示	273

13.2.2 普通字符	273	14.10 其他有用的工具	309
13.2.3 空白字符	273	14.11 风格	310
13.2.4 转换说明	273	14.12 常见的编程错误	310
13.2.5 输入流中的浮点数	275	14.13 系统考虑	311
13.2.6 使用扫描集	275	小结	311
13.2.7 返回值	276	练习	312
13.2.8 一个scanf()的例子	276	第15章 从C到C++	315
13.3 函数sprintf()和sscanf()	277	15.1 为什么转到C++	315
13.4 函数fprintf()和fscanf()	277	15.2 类和抽象数据类型	317
13.5 访问文件	278	15.3 重载	318
13.6 例子:对文件行距加倍	279	15.4 构造器和析构器	320
13.7 使用临时文件和得体的函数	281	15.5 继承	321
13.8 随机地访问文件	283	15.6 多态性	322
13.9 风格	284	15.7 模板	324
13.10 常见的编程错误	285	15.8 C++中的异常	325
13.11 系统考虑	286	15.9 面向对象编程的益处	325
13.12 转向C++	287	15.10 风格	325
小结	289	15.11 常见的编程错误	326
练习	290	15.12 系统考虑	326
第14章 软件工具	295	小结	326
14.1 在C程序中执行命令	295	练习	327
14.2 环境变量	296	附录A 标准库	329
14.3 C编译器	297	附录B 预处理器	354
14.4 创建库	299	附录C 位运算符	363
14.5 使用profiler	300	附录D ANSI C与传统C的比较	375
14.6 关于时间的编码	303	附录E ASCII字符编码表	380
14.7 dbx的用法	305	附录F 运算符的优先级和结合性	381
14.8 make的用法	306	索引	382
14.9 touch的用法	309		

第1章 编写ANSI C程序

本章向读者介绍ANSI C编程世界。本章要讨论一些关于编程的一般思想，并要对一些基本的程序详尽地进行解释。本章提出的基本思想是后面各章中出现的更完整的说明的基础。C的基本输入/输出函数是本章的一个重点。把信息输入到机器和从机器输出信息是掌握任何编程语言的的第一项任务。

C广泛地把printf()和scanf()函数分别用于输入和输出。本章对这两个函数的用法进行了解释。本章中讨论的其他主题包括用于存储值的变量的用法以及用于改变变量值的表达式和赋值语句的用法。本章也包括对while语句的讨论。一个例子用于表明while语句如何提供重复的操作。

本章和本书给出了很多例子，包括很多要经常用于解析的完整程序，这使得读者能详细地理解每个结构如何工作。本章介绍的主题在后续章节中也会出现，在适当的地方会有更详细的解释。这种螺旋式的学习方法向程序员强调C编程的思想和技术本质。

每章以标题为“转向C++”的一节结束。C++大部分是C的超集。通过学习C++，你也可以学习到C的核心语言。Ira Pohl所著的本书的续集《C for C++ Programmers》第2版，讲述本书中没出现的C++的其余部分。大多数章有基于Java的练习。Java是部分基于C的，然而，与C++不同，一些C的概念不适合于Java或在Java中有不同的含义。正如在姊妹篇——Ira Pohl和Charlie McDowell所著的《Java by Dissection》——中所发现的那样，越来越多的有Java背景的人开始用C编程。现代的程序员在使用所有3种基于C的语言中必须是得心应手的。

1.1 准备编程

编写程序用于指示计算机完成特定的任务或解决特定的问题。步进式地完成所需任务的过程被称为算法(algorithm)。因而编程是沟通算法和计算机的活动。我们都习惯于用语言给别人下指令，让其完成该指令。编程的过程与此类似，只是机器不能容忍多义性，必须要用精确的语言不厌其烦地详述机器要执行的所有步骤。

编程过程

- 1) 描述任务。
- 2) 找出解决问题的算法。
- 3) 用C对算法编码。
- 4) 测试代码。

计算机是由处理器、内存和输入/输出设备这三个主要部件组成的数字电子机器。处理器也称作中央处理单元(central processing unit, CPU)。处理器执行存储在内存中的指令。数据和指令一样也存储在内存中，按一定的要求方式，通常处理器按指令操纵数据。输入/输出设备从机器的外部介质获取信息，并向外部介质提供信息。输入设备一般是终端键盘、磁盘驱动器和磁带机。输出设备一般是终端屏幕、打印机、磁盘驱动器和磁带机。机器的物理组成可

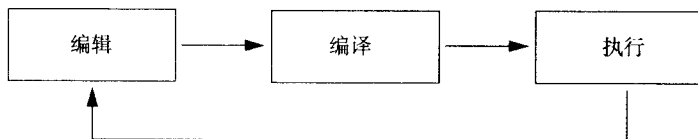
能是很复杂的，但用户不必关心这些细节。机器的操作系统负责协调机器的资源。

操作系统由一些专门的程序组成，有两个主要的用途。首先，操作系统在整体上监视和协调机器的资源。例如，当在磁盘上创建文件时，操作系统负责在磁盘的适当位置上定位文件，以及掌管文件名、文件的尺寸和创建的日期这样的细节。此外，操作系统向用户提供工具，其中的很多工具对于C程序员来说是很有用的。文本编辑器和C编译器就是其中的两个极为重要的工具。

我们假设读者能够用文本编辑器创建和修改含有C代码的文件。C代码也称作源代码(source code)，含有源代码的文件称为源文件(source file)。在创建了含有源代码的文件(程序)后，就可以调用C编译器。这个过程是与系统相关的(参见1.11.1节“编写和运行C程序”)。例如，在一些系统中，我们能利用命令`cc pgm.c`调用编译器，其中`pgm.c`是含有程序的文件名。如果在`pgm.c`中没有错误，该命令产生一个能够运行或执行的可执行文件(executable file)。虽然我们把这看作是在编译程序，但实际上发生的事情比这要复杂得多。

在第14章“软件工具”中，我们将更详细地讨论编译过程。在这里我们仅想提及基础知识。当我们编译一个简单程序时，要发生三个独立的操作：首先调用预处理器，然后调用编译器，最后调用加载器。通过加进其他文件和做一些其他的改变，预处理器修改源代码的拷贝。(在1.5节“`#define`和`#include`的用法”中，我们要进一步讨论预处理器。)编译器把编辑的结果转换成目标代码(object code)，加载器产生最终可执行的文件。含有目标代码的文件称为目标文件(object file)。目标文件与源代码文件不同，它通常是不可读的。当我们说编译程序时，实际意味着是调用预处理器、编译器和加载器。对于简单程序而言，用简单的命令做这些就足够了。

在程序员编写一个程序后，他必须要编译和测试他的程序。如果需要修改，就必须再次编辑源代码。这样，编程过程由下述循环组成：



在程序员对程序的性能满意后，就结束循环。

1.2 第一个程序

对任何人而言，学习编程的第一项工作都是在屏幕上显示。让我们从编写在屏幕上显示短语“Hello, world!”的Kernighan和Ritchie程序开始。完整的程序如下：

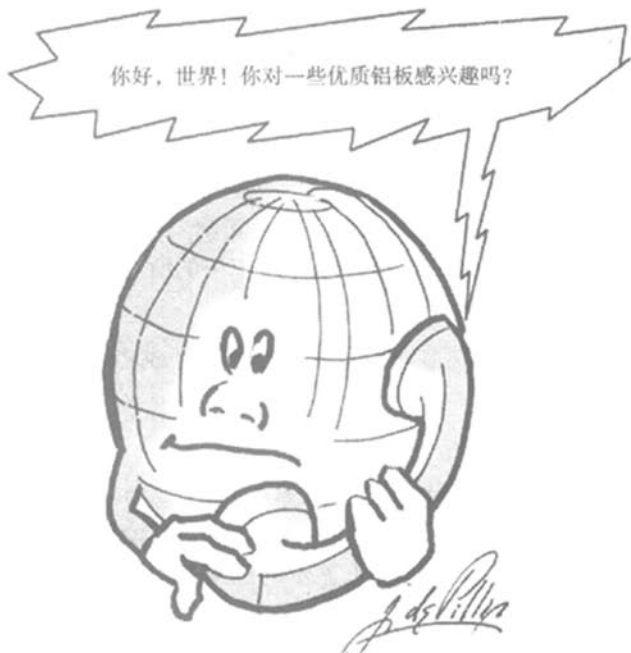
```
/* The traditional first program in honor of
   Dennis Ritchie who invented C at Bell Labs in 1972 */
#include <stdio.h>

int main(void)
{
    printf("Hello, world!\n");
    return 0;
}
```

用文本编辑器，程序员把这段程序键入到一个以.c结尾的文件。所选择的文件名应该有助于记忆。假设我们用`hello.c`作为已经编写的程序的文件名。当编译后执行该程序时，在屏幕

上会出现：

Hello, world!



计算机编程界的最流行习语是故事未完。

对程序hello的解析

• /*The traditional first program in honor of Dennis Ritchie who invented C at Bell Labs in 1972*/

这行是注释。编译器忽略了由开始的符号对/*和结束的符号对*/括起来的文本。它是为阅读程序的人提供的文档。

• #include <stdio.h >

用#开始的行被称为预处理指令(preprocessing directive)。它们用于与预处理器通信。#include指令使得预处理器在代码的当前位置上加入标准头文件stdio.h的拷贝。这个头文件是C系统提供的。<stdio.h>的尖括号指明该文件能在“通常的位置”找到，这是与系统相关的。我们引入这个文件是因为它含有关于printf()函数的信息。

• int main(void)

每一个程序都有一个命名为main的函数，程序要从这里开始执行。main后面的圆括号告诉编译器这是一个函数，关键字int声明返回类型的值是整型值。关键字void表明该函数没有参数。

• {

各函数体以左花括号开始，相应的右花括号必须出现在函数末尾。我们的风格是每个括号单占一行，并靠左放置。这种括号也用于把语句组织在一起。

- printf()

C系统包含一个能用于程序中的标准函数库。该函数来自于在屏幕上显示的函数库。我们引入头文件stdio.h是因为它向编译器提供了一定的关于函数printf()的信息，即它的函数原型。(4.4节“函数原型”中讨论了关于函数原型的内容。)

- printf("hello,world!\n");

这是用一个单参数(即串"Hello,world!\n")对函数printf()的调用或引用。C中的串常量是用双引号括起来的一串字符。本串是函数printf()的一个参数,它控制显示内容。在串尾的两个字符\n代表一个被称为换行的单字符,这是一个非打印字符,其作用是把光标移到屏幕的下一行的开头。注意本行用一个分号结尾。C中的所有声明和语句都用分号结尾。

- return 0;

main()向操作系统返回整数值零。零意味着程序已经成功结束。非零值用于告诉操作系统main()没有成功。

- }

该右花括号与上面的左花括号相匹配,它结束了函数main()。

函数printf()产生的效果是在屏幕上连续地显示,当读到一个换行字符时,会移动到一个新行。屏幕是一个从左到右和从上到下显示的二维显示器。为了读起来方便起见,程序员必须考虑输出在屏幕上的布局。

下面用两个printf()语句重写我们的程序。虽然程序看起来不同,但输出是一样的。

```
#include <stdio.h>

int main(void)
{
    printf("Hello, ");
    printf("world!\n");
    return 0;
}
```

注意作为第一个printf()语句的参数的串用一个空格字符结尾。如果此处没有这个空格字符,输出的Hello world!间将没有空格。

作为该程序的最后一个变型,让我们再增加一个短语“Hello,universe!”,并在两行显示。

```
#include <stdio.h>

int main(void)
{
    printf("Hello, world!\n");
    printf("Hello, universe!\n");
    return 0;
}
```

执行这个程序时,如下信息会出现在屏幕上:

```
Hello, world!
Hello, universe!
```

注意在main()体内的两个printf()语句可以用下面的一个语句代替:

```
printf("Hello, world!\nHello, universe!\n");
```

在标准库中像printf()这样的有用函数的可用性是C的一个重要特征。虽然在技术上标准库不是C语言的一部分,但它是C系统的一部分,因为在标准库中的函数在C系统中随处可